

Cognitive Services de Microsoft

Projet de semestre

Etienne Guignard

Année 2017-2018

02.09.2017

Version 1.5



Table des matières

1	Introduction	3
1.1	Objectifs	3
2	Cahier des charges	3
3	Etude des Cognitive Services	4
3.1	Service Vision	4
3.2	Service Speech	4
3.3	Service Knowledge	5
3.4	Service Search	5
3.5	Service Language.....	5
3.6	Coûts	6
3.6.1	API Face.....	6
3.6.2	API Vision	7
4	Conception et réalisation.....	8
4.1	Concept	8
4.2	Environnement de développement.....	8
4.3	Prérequis.....	9
4.3.1	Librairies Cognitives Services	9
4.3.2	Librairies ImageView	11
4.4	Base de données	13
4.5	Test préliminaire des APIs.....	14
4.6	Programme Photo Identifier	15
4.6.1	Association (ajout)	15
4.6.2	Identification	16
4.6.3	Enregistrement	17
4.7	Programme Photo Finder.....	17
4.8	Fichier de configuration	17
4.9	OpenCV	18
5	Conclusion.....	18
6	Références	19
7	Table des illustrations	19
8	Table des tableaux	19

1 Introduction

À ce jour, l'intelligence artificielle, dont l'objectif est de permettre à la machine de simuler l'intelligence humaine, est un domaine en pleine expansion. C'est pourquoi Microsoft s'est notamment doté d'un département de recherche dénommé *Cognitive Service* afin d'approfondir le développement d'algorithmes en relation avec l'AI et les sciences cognitives (1). Dans ce projet, il sera donc question d'étudier dans un premier temps les différentes solutions et services proposés par Microsoft dans ce domaine puis, de développer une application à partir d'un de ces services afin de voir concrètement et dans la pratique quelles sont ses fonctionnalités et ses capacités.

1.1 Objectifs

L'objectif premier de ce projet est d'effectuer une étude de l'API *Vision* des *Cognitive Services* de Microsoft Azure à savoir, la détection de mouvement, le marquage de vidéos et la stabilisation de rendu vidéo. Après avoir discuté avec le professeur en charge de ce projet, l'objectif a dû être redéfini afin de pouvoir réaliser une application plus concrète utilisant les services de Microsoft. Par conséquent, le nouvel objectif de ce projet consiste à se familiariser avec les différentes API du service *Vision* proposé par les *Cognitive Services* de Microsoft et à développer une application en C# portant sur la détection de personnes dans une image et la classification (recherche) de photos.

2 Cahier des charges

Comme mentionné dans le paragraphe précédent, le cahier des charges a été redéfini. Par conséquent, voici les différents points effectués pour ce travail de semestre :

- Analyse et test des *Cognitive Services* de Microsoft.
- Choix du projet à réaliser en relation avec les *Cognitive Services* de Microsoft.
- Réalisation d'une application de détection de personnes dans une photo utilisant :
 1. L'API *Face* pour la détection de personnes dans une photo.
 2. L'API *Emotion* pour récupérer des paramètres relatifs à une personne tels que l'âge, la couleur de cheveux, le genre ou encore l'expression faciale.
 3. L'API *Computer Vision* afin de détecter divers éléments tels les couleur dominante, l'environnement où la photo a été prise, le texte présent dans la photo etc.
- Réalisation d'une application de classification (recherche) de photos.

3 Etude des *Cognitive Services*

Les *Cognitive Services* (anciennement Project Oxford), sont un ensemble d'API et de SDK qui mettent à disposition des algorithmes d'apprentissage automatique que Microsoft a développé afin de résoudre des problèmes dans le domaine de l'intelligence artificielle. L'objectif de ces services est de démocratiser l'IA pour que les développeurs puissent plus facilement les intégrer et les utiliser au sein de leurs propres applications. Ces services sont disponibles par le biais d'Azure (2), la plateforme applicative du cloud de Microsoft, et sont divisés en cinq grandes catégories :

- *Vision*
- *Knowledge*
- *Language*
- *Speech*
- *Search*

La grande majorité de ces services peuvent être utilisés avec des appels standard REST, documentés pour chaque API, ce qui permet d'utiliser facilement ces fonctionnalités indépendamment du langage et de l'OS employés.

Toutefois, il faut savoir que la plupart des API des *Cognitive Services* sont actuellement en version préliminaire et que par conséquent, elles sont en cours d'amélioration et de modification en fonction des commentaires des utilisateurs.

3.1 Service *Vision*

Le service *Vision* est composé d'algorithmes intelligents permettant de récupérer des informations exploitables à partir d'images et de vidéos. Ce service est composé des sous-ensembles suivants :

- *Computer Vision* : récupère des informations pertinentes, comme par exemple du texte, à partir d'une image.
- *Content moderator* : détecte des informations indésirables, dans des images, vidéos et textes, comme du contenu pour adulte ou de la vulgarité.
- *Custom Vision Service* : personnalise des modèles de *Vision* par ordinateur pour les adapter à un cas d'utilisation précis.
- *Face* : détecte, analyse, organise et identifie des visages dans une photo.
- *Emotion* : détecte une expression faciale et l'associe à un ensemble d'émotions.
- *Video Indexer* : extrait et trie de données dans une vidéo.

