



# HandiControl2

Tschopp Bruno

Projet de semestre 5 2018-2019

## Rapport

---

### Mandant :

Home Atelier de La Colline

### Filière :

Télécommunications

### Responsables internes :

Wagen Jean-Frédéric ; Hennebert Jean

### Version :

0.1

**Hes·so**  
Haute Ecole Spécialisée  
de Suisse occidentale  
Fachhochschule Westschweiz  
University of Applied Sciences and Arts  
Western Switzerland



## Table des matières

---

<b>RAPPORT .....</b>	<b>1</b>
<b>CONTEXTE .....</b>	<b>3</b>
PARTIES PRENANTES .....	3
OBJECTIFS.....	3
CONFIDENTIALITÉ .....	4
<b>STRUCTURE DU RAPPORT.....</b>	<b>4</b>
<b>DÉBUT DU PROJET .....</b>	<b>4</b>
<b>LISTE DES TÂCHES .....</b>	<b>5</b>
<b>PRISE EN MAIN DU PROJET : .....</b>	<b>5</b>
REPRODUCTION DU SYSTÈME HANDICONTROL .....	6
<b>ANALYSE .....</b>	<b>7</b>
TECHNOLOGIES .....	8
DÉTECTION DE VISAGES .....	10
DÉTECTION DE CARACTÉRISTIQUES .....	19
<b>TEST.....</b>	<b>21</b>
FACTEURS D'INTERPRÉTATIONS DES INTENTIONS DE MR. F. ....	21
<b>CONCEPTION .....</b>	<b>22</b>
<b>BUSINESS PLAN .....</b>	<b>22</b>
<b>CONCLUSION.....</b>	<b>22</b>
REMERCIEMENTS .....	22
SIGNATURE .....	23
<b>DÉCLARATION D'HONNEUR .....</b>	<b>23</b>
<b>GLOSSAIRE.....</b>	<b>23</b>
<b>LOGICIELS UTILISÉS.....</b>	<b>24</b>
<b>ANNEXES.....</b>	<b>24</b>
PROCÉDURE D'INSTALLATION DU SYSTÈME HANDICONTROL .....	24
RÉFÉRENCES :.....	27
<b><u>BIBLIOGRAPHIE .....</u></b>	<b><u>27</u></b>

## Contexte

Monsieur F., résident à la Colline, est atteint de polyhandicap. Sa mobilité, ses capacités de communications et d'interactions sont fortement réduites. Le mandant souhaiterait lui fournir des outils lui permettant d'être plus autonome et de pouvoir agir sur son environnement. Le projet HandiControl a pour but de réaliser un système permettant à Monsieur F. de pouvoir écouter de la musique et gérer l'éclairage de sa chambre par lui-même.

A ce stade, une première phase du projet (HandiControl1) a déjà été réalisée durant le semestre précédent (automne 18-19). Les étudiants ont conçu un système permettant d'enregistrer et de labéliser les différentes expressions du patient. De plus une application à l'aide duquel le personnel soignant peut interagir sur l'environnement de Monsieur F (changer la musique) a également été développée.

L'objectif désormais, durant cette deuxième phase du projet, est de rendre ce système autonome. Le Machine Learning est l'outil qui sera utilisé pour supprimer tout acteur humain entre le patient et le système. L'algorithme utilisera les données fournies par le système de labélisation de HandiControl1 pour évoluer.

## Parties prenantes

**Mandants** : Home Atelier La Colline Mme Rita Caridade, référente du patient.

**Responsables internes** : Mrs Jean-Frédéric Wagen et Jean Hennebert, professeurs à la HEIA-FR.

**Mandataires** : Bruno Tschopp, étudiant à la HEIA-FR.

## Objectifs

Objectifs principaux :

- Reproduire le système HandiControl1 et mettre à jour sa documentation.
- Analyser l'état de l'art Machine Learning pour la détection de visage et d'émotion.
- Tester un ou plusieurs algorithmes existants de détection de visage/caractéristique avec les datas de HandiControl1.
- Définir avec le personnel soignant quels sont les facteurs importants dans

l'interprétation des intentions du patient. Savoir quelles sont les émotions à reconnaître sur le visage.

- Élaborations d'une demande de financement.

Objectifs secondaires :

- Modifier (ré-entraîner) les algorithmes de détections de visage, dans le cas où les performances des algorithmes existants ne sont pas satisfaisantes.
- Mettre en place la reconnaissance d'émotions.

## Confidentialité

Ce projet est intimement lié au milieu médical, par conséquent le secret médical est de rigueur. Afin de préserver la vie privée du patient son nom sera remplacé par Mr. F. De plus un accord de confidentialité a été signé, celui-ci est disponible en annexe.

