

Rapport

E1

Hauet Joffrey

Sommaire :

Préambule : 1

Mise en place d'un projet 2:

Code : 3

Stratégie : 4

Définition du projet : 5 -6

Projet Sprint 1: 7-11

E3 : Etat de l'art 12 - 15:

E2 : Intégration d'IA dans une application : 16 - 28

Projet : Sprint 2 : 29-31

Compte rendu : 32

Remerciement : 33

Annexe :

Preamble :

Le projet aurait pu être hypothétiquement être professionnel, le contexte sanitaire, et la réglementation forte de la donnée de mon employeur, font que le projet de secours fut utilisé..

Les projets professionnel, implique 7 mois de requêtes SQL, de nettoyage, de vérification, de liaison de table multiple et inters système d'information et 5 Mois d'une développement d'une application sous VBA et SQL appelé Tableau de bord, ou j'ai du créer divers outils utilisateurs permettant le suivie, la mise a jours, la correction, la suppression de divers indicateurs.

Le projet IA avait pour thème une exploration de la donnée inter-service, afin d'extraire des éventuels composants déterminants des clusters.

La technologie utilisée était Zeus, dans son environnement Python, m'a permis d'accéder au librairie de machine learning, celle de deep learning n'étant pas accessible dans l'environnement durant l'alternance..

J'ai eu l'occasion en conséquent de faire des analyses par composantes principales, de réaliser des classifications ascendantes hiérarchiques tout cela joint pour limiter les centres initiaux de la kmeans, qui s'effectueront sur les axes principaux de l'acp, la mission suit toujours son cours, mais m'a permis d'utiliser un fragment du machine learning.

Au vue de la mission, ce projet personnel a pour objectif d'utiliser la seconde partie de l'apprentissage de la formation Microsoft by simplon, à savoir le deep learning.

Ce projet de secours fut défini courant décembre, avant le changement de fond certificat vers titres.

De nombreuses informations étant divergentes, durant l'année j'ai surtout effectué des recherches de compréhensions et de la consolidation au vue du caractère incertains de ce qui était attendu pour le passage de ce titre.

Après l'intervention d'un élève titulaire du titre, j'ai posé des congés pour lancer pleinement mon projet, cela m'a permis de restituer une timeline cohérente au niveau des attentes backlogs, et me poser comme dans une situation professionnelle.

L'école n'ayant toujours pas communiqué les codes Azures pour l'E2 lors de l'écritures de ce rapport, mon E3 et E2 seront également présents dans cette E1, la E2 étant ajustée pour ne pas dépendre du service Azure et s'appliquera à l'évolution de ma propre application.

D'un point de vue SQL, le projet créera une base SQL avec une seule table car celle-ci n'était pas vraiment nécessaire.

MISE EN PLACE D'UN PROJET

Comme le définit le préambule, il s'agit d'un projet personnel avec un effectif de 1 salarié, cependant des procédures de projet sont à mettre en place dans le cadre du développement de celui-ci.

Tout prototype implique du versioning, avec une version stable, des versions d'essais, et de développement.

Dans le cadre de mon alternance au CAPG*, les contraintes faisaient que les fichiers étaient en production sans copie, et sans versionning ce qui suffisait mais crée des multitudes de fichiers même pour les utilisateurs qui avait tendance à conserver les divers programmes.

Le développement du programme implique un périmètre, une stratégie et des choix.

De part le caractère du projet, l'état de l'art et les sujets s'orientent sur les compétences acquises durant la période de formation, avec des logiciels gratuit et centrer sur python.

Un projet se caractérise par un bassin de compétence et des infrastructures, le bassin de compétence est l'ensemble des leviers que dispose un acteur pour réaliser une opération.

L'infrastructure ne permet pas l'accès à de multiple ordinateur, avec des composants à jour et permettant d'exploiter les technologies au maximum.

Ce projet se conçoit donc avec la perspective d'un bassin de compétence assignable à un alternant, et l'infrastructure se limitera à l'ordinateur fourni pour la formation à savoir un thinkpad w540..

Dans les faits, des éléments tiers peuvent influencer le projets, comme des termes contractuels, des environnements spécifiques au traitement de la donnée, des obligations de travail sur site sécurisé et en conséquence une limitation de l'environnement d'utilisation à travers le blocage de certaines librairies ou un retard dans les mises à jours de celle-ci.

Pour une comparaison au CAPG : les ordinateurs sont exclusifs au site, avec une utilisation de Windows, et des connections sécurisés pour l'utilisation de la donnée.

Ce type de protocole peut également être technologique à travers la reprise d'un projet en cours, c'est à dire savoir compiler avec des anciennes librairies et ne pas mettre à l'arrêt des anciennes fonctions, ce type de version peut impliquer une restriction des ressources connues et des recherches alternatives à la mises en place.

CODE

Qu'est ce que GitHub ?

GitHUB c'est avant tout Gît, un outil local développé par le créateur de linux, Linus Torvalds. Il peut s'effectuer uniquement localement, est de la est née github, en proposant une plateforme de stockage et partage du code de projet.

Celui-ci est actuellement détenu par Microsoft, il est un outil gratuit avec des options d'entreprise payante, celui-ci permet de faire du versionning et des sauvegardes en ligne de vos codes de projets.

Les projets peuvent être spécifiés privés ou ouverts.

Les projets dit “open source” sont des situations où le développeur est transparent cotés serveur ou / et client, cette situation met en avant l'accès libre à vos travaux.

Les points positifs sont les suivants : Vous pouvez donner une vitrine à vos travaux, aider des développeurs ou être aidé par ceux-ci.

Les points négatifs sont les suivants : vous pouvez être copié, votre idée devient publique.

Voici un des exemples d'alternative à github :

Local : Git , Gogs

Online : Gitlab, AWS CodeCommit, Google Cloud Source Repositories

Dans le cadre de github, celui-ci vous donne la procédure, vous crée un dossier appelé repos avec les droits et options souhaités, vous aurez une suite de ligne de commande d'exemple pour mettre en ligne vos premiers fichiers (*Annexe 1*).

Une fois réalisés, vous pouvez commencer votre dossier commun nécessaire aux développeurs (*Annexe 2*).

STRATÉGIE

Une fois le cadre de la partie code mise en place, il faut être d'accord sur qui fait quoi et sur ce qui est à faire.

Ce cadrage du déroulement des tâches et de l'assignation se basera sur un concept d'agilité, et plus spécifiquement Kanban.

Les missions seront donc décomposées dans les rôles suivants :

- Un product Owner (PO)
- Un Scrum Master (SM)
- L'équipe

Au vu des contraintes je serais les 3 rôles à la fois.

L'objectif de l'application de ce genre de concept est d'éviter une saturation de l'équipe appelée Crunch dans le monde du développement.

Le crunch résulte des contraintes suivantes : un rendu final défini, avec une barrière financière et un temps imposé.

La stratégie est donc d'agrégérer la seule variable sur lequel on peut travailler : le rendu final.

Le PO assure qu'un livrable sera rendu, il va maximiser la valeur ajoutée en triant les demandes de son clients, le but étant de satisfaire au mieux le client tout en respectant les délais.

Les tâches sont appelées backlogs et celle-ci peuvent disposer d'un poids, d'une priorité et d'un temps impartis.

Le Scrum Master met en place des points d'équipe pour faire éventuellement remonter les problèmes non envisagés en amont par le PO, trouver des solutions au sein des équipes et faire remonter les éventuels dysfonctionnements.

L'équipe tire éléments du backlogs et s'assigne la tâche en fonction de ses compétences et de la prioritisation.

Le client n'aura pas le produit de ses rêves mais le produit avancera de façon continue et permettra une transparence au niveau du client qui pourra affiner ses demandes et apporter les éventuelles corrections / nouvelles demandes.

L'outil pour appliquer cette stratégie sera Trello, un outil qui dispose d'une version gratuite, et une payante destinée à certaines fonctions et aux entreprises.

La création d'un compte est rapide et le site dispose de template selon votre objectif.

Pour le cas du projet Foodies l'annexe 3 met en avant que celui-ci n'a plus qu'à être ajusté.

DÉFINITION DU PROJET

Le projet est issu d'un besoin, ce besoin est communiquer soit en interne, soit en externe par un client.

Cette demande sera exprimée par un client fictif dans le cadre du projet.

Voici donc une synthèse de ce que souhaite le client :

Le client souhaite déterminer la faisabilité d'une application mêlant information et santé.
A partir d'un prototype il souhaiterait avoir un aperçu rapide des possibilités en la matière.

- L'application pourrait avoir des petits outils gadgets tels qu' un calculateur d'imc.
- L'application doit pouvoir reconnaître une image dans un périmètre à définir.
- L'application devrait pouvoir indiquer les informations nutritives du produits.
- Le client imagine une interface compacte et intuitive.
- L'application pourrait être interactive.
- Le client n'a pas de donnée à proposer, il s'agit d'un concept.

Traduction en tant que Product Owner :

Must :

La plus forte valeur ajoutée se traduit sur la démonstration d'une reconnaissance d'image.

Trouver un substitut à la donnée que n'amène pas le client

Should :

L'application devrait pouvoir indiquer les informations nutritives du produits.

L'application devra avoir une interface intuitive et compacte

Could :

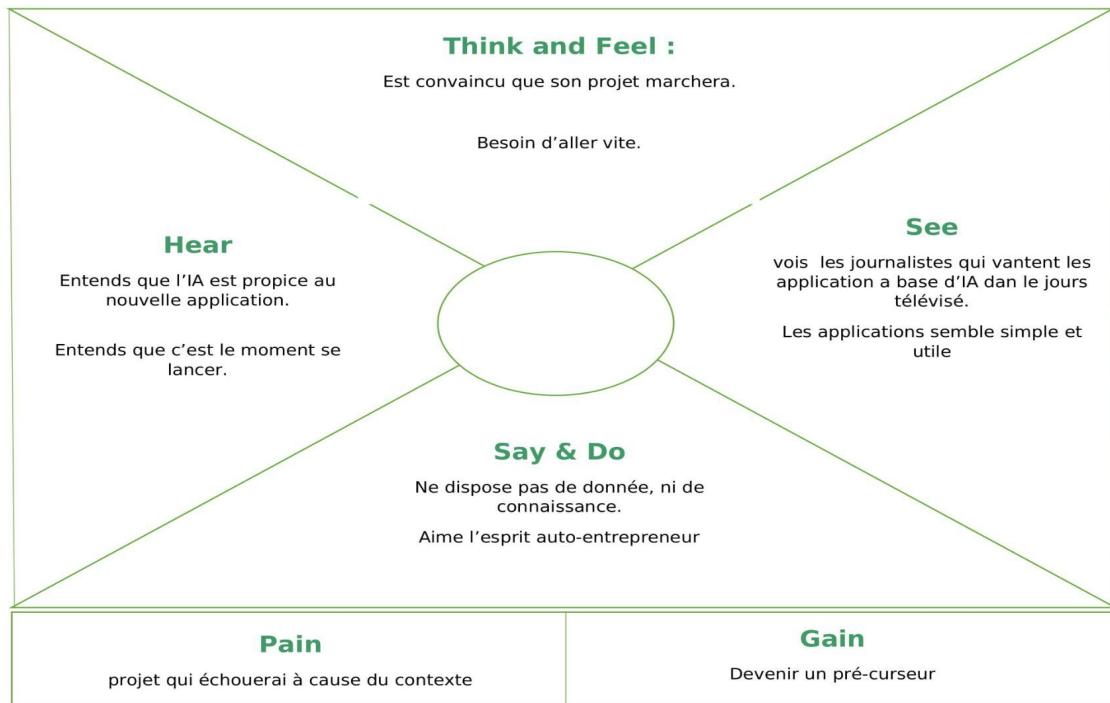
Elargissement du périmètre de la reconnaissance

Calculateur d'imc

Shouln't :

L'application pourrait être interactive.