3.2 Service *Speech*

Le service *Speech* permet notamment de convertir du contenu audio en texte ou de comparer la voix de plusieurs personnes pour chercher des similitudes entre elles. Ce service est composé des sous-ensembles suivants :

- *Translator Speech* : traduction orale en temps réel.
- *Bing Speech* : reconnaître l'intention d'un utilisateur.
- *Speaker Recognition* : identifier et authentifier une personne en fonction de sa voix.
- *Custom Speech Service* : filtrer les bruits de fond ou le style d'élocution.

3.3 Service Knowledge

Le service *Knowledge* permet de faire correspondre entre elles des données et des informations complexes pour résoudre diverses tâches telles qu'effectuer des suggestions intelligentes ou faire de la recherche sémantique. Ce service est composé des sous-ensembles suivants :

- *QnA Maker* : gestion de questions et de réponses de style conversationnel.
- *Custom Decision Service* : prise de décision en fonction d'un contexte.
- *Project Knowledge Exploration* : recherche sur de données structurées en fonction d'une grammaire prédéfinie.
- *Project Academic Knowledge* : recherche de similitudes dans des documents, interprétation et anticipation des recherches de l'utilisateur.
- *Project Entity Linking* : en fonction du contexte, détermine quel est le sens correct d'un mot pouvant en avoir plusieurs.

3.4 Service Search

Le service *Search* permet de rechercher et de comparer des page web, des images et des vidéos. Ce service est composé des sous-ensembles suivants :

- *Autosuggest* : suggestion automatique pour les recherches.
- *News Search* : recherche d'articles traitant d'actualité sur Internet.
- *Web Search* : recherche de documents web indexés.
- *Entity Search* : recherche en fonction d'un lieu ou d'un service à proximité.
- *Image Search* : recherche d'images en fonction de critères précis.
- *Video Search* : recherche de vidéos en fonction de critères précis.

3.5 Service Language

Le service *Language*, par le biais du traitement de texte, permet d'évaluer les sentiments et de reconnaître les attentes d'un utilisateur. Ce service est composé des sous-ensembles suivants :

- *Language Understanding (LUIS)* : "entraîner" une application afin qu'elle soit capable de reconnaître et de comprendre les commandes d'un utilisateur.
- *Spell Check* : détecter et corriger les fautes d'orthographe.
- *Web Language Model* : décomposer une chaîne de mots en ajoutant des espaces, comme par exemple les mots se trouvant après un hashtag.
- *Text Analytics* : analyser des émotions présentes dans un texte par le biais de l'étude de la langue et des phrases clés.
- *Translator Text* : traduire facilement du texte.
- *Linguistic Analysis* : déterminer la signification d'une phrase en la fractionnant en plusieurs unités permettant d'identifier notamment le sujet de la phrase ou encore son objet.

3.6 Coûts

L'utilisation de ces services est payante. Cependant, il y a la possibilité de souscrire à une période d'essai de 12 mois comprenant notamment 190 CHF de crédit pendant les 30 premiers jours ainsi que la gratuité de 25 services même après la fin de cette période. Dans notre cas, nous avons pu bénéficier d'un pass valable pendant six mois grâce au contact de l'HEPIA avec un ingénieur de Microsoft.

Le calcul des coûts de ces différents services varie beaucoup en fonction de plusieurs paramètres, comme par exemple le nombre de transactions par minute, par heure ou par mois. Nous n'avons pas jugé nécessaire de tous les mentionner pour chaque service car sur le site de Microsoft, il y a à disposition une page (3) permettant de les calculer facilement.

En niveau Standard, Microsoft garantit que les *Cognitive Services* exécutés sont disponibles au moins dans 99,9 % des cas. Cependant, pour le niveau gratuit, il n'y a aucune garantie.

3.6.1 API Face

Niveau	Fonctionnalités	Tarif
<i>Face API - Gratuit</i>	Jusqu'à 20 transactions par minute	30 000 transactions gratuites par mois.
<i>Face API - Standard</i>	Jusqu'à 10 transactions par seconde	
	0-1 000 000 transactions	CHF 0,904 toutes les 1000 transactions
	1000001–5000000 transactions	CHF 0,723 toutes les 1000 transactions
	5000001–100000000 transactions	CHF 0,542 toutes les 1000 transactions
	Plus de 100000000 transactions	CHF 0,362 toutes les 1000 transactions
<i>Face storage</i>	Stocke les images jusqu'à 4 Mo chacune	CHF 0,226 toutes les 1000 images par mois

Tableau 1: Coûts de l'utilisation de l'API Face

3.6.2 API Vision

Niveau	Fonctionnalités	Tarif
<i>Computer Vision API</i> - Gratuit		5000 transactions gratuites par mois.
S1 - Jusqu'à 10 transactions par seconde pour les fonctionnalités suivantes	Balise, Visage, Couleur de miniatures et Type d'images	De 0-1 million de transactions : CHF 0,904 toutes les 1000 transactions. De 1 millions à 5 millions de transactions : CHF 0,723 toutes les 1000 transactions. Plus de 5 million de transactions : CHF 0,588 toutes les 1 000 transactions.
	OCR (avec impression), Contenu adulte, Célébrités et Élément géographique	De 0-1 million de transactions : CHF 1,355 toutes les 1000 transactions. De 1 à 5 millions de transactions : CHF 0,904 toutes les 1000 transactions. Plus de 5 millions de transactions : CHF 0,588 toutes les 1 000 transactions.
	Description et OCR (écriture manuscrite)	CHF 2,258 toutes les 1000 transactions.