## Structure du rapport

Le présent document présente l'ensemble des travaux, tâches et recherches effectués durant ce projet. Sa structure est la suivante :

- Prises en main : Reproduction du système HandiControl.
- Analyse : Étude des technologie de Machine Learning existantes.
- Benchmark : Mise en place d'un Benchmark dans le but d'effectuer des test.
- Test : Test du système implémenté.
- Conception : Proposition d'un système pour la détection de visage et d'indentification des émotions.
- Business plan : Présentation d'un business plan dans le but d'obtenir du financement.
- Conclusion : Conclusion personnel du projet.
- Déclaration d'honneur
- Glossaire
- Logiciels utilisés
- Annexes

## Début du projet

Durant les premières semaines qui ont suivi les attributions des projets, il m'a fallu rédiger un cahier des charges dans lequel j'ai regroupé les informations suivantes : une mise en contexte (ci-dessus Contexte); une liste des objectifs à atteindre; une liste des tâches; une planification complète sur la durée totale du projet; les liens concernant les rendus administratifs (cf. (Cahier des charges - HandiControl2)).

## Liste des tâches

### *1) Reproduction du système HandiControl1 Pour l'acquisition de données :*

- Prise de connaissance du système et de son fonctionnement global.
- Mise en place du système.
- Adaptation/rédaction d'un manuel d'installation.

### *2) Analyse de l'état de l'art du machine Learning (détection visage/émotion) :*

- Assimilation des principes de base.
- Définir quels sont les types d'algorithmes/technologies les plus adaptés au système de HandiControl1 (type d'input/output, modules, frameworks, etc.).

### *3) Benchmark d'un système de Machine Learning :*

- Acquisition des données pour le benchmark.
- Tester le ou les systèmes avec un set de données spécifique pour en étudier la pertinence.

### *4) Proposition/conception d'un système pour la détection de visage et d'indentification des émotions.*

### *5) Réalisation d'une partie Business*

- Rédaction d'un business plan dans le but d'obtenir du financement.

### *6) Réalisation de la documentation :*

- Fournir une documentation complète du projet et de son processus de réalisation (Cahier des charges, PVs, Rapport, Planning, etc.)

## Prise en main du projet :

Afin de poursuivre ce projet dans de bonnes conditions, il a été primordial de me familiariser avec le travail fourni par les étudiants (Mr. Beffa et Donzallaz) durant

le semestre précédent. En effet c'est à l'aide de leur système que se fera l'acquisition des données pour le Machine Learning.

## Reproduction du système HandiControl

C'est donc à l'aide du rapport (cf. (Rapport HandiControl1)) rédigé par leurs soins que je me suis attaqué à la reproduction du système issu de la première phase du projet. De plus, une procédure d'installation complète du système a été rédigée (cf. (Procédure d'installation - HandiControl1)).

### *Améliorations possibles :*

Je rassemble ici dans cette section, les diverses possibilités d'optimisations (autre que celles déjà proposées dans le rapport HandiControl1) des applications issues de mon expérience utilisatrice et de celle du personnel soignant.

### *Réduction du délai de synchronisation des appareils :*

Une des remarques le plus souvent faite par le personnel soignant est que l'application de contrôle présente souvent un écran noir au démarrage. Cela peut être dû à des difficultés de synchronisation entre les deux appareils. La mise en place d'un Access point (plus stable que l'architecture réseau présente dans les locaux) sur le terrain pourrait réduire ce problème.

### *Ajout de label sur la durée dans l'application dataview :*

Lorsque l'on visionne une session à l'aide de l'application dataview, il serait intéressant de pouvoir ajouter un label sur une période de temps donnée. À ce stade cela est possible uniquement durant la prise de données avec le téléphone de contrôle. Ce point rejoint la critique déjà émise par le groupe (cf. (Rapport HandiControl1) chapitre 6.4).

### *Changement de couleur pour le bouton DEMARRER/ARRÊTER :*

Afin de faciliter l'expérience utilisatrice du personnel médical, il serait intéressant d'indiquer en changeant la couleur du bouton (bleu → rouge) si oui ou non une saisie de données est en cours.



Figure 1 : Bouton DEMARRER de l'application de saisie HANDICONTROL



Figure 2 : Bouton ARRETER de l'application de saisie HANDICONTROL

### *Mise en place d'un access point :*

Nous avons fourni à l'équipe éducatif un access point, afin de réduire le délai de synchronisation entre les deux appareils Androids.



Figure 3 : Équipement Zyxel servant d'access point pour le home de la colline

## Analyse

Ce chapitre regroupe le travail de recherche effectué. L'objectif est d'y présenter l'état de l'art actuel du machine learning en matière de détection de visages et d'émotions/expressions faciales.

En fin d'analyse, suffisamment d'éléments devront être présentés pour pouvoir faire un choix technique quant à la stratégie à adopter. Deux éléments principaux sont à déterminer pour répondre à la demande du Home de la Colline :

- Le type d'approche :
  - Détection de visage.
  - Détection de caractéristiques.