L'expression du client est également assortie de l'établissement d'une empathie map.
Ce document peut s'établir entre chaque rush.



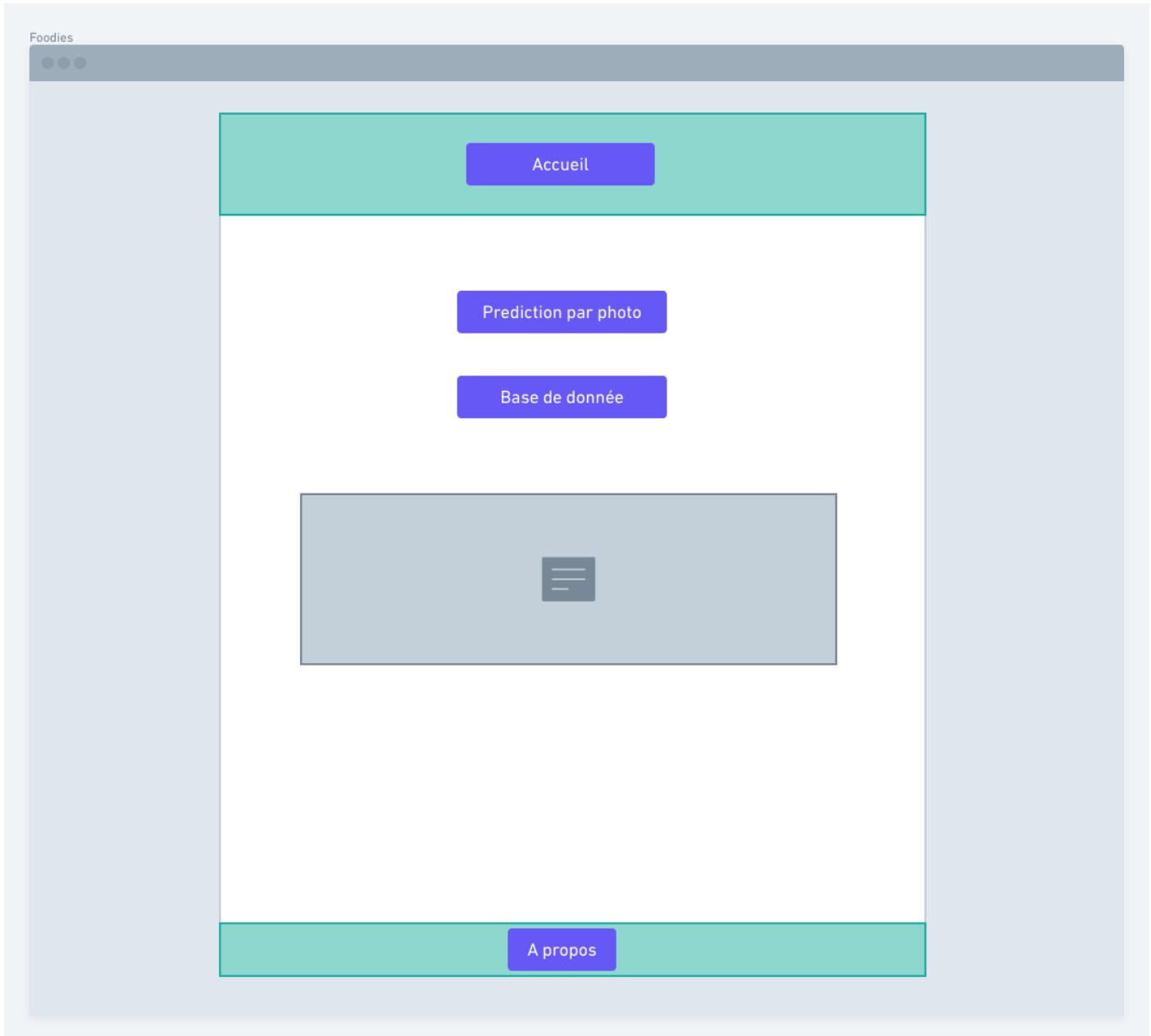
L'application sera réalisée avec une optique suggérée par un intervenant Ux designer dans la formation, celle-ci implique une utilisation libre de l'application sur un principe de fidélisation de l'utilisateur.

L'optique de fidélisation implique que l'enregistrement devrait donner accès à des bonus, les bonus ne seront pas réalisés durant la phase de prototypage.

Le bandeau inférieur et le footer seront limités au minimum pour les mêmes raisons.

Premier Rush

L'utilisation doit être simple, avec une interface minimalist



Le premier design du devra tenir compte du template, qui ce veut tenir compte de l'acquisition de la base donnée.

Il tient compte également d'une fonction : calcul de l'imc.

Cette fonction à un gain minime sur l'application mais représente une très légère charge de travail.

Le projet s'initie avec une version, le développement pouvant mettre du temps et le produit devant perdurer, il faut figer les ressources utilisées.

Cet environnement peut se faire localement, mais également avec un container, ce qui équivaut à une machine virtuelle.

Le développement peut se faire en production à l'aide de l'outil Docker.

Tout commence avec un fichier appelé Dockerfile :

```
duck> dockerfile > ...
1  # syntax=docker/dockerfile:1
2 |
3  FROM python:3.9-slim
4  LABEL version ="0.0"
5
6  ENV FLASK_APP=app.py
7  ENV FLASK_RUN_HOST=0.0.0.0
8
9
10 RUN pip install --upgrade pip
11
12 WORKDIR /opt/app
13 COPY requirements.txt /opt/app
14 RUN pip install -r requirements.txt
15
16 WORKDIR /foodies
17 COPY . .
18
19 EXPOSE 5000
20
21 CMD ["flask", "run"]
```

Le From stipule toujours l'image, on peut utiliser des fonctions ou se crée un ubuntu.

Pour du Flask j'ai opté pour la dernière mise à jour connue de python dans sa version allégé.

les versions et leurs particularité sont disponible sur le site source de docker : <https://hub.docker.com/>

La label définit le versionning de notre container actuellement le container est dans son image la plus récente et la plus initial la 0.0

Cette partie manuel est donc définie par le créateur qui met l'image, il peut y indiquer une version DEV, ou une production.

La numérotation est la suivante :

1.00 version dite de lancement d'un logiciel dans sa version stable.

< 1 alpha / beta

0.0X : évolution ou correctif mineur, ce qui implique un faible risque d'incompatibilité

0.X0 : évolution ou correctif majeur, ce qui implique un risque probable d'incompatibilité

X.00 : Mise à jours critiques implique des risques d'incompatibilité

Le but est donc de faire évoluer notre application, tout en effectuant les mises à jour et les correctifs des dépendances nécessaires.

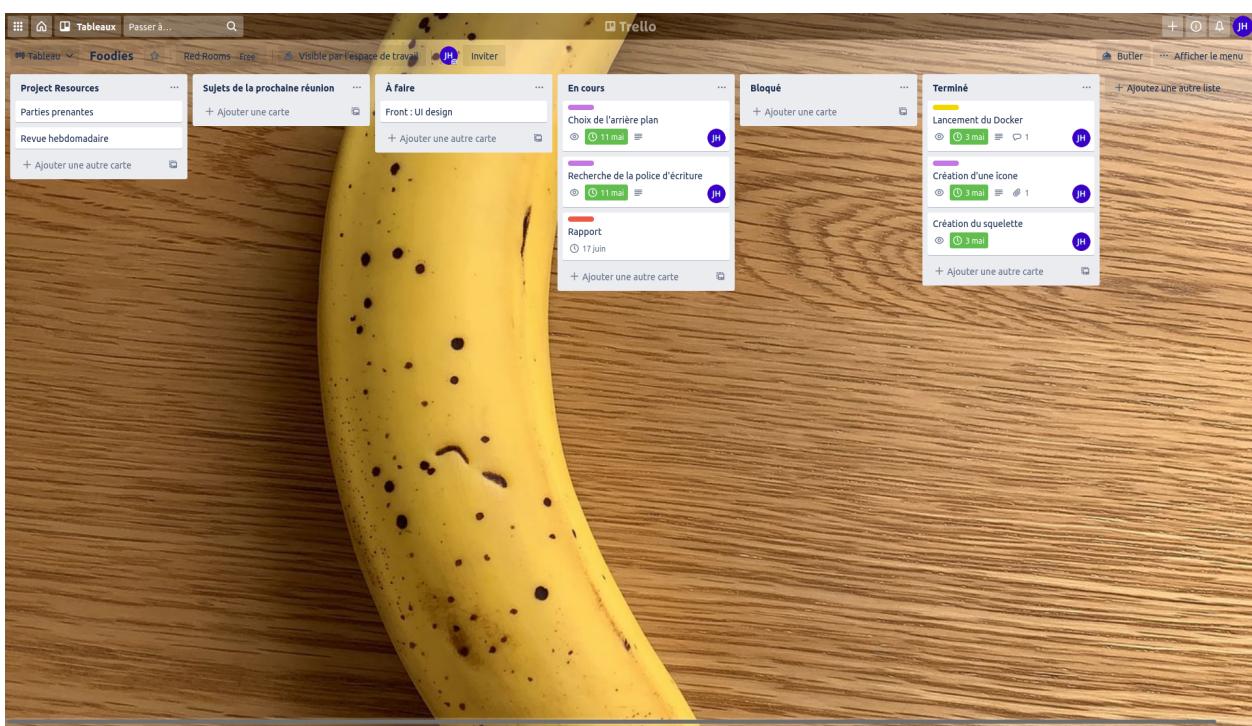
Les dépendances se trouvent dans le fichier requirements.txt

actuellement il n'y a que Flask grâce à la ligne : flask==1.1.2

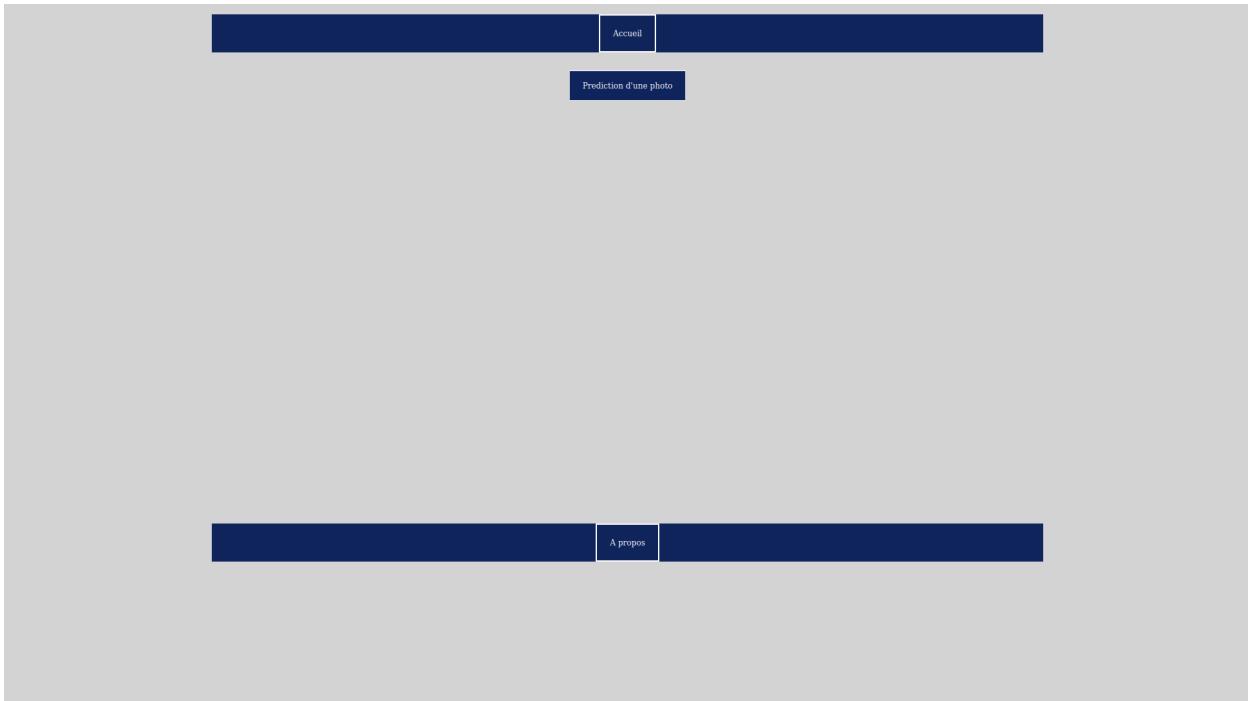
En l'état Flask à sortie une mise à jour majeure durant la réalisation du projet, la gestion de l'environnement permet une stabilité.