Tableau 2: Coûts de l'utilisation de l'API Vision

4 Conception et réalisation

Après avoir lu et documenté les différentes options proposées par les *Cognitive Services* de Microsoft, le développement d'une application concrète utilisant ces services nous a paru un projet intéressant. Par conséquent, nous avons décidé de réaliser deux programmes. Le premier (*Photo Identifier*) permet de récupérer toutes les informations relatives à une photo au moyen de l'API *Face* et l'API *Visio* pour finalement les ajouter dans une base de données. Le deuxième (*Photo Finder*) a pour but de rechercher des photos en fonction de critères en relation avec les photos préalablement identifiées par le premier programme et de ce fait, disponibles dans la base de données.

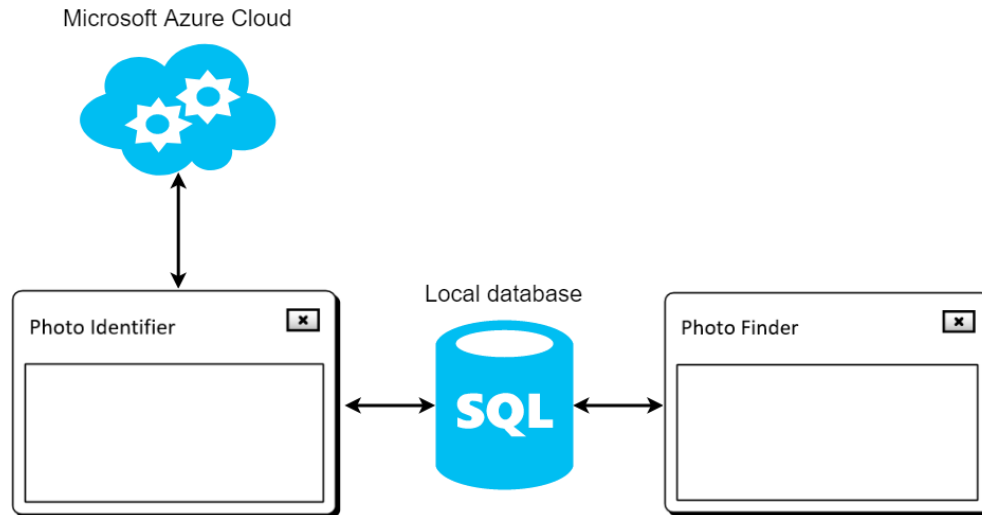


Figure 1: Schéma de base des applications

4.1 Concept

A l'ère du tout numérique, nous prenons des quantités de plus en plus importantes de photos. De même, la publication de ces photos sur des sites Internet tels que Facebook, Instagram ou même à la création d'album photos virtuel est désormais chose courante. Par conséquent, dans la masse d'images à notre disposition, il est souvent difficile de retrouver rapidement une photo selon des critères spécifiques tels que l'expression faciale ou le visage. Pour pallier à ce problème, nous avons ainsi décidé de créer deux applications : l'une permettant l'identification de personnes et d'éléments dans une photo et l'autre la recherche après identification.

4.2 Environnement de développement

Pour le développement des deux applications, nous avons travaillé avec le même environnement tout au long du projet :

- Système d'exploitation : Windows 10 Pro x64
- Environnement de développement : Visual studio 2017 Entreprise
- Framework : Microsoft .NET 4.6.1
- Langage : C#
- Versioning du code : Github

Pour protéger les clés d'API utilisées dans une des applications, j'ai décidé de ne pas rendre ce projet (4) public sur GitHub. Par conséquent, pour y avoir accès, il suffit de me contacter.

4.3 Prérequis

Pour que les programmes fonctionnent dans des conditions similaires à celles employées dans le cadre de ce travail de semestre, nous avons dû installer plusieurs librairies qui seront détaillées ci-dessous.

4.3.1 Librairies *Cognitive Services*

4.3.1.1 Azure

Pour accéder à ces services, il est nécessaire de créer ou de posséder un compte Microsoft puis, de s'identifier sur le portail Azure. D'autre part, pour utiliser un de ces service, il faut d'abord effectuer trois manipulations sur la plateforme Azure :

- Sélectionner et souscrire à un service
- Récupérer la clé d'API
- Récupérer l'URL permettant d'accéder à un *endpoint*

La clé d'API permet de s'authentifier pour chaque requête effectuée sur le serveur. C'est pour cela qu'il est important de la garder secrète. L'URL permettant d'accéder au *endpoint* peut être différent en fonction du pays où l'on se trouve. En effet, les API des *Cognitive Services* sont hébergés sur un réseau composé de data center en pleine expansion et par conséquent, l'URL peut différer entre les régions.

Dans ce projet, nous avons souscrits à deux services :

- L'API *Face*



Congratulations! Your keys are ready.

Now explore the Quickstart guidance to get up and running with Face API.