Il s'agira ici de déterminer quel sera la manière la plus efficace d'interpréter les intentions de Mr. F. Dans les conditions actuelles sur le terrain est-il possible de détecter le visage du patient à l'aide des technologies existantes ? Au niveau de l'interprétation du vocabulaire du patient, est-il nécessaire de reconnaître ses expressions faciales dans le détail ou le visage suffit-il ?

- Quelles technologies utiliser :
  - Frameworks, modules, etc.

Quelle est la technologie la plus adaptée au système déjà en place et aux particularités de Mr. F. ?

Finalement, je précise ici qu'à ce stade du projet le focus est donné sur l'utilisation de la vidéo comme input. Les applications de prises d'images étant déjà fonctionnelles. Toutes fois il se pourrait que plus tard dans le projet apparaisse le besoin d'ajouter des inputs différents (signaux physiologiques, sons, etc.) dans un but de performances et de pertinences.

## Technologies

Vous trouvez ci-dessous un récapitulatif des technologies entourant ce projet. Le machine learning et le *deep learning* sont tous deux des champs sous-jacents au domaine de l'intelligence artificielle.

### *Machine Learning*

Le machine learning est un ensemble d'algorithmes offrant la capacité à un ordinateur d'apprendre des données qu'il traite et d'en tirer des conclusions, plus exactement être capable de faire des prédictions sur des résultats futurs. Ce type d'algorithmes a besoin de modèles pour apprendre. Les modèles sont l'ensemble des caractéristiques à évaluer et à reconnaître.

Par exemple sur l'image ci-dessous (cf. Figure 4 : Machine Learning Workflow), nous voulons créer un modèle permettant de reconnaître une image de chat, nous allons donc fournir à l'algorithme tous les détails (caractéristiques, features) servant à distinguer un chat d'un autre animal (forme des oreilles, moustaches, forme des pâtes, etc.).



