

# 5A ILC Sujet de TP/Projet

2021- Détection de port de masque

Brousse Olivier

26/09/2021

## 1 Objectif pédagogique du projet

L'objectif de ce projet est de permettre aux étudiants de voir l'ensemble de la chaîne permettant de concevoir une application à base de Machine Learning (incluant les méthodes de Deep-Learning présentées lors des cours).

Le projet porte donc sur les 3 axes principaux suivants :

1. **Conception d'une base d'images annotées** destinée à l'entraînement des modèles/algorithmes de Machine Learning.
2. **Sélection et entraînement** d'un **modèle** adapté à l'application choisie.
3. **Mise en œuvre temps réel** de ce modèle pour présentation des résultats obtenus.

## 2 Application de détection de port de masque

L'application sélectionnée pour cette année 2020 est le port du masque de protection individuelle.

La fiabilité des algorithmes sera évaluée sur le dataset kaggle face mask détection <https://www.kaggle.com/andrewmvd/face-mask-detection>.

Il se compose de 853 images annotées à l'aide de boîtes englobantes et se focalise sur 3 classes d'objets :

1. Label "**without\_mask**" : Visages sans masque
2. Label "**mask\_wearred\_incorrect**" : Visages avec un masque mal porté
3. Label "**with\_mask**" : Visages avec un masque (bien porté)

Les annotations sont faites de boîtes englobantes et disponibles au format Pascal/VOC.

Ce dataset ne doit en aucun cas être utilisé pour entrainer l'algorithme final sous peine de disqualification (et donc d'une note en conséquence).

### 2.1 Etape 1 Créer le dataset d'entraînement

Le dataset pour l'entraînement est donc à concevoir. Une base collective d'environ 2000 images annotées doit être mise en place de façon à former un socle pour l'entraînement des algorithmes ou modèles utilisés. Dans ce cadre la promotion doit :

- **Collectivement** et équitablement composer la base
  - **La répartition des tâches ne doit pas incomber au même personnes**
  - Chacun participe aux différentes étapes de la création de la base :
    1. Recherche d'image ou prises de vues : **chacun apporte des images** à mettre en commun.
    2. **Validation** des images apportées **par un autre étudiant**
    3. Annotation des **images validées** : une fois les images sélectionnées chacun prend sa part d'image et les annotes (soit environ 2000/28 images chacun)
    4. **Validation** et correction des **annotations faites par un autre étudiant**
- **Aider** les camarades qui rédigent les **scripts utilitaires**
  - Recherche de doublons et renommage
    - **le nom de l'image est le hash md5 des données brutes de l'image**
    - on recherche les **images similaires** par exemple à l'aide de <https://betterprogramming.pub/how-to-measure-image-similarities-in-python-12f1cb2b7281> et on **choisi un bon descripteur pour éliminer les doublons**
  - conversion des format d'annotation
  - Téléchargement
  - ...

- Respecter les bonnes pratiques de la conception de base
  - Pas d'images en doublon
  - Des images de taille et qualité suffisante (on évite les timbres poste l'e images floues...)
  - L'annotation doit être la plus similaire à celle du dataset de test en terme de forme et placement des boîtes englobantes. Il en va de la capacité des modèles à bien réussir le test final !
- Cette base collective n'est pas forcément la seule que vous pouvez utiliser...  
**Faites vos ajouts personnels cela peut vous aider.**

## 2.2 Etape 2 choix de l'algorithme ou modèle

Le but final de l'application est de détecter les visages présents dans une scène et de déterminer s'ils sont masqués correctement ou non. Les détecteurs d'objets sont donc une possibilité à envisager mais ce n'est pas la seule option. Par binôme vous allez devoir choisir la solution que vous estimez être la plus efficace pour mener à bien la tâche demandée.

Quelle que soit votre choix il vous faudra :

1. Sélectionner les **outils et framework** que vous utiliserez
2. Historiser vos développements à l'aide un service comme **github**
3. Garder des traces des différents entraînements effectués avec à chaque fois
  - a. ce que vous avez changé entre les entraînement : ajout d'images à la base, changement dans la structure du réseau, changement dans les hyperparamètres...)
  - b. ce que cela apporte ou enlève aux performances de votre solution

L'évaluation de la solution se fait suivant 3 critères principaux **rapidité, performance de détection** et votre **gestion du projet et son utilisabilité**. L'évaluation est détaillée dans le chapitre 3 Evaluation.

Les notes de rapidité et performance de détection seront données par deux challenges (le meilleur aura la meilleure note et ainsi de suite). La rapidité sera évaluée sur une cible unique (PC sous windows 10 a priori). Il vous faudra donc vous assurer le pouvoir déployer votre application sur cette plateforme car les sessions Google Colab ne sont pas équilibrées en ressources ce qui induirait de grandes disparités dans les résultats.

## 2.3 Etape 3 Mise en œuvre et évaluation

L'application doit être utilisable en 2 modes :

1. Evaluer la solution sur le dataset de test pour fournir les informations de rapidité et de performance de détection. L'évaluation se fait au moyen des fonctions utilisées pour le projet VOC disponibles dans le lien suivant : <https://github.com/Cartucho/mAP> .
2. Faire des inférences et affichage des résultats, en temps réel. Les images provenant d'une caméra (webcam ou capteur photo d'un téléphone portable). L'affichage doit faire apparaître en incrustation

- a. les résultats de détection dans l'image (une couleur par classe, incrustation du label sous forme de texte)
- b. le FPS calculé (sous forme de texte)

c.

### 3 Evaluation

Pour l'évaluation de vos projet il vous faut :

1. Un rapport de projet (note d'écrit sur 20)
2. Une présentation orale (note d'oral sur 20) lors d'une séance de présentation
3. L'application (note de challenge IA sur 20)
  - a. Installer windows pour les tests en live
  - b. Lien vers le repo Github pour permettre
    - i. une validation des entraînements
    - ii. Une reproduction de l'évaluation

Pour la note de challenge la performance de détection du réseau sera évaluée sur 10 et la rapidité d'inférence sera évaluée sur 10.

