BRIEF 2B - Démarche projet

Classification d’images pour un end-user

Laurent Vergoz

Présentation

- **Contexte** :

  L'entreprise Classifr souhaite m'engager en tant que freelance pour une mission d'amélioration d'une application de classification d'images.

  - Leur algorithme fonctionne correctement, mais ils aimeraient ajouter deux catégories d'images qui ne sont pas prises en charge actuellement.
  - De plus, ils aimeraient que l'application à destination des utilisateur·rice·s permettant d'interagir avec cette IA de classification soit monitorée.

Objectifs

- Ajouter 2 nouvelles catégories au classificateur : 2 catégories suggérées : gâteau et pizza
- Introduire un système de monitoring de l'application afin de détecter les bugs et être proactif sur la maintenance du modèle
- Clarifier la démarche complète que je souhaite mettre en œuvre dans une application démo destinée à un·e utilisateur·rice
- Présenter la **conception** de cette application démo : elle doit assurer la classification d'images et son monitoring
- Réaliser les prémices de cette application de démo

Démarche envisagée

Dans la recherche du meilleur modèle d'Intelligence Artificielle de Classification d'image, plusieurs approches sont possibles, allant de la création de modèle "From scratch" avec les bibliothèques Keras / TensorFlow ou Pytorch plus orienté pour des développeurs / Data Scientist jusqu'au outils clé en main d'Azure Custom Vision (plus axé grand public). Cependant, dans les outils d'Azure Custom Vision, on en distingue 3, un Kit de développement (SDK) logiciel, un API REST ou bien sur le portail web Custom Vision.

L'utilisation du service d'Azure Custom Vision à cet instant a été la plus pertinente pour moi dans le cadre du projet Classifr.

En effet, pour donner suite à mes apprentissages du MicrosoftLearn ainsi que de mon passage à l'épreuve de l'AI-900 durant le déroulement de ce projet, cela m'a permis de me perfectionner et ainsi de retranscrire mes connaissances dans ce projet.

Donc m'a démarche va consister à utiliser le Custom Vision d'Azure, avec le Kit de Développement logiciel (SDK) en python, afin de pouvoir entraîner un modèle de classification d'images et d'ajouter de nouvelles catégories.

D’autres choix seront à l’études par la suite, en partant “From scratch” avec certaines librairies tels que Keras/ Tensorflow ou Pytorch

Démarche détaillée

B1. Identifier et qualifier les orientations possibles

Azure Custom Vision

Azure Custom Vision est un service de reconnaissance d’image qui nous permet de créer, de déployer et d’améliorer nos propres modèles d’identificateurs d’images.

Un identificateur d’images applique des étiquettes à des images en fonction de leurs caractéristiques visuelles. Chaque étiquette représente une classification ou un objet.

Le service Custom Vision nous permet de spécifier nos propres étiquettes et d’entraîner des modèles personnalisés pour les détecter.

On peut entraîner notre modèle soit en utilisant l’interface web de Custom Vision où bien le Kit de développement logiciel (SDK) de Custom Vision

Je me suis orienté vers l’utilisation du SDK pour des raisons de compréhension de tout le process allant de la création du projet à la prédiction et de son utilisation

Keras / TensorFlow

Keras est l'API de haut niveau de TensorFlow permettant de créer et d'entraîner des modèles de Deep Learning. Elle est utilisée dans le cadre du prototypage rapide, de la recherche de pointe et du passage en production.

Elle présente trois avantages majeurs :

- **Convivialité**
  Keras dispose d'une interface simple et cohérente, optimisée pour les cas d'utilisation courants. Elle fournit des informations claires et concrètes concernant les erreurs des utilisateurs.
- **Modularité et facilité de composition**
  Les modèles Keras sont créés en connectant des composants configurables, avec quelques restrictions.

- **Facilité d'extension**
  Composez des éléments de base personnalisés pour exprimer de nouvelles idées de recherche. Créez des calques, des métriques et des fonctions de perte, et développez des modèles de pointe.