1

Grab your keys

Every call to the Face API requires a subscription key. This key needs to be either passed through a query string parameter or specified in the request header. You can find your keys in the API resource 'Overview' or 'Keys' from the left menu.

[Keys](#)

2

Make an API call to endpoint

<https://westeurope.api.cognitive.microsoft.com/face/v1.0>

Figure 2: Souscription API Face

- L'API *Vision*



Congratulations! Your keys are ready.

Now explore the Quickstart guidance to get up and running with Computer Vision API.

1

Grab your keys

Every call to the Computer Vision API requires a subscription key. This key needs to be either passed through a query string parameter or specified in the request header. You can find your keys in the API resource 'Overview' or 'Keys' from the left menu.

[Keys](#)

2

Make an API call to endpoint

<https://westeurope.api.cognitive.microsoft.com/vision/v1.0>

Figure 3: Souscription API Vision

Après avoir effectué ces trois étapes, nous sommes enfin prêts à utiliser le service. Des tutoriaux, outils de développement et exemples sont mis à disposition pour chaque API ce qui permet une compréhension rapide de leur utilisation. Dans la plupart des cas, pour chaque API plusieurs SDK sont disponibles pour les plateformes et langages suivants :

- Android
- Windows
- IOS
- Python

4.3.1.2 *Visual studio*

Afin de pouvoir utiliser un de ces services avec Visual Studio, il est nécessaire d'installer les quatre *NuGet* paquets suivants :

- Microsoft.ProjectOxford.Vision
- Microsoft.ProjectOxford.Common
- Microsoft.ProjectOxford.Face
- Newtonsoft.Json

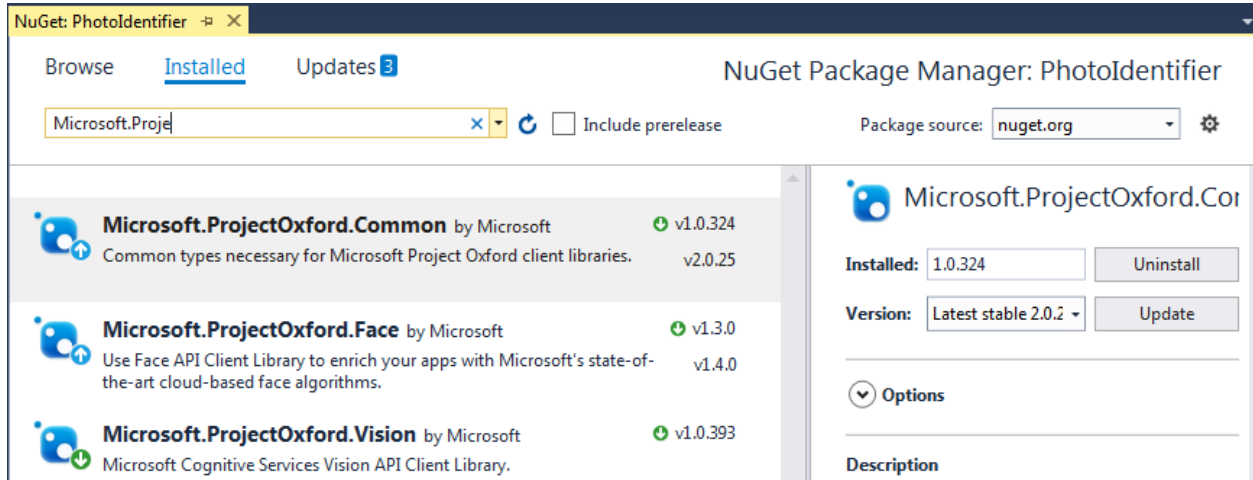


Figure 4: Installation paquet NuGet Project Oxford

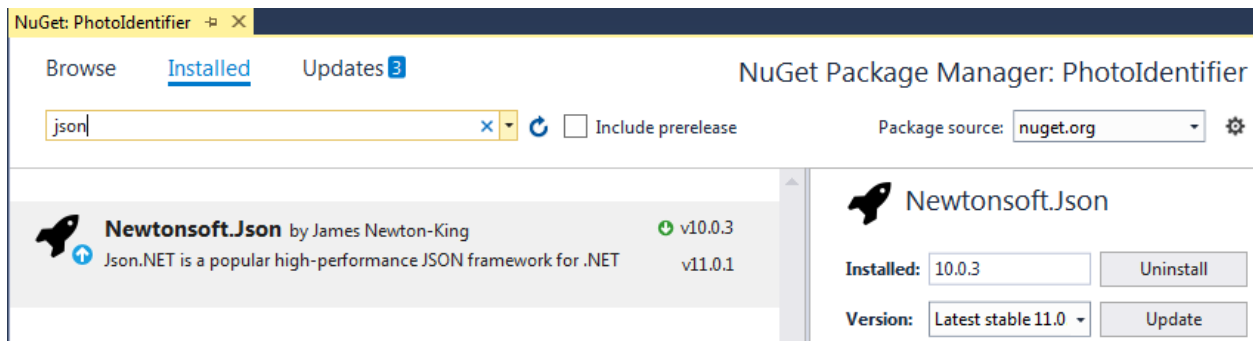


Figure 5: Installation paquet NuGet Newtonsoft

4.3.2 Librairies ImageListView

Pour l'affichage des photos dans les deux programmes, nous avons utilisé un composant externe car celui de base n'est pas très intuitif pour l'utilisateur et graphiquement pas très esthétique.