Après la partie code, il faut également un squelette qui servira de base entre la partie design et le développement du back, à savoir le fonctionnement de l'application pour ce faire le trello dispose d'étiquette pour répartir les rôles en fonction des compétences.

Niveau Trello l'avancement est le suivant :



Voici le squelette :



La recherche d'écriture pour l'application fut réalisés en parallèle que le squelette, la police retenu est le garamond, l'utilisation sur cette écriture implique pour citer l'article de la recette du web ci-joint en source :

Voici ce qui a été constaté en matière de lisibilité :

- La police Garamond (serif) a obtenu 67%,
- La police Times New Roman (serif) a obtenu 32%,
- La police Helvetica (sans-serif) similaire à Arial a obtenu 12%.

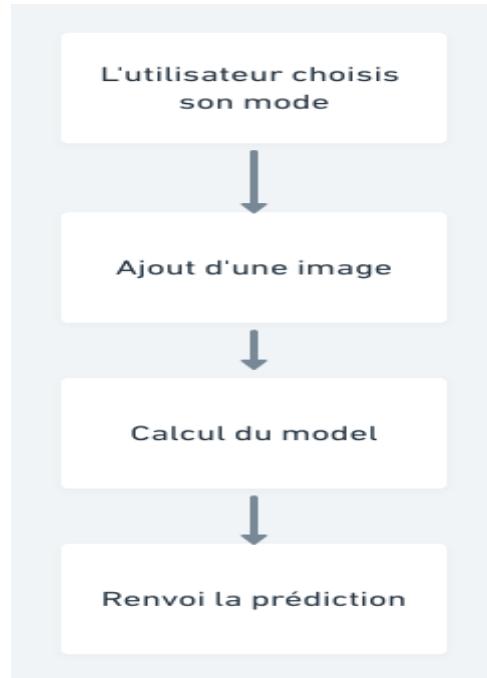
Les polices les plus faciles à lire pour les textes sont les polices à empattement.

Cette donnée fut croisée avec un article de 99 design disponible également en source.

La police n'étant pas native, il faut la mettre en liaison dans le CSS :

```
@font-face {  
    font-family: garamond;  
    src: url('garamond.ttf');  
}
```

Le working flow est le suivant :



Le code du squelette est le suivant :

```
1  #!/usr/bin/python3
2
3  #Import
4  from flask import Flask, render_template
5
6
7  #Definition APP
8  app = Flask(__name__, static_url_path='/static')
9
10 #Route definition
11 @app.route('/', methods=['GET', 'POST'])
12 def home():
13     return render_template('home.html')
14
15 @app.route('/prediction-picture', methods=['GET', 'POST'])
16 def prediction_photo():
17     return render_template('prediction-photo.html')
18
19 @app.route('/about')
20 def about():
21     return render_template('about.html')
22
23
24
25 #Launch Flask
26 if __name__ == "__main__":
27     app.run(debug=True, host='0.0.0.0')
```

Etat de l'art - Computer Vision (E3)

Présentation

Qu'est ce qu'un ordinateur ?

Un ordinateur est un outil inventé à diverses fins, stockages, ludiques, professionnels, et de divertissement.

Le noyau est l'unité de calcul, celui-ci s'appelle processeur, on communique avec lui grâce au noyau qui est le centre même d'un système d'exploitation.

Le noyau ne fait qu'exécuter une suite de code dans un langage compréhensible par celui-ci.

Cette communication s'interprète à travers un langage, celui-ci s'effectue à travers du code et ceux-ci peuvent être saisis en direct, ou par l'intermédiaire d'application.

Les différents langages qui existent ont tous leurs spécificités, voici une liste exhaustive : C++, Java, Python et bien d'autres.

L'ordinateur dispose d'une interface graphique, celui-ci restitue donc ce qu'un programme lui demande, il peut restituer des données, des images, des sons, les 2 combinées avec les vidéos. Ce sont des encodages qui permettent les restitutions de ceux-ci, dans les faits, celui-ci effectue simplement la demande de restitution.

Qu'est ce qu'une image ?

Une image se décrit par une suite de donnée numérique sur un axe x,y allant de 0 à 1 pour déterminer un nuancier de colorimétrie grise.

Elle peut être accompagnée de 2 axes complémentaires venant enrichir le rendu visuel, l'axe de colorimétrie RVB ou RGB en anglais, et un quatrième axe qui détermine la luminosité appeler Alpha.

L'ordinateur voit donc une suite de caractères binaire au format array constituant l'image, avec une colorimétrie et les formes sont donc un nuancier de nombre qui de part un contraste élevé peut refléter la présence ou non d'objet ou personne.

Problématique :

Un processeur de par sa nature calcul, il faut donc essayer de représenter mathématiquement une image.

Cas courants

Deux types de solutions sont envisageable :

La première est une approche de sollicitation, on utilisera un panel d'offres celle-ci se retrouve auprès des 3 grands prestataires dans le domaine de la donnée : amazon, google et microsoft.

Les solutions peuvent être clé en main et ne requiert pas forcément la compétence technique et ou mathématique.

Il existe de multiples solutions dans le domaine du cloud computing que ce soit:

- Amazon (AWS) - AMAZON RECOGNITION IMAGE
- Microsoft(Azure) - IMAGE RECOGNITION SERVICE
- Google - VISION AI

Les 3 propose une approche en mode gratuit, et un en mode payant, il s'agit de volumétrie et de transaction / token.

Dans tous les cas le prix se compose de trois segments : entre 1 à 2 € toutes les X requêtes utilisant token / transaction.

entre 10 à 20 euros l'heure de calcul selon où se trouve le serveur et toutes les requêtes brutes d'utilisations de l'api avec entre 0,30 à 0,60 € les 1000 images.

Cette approche permet d'obtenir une production mise à jour de façon automatique.

Elle est efficace avec très peu de données, et peut se réaliser en l'absence d'un technicien permanent dans les effectifs.

La solution par création du modèle :

Les solutions qui commencent à 0 ou tout se construit sont réalisables, elles impliquent des infrastructures supplémentaires, des compétences pour établir le modèle, et pour le maintenir ou le mettre en production.

Tout d'abord avant tout développement, il faut se constituer une base d'image labellisé, de nombreuses images sont disponibles sur internet, et des sets de données sont mis à disposition.

Celles-ci peuvent être libres de droit, mis à disposition pour des concours.

Selon l'objectif que doit remplir le modèle, la volumétrie de la donnée peut varier, certains modèles peuvent être viable avec des centaines d'images, d'autres des milliers, cela dépend principalement du nombre de composantes et de variables qui peuvent être attendues.

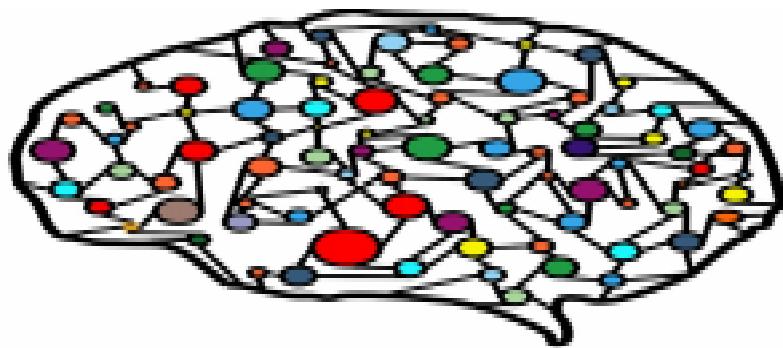
Cette donnée labellisée sert d'apprentissage, il faut donc réaliser une approche d'apprentissage à la machine.

Cet apprentissage peut s'effectuer à travers du machine learning ou du deep learning.

Les principaux travaux s'axent sur le deep learning :

- La librairie Pytorch est la plus en vogue pour l'année 2020 suite à sa flexibilité, elle commence à se mettre en valeur pour des groupes tels que Disney.
- La librairie Tensorflow reste la plus utilisée par les professionnels et elle est produite par Google.

Le deep learning fonctionne sur la réplication du fonctionnement du cerveau humain dans la machine.



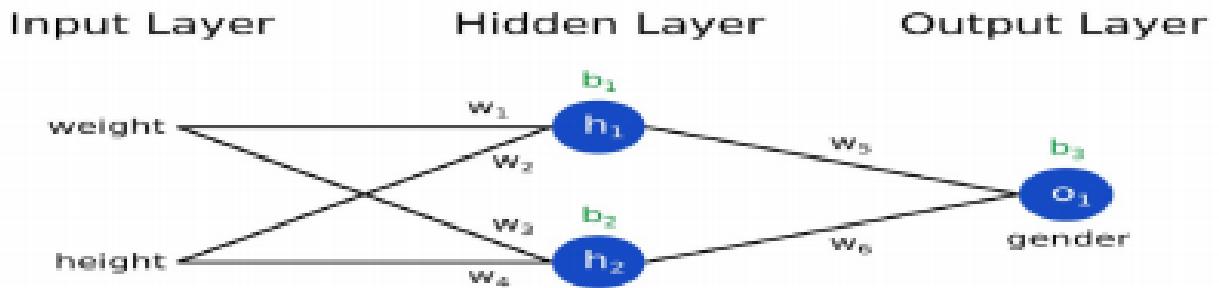
Ce processus de réplication s'appelle un **réseau de neurones artificiels**.

Celui-ci fut créé par M. McCulloch et M. Pitts (2) dans leur thèse de 1960, permettant mathématiquement et à travers des algorithmes de se rapprocher du fonctionnement de notre cerveau en simulant le fonctionnement de ses neurones.