Pytorch

PyTorch est une bibliothèque d'IA, développée par Meta (ex-Facebook), écrite en Python pour se lancer dans le Deep Learning (ou apprentissage profond) et le développement de réseaux de neurones artificiels. À partir de plusieurs variables, elle peut servir à réaliser des calculs de gradients ou à utiliser des tableaux multidimensionnels obtenus grâce à des tenseurs.

PyTorch se veut d'abord simple à prendre en main. Ce qui le rend à la fois facile d'accès et productif. Ensuite, il s'articule autour d'une architecture dynamique. Résultat : ses réseaux de neurones sont évolutifs au fur et à mesure de leur phase d'apprentissage. PyTorch permet d'ajouter de nouveaux nœuds, modifier les connexions entre eux, y compris d'une couche à l'autre. En aval, le traitement du graph du modèle étant directement adossé au moteur d'exécution de PyTorch, le Framework peut aisément être utilisé avec divers environnements de débogage tiers comme PyCharm ou ipdb.

B2. Timing envisagé

Afin de pouvoir réaliser les différents labos, et ainsi optimiser le choix de la meilleure solution, je me donne 2 à 3 jours par labos sur une période de 10 jrs ouvrés

B3-1. Orientation1. Labos réalisés, conclusions

Avec Custom Vision d'Azure

En passant par le SDK, cela demande un temps de compréhension du code utilisé afin de pouvoir l'exploiter

Au travers du code, j'ai pu me connecter à la ressource sur customvision.ai, créer le projet, uploader 2 catégories existantes de l'entreprise "gnocchi et rissotto"

Entraîner le modèle avec ces catégories et test de prédiction

Ensuite, ajouter 2 nouvelles catégories "tiramisu et pizza", puis entrainer le modèle sur ces nouvelles catégories et enfin prédire

Conclusion : l'utilisation du SDK combiné avec l'interface web de custom vision est un excellent compromis et une rapidité dans la création et l'entraînement d'un modèle de classification d'images

B3-2. Orientation2. Labos réalisés, conclusions

Keras

Pas eu le temps de pouvoir le réaliser

B3-3. Orientation3. Labos réalisés, conclusions

Pytorch

Pas eu le temps de pouvoir le réaliser

B4. Choix pour la solution finale

Mon choix c'est porté sur la solution du Custom Vision

Un manque de temps afin d'explorer les 2 autres solutions ont eu raison de ce choix

Solution finale envisagée

C1. Le site web prévu

Cas d'utilisation

Un utilisateur à la suite d'un login, pourra uploader une image, voir la prédiction affichée et validé ou non la bonne prédiction

Un admin à la suite d'un login, pourra faire la même chose que l'utilisateur, avec en plus une page dédiée ou il y aura les images qui n'ont pas été bien prédit ainsi que les informations de prédiction