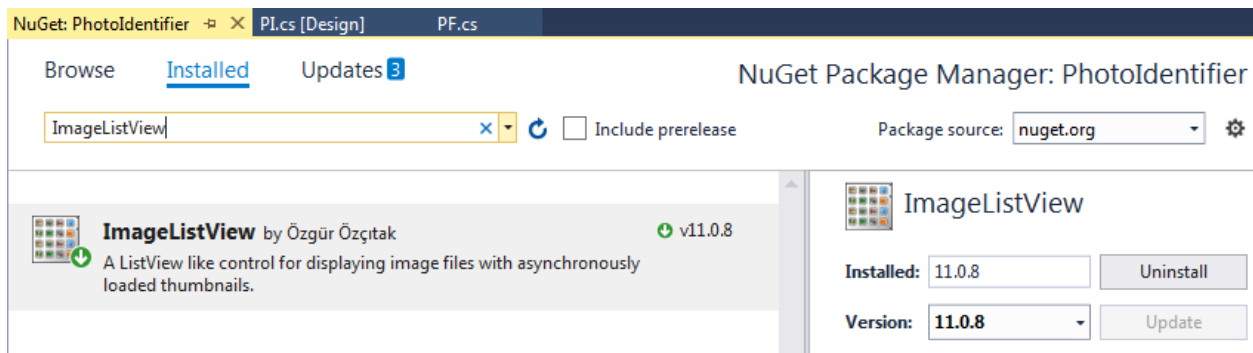


Figure 6 : Installation paquet NuGet ImageListView

A la suite de l'installation du paquet *NuGet ImageListView*, il faudra encore ajouter le composant dans la Barre d'outils de Visual Studio pour pouvoir l'utiliser.

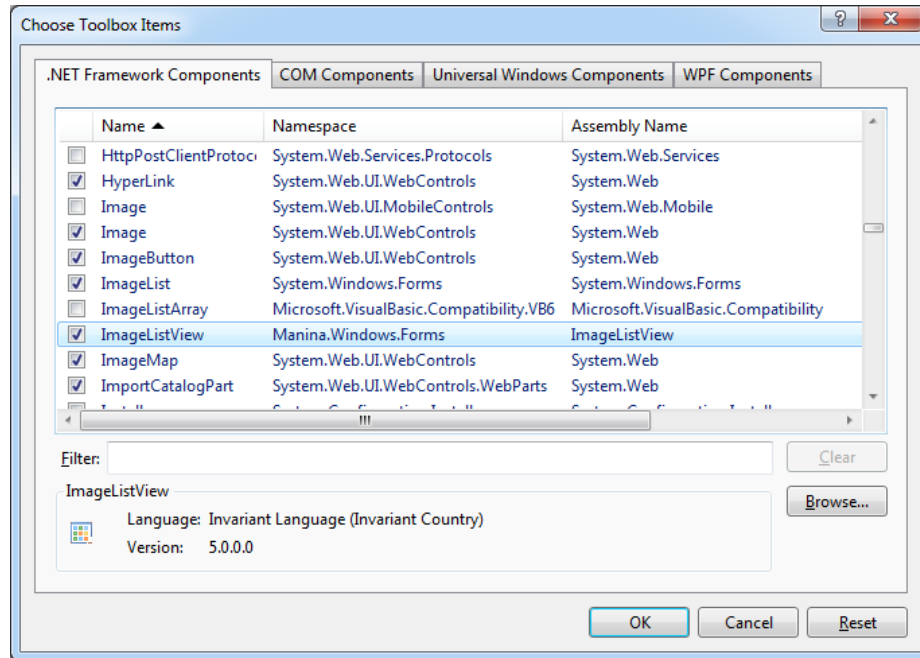


Figure 7: Ajout du composant *ImageListView* dans la boîte à outils de Visual Studio

Pendant le processus de déploiement de l'application, il est important que les librairies (dll) soit correctement rattachées au projet et au exécutable. Dans le cas où cette opération n'est pas réalisée correctement, l'application ne pourra pas se lancer sans que des erreurs soient notifiées.

4.4 Base de données

Dans ce projet, nous avons décidé d'utiliser une base de données type SQL contenue dans un fichier¹ local. Cela permet d'éviter d'interroger une base de données externe et par conséquent, de devoir mettre en place une infrastructure supplémentaire. Cela est aussi pratique car l'utilisateur a la possibilité d'effectuer des recherches dans les photos identifiées via l'application sans avoir ainsi à disposer d'une connexion à Internet.