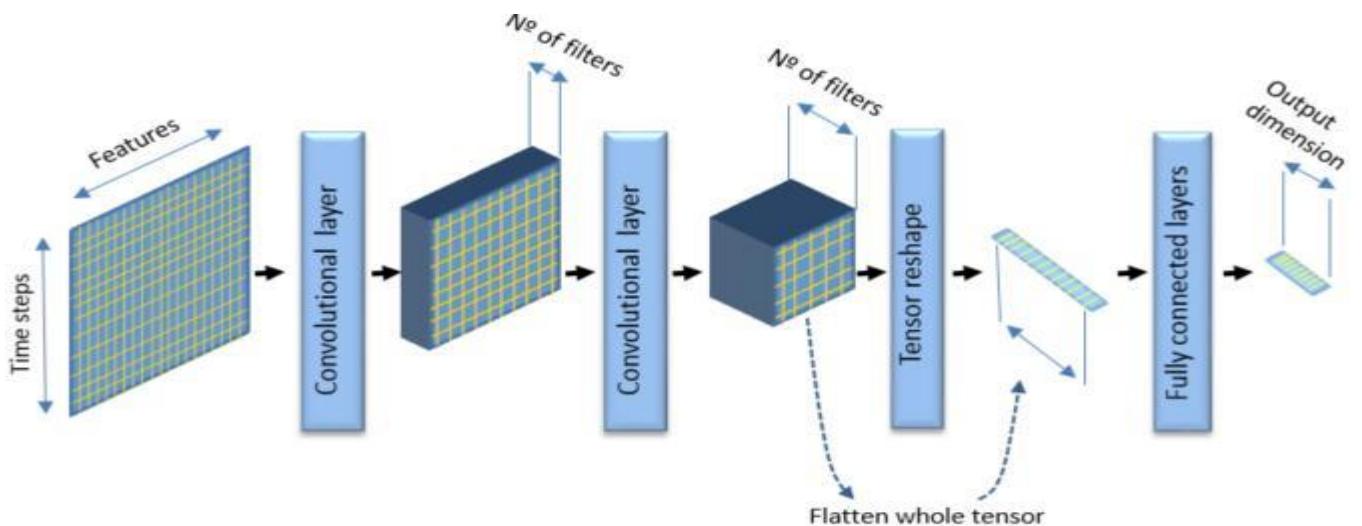
L'avantage est que celui-ci ne rentre pas dans le cadre d'un processus supervisé, suivant des lois mathématiques fixes, chaque neurone pour obtenir la réponse souhaitée, va chercher à appliquer des lois et méthodes mathématiques permettant de converger.

Cette convergence s'analyse de par les paramètres de précisions et de la perte.

Les composantes déterminent les caractéristiques d'entrée dans le neurone, voici une illustration sur un neurone avec 2 composantes (crédit Victor Zhou) :



La représentation d'une image peut se traduire comme cela visuellement une image plate, se transforme en multiple facette selon le nombre de filtre retenue, colorimétrie et alpha :



Synthèse technologique

Dans le monde professionnel, le constat présenté en 2015 par la présentatrice Fei Fei Li(4) qui a fait de la reconnaissance d'images sa spécialité, démontre que par rapport au fonctionnement du cerveau, les meilleurs modèles simulent le raisonnement d'un enfant de 4 ans malgré de nombreuses erreurs.

A 4 ans nous sommes capables d'effectuer une phrase simple de contexte en tenant compte de l'analyse des éléments principaux constituant une image.

En dehors de la levée de fonds, conséquente levée pour le projet Open source de labellisation d'une base de données de l'organisme de Mme Li, des géants de l'imageries entraînent des IA à répliquer les images et leurs formes pour alléger les performances et les bandes passantes destinées au consommateur de contenu au format 4K pour les jeux et les films.

Ce qui démontre un engouement global des diverses parties pour cette technologie. Dans l'état du projet nous n'aurons pas besoin d'une phrase descriptive, mais uniquement d'une réponse verbale pour identifier, les outils conventionnels répondront donc à l'objectif du projet.

E2 - Insertion d'I.A dans une application

Prototype

Pour la base de données d'image et pour le premier jet de l'application, la base populaire fruit-360 constitue plusieurs échantillons de plusieurs fruits.

Les échantillons sont déjà pré-traiter au format 100*100, l'avantage de ce type d'image est qu'elle facilite l'apprentissage de par leurs natures elles sont prétraitées et sont extraites de tout élément pouvant perturber l'apprentissage, de plus cela s'appuient sur des traits distinctifs.

En conséquent, une résolution inférieur pour l'oeil humain, permets de distinguer les composantes majoritaires définissant l'élément, cela peut être une alternative de couleur, une forme géométrique et ainsi de suite.

Pour le prototype voici le socle large des outils Keras :

```
import tensorflow.keras as keras
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.losses import SparseCategoricalCrossentropy
```

Pour le premier jet, j'ai utilisé la fonction `image dataset_from_directory`, cette fonction utilise l'arborescence de 2 dossiers, sous un format précis, entraînement et test qui contiennent la même structure.

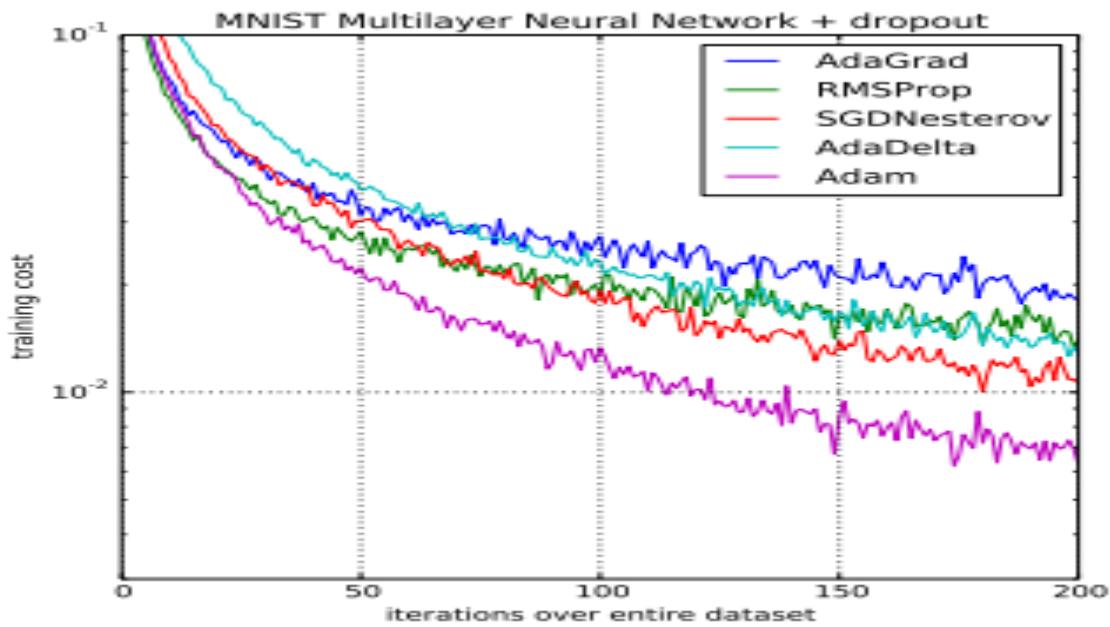
Cette fonction travaille directement sur la donnée avec un traitement de l'image en paramétrage utilisateur, cependant elle dépend énormément de la volumétrie de la donnée.

Dans tous les cas, l'outil keras attribue une note à son environnement pour une utilisation CPU (longue) ou GPU (rapide, intérêt principal de keras), l'environnement est évalué à 3.0 et keras requiert 3.1

Voici les choix de paramètres :

- La colorimétrie est maintenue avec le paramètre RGB.
- La taille du set est limitée à 100 * 100
- mélange de la donnée pour l'entraînement
- méthode de classement : par catégorie au vu de l'optique du projet
- l'infrastructure ne dispose pas d'un ordinateur permettant les calculs sur le GPU de par l'obsolescence de celui-ci en conséquent les traitements seront effectués pour converger rapidement vers un résultat.
- optimiser : "adam" : les avis internets recommandent de prendre systématiquement le plus récent suite au

- performance sur la convergence.
- perte : categorical crossentropy , il s'agit d'une corrélation avec la méthode



extrait du site medium : <https://medium.com/mdr-inc/from-sgd-to-adam-c9fce513c4bb>

Le modèle étant sorti avec une précision sur test de 50 % est ayant échoué tous les tests des pommes.

Ce cas s'observe de la perte d'apprentissage élevé sur un nombre successif d'epoch et une précision élevée, qui ne correspond pas à la précision sur les tests validants.

Les problèmes générant ceci, peuvent être une volumétrie de donnée faible, ou un sur apprentissage.

Pour compenser le manque de donnée il est possible en premier temps d'en générer de façon artificiel, il s'agit du preprocessing appeler ImageDataGenerator, il s'agit d'un module qui va générer des images en fonction d'un paramètre de, distorsion, décalage, retournement de l'image.

```
train_datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    vertical_flip= True)

test_datagen = ImageDataGenerator(
    horizontal_flip=False)
```

Pour l'exemple d'une pomme sa forme n'est pas tout à fait ronde, et peut disposer d'une tige, il faut prévoir durant l'entraînement que le modèle prévoit ce genre de situation, les images standards peuvent être donc multiplier par 2-3-4.

Après les essais le modèle dispose avec une perte convenable, et une précision sur la validation de l'ordre de 60 % sur un tirage d'une chance sur 2 et 2 tests sur 6 réussis c'est mieux que l'aléatoire en terme de maïs les bananes ne répondent pas.

corpus des paramètres flow_from_directory :

```
path="/media/jhy/46AE-6494/DB/img/food/fruits-360/"

directory = path + 'Training-light'
Train = train_datagen.flow_from_directory(
    directory= directory,
    target_size=(100, 100),
    color_mode="rgb",
    batch_size=32,
    class_mode="categorical",
    shuffle=True,
)

directory = path + 'Test-light'
Test = test_datagen.flow_from_directory(
    directory= directory,
    target_size=(100, 100),
    color_mode="rgb",
    batch_size=32,
    class_mode="categorical",
    shuffle=False,
)

Found 982 images belonging to 2 classes.
Found 330 images belonging to 2 classes.
```

La fonction issue du générateur permet de récupérer sous formes de dictionnaires les clés des valeurs, par exemple la clé Apple red 1 correspond à la valeur 0, la prédiction étant une valeur, il s'agit d'itérer pour récupérer la valeur de la classe qui nous parlera.

```

▷ ▶ M4
label_map = (Train.class_indices)

▷ ▶ M4
print(label_map)
{'Apple Red 1': 0, 'Banana': 1}

```

De par les performances MobileNetV2 est optimisé pour les ordinateurs à faibles performances, contrairement à la VGG16. Celui est lancé avec le paramètre d'input le plus proche du format des images sur un premier jet de donnée.

```

# Load the pretrained model
pretrained_model = keras.applications.MobileNetV2(
    input_shape=(96, 96, 3),
    include_top=False,
    weights='imagenet',
    pooling='avg'
)

```

Le modèle consiste à récupérer les éléments de sortie et d'entrée, en ajoutant la couche standard de 128 éléments qui est la valeur d'entrée cohérente avec le MobileNetV2. La valeur de sortie correspond à nos données Foods alléger sur 2 possibilités : Pomme rouge, Banane et le nombre de sortie attendu, pour un mode qui vérifie que cela soit un objet ou l'autre cela correspond à 2 possibilités.

```

inputs = pretrained_model.input

x = keras.layers.Dense(128, activation='relu')(pretrained_model.output)
x = keras.layers.Dense(128, activation='relu')(x)

outputs = keras.layers.Dense(2, activation='softmax')(x)

model = keras.Model(inputs=inputs, outputs=outputs)

model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

model.summary()

```

La variable history permet de restituer graphiquement si cela est nécessaire pour distinguer si le modèle

progresse ou régresse au cours de l'entraînement.

```
history = model.fit(  
    Train,  
    validation_data=Test,  
    epochs = 5)
```

Voici le résultat de la dernière époche déterminant la réel précision et la val_loss pouvant encore être améliorer.

```
loss: 0.0040 - accuracy: 0.9980 - val loss: 1.4058 - val accuracy: 0.8485
```