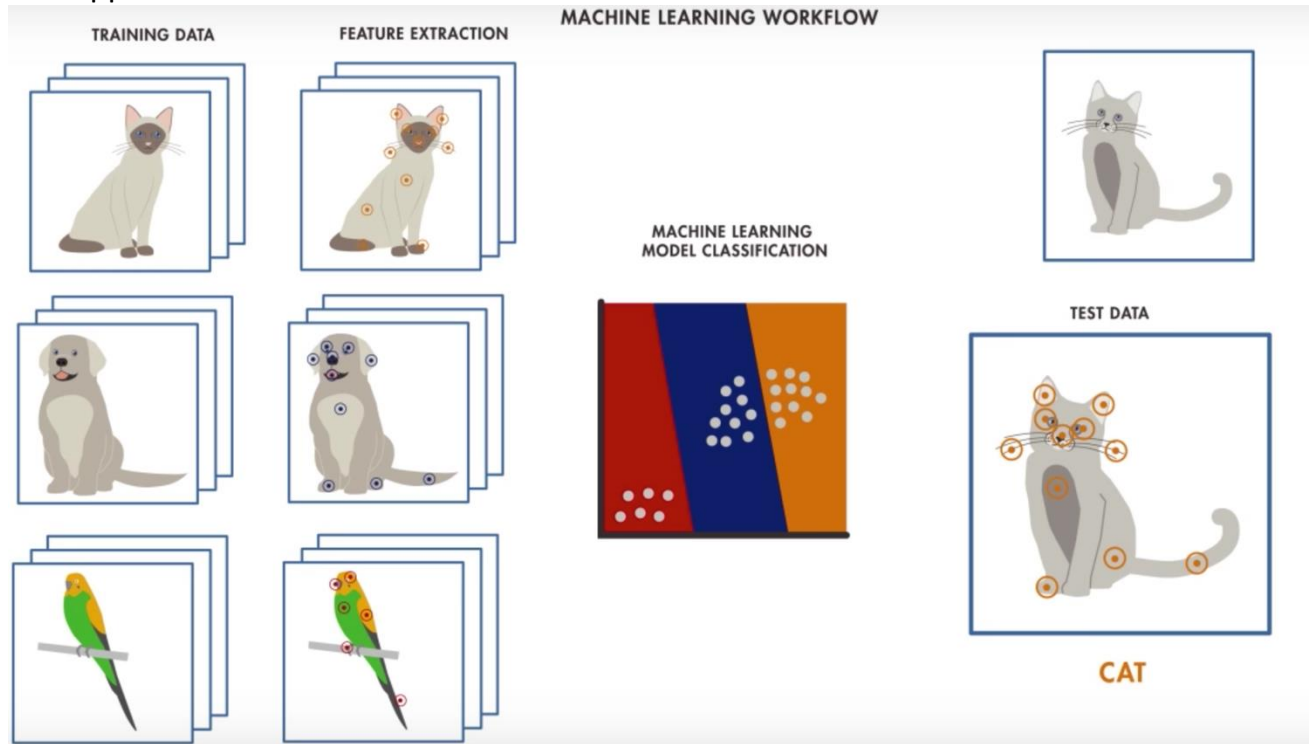


Figure 4 : Machine Learning Workflow

## Deep Learning

Le deep learning est un ensemble d'algorithmes et de méthodes d'apprentissage automatisés, inspiré de la structure et du fonctionnement du cerveau. Cette technologie utilise ce que l'on nomme des *réseaux de neurones*, équivalent aux modèles du machine learning. Ces réseaux ont un rôle de filtre, l'algorithme y passe un grand ensemble de données au travers, contrairement au machine learning, ce type d'algorithme évolue de lui-même en ajustant son réseau.

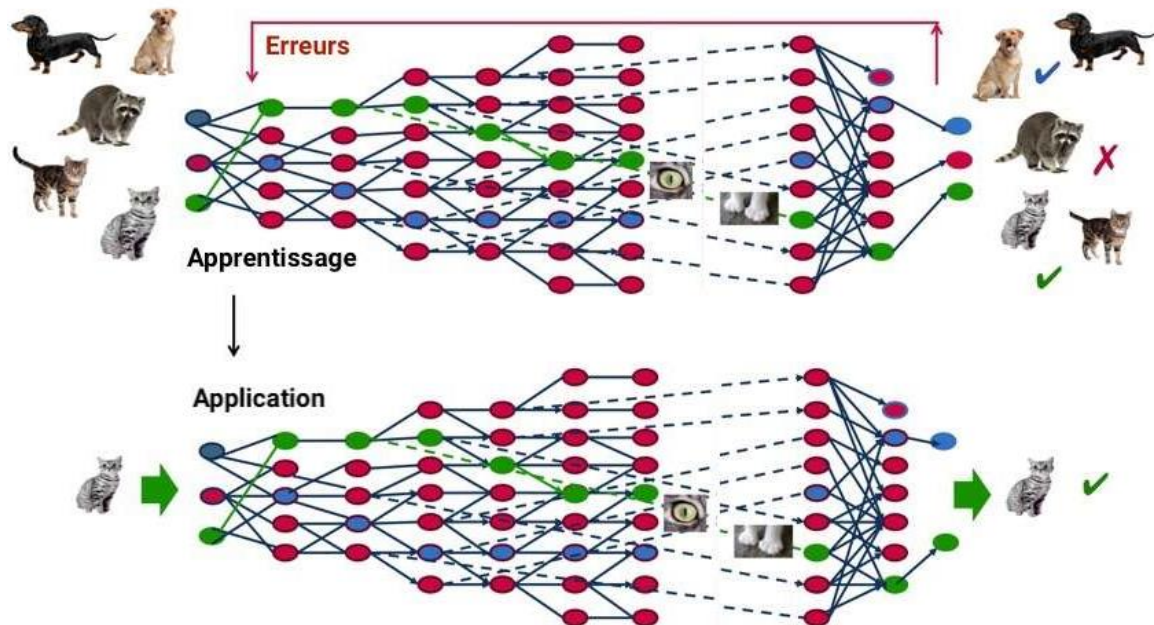


Figure 5 : Deep Learning Workflow

## Détection de visages

### *Critères initiaux*

Le large champ de possibilités offert par le machine learning en fait un outil extrêmement puissant, c'est pourquoi un nombre important d'acteurs sont sur le marché. Une quantité importante de framework et d'API sont donc disponible, il s'agira ici d'en faire une sélection.

Tout d'abord, j'établis ici des critères de recherche servant de filtre :

- 1) Compatibilité avec le système d'exploitation Android.
- 2) Capable d'analyser un input vidéo en temps réel.
- 3) Dans la configuration actuel du système de prises de données, le visage du patient est parfois de travers ou éloigné du téléphone. La technologie implémenté devra permettre une certaine souplesse dans la qualité de prise d'image.
- 4) De par sa condition, le visage de Mr. F. présente quelques particularités. Le système de détection de visage devra être capable de reconnaître celui du patient.

## Tests/essaies préliminaires

Après avoir effectué une première recherche sur les possibilités existantes, il m'a semblé judicieux de tester si des applications de reconnaissance faciale natives (faceid, etc.) étaient aptes à reconnaître le visage de Mr. F. J'ai donc demandé à Mme Caridade d'effectuer des essais avec le patient. Ces tests n'ont malheureusement pas pu être effectués pour des raisons techniques.