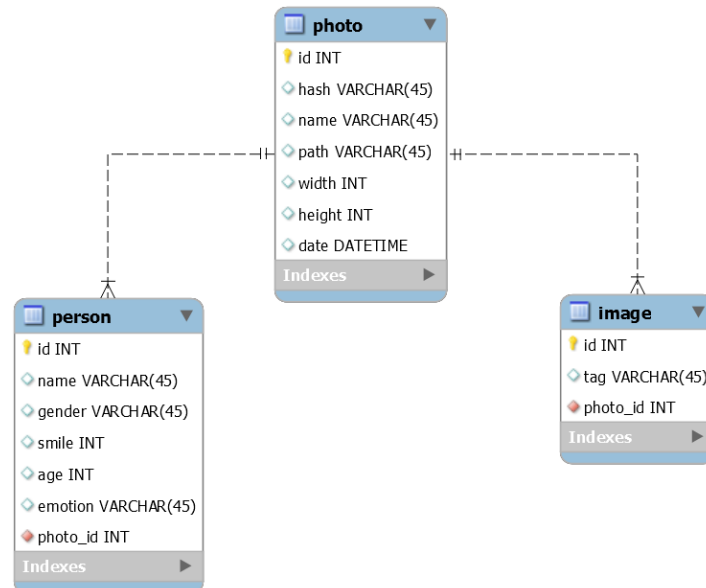


Figure 8: Base de données de l'application

La base de données est constituée de trois tables contenant les informations recueillies à partir de chaque image où des visages ont été reconnus. La première table (*photo*) réunit les caractéristiques de la photo en elle-même et permet de faire la liaison entre les deux autres tables. La deuxième table (*person*) contient les caractéristiques physiques d'une ou plusieurs personnes identifiées dans l'image. Enfin, la dernière table (*image*) permet de stocker des mots clés pertinents à partir de cette même image.

¹ photos.mdf

4.5 Test préliminaire des APIs

Avant de démarrer le développement d'une application plus complexe mais aussi pour se familiariser avec les l'API *Face* et *Visio*, nous avons effectué quelques tests en créant une application WPF. Cette application permet notamment de détecter les visages, l'âge, l'expression faciale des personnes présentes sur l'image ainsi que bien d'autres caractéristiques physiques. Voici un exemple d'exécution avec une image (5) de la famille Obama téléchargée depuis Internet :

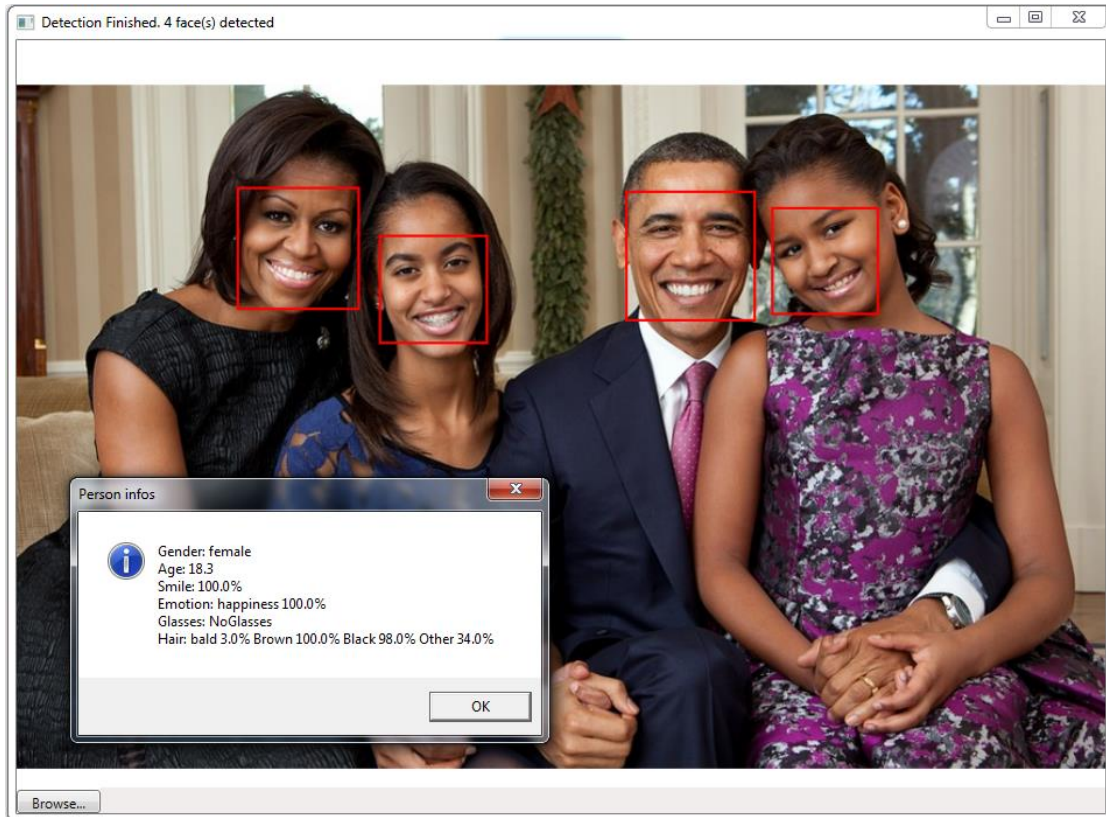


Figure 9: Famille Obama API Face

Afin de constituer une banque d'images de visages pour la suite du développement, nous avons sauvé chaque visage identifié.