Voici la fonction ajouter dans le squelette pour faire la prédiction, en version ébauche.

```
def prediction(image, size = (96,96)):  
    """  
    img : path to file  
    size :  
    weight : weight of model  
    height : height of model  
    """  
    from tensorflow import keras  
  
    image_for_keras = keras.preprocessing.image.load_img(image, target_size=size)  
    image_for_keras = keras.preprocessing.image.img_to_array(image_for_keras)  
  
    import numpy as np  
    image_for_keras = np.expand_dims(image_for_keras, axis = 0)  
  
    model = keras.models.load_model('models/model-prototype')  
    prediction = model.predict(image_for_keras)  
  
    y_classes = prediction.argmax(axis=-1)  
  
    if y_classes == 0 :  
        result = "Pomme"  
    elif y_classes == 1:  
        result = "Banane"  
    else :  
        result = "erreur"  
  
    return result
```

L'image de l'utilisateur est stockée avec cette fonction :

```
def saves_pictures(UPLOAD_FOLDER,image,app,users = False):
    import os
    from datetime import datetime
    current_time = datetime.now().strftime("%y-%m-%d_%H-%M-%S")
    #stockage - image

    if users is not False :
        save_path = UPLOAD_FOLDER+ "/" + users

        if not os.path.exists(save_path):
            os.makedirs(save_path)

        stockage = save_path +"/" +str(current_time)
    else :
        save_path = UPLOAD_FOLDER+ "/"
        stockage = save_path +"/" +str(current_time)

    # save local
    filename = stockage + ".jpg"
    image.save(os.path.join(app.root_path, filename))
    return filename
```

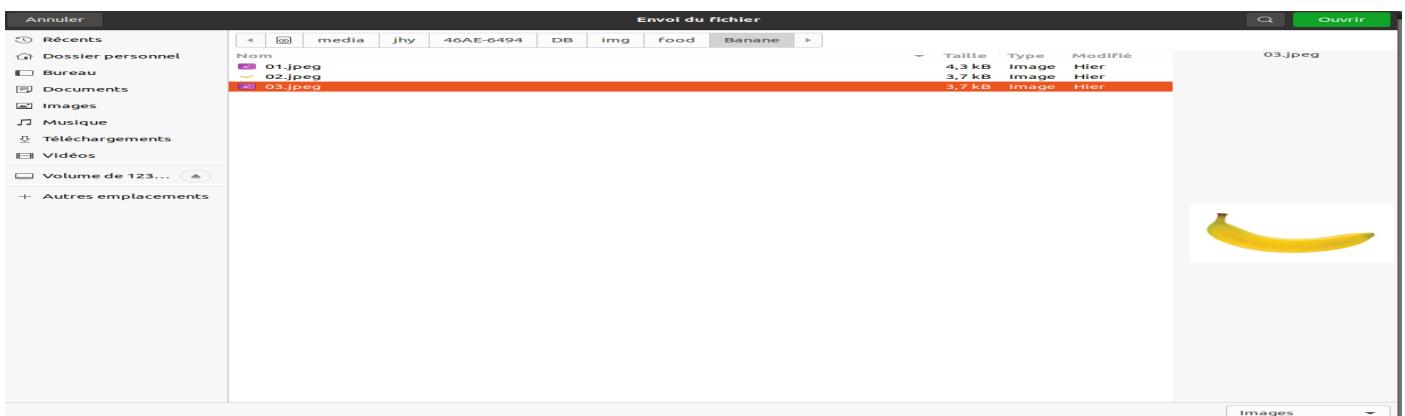
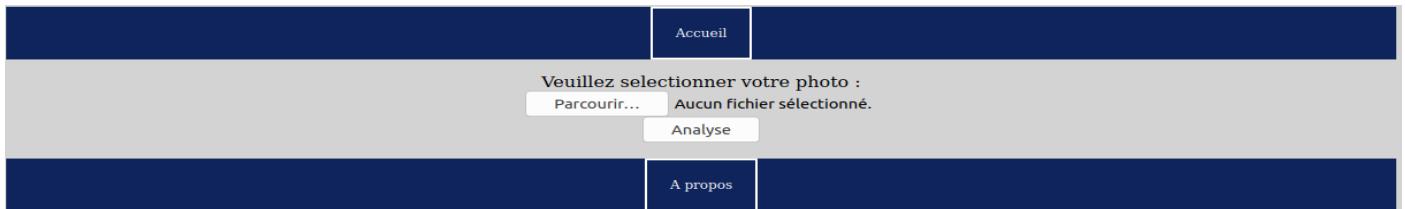
Comme vu en amont, il faut transformer chaque pixel dans un tableau array qui récapitule la donnée au format numérique pour que celle-ci soit analysée dans un modèle. Cette transformation nécessite une image, en conséquent l'utilisation de la donnée directement depuis le formulaire n'était pas compatible..

Il a été imaginé de stocker les photos sous SQL, cependant d'un point de vue performance est accordement à ce post très approuver sur stack overflow (voir annexe),, le modèle se définit sur une base pré-traiter avec des images de faible résolution, il n'y a pas de filtre prévu sur la photo prévue à tester.

Une fonction keras img_to_array, effectue cette conversion, la fonction numpy ajuste le format array avec le nombre de dimensions nécessaire à l'usage du modèle. Pour le stockage d'une image au meilleur format, le nosql semble être la meilleure solution, et pour l'instant elle reste hypothétique et surtout non nécessaire à l'établissement du prototype.

On charge le modèle précédemment établi, et de manière temporaire on indique le résultat traduit pour l'utilisateur.

Présentation du prototype au client avec démonstration :



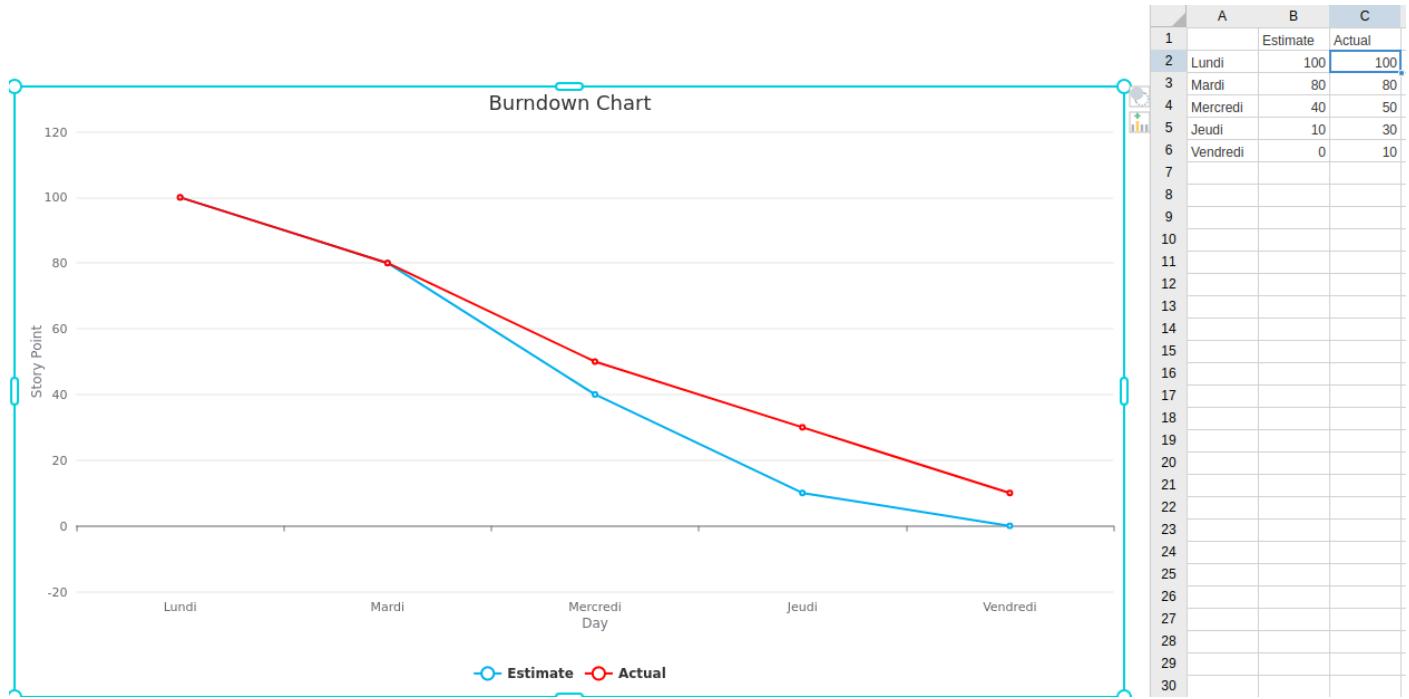
Le sprint étant finis, la démonstration effectuée, voici le compte rendu du client :

Le client semble satisfait, cela fonctionne, il n'a pas obtenu toutes les fonctions qu'il souhaite mais le prototype dans sa phrase squelette réagit plutôt bien et réussit les tests.

Le client attend désormais que la partie "should" soit engagée, avec une refonte esthétique dans la mesure du possible.

Le client rajoute une demande pour l'application en should (impératif de formation), la création d'une base de données utilisateur ou alimentaire.

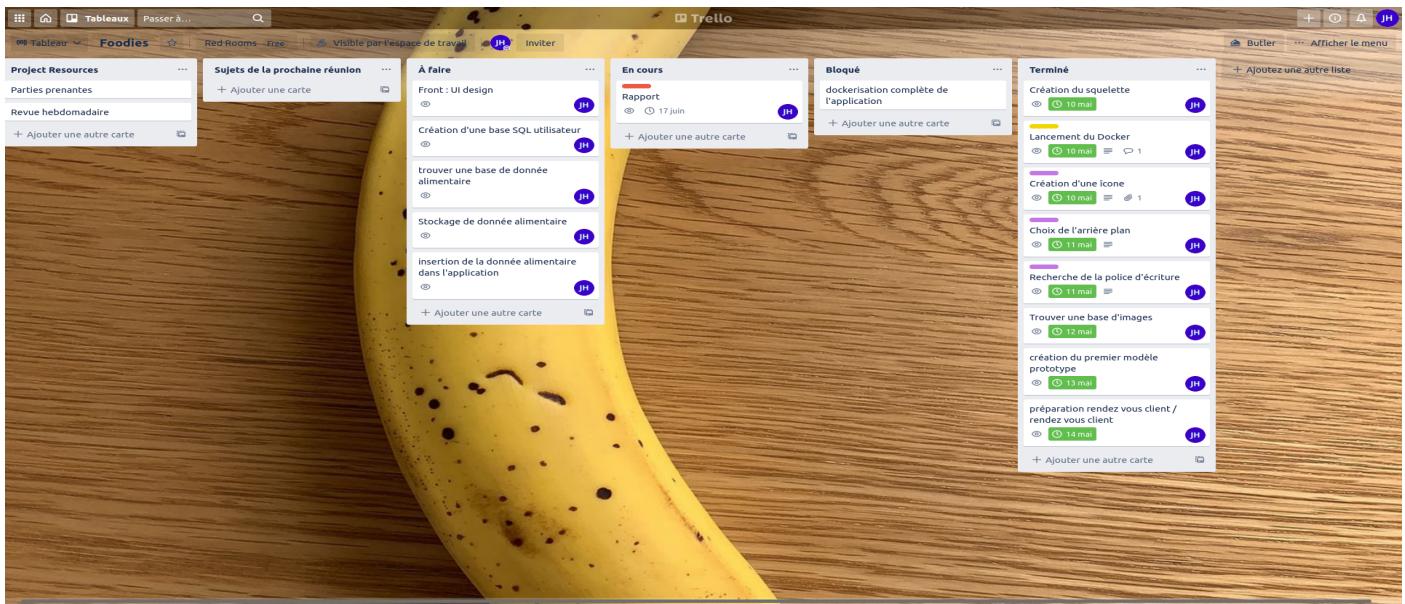
Point de vue du PO



Les soucis de dockerisation ont pesé sur le développement dans la semaine.