J'ai donc profité de la visite du Home de la Colline (samedi 09.03.2019), pour effectuer ses tests directement sur le terrain. Mon smartphone (S7 Android 8.0.0) n'implémentant pas la reconnaissance faciale de manière native, j'ai donc parcouru le PlayStore pour y trouver des applications de détection. Mon choix s'est porté sur les applications suivantes :

### Face Detection -AI



Figure 6 : Application Android Face Detection -AI

- Développeurs : GetUbusiness
- Version 1.9.4
- Compatibilité Android : 5.1 ou +
- Framework/API : Google Vision et Azure

Résultats des tests :

Critères	Détection du visage	Souplesse de la prise de vue (biais)	Distance
Résultats	Oui	Forte	Grande

Lien PlayStore :

<https://play.google.com/store/apps/details?id=com.rebel.user.findfaces>



*Figure 7 : Application Android Luxand Face Recognition*

- Développeurs : Luxand, Inc.
- Version 1.12
- Compatibilité Android : 4.0 ou +
- Framework/API : Luxand FaceSDK

Résultats des tests :

Critères	Détection du visage	Souplesse de la prise de vue (biais)	Distance
Résultats	Oui	Leger	Proche

Lien PlayStore :

<https://play.google.com/store/apps/details?id=com.luxand.facerecognition>

## Face Recognition Demo with Opencv Manager



*Figure 8 : Application Android Face Recognition Demo with OpenCV Manager*

- Développeurs : Preeti
- Version 1.0
- Compatibilité Android : 4.4 ou +
- Framework : Opencv

Critères	Détection du visage	Souplesse de la prise de vue (biais)	Distance
Résultats	Oui	Forte	Grande

Lien PlayStore :

<https://play.google.com/store/apps/details?id=demo.recgonition.face.facerecognitiondemo>

Ces tests m'ont donc permis de répondre à certaines questions posées par les deux derniers critères de sélection (cf. Critères initiaux 3 et 4). L'environnement de prises d'images actuellement mis en place ainsi que le visage de Mr. F. ne sont pas un problème pour les technologies existantes.

### *Frameworks/APIs relevés*

Je regroupe dans cette section, les frameworks ou APIs trouvés sur internet répondant aux critères initiaux. À noter que durant mes recherches, des APIs intéressantes mais malheureuses payantes ont été trouvées, j'ai choisi de ne pas présenter ces dernières ci-dessous (Amazon Rekognition, IBM Watson).

### **OpenCV**



*Figure 9 : Open Computer Vision*

Open Computer Vision est une bibliothèque graphique libre, optimisé pour le traitement d'image en temps réel et disponible sous *licence BSD*.

**Spécificités :**

- Traitement d'images :
  - Lecture, écriture et affichage d'une image.
  - Calcul de l'*histogramme*.
  - Opérations de traitement d'image (lissage, filtrage, etc.).
  - Analyse de la structure de l'image.
- Traitement de vidéos :
  - Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra)
  - Détection de visages
  - Détection de mouvement et poursuite d'objets.
  - Estimation de *flux optique*.
- Grands nombre d'algorithmes d'apprentissage disponibles.

**Liens :**

<https://opencv.org/>

**TensorFlow/TensorFlow Lite**

Librairie logiciel open-source pour le machine Learning développé par Google, orienté back-end.



Figure 10 : TensorFlow

**Spécificités :**

- Communauté très active et de nombreux produits basés sur cette librairie sont déjà existants.
- Flexibilité et modularité dans la création des modèles d'apprentissage.
- Hautement compatible, peut fonctionner sur divers support (GPU ou CPU, serveur, desktop, mobile) sans avoir à réécrire le code.
- Version Lite pour Android.
- Performant.
- Fournit l'outil de visualisation de données TensorBoard.

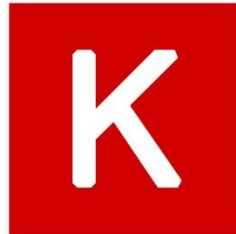
*Liens :*

<https://github.com/tensorflow/tensorflow>

<https://www.tensorflow.org/>

<http://codeinpython.com/tutorials/deep-learning-tensorflow-keras-pytorch/>

## Keras



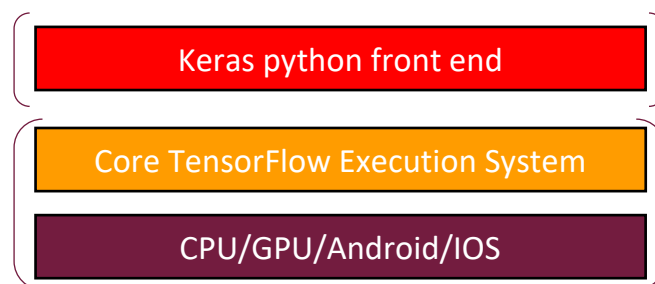
*Figure 11 : Keras*

Librairie open-source implémenté en python, pensé pour l'interaction front-end avec d'autre framework de machine learning existants.