Technologies envisagées

Custom Vision SDK

Visual Studio

HTML

CSS

Bootstrap

Python

Django

PostgreSQL

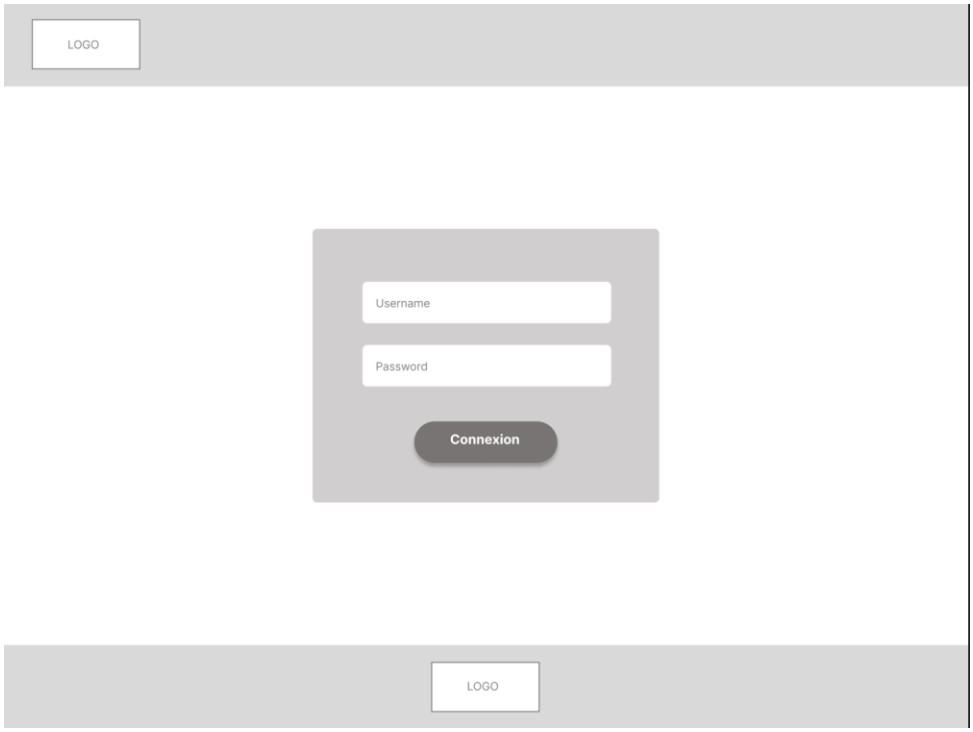
Architecture

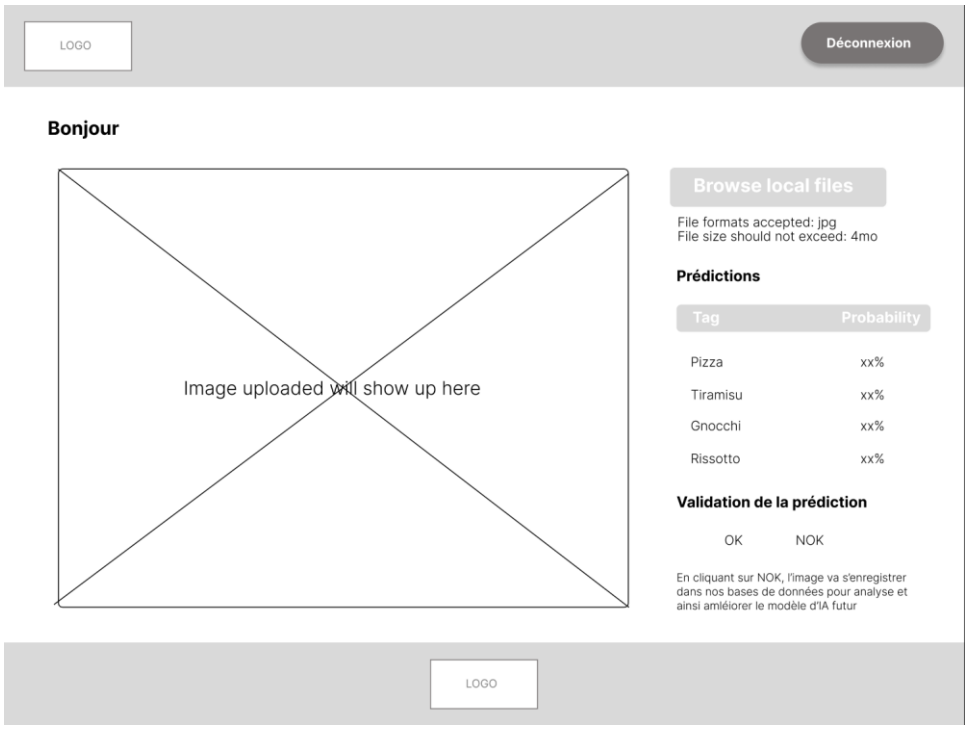
L'utilisateur, via la page web (front End) pourra uploader une image.