La partie design est squelettique, les fonctions shoulds n'étaient pas prévues pour ce run, mais la partie production n'est pas sortie en conséquent le run n'est pas boucler, cependant la majorité de la valeur ajouté fut créé.

Voici l'évolution du trello, les backlogs sont apparus, le problème de docker et la liste des tâches déjà effectuées.



Les données alimentaires ont été trouvées, il s'agit de la base de données publique ciqual, celle-ci est disponible avec un csv, le prototype va être brancher dessus.

Voici le branchement de la fonction local :

```
def request_food_local(recherche = False, flask = False, type_pred = "fruits"):  
    """  
    recherche : False for all df : True for Only name, str for specific research  
    """  
    from pandas import read_excel  
    df = read_excel("ressources/data/Table Ciqual 2020_FR_2020 07 07.xls", header = 0)  
  
    df = df[df['alim_ssgrp_nom_fr']== type_pred]  
  
    cols_id = ["alim_grp_code","alim_ssgrp_code","alim_ssssgroup_code"]  
    to_drop = cols_id  
    df = df.drop(columns = to_drop)  
  
    if recherche is False :  
        return df  
    elif recherche is True :  
        return df['alim_nom_fr']  
    else :  
        if flask is True :  
            # Result = df[df['alim_nom_fr'].str.contains(recherche)].transpose() #  
            terms = recherche.split()  
            Result = df[df['alim_nom_fr'].str.contains('|'.join(terms))].transpose()  
            columns = Result.columns.values  
  
            row = [Result.to_html(classes='data')]  
            return columns, row  
        else :  
            return ""
```

Voici le rapport du modèle après 1 jours de compilation :

```
1220s 577ms/step - loss: 0.0557 - accuracy: 0.9855 - val loss: 1.4644 - val accuracy: 0.7641
```

Les données varient entre 70 à 80 % accuracy, mais 0 à 2 sur 5 tests unitaires ne sont pas validés, ils retournent des résultats tels que : Banane pour Pomme, Cerise pour Pomme, Pomelo pour banane et bien d'autres.

Les paramètres suivants ont été ajustés : Batch, Size du MobileNet.

Après lecture de différents rapport présentés en annexe, présenter la problématique d'un batch size à ajuster, un batch élevé avait tendance à rendre le modèle moins précis en % et moins précis dans les tests unitaires.

Si celui-ci est trop faible cela à la même tendance, une imprécision, le compromis est de trouver le juste milieu, le modèle doit trouver rapidement une réponse simple aux données qui lui sont soumises, pour en extraire les features essentiels, mais la quantité de celle-ci doivent être suffisamment volumineuse pour distinguer

Création de la table SQL sous docker:

Pull d'image sql:latest

```
sudo docker run -u root -e MYSQL_ROOT_PASSWORD=123 -d -p 127.0.0.1:3307:3306 mysql:latest
```

```
jhy@Red:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
314bce7e9aaa        mysql:latest       "docker-entrypoint.s..."   10 minutes ago   Up 10 minutes   3306/tcp, 127.0.0.1:3308->3306/tcp
NAMES
kind_dewdney
```

Création de la fonction pour créer la table à distance (remote) :

```
import mysql.connector

""" # local
mydb = mysql.connector.connect(
    host="localhost"
, user="root"
, password="Umbrell@2325"
, database = "FOOD_ALPHA"
)
"""

try : #Docker configured
    mydb = mysql.connector.connect(
        host="127.0.0.1"
, port = "3308"
, user="root"
, password="123"
, database = "Foodies"
)
except : #docker base
    mydb = mysql.connector.connect(
        host="127.0.0.1"
, port = "3308"
, user="root"
, password="123"
    )

def create_table_users():
    mycursor = mydb.cursor()

    request_init = "CREATE TABLE USERS "

    ID_USERS = "ID_USERS INT PRIMARY KEY NOT NULL AUTO_INCREMENT"
    CREATION = "CREATION DATETIME DEFAULT (CURRENT_TIMESTAMP()) "
    E_MAIL = "E_MAIL VARCHAR(25) UNIQUE "
    TEL = "TEL VARCHAR(10) UNIQUE "
    USERNAME = "USERNAME VARCHAR(25) NOT NULL UNIQUE "
    PASSWORD = "PASSWORD VARCHAR(255) NOT NULL "
    AGE = "AGE INT(3) DEFAULT 0 "
    POIDS = "KG INT(3) DEFAULT 0"
    SEXE = "SEXE VARCHAR(1) DEFAULT 'X' "
    NOM = "NOM VARCHAR(25) "

    request_SQL_LOG      = ID_USERS + "," + CREATION + "," + E_MAIL + "," + TEL + "," + USERNAME + "," + PASSWORD
    request_SQL_APP      = AGE + "," + POIDS + "," + SEXE
    request_SQL_BONUS    = NOM

    request = request_init + "(" + request_SQL_LOG + "," + request_SQL_APP + "," + request_SQL_BONUS + ")"
    mycursor.execute(request)
    mycursor.close()
```

Squelette création utilisateur :

The screenshot shows a user creation form. At the top, there's a dark blue header bar with a "Accueil" button. Below it is a light gray form area with three input fields: "Mail" containing "Elrond@redmon.fe", "Pseudo" containing "Test1", and "Mot de passe" containing "*****". At the bottom right of the form is a "Création" button. At the very bottom of the page, there's a dark blue footer bar with a "A propos" link.

Exemple du hash password sur la création d'un utilisateur avec mots de passe “Elsa”

```
jhy@Red:/media/jhy/46AE-6494/Projet/Foodies$ /usr/bin/python3 /media/jhy/46AE-6494/Projet/Foodies/ressources/SQL_APP.py [(1, datetime.datetime(2021, 5, 15, 16, 30, 16), 'jhyeon@hotmail.fr', None, 'Jhyeon', '4652f63873dd1c36d8912684df39b971', 0, 0, 'X', None)] jhy@Red:/media/jhy/46AE-6494/Projet/Foodies$
```

Sécurisation code :

```
try : #Docker configured
    mydb = mysql.connector.connect(
        host="127.0.0.1"
        , port = "3308"
        , user=secret_data[0].strip()
        , password= secret_data[1].strip()
        , database = secret_data[2].strip()
    )
```

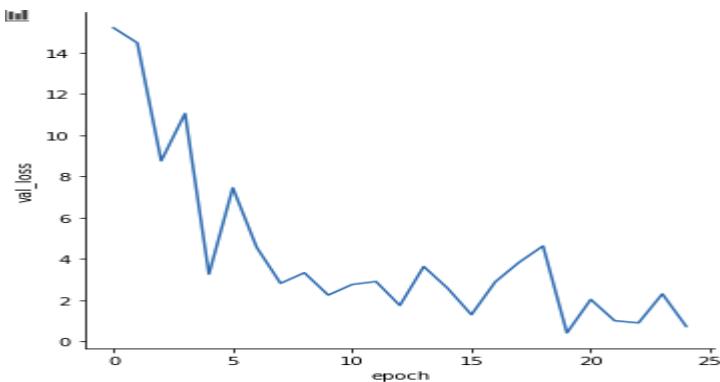
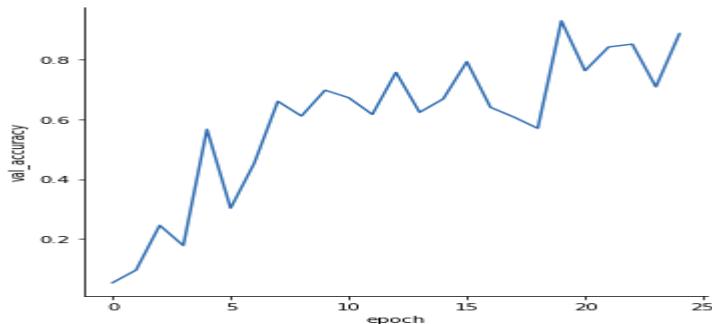
L'application est sécurisée de façon légère, de par la saisie d'un mot de passe dissimulé, d'un point de vue structurel., le mot de passe est hash, ce choix à était fait par simplicité, un mot de passe hash donne toujours le même en conséquent si l'utilisateur tape son mot de passe “Elsa” : la clé de Hash sera systématiquement la même et une éventuelle connexion pourra toujours s'effectuer.

Si l'utilisateur perd son mot de passe il peut tout de même être modifier par après si l'application le conçoit mais on ne pourra pas lui restituer son mot de passe comme le prévoient certains sites webs il y a de nombreuses années.

Un hash est une donnée clé calculée de façon spécifique ce qui fait qu'on peut inversement la recalculer mais celle-ci offre plusieurs combinaisons possibles.

Un encryptage permet de restituer le mot de passe, mais une fois la clé décrypter, le mot de passe apparaît en clair si la base de donnée est compromise, côté gestion de connections un mot de passe telle que “Elsa” donnera de multiple encryptage possible, imposant de décrypter la base de donnée pour faire correspondre la donnée à la saisie utilisateur.

Les représentations historise suivants sont le reflets de la précision, par rapport au nombre d'epoch pour l'accuracy et la loss



Malgré les courbes en progression, la problématique persiste, les tests unitaires ne sont pas cohérent avec les résultats attendus de la précision sur les tests.

Pour déterminer si je pouvais utiliser plus de paramètres sur le modèle pré-entraîné, j'ai découvert que la version 3 était disponible, elle propose une convergence plus rapide et plus efficace, mais de nombreux paramètres ne sont plus disponibles, notamment le paramètre size.

Par défaut il est au maximum de l'ancien modèle soit 224,,224, 3 pour la colorimétrie, le paramètre alpha semble pris en compte mais la dimension ne correspond pas au sortie pour la luminosité, la documentation du modèle prévoit également 2 version, light et large, ceux ci disposent de compte rendu, les modèles large sont plus lourd et dispose d'une charge de travail plus lourde.

Malgré les problématiques de performance de par le fait que comparer au V2 l'ordinateur est moins utilisable de par le ralentissement engendré en cas d'utilisation, les performances de celui-ci justifient l'investissement temps.

Si l'infrastructure était différente avec des moyens financiers, il aurait été possible d'effectuer les calculs dans le cloud, sur des serveurs dédiées, cela aurait représenté un coût modéré au vue de l'utilisation GPU de ce type

d'ordinateur, cependant en cas de réel projet, la confidentialité des données aurait pu être compromise si les données étaient sensibles.

La seconde aurait été d'effectuer les calculs sur un ordinateur dédié.

Modèle : ImageNetV3-Large :

```
2116/2116 [=====] - 2142s 1s/step - loss: 0.0853 - accuracy: 0.9783 - val loss: 0.7572 - val accuracy: 0.8619
```

test unitaire : 3/6, ressemblant 1/6

Le dernier modèle V3 en spécification large, fut relativement satisfaisant, suffisamment pour le laisser dans le prototype, pour une éventuelle démonstration.

Dans l'objectif de fournir de manière progressif l'évolution des modèles les performances en test unitaires n'étant pas idéales, un modèle intermédiaire sur 10 fruits fut créé dans un modèle avec le moins de composantes possibles..

```
Epoch 2/5
242/242 [=====] - 23s 95ms/step - loss: 0.0381 - accuracy: 0.9861 - val_loss: 0.0032 - val_accuracy: 1.0000
Epoch 00002: val_loss improved from 0.02189 to 0.00325, saving model to /media/jhy/46AE-6494/DB/img/food/fruits_base_light/fruits_fresh_cnn_1.h5
Epoch 2/5
```

Dans un souci d'optimisation, une procédure callback fut insérée dans le modèle, celle-ci permet une progression systématique du modèle en ne retenant pas les epochs où celui-ci n'évolue pas.