Figure 10: Famille Obama visage

4.6 Programme *Photo Identifier*

Cette application va permettre d'identifier des éléments et des personnes dans une image puis, de les ajouter dans une base de données locale. Le programme comporte ainsi trois grandes fonctionnalités :

- Association : Image (visage, tête) à un nom
- Identification : photo en tout genre
- Enregistrement : des informations identifiées dans la base de données

Il est important de noter que conformément au concept du projet, seules les photos où au moins une personne est identifiée (reconnue ou non), sera enregistrée et donc traitée par la dernière fonctionnalité.

4.6.1 Association (ajout)

La première étape consiste à associer à un nom unique une ou plusieurs images composées d'un seul visage ou d'une seule tête. En effet, si sur une image il y a plus d'une tête ou d'un visage reconnu, alors celle-ci ne sera pas prise en compte lors de l'identification à l'étape suivante.

L'association s'effectue dans le cloud de Microsoft de la manière suivante :

- Création d'un groupe (hepia²)
- Création d'une personne
- *Upload* des photos
- Identification des visages
- Association de ces photos au nom et au groupe

Voici comment nous avons décidé d'effectuer l'opération d'ajout de personnes dans notre programme. Tout d'abord, cette opération peut être exécutée autant de fois qu'il y a de nouvelles personnes. Nous avons ensuite mis en place une interface qui permet de gérer les personnes déjà ajoutées. Cette interface permet uniquement l'ajout de personnes, et non l'association, car elle est précisément faite au moment de l'identification pour pallier au problème des personnes ajoutées au fur et à mesure.

² Chaque personne sera ajoutée dans ce même groupe.



Figure 11: Ajout de personne

4.6.2 Identification

La deuxième étape consiste à identifier un groupe d'images à partir des personnes ajoutées à l'étape précédente. L'identification s'effectue dans le cloud de Microsoft de la manière suivante :

- *Upload* du groupe d'image
- Identification des personnes et de leurs caractéristiques physiques pour chaque photo du groupe d'images comportant les personnes associées précédemment en fonction du groupe.
- Identification des divers éléments de chaque photo du groupe d'images tels que le texte, la couleur ou l'environnement.

Voici maintenant comment nous avons décidé d'effectuer l'opération d'identification dans notre programme. De la même façon qu'à l'étape précédente, il faudra en premier lieu ajouter des photos à identifier, puis le programme s'occupera d'effectuer le traitement et l'envoi au cloud de Microsoft.

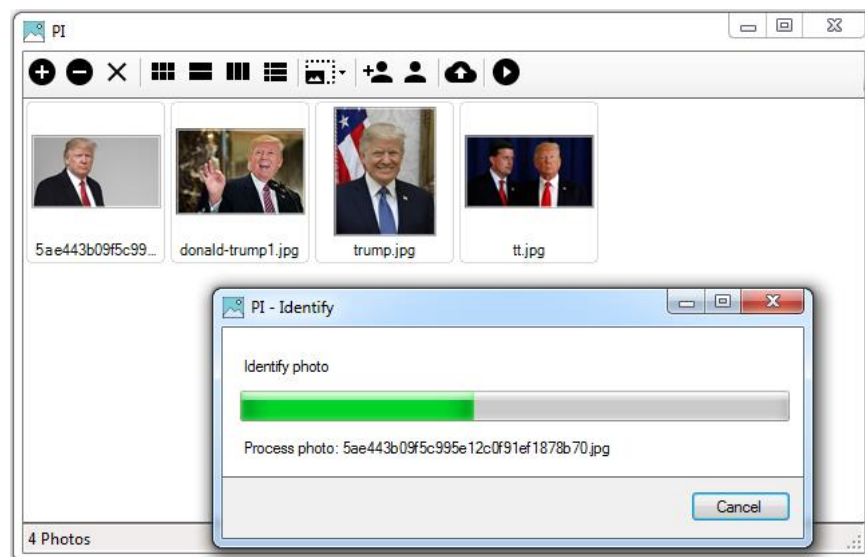


Figure 12: Identification

4.6.3 Enregistrement

La dernière étape consiste à enregistrer dans une base de données locale toutes les informations identifiées et récoltées à l'étape précédente. Toutefois, avant d'ajouter ces informations dans la base de données, on effectue un pré-traitement afin de les rendre facilement compréhensibles pour l'utilisateur.

4.7 Programme *Photo Finder*

Cette application va permettre de rechercher et de sauver des photos identifiées selon des critères spécifiques :

- L'âge ou/et le nom
- Une émotion ou/et un nom
- Un genre (homme, femme) ou/et un nom
- Un nom ou/et un nom
- Un tag (description de l'image)

Il est important de noter que les deux applications se basent sur un chemin prédéfini dans Windows. Par conséquent, il faudra effectuer l'identification des photos issues de ce chemin.

L'application va traiter la requête de l'utilisateur et récupérer les photos correspondantes avec leur chemin pour ensuite les afficher à l'utilisateur.