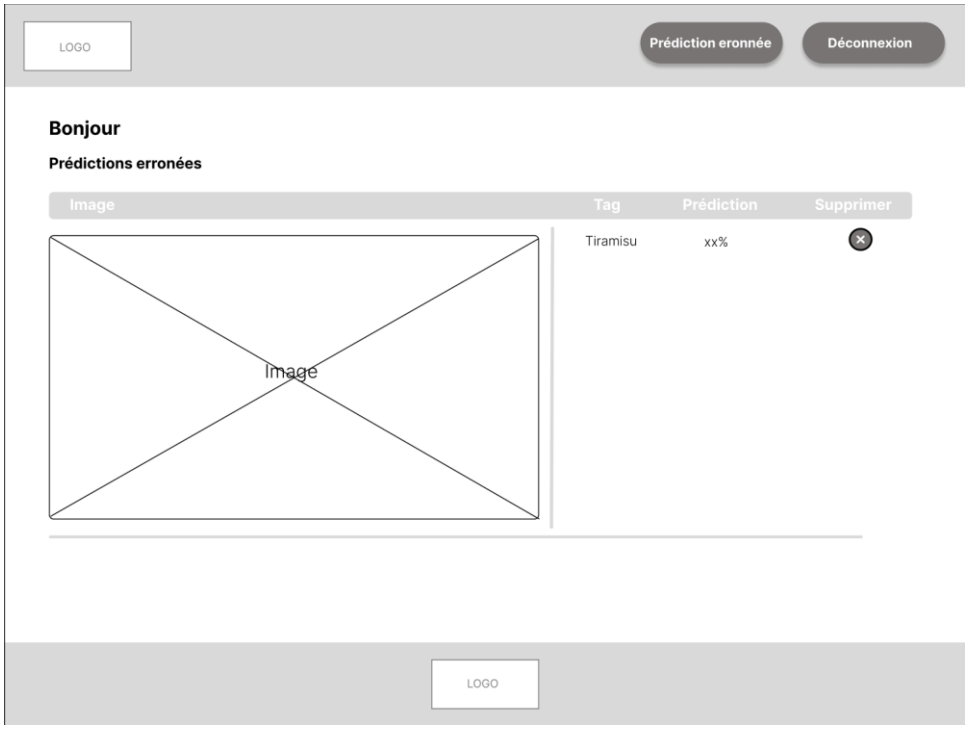
L'upload de l'image lancera une requête vers l'API de prédiction de Custom Vision (Back End) pour un retour d'information de prédiction de l'image uploadé vers la page web.

Si la prédiction n'est pas bonne, l'utilisateur à la possibilité de l'identifier (bouton NOK). Cette action entrainera la sauvegarde de l'image dans le Blob Storage d'Azure qui par la suite, le développeur pourra la retrouver sur la page web dédié

Maquettes


LOGO
Username
Password
Connexion
LOGO


LOGO
Déconnexion
Bonjour
Image uploaded will show up here
Browse local files
File formats accepted: jpg
File size should not exceed: 4mo
Prédictions
Tag
Probability
Pizza
xx%
Tiramisu
xx%
Gnocchi
xx%
Rissotto
xx%
Validation de la prédiction
OK
NOK
En cliquant sur NOK, l'image va s'enregistrer dans nos bases de données pour analyse et ainsi améliorer le modèle d'IA futur
LOGO



C2. Monitoring

Le monitoring sera basé sur l'interaction de l'utilisateur avec la bonne ou mauvaise prédiction de son image uploader.

Il pourra cliquer sur un bouton ou autre, confirmant ou non la bonne prédiction

Autre monitoring envisagée, celui fournit par le portail web de Custom Vision

C3. Base de données et Persistance des Données

Une base de données pouvant stocker les images qui n'ont pas été prédit comme il faut avec les informations de prédiction afin de pouvoir procéder à une analyse ultérieur par le développeur pour pouvoir améliorer par la suite le modèle de classification d'images.

C4. Organisation du code. Points délicats

Une approche avec Django pour la réalisation de l'application

Process de connexion de l'application Django vers Custom Vision sera détaillé dans le dossier Conception Technique

Réalisation

D1. Ce qui a été fait

Création du projet via le Kit de développement de Custom vision Classification d'image

Ajout de nouvelles catégories et test de prédiction

D2. Ce qui reste à faire

Tester de nouvelles approches avec les librairies Keras / tensorFlow et Pytorch