Résultat de l'entraînement en précision :

Classification Report				
	precision	recall	f1-score	support
Apple	1.00	1.00	1.00	164
Avocado	1.00	1.00	1.00	143
Banana	1.00	1.00	1.00	166
Kiwi	1.00	1.00	1.00	156
Nut Pecan	1.00	1.00	1.00	178
Orange	1.00	1.00	1.00	160
Passion Fruit	1.00	1.00	1.00	166
Pear	1.00	1.00	1.00	164
Pineapple	1.00	1.00	1.00	166
Watermelon	1.00	1.00	1.00	157
accuracy			1.00	1620
macro avg	1.00	1.00	1.00	1620
weighted avg	1.00	1.00	1.00	1620

Conclusion sur la partie de l'implémentation d'une IA

Objectivement, ce modèle obtient le maximum d'accuracy, c'est les tests unitaires qui mettent celui-ci en défaut. Le set Kaggle prévoit des situations spécifiques pour la reconnaissance de certains fruits, absence de tige, absence de feuille, colorimétrique spécifique.

La finalité pour l'obtention d'un résultat plus cohérent avec le parcours d'un utilisateur, implique que le test de validation devrait être moins homogène ou que la donnée d'entraînement et de validations se voit alors incorporer plusieurs sous-famille au sein de son même dossier.

Par exemple, une pomme peut avoir plusieurs couleurs, sa tige ou non, avoir des tâches et ainsi de suite, un ananas peut être avec ou sans sa tige, les fruits peuvent avoir des formes plus variées.

Le set de donnée se constitue du principe suivant : un produit pris sous tous les angles possibles cela se traduit de par un modèle traitera correctement la donnée si la ressemblance était propre à la forme reçue car il n'y a pas assez de versatilité

Il serait envisageable de représenter graphiquement avec un min x,y max x,y pour comprendre les critères que retient le modèle pour déterminer son choix, l'image apparaît avec un encadrement sur la composante discriminantes.

Le modèle fut optimiser à partir d'un modèle local :

```
model = Sequential([
    Conv2D(num_classes * 1, kernel_size=(5,5), activation='relu', input_shape=shape),
    MaxPooling2D(),
    Conv2D(num_classes * 2, kernel_size=(5,5), activation='relu'),
    MaxPooling2D(),
    Dropout(0.20),
    Conv2D(num_classes * 3, kernel_size=(5,5), activation='relu'),
    MaxPooling2D(),
    Dropout(0.30),
    Flatten(),
    Dense(num_classes , activation = "softmax")
])
```

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 96, 96, 10)	760
max_pooling2d (MaxPooling2D)	(None, 48, 48, 10)	0
conv2d_1 (Conv2D)	(None, 44, 44, 20)	5020
max_pooling2d_1 (MaxPooling2D)	(None, 22, 22, 20)	0
dropout (Dropout)	(None, 22, 22, 20)	0
conv2d_2 (Conv2D)	(None, 18, 18, 30)	15030
max_pooling2d_2 (MaxPooling2D)	(None, 9, 9, 30)	0
dropout_1 (Dropout)	(None, 9, 9, 30)	0
flatten (Flatten)	(None, 2430)	0
dense (Dense)	(None, 10)	24310
Total params: 45,120		

Sprint 2

Dans un premier temps la restitution de la table ciqual sous format dataframe, contente le prototype, certains retraitement ne marchent pas pour des raisons inconnues.

Les anomalies sont laisser telle quelle car elles n'impacte pas de façon majeur l'outils, de plus il était imaginer à un moment donner de stocker les données également en nosql, pour éventuellement sur un axes totalement imaginaire aurait permis une évolution de l'outils en reliant les aliments directements à des recettes de cuisines customiser , pour en faire prédire l'apport nutritif d'un plat et non des aliments au cas par cas.

La base Ciqual de par sa fonction dispose également de plat, mais celle-ci pourrait se renforcer avec la base de donnée foods qui est un csv de 1 giga disponible également sur Kaggle.

Les pistes de développement pour fidéliser l'utilisateur sont nombreuses mais sont mises de côté pour remplir au besoin la formation et la problématique du temps.

Les fonctions nécessaires à la formation impliquent l'utilisation d'une base de donnée en conséquent la fonction IMC se voit couplé à un besoin pour créer un intérêt pour l'utilisateur, cette fonction superficiel à 2 rôles, occuper un espace de l'application, créer une utilisation par l'utilisateur, remplir une partie de l'interface à disposition.

Les bases de données sont les suivantes :

USERS et IMC.

The diagram illustrates a relationship between two database tables: **USERS** and **IMC**. A pink arrow points from the **USERS** table to the **IMC** table, indicating a connection or dependency between them.

USERS			
	ID_USERS	integer	
NOT NULL	CREATION	date default(CURRENT_TIMESTAMP())	
NOT NULL	E_MAIL	varchar(25)	
NOT NULL	USERNAME	varchar(25)	
NOT NULL	PASSWORD	varchar(255)	
	AGE	integer(3) default(0)	
	POIDS	integer(3) default(0)	
	SEXE	varchar(1) default(0)	
	NOM	varchar(25) default(0)	
Add field			

IMC			
NOT NULL	USERS_ID	integer	
NOT NULL	CREATION	date	
NOT NULL	IMC	float	
Add field			

L'utilisation permet de se connecter en sollicitant le module login de Flask, pour rappel Flask utilise déjà une gestion des cookies, mais ce module permet d'importer des données et de donner une valeur à chaque utilisateur.

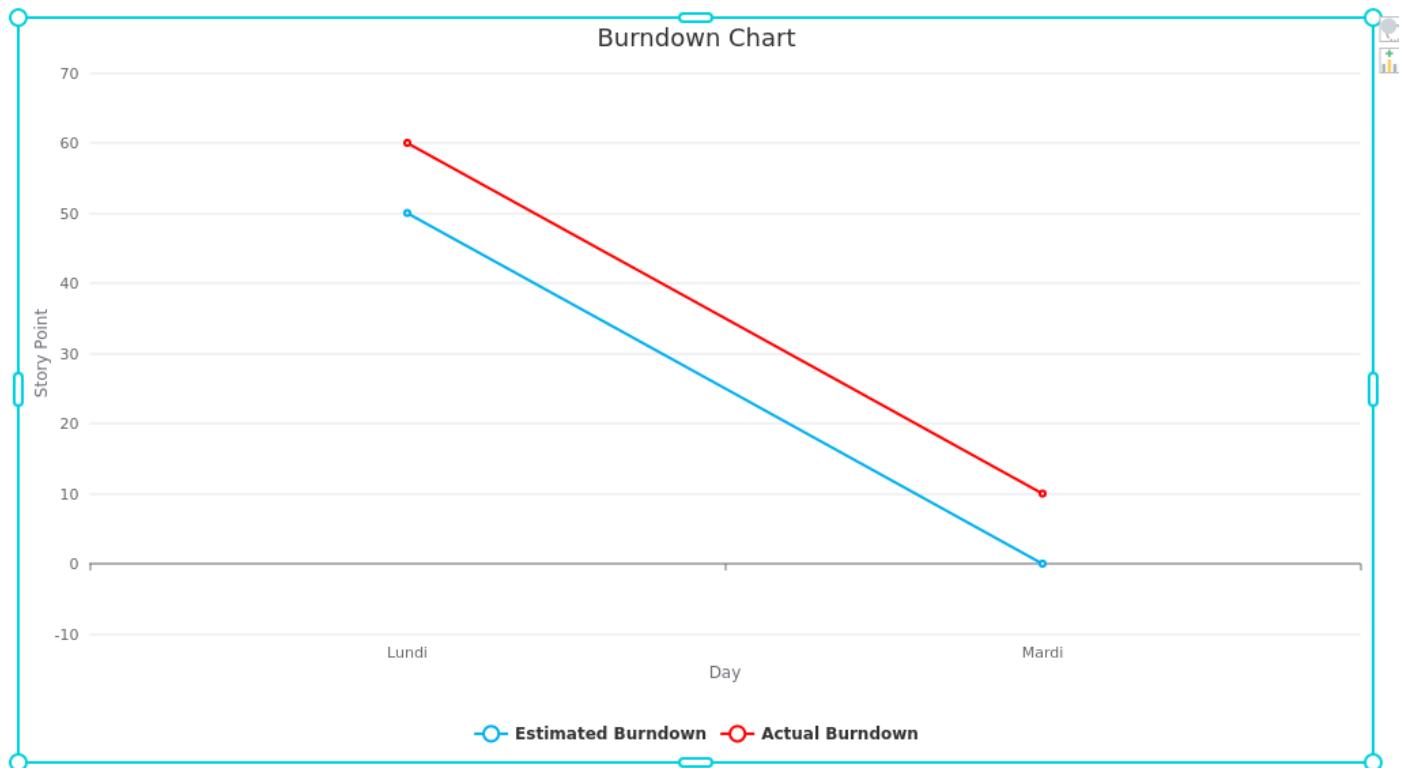
Dans l'utilisation la table IMC contient 3 champs, ID_USERS en clé étrangère, IMC, et DATE_CREATION, En conséquence, si l'utilisateur utilise l'imc en étant connecter, une action supplémentaire SQL se crée, et enregistre la donnée dans la table IMC.

Il peut être imaginer que l'utilisateur puisse suivre la différence entre le début de l'application, et son utilisation courante comme un suivie chronologique de sa progression et si éventuellement, quelqu'un qui s'intéresse ou surveille ses apports nutritifs peut grossir ou maigrir.

Une fonction de test serveur fut créée en supplément, celle-ci vérifie uniquement que toutes les pages de l'application renvoient un code 200 et signal si ce n'est pas le cas.

Voici le burndown chart de sprint 2.

Tout fut dans les temps à l'exception de la mise en production, ce qui est en soit non requis par le client, les travaux peuvent être livrés en l'état. (le référentiel n'exige pas la mise en production, cela aurait été un plus)



Compte rendu de clôture

Le client est satisfait du produit reçu, il cherchera à l'améliorer avec la sollicitation de spécialistes dédiés à sa partie Front et avec de la chance pourra nous solliciter pour un approfondissement du back

Le temps à était respecté, et globalement certains axes seront améliorés entre le temps de la remise du rapport final, cependant le corps du projet est déjà présent.

De nombreux axes peuvent être améliorés comme la mise en place de bootstrap pour gérer facilement la partie front en l'absence d'un investissement dans le css, l'avantage de celui-ci est qu'il propose travers divers templates, initialement bootstrap était un précurseur sur les avancées du CSS et en conséquent il s'agit de moyen dérivé pour obtenir des fonctions que fais aujourd'hui CSS si on s'investit.

Son implication peut donc peser en cas d'utilisation de celui-ci, le template approfondis utilisateur devrait le justifier mais il reste une porte de sortie pour les développeurs n'aimant pas le front.

L'amélioration du modèle avec des instances de calcul sur un pc supportant le GPU pour effectuer des différentes comparaison que cela soit sur le batch, les epochs, les analyses de convergences des modèles et déterminer lequel répondait le mieux à cette problématique.

Il était également possible de faire une approche par une classification avec du machine learning, la tendance est au Deep learning pour ce genre de modèle de par leurs précisions, mais une comparaison des résultats aurait pu être intéressante.

La partie docker et son déploiement via dockerfile pourront être un bonus d'ici la présentation finale de l'application.

En axe d'amélioration la partie base de donnée, les images pourraient être stockés au format array sur une base NOSQL, avec la labellisation attendu, la labellisation utilisateur et la labellisation, cela permettrait par après de faire s'entraîner le modèle naturellement avec des images de qualités sans intermédiaire ce qui lui permettrait de s'enrichir avec la durée de son utilisation.