4.8 Fichier de configuration

Les deux programmes chargent au démarrage un fichier³ de configuration afin de récupérer les deux paramètres suivants :

- Le groupe auquel les personnes à associer font partie
- Le chemin prédéfini par Windows auquel les photos identifiées sont issues

³ app_conf.xml

4.9 OpenCV

Dans le projet que nous sommes en train de réaliser avec Monsieur Zeller, nous avons eu l'occasion d'utiliser OpenCV et notamment, sa méthode de détection de visages à l'aide de classificateurs déjà pré-entraînés. Cette méthode basée sur les fonctions de Haar (6) permet en effet une détection efficace de divers objets comme par exemple, des voitures, des vélos ou encore des visages comme mentionné ci-dessus. Il s'agit d'une approche basée sur le principe des *machine learning* consistant à entraîner une fonction à partir de nombreuses images positives et négatives pour ensuite les utiliser et détecter des objets dans d'autres images. Voici le résultat obtenu avec le flux vidéo de ma webcam :

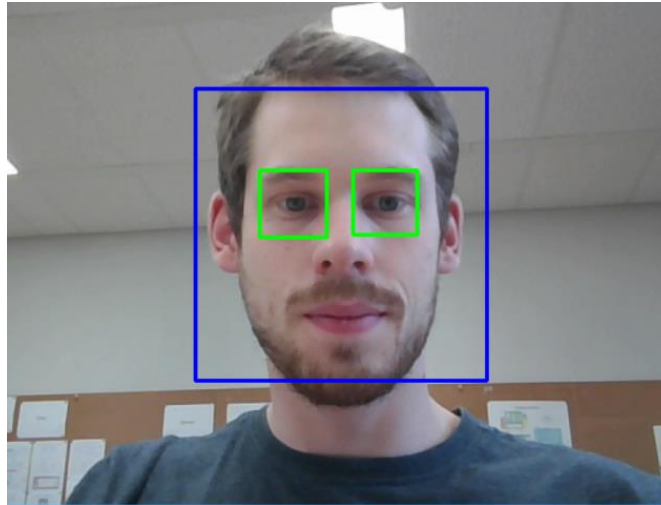


Figure 13: OpenCV détection

5 Conclusion

Ce travail de semestre a été très enrichissant car il m'a notamment permis de découvrir et de me familiariser avec le cloud de Microsoft et plus précisément, avec ses nombreuses fonctionnalités sur l'AI. En effet, c'est un domaine que je ne connaissais pas bien jusque-là. Ce projet m'a donc permis d'élargir mon domaine de connaissances.

Dans l'ensemble, les recherches et le développement de l'application se sont déroulés sans grande difficulté. Cependant, j'ai quand même rencontré des problèmes un peu étranges avec la base de données SQL. J'ai pu les résoudre mais sans toutefois comprendre vraiment comment. Pour une implémentation future, j'utiliserai donc une base de données MySQL que j'ai déjà employée dans d'autres projets.

Comme le programme n'est pas complètement abouti, on pourrait encore apporter de nombreux ajouts et modifications. Au niveau de l'interface graphique, j'aurais pu utiliser Blend pour personnaliser et revoir entièrement la ligne graphique des deux applications. D'autre part, la recherche des photos identifiées pourrait être améliorée en proposant à l'utilisateur un nombre de critères de recherche plus important. De même, au lieu de développer deux applications séparément, j'aurais pu les combiner pour rendre son utilisation plus simple. Enfin, les deux programmes, la base de données ainsi que les photos ne sont pas prévues pour être transférées d'un ordinateur à un autre. J'aurais donc pu mettre en place cette fonctionnalité.

6 Références

1. Sciences cognitives wiki. [En ligne] https://fr.wikipedia.org/wiki/Sciences_cognitives.
2. Microsoft Azure portail. [En ligne] <https://portal.azure.com/>.
3. Microsoft Azure services calculatrice. [En ligne] <https://azure.microsoft.com/fr-fr/pricing/calculator/?service=cognitive-services>.
4. Github projet de semestre. [En ligne] https://github.com/raccoonmaster/semester_project.git.
5. Obama lánya Steven Spielbergnek dolgozik. [En ligne] <http://www.instyle.hu/obama-lanya-steven-spielbergnek-dolgozik>.
6. Haar wavelet. [En ligne] https://en.wikipedia.org/wiki/Haar_wavelet.
7. Cognitive Services APIs: Vision. [En ligne] <https://blogs.windows.com/buildingapps/2017/02/13/cognitive-services-apis-vision/>.
8. What is Cognitive Services? [En ligne] <https://docs.microsoft.com/en-us/azure/cognitive-services/welcome>.
9. Github imagelistview. [En ligne] <https://github.com/oozcitak/imagelistview/wiki>.

7 Table des illustrations

Figure 1: Schéma de base des applications	8
Figure 2: Souscription API Face	9
Figure 3: Souscription API Vision	10
Figure 4: Installation paquet NuGet Project Oxford	11
Figure 5: Installation paquet NuGet Newtonsoft.....	11
Figure 6 : Installation paquet NuGet ImageListView	11
Figure 7: Ajout du composant ImageListView dans la boîte à outils de Visual Studio	12
Figure 8: Base de données de l'application	13
Figure 9: Famille Obama API Face.....	14
Figure 10: Famille Obama visage	15
Figure 11: Ajout de personne	16
Figure 12: Identification.....	16
Figure 13: OpenCV détection.....	18

8 Table des tableaux

Tableau 1: Coûts de l'utilisation de l'API Face	6
Tableau 2: Coûts de l'utilisation de l'API Vision.....	7