### *Spécificités :*

- Apport un niveau d'abstraction supplémentaire pour d'autre framework comme TensorFlow.
- Conçu pour permettre une expérimentation rapide avec des réseaux de neurones complexes.
- Modulable, extensible et simplifié.

### *Architecture :*



*Figure 12 : Architecture des couches liant TensorFlow et Keras*

*Liens :*

<http://codeinpython.com/tutorials/deep-learning-tensorflow-keras-pytorch/>  
<https://github.com/keras-team/keras/tree/master/keras>  
<https://keras.io/>

## YOLO Real-Time Object Detection

YOLO pour You Only Look Once est un logiciel de détection d'objet en temps réels.



Figure 13 : YOLO Real-Time Object Detection

### Spécificités :

- Vitesse de traitement élevés.
- Traitement d'images multiples.
- Modèle pré-entraîné existants.
- Détections en temps réels avec une webcam.

### Restrictions :

- Processing sur ordinateur.
- Utilise Opencv et CUDA pour la détection en temps réels.

### Tests :

De manière similaires aux tests effectués avec les applications mobiles, j'ai voulu m'assurer que le frameworks soit apte à détecter le visage de Mr. F. J'ai donc utilisé le modèle pré-entraîné sur des photos de moi-même ainsi que du résident. Le logiciel détecter nos deux visages en quelques secondes !



Figure 14 : Test du modèle pré-entraîné de YOLO



*Liens :*

<https://pjreddie.com/darknet/yolo/>

## Firestore ML kit



*Figure 15 : Firestore*

Développée par Google et inscrite dans les services d'hébergement Firestore, ML kit est une API conçue pour les applications mobiles Android et iOS. Elle se veut accessible à tous indépendamment du niveau de familiarité avec le machine learning. ML kit propose un nombre important de modèles pré-entraînés (reconnaissance de texte, détection de visage, labélisation d'image, etc.)

### *Spécificités :*

- Peu de prérequis.
- Compatibilité Android, iOS.
- Mobile design.
- Modèles pré-entraînés.
- Mise à disposition du cloud Google pour une plus grande puissance de calcul.
- Compatibilité avec les modèles produits sur TensorFlow Lite.

### *Restrictions :*

- Firestore dispose d'une offre gratuite, limitant les interactions avec le cloud en terme de volume de données, services, etc.

*Liens :*

<https://firebase.google.com/products/ml-kit/>

## Luxand FaceSDK



Figure 16 : Luxand

SDK utilisé pour développer une des applications testées plus haut (cf. Tests/essaies préliminaires - Luxand Face Recognition). Il se veut centré sur la reconnaissance faciale, la détection de caractéristiques et d'expression facial.

*Spécificités :*

- Cross-platform (Windows, linux, mac OS, iOS, Android)
- Work on device → aucune connexion internet requise.

*Restrictions :*

- Payante à hauteur de 99\$ par mois (période d'essai gratuite de 2 semaines)

*Liens :*

<https://www.luxand.com/facesdk/>

## Microsoft Azure Face



Figure 17 : Microsoft Azure

Issue de la suite de service cloud Azure de Microsoft, Face permet la détection et la comparaison de visage humain et est beaucoup utilisé dans le domaine de la sécurité. Cet outil a permis de développer l'application Face Detection -AI (cf. Tests/essaies préliminaires - Face Detection -AI).

*Spécificités :*

- Compatibilité C#, Java, Node.js, Python, GO.

- API Rest disponible.

*Liens :*

<https://azure.microsoft.com/en-us/services/cognitive-services/face/>

## Google Cloud Vision



*Figure 18 : Google Cloud*

API proposant des modèles pré-entraînés d'analyse d'images (modération de contenus, extraction de texte, détection de visage et de caractéristiques, etc.). S'inscrit dans la suite de services Google Cloud. Elle a également été utilisée pour développer l'application Face Detection -AI (cf. Tests/essaies préliminaires - Face Detection -AI)

*Spécificités :*

- Accessible aux utilisateurs ayant peu d'expérience dans le domaine du machine learning.

*Restrictions :*

- Analyse des images et non pas des vidéos (cf. Critères initiaux - 2). Une seconde API d'analyse de vidéo (Vidéo Intelligence) est mise à disposition dans le service Cloud mais elle ne semble pas apte à effectuer les tâches que nous souhaitons, Cette dernière se concentre sur la détection de plans, le suivi d'objet etc.
- Essaie gratuit, puis tarification à l'image dépendante des fonctionnalités implémentées.

*Liens :*

<https://cloud.google.com/vision/>

## Détection de caractéristiques

Dans ce chapitre, se trouve les descriptions des différents framework offrant des outils d'analyse de caractéristique d'un visage. Ces données une fois rassemblées permettent aux modèles de machine Learning de nous dire plus sur l'état émotionnel d'une personne.