D'un point de vue PO, le produit est vendu mais la cadence de temps aurait pu être optimiser, les principes ce sont voulu respecter, mais pour la deadline et car l'équipe était motivée, un peu de crunch fut mis en place.

D'un point de vue équipe : les tickets étaient pris et malgré de nombreuses difficultés, les solutions furent trouver tant bien que mal.

Le test de régression n'indique aucune perte de performance du prototype du sans modèle, au modèle 2 sorties, et du modèle 2 couches à 10 couches / sorties.

Remerciement

Je tiens à remercier dans ce rapport spécifiquement Arnaud de Mouhy qui sur son temps personnel de développer m'a permis de résoudre les problématiques de containers.

Mais également je remercie le CAPG pour m'avoir accueillie et à Monsieur Brand le data scientiste du groupe chargé de superviser les alternants qui était dans le groupe, ce qui m'a permis de consolider une partie du machine learning.

Ainsi que les formateurs de la formation et les intervenants tiers que la promotion a reçu durant les 19 mois.

ANNEXE

Lexique : *

*CAPG : [Crédit Agricole Pyrénées Gascogne](#)

Source Garamond :

<https://larecetteduweb.fr/conception-de-site-web/police-de-caractere/>

<https://99designs.fr/blog/conseils-design/meilleures-et-pires-polices-de-caracteres/>

Source medium :

<https://medium.com/mdr-inc/from-sgd-to-adam-c9fce513c4bb>

source : batch size : <https://arxiv.org/pdf/1609.04836.pdf>

Stack overflow :

image et sql : <https://stackoverflow.com/questions/5613898/storing-images-in-sql-server/5613926>

extrait : There's a really good paper by Microsoft Research called [To Blob or Not To Blob](#).

Their conclusion after a large number of performance tests and analysis is this:

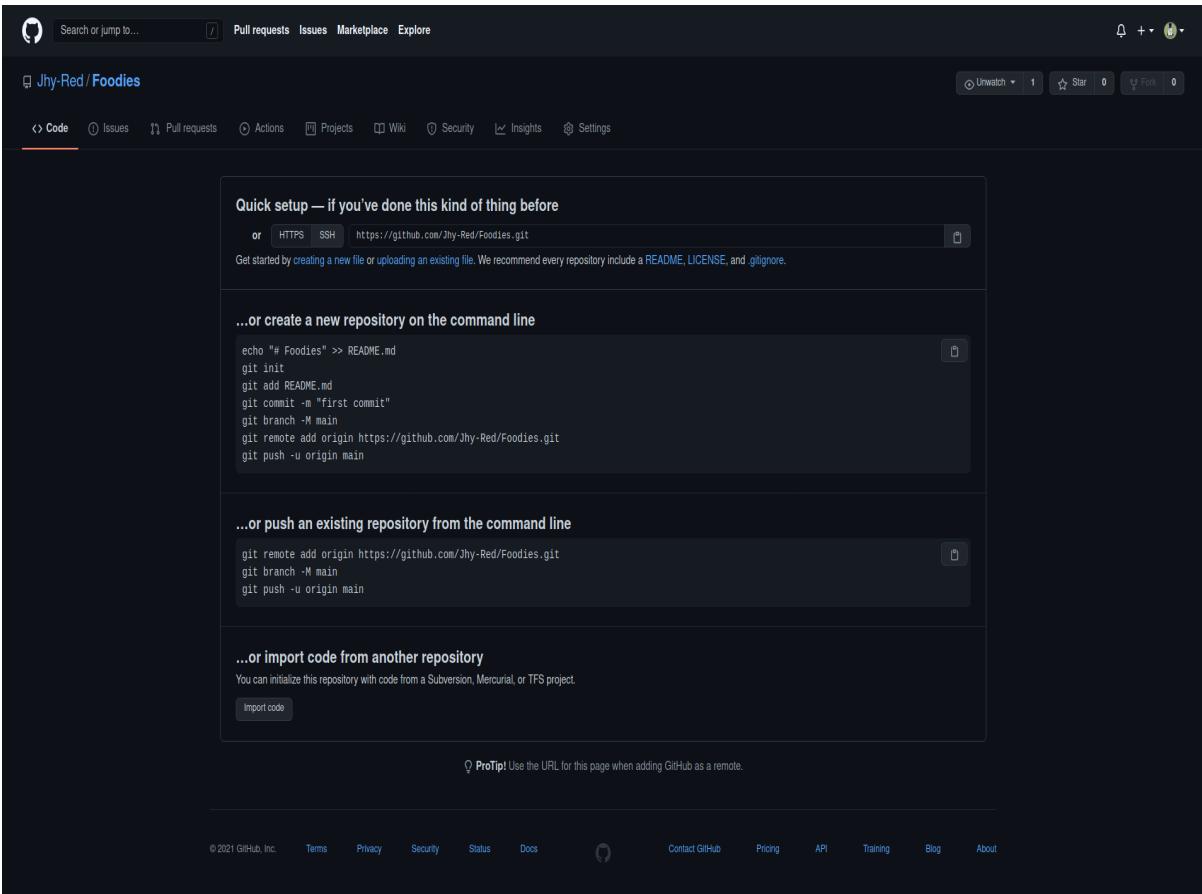
- if your pictures or document are typically below 256KB in size, storing them in a database VARBINARY column is more efficient
- if your pictures or document are typically over 1 MB in size, storing them in the filesystem is more efficient (and with SQL Server 2008's FILESTREAM attribute, they're still under transactional control and part of the database)
- in between those two, it's a bit of a toss-up depending on your use

If you decide to put your pictures into a SQL Server table, I would strongly recommend using a separate table for storing those pictures - do not store the employee photo in the employee table - keep them in a separate table. That way, the Employee table can stay lean and mean and very efficient, assuming you don't always need to select the employee photo, too, as part of your queries.

For filegroups, check out [Files and Filegroup Architecture](#) for an intro. Basically, you would either create your database with a separate filegroup for large data structures right from the beginning, or add an additional filegroup later. Let's call it "LARGE_DATA".

Now, whenever you have a new table to create which needs to store VARCHAR(MAX) or VARBINARY(MAX) columns, you can specify this file group for the large data:

GITHUB



The screenshot shows the GitHub interface for a new repository named "Foodies". The top navigation bar includes links for "Pull requests", "Issues", "Marketplace", and "Explore". The repository header shows "Unwatch 1", "Star 0", and "Fork 0". Below the header, there are tabs for "Code", "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". A "Code" tab is selected.

Quick setup — if you've done this kind of thing before

or [HTTPS](https://github.com/Jhy-Red/Foodies.git) [SSH](ssh://github.com/Jhy-Red/Foodies.git) <https://github.com/Jhy-Red/Foodies.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# Foodies" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Jhy-Red/Foodies.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/Jhy-Red/Foodies.git
git branch -M main
git push -u origin main
```

...or import code from another repository

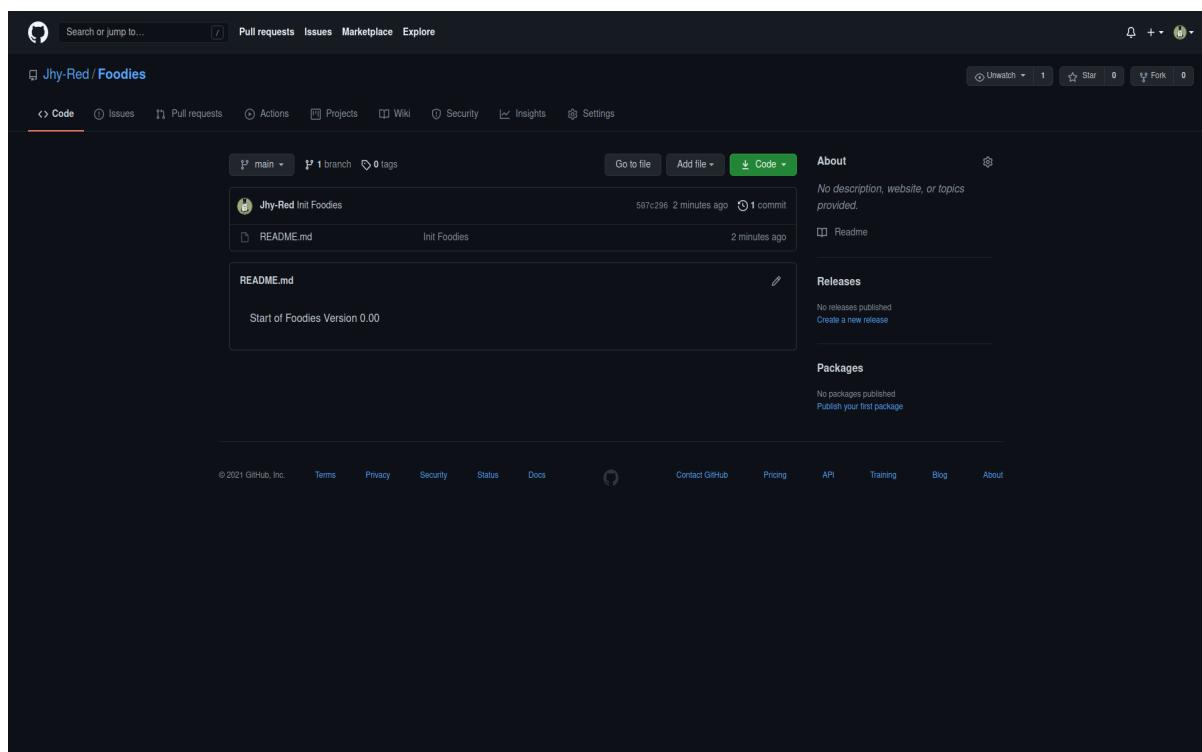
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

ProTip! Use the URL for this page when adding GitHub as a remote.

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Annexe 1 :



The screenshot shows the GitHub repository page for "Foodies". The top navigation bar includes links for "Pull requests", "Issues", "Marketplace", and "Explore". The repository header shows "Unwatch 1", "Star 0", and "Fork 0". Below the header, there are tabs for "Code", "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". A "Code" tab is selected.

Code main 1 branch 0 tags Go to file Add file Codes

Jhy-Red Init Foodies 507c296 2 minutes ago 1 commit

README.md Init Foodies 2 minutes ago

README.md Start of Foodies Version 0.00

About No description, website, or topics provided. Readme

Releases No releases published Create a new release

Packages No packages published Publish your first package

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Annexe 2 :

Trello :

The screenshot shows a Trello board titled "Foodies". The board has four main lists:

- À faire**: Contains cards like "Astuce Trello: Ajoutez vos questions et sujets à aborder pour ne plus rien oublier!", "Qui peut m'aider à résoudre mon problème HTML d'alignement de texte?", and "Créer la bannière du site".
- En cours**: Contains cards like "Astuce Trello: Pour les tâches presque terminées mais qui dépendent d'une dernière étape.", "Revue des termes légaux", and "Écrire une première ébauche des e-mails".
- Bloqué**: Contains cards like "Astuce Trello: Alerte rouge ! Falles remonter les goulots d'étranglement qui empêchent votre équipe d'avancer sur le projet.", "Contracts avec les freelances", and "Validation du budget".
- Terminé**: Contains cards like "Astuce Trello: Bravo! Vous avez terminé La liste de tous les projets menés au bout avec succès. Votre palmarès.", "Finalisation de la campagne: La perfection du travail en équipe.", "Envoyer le rapport du T1", and "Proposition de campagne".

On the left side, there is a sidebar with project resources and a search bar at the top.