### *Critères initiaux*

Les critères sont similaires à ceux détaillés au chapitre précédent (cf. Détection de visages - Critères initiaux).

### *Frameworks/APIs relevés :*

De manière similaire au chapitre précédent (cf. . Détection de visages - Frameworks/APIs relevés), ci-dessus le descriptif des outils permettant la détection et l'interprétation des caractéristiques d'un faciès.

### **OpenFace**

API python conçue pour la reconnaissance de visage basé sur FaceNet (FaceNet: A Unified Embedding for Face Recognition and Clustering). Elle se veut très précise, en utilisant un nombre élevé de landmarks.

#### *Spécificités :*

- Utilisation des librairies OpenCV et dlib.
- Open-source.

#### *Liens :*

<https://cmusatyalab.github.io/openface/>

### **Affectiva Emotion SDK**



*Figure 19 : Affectiva*

Affectiva est une entreprise développant des solutions technologiques rattachée au MIT. Elle est active dans le domaine de la science affective (mesure de l'état émotionnel). Elle propose un SDK permettant de détecter sept émotions distinctes et vingt expressions.

*Spécificités :*

- Cross-platform : iOS, Android, Web, Windows, Linux, Mac OS.

*Liens :*

<https://www.affectiva.com/product/emotion-sdk/>

D'autres frameworks analysés plus haut (cf. Détection de visages - Frameworks/APIs relevés) fournissent également la détection des caractéristiques. Il s'agit des suivants :

- ML- kit
- Luxand
- TensorFlow
- Opencv
- Microsoft Azure

*Sélection*

Nous constatons ici que la majeure partie des frameworks permettant la détection de visage sont également aptes à détecter les caractéristiques d'un visage. De plus, suite aux discussions tenues durant les séances des semaines P03 et P04 (cf. PVs), le focus a rapidement été dirigé non plus vers la détection du visage mais la détection des caractéristiques. Une sélection de deux de ces outils a donc été faite pour la partie de test :

- Openface : API puissante, elle a été recommandée par Mr. Carl Daher, doctorant à l'HEIA-FR.
- ML-kit : User friendly, optimisé pour les mobiles, possibilité d'ajout des modèles personnalisés avec TensorFlow Lite, documentation riche.

## Test

L'objectif de ce chapitre est de tester les trois frameworks sélectionnés dans la partie précédente, avec les données fournies par le système HandiControl et de comparer les résultats obtenus.

## Facteurs d'interprétations des intentions de Mr. F.

**Focus sur les labels Encore/change et stop :**

**Spécificité du visage de Mr. F.**

*Interprétation différentes par les modèles ?*

*Emotions vs Micro-expressions*

**Labels ordre/ressentie :**

## Conception

## Business plan

Offre de financement

## Conclusion

## Remerciements

- Beffa et donzollaz
- Mme. Carridade, l'ensemble de l'équipe médicale ainsi que le Home Atelier de la Colline
- Mr. wagen et Hennebert
- Mr. Daher
- Mr. Roche et Mr. Schaer

Signature

Tschopp Bruno

Déclaration d'honneur

Je, soussigné, Bruno Tschopp, déclare sur l'honneur que le travail rendu est le fruit d'un travail personnel. Je certifie ne pas avoir eu recours au plagiat ou à toutes autres formes de fraudes. Toutes les sources d'information utilisées et les citations d'auteur ont été clairement mentionnées.

Glossaire

Nom	Définition	Contexte
Micro-expression		
Émotions		
Caractéristiques	Ensemble de points sur un visage permettant de déduire l'évolution de l'état émotionnel d'une personne. Par ex. la position des sourcils, etc.	
FrameRate		
Framework	Ensemble cohérent de composants logiciels structurels	
API	Ensemble de fonctions, routines et méthodes ayant le but de faciliter l'implémentation.	
SDK	Software Development kit	
ROI		
Landmark		
Modèle		Deep Learning / Machine Learning
Réseau de neurones		Deep Learning
Label		
Histogramme	Représente la distribution des intensités ou des couleurs d'un image donnée.	Imagerie numérique
Flux optique	Mouvement apparent des objets causé par le mouvement entre	

	un observateur (l'œil ou une caméra) et l'objet observé.	
BSD	License libre, permettant la réutilisation d'une partie ou de la totalité d'un logiciel donné sans restriction.	
CUDA	Compute Unified Device Architecture, technologie utilisant un processeur graphique pour exécuter des calculs.	

## Logiciels utilisés

Les logiciels utilisés sont les suivants :

- Android File Transfer, Version 1.0.12

## Annexes

### Procédure d'installation du système HandiControl

#### *Liste du matériel*

- Deux smartphones Android (Android 6.0.1 ou plus)
- Une smart watch
- Un haut-parleur Bluetooth

#### *Compilation et installation des applications*

1. Cloner le git - <https://gitlab.com/jdonzallaz/handicontrol>
2. Activer le mode développeur sur le téléphone et autoriser le débogage USB.
3. Connecter le téléphone de saisie.
4. Naviguer dans le répertoire *HandiContro*.
5. Exécuter en ligne de commande *npm install*
6. Exécuter en ligne de commande *npm install -g react-native-cli*
7. Exécuter en ligne de commande *npm install -g expo-cli*



8. Créer le fichier *local.properties* dans le répertoire *android* et lui ajouter les lignes suivantes :

in Windows `sdk.dir = C:\\Users\\USERNAME\\AppData\\Local\\Android\\sdk`

in macOS `sdk.dir = /Users/USERNAME/Library/Android/sdk`

in linux `sdk.dir = /home/USERNAME/Android/Sdk`

Figure 20 : fichier *local.properties* (source StackOverflow)

9. Assurez-vous d'avoir les droits sur les répertoires *android* et *node\_modules* (`sudo chmod -R 777 dir`)
10. Exécuter en ligne de commande *react-native run-android*
11. Une fois que l'application est bien apparue, il faut déconnecter le téléphone de saisie.
12. Connecter le téléphone de contrôle
13. Naviguer dans le répertoire frère *HandiControlRemote*
14. Répéter les opérations 5 à 10
15. Une fois que l'application est bien apparue, il faut déconnecter le téléphone de contrôle

### Installation du serveur RTSP

1. Installer l'application Dory (d/node.js) depuis le Play Store sur le téléphone de saisie.
2. Donner les permissions à l'application sur le Storage.
3. Ouvrir l'application.
4. Cliquer sur le menu en haut à droite.
5. Sélectionner git clone.
6. Mettre le lien du repository git contenant le serveur RTSP : <https://gitlab.com/yannivain/rtsp-server.git>
7. Sélectionner le lieu d'installation.
8. Appuyer sur "ok".
9. Appuyer ensuite sur le bouton + en bas à droite.
10. Appuyer sur le bouton "Add Directory (package.json's main)".
11. Naviguer jusqu'au répertoire où le git a été cloné.
12. Votre serveur devrait apparaître sur l'écran principal de l'application Dory..
13. Sélectionner Open, puis dans le menu en haut à droite "npm install".
14. Ensuite dans les settings, ajouter "server.js" dans le champ "node option 1".
15. Activer le "start on boot|update".

## *Connecter les périphériques*

Il faut encore connecter les haut-parleurs et la smart-watch en bluetooth au téléphone de saisie. A noter que ces deux derniers périphériques ne sont pas impératifs pour le fonctionnement du système. Une session d'enregistrement est réalisable uniquement avec les deux smartphone Android.

## *Application Dataview*

- Windows :
  - Cloner le git suivant : <https://gitlab.com/jdonzallaz/handicontrol-dataview>
  - Suivre les instructions.(README.md)
- Mac OS :
  - Cloner le git suivant : <https://gitlab.com/jdonzallaz/handicontrol-dataview>
  - Suivre les instructions.(README.md)
  - En cas de bug/problème utiliser le .dmg disponible dans les annexes du projet.

## *Récupération des données*

- Par usb
  - Connecter le téléphone de saisie à l'ordinateur et autoriser l'accès aux données. Si l'ordinateur ne demande pas l'accès par lui-même, dans la barre de notification du téléphone, cliquer sur "*Recharge par USB*" puis choisir "*Utiliser la connexion USB pour Transférer des fichiers*".
  - Ouvrir l'explorateur de fichier ou *Android File Transfer* (pour mac <https://www.android.com/filetransfer/>) et ouvrir le dossier "*Mémoire de stockage interne*".
  - *Les vidéos se trouve dans le dossier /HandiControl-Videos/*
- Serveur SFTP
  - Créer un compte d'accès (cf. Mr. Roche Jean-François HEAI-FR)

Ps : ce document se base sur l'annexe 1 disponible dans le rapport de projet HandiControl rédigé par Mr Beffa et Mr Donzallaz.

## Références :

## Bibliographie

---

- Bruno, T. (2019). *Cahier des charges - HandiControl2*. HEIA-FR.
- Bruno, T. (s.d.). *Procédure d'installation - HandiControl1*. Marche à suivre, HEIA-FR.
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. *CVPR2015*.
- Yann-Ivain Beffa, J. D. (2018/2019). *Rapport HandiControl1*. Projet de semestre 5, HEIA-FR.

### *Figures :*

Figure 1 : Bouton DEMARRER de l'application de saisie HANDICONTROL .....	6
Figure 2 : Bouton ARRETER de l'application de saisie HANDICONTROL .....	7
Figure 3 : Équipement Zyxel servant d'access point pour le home de la colline .....	7
Figure 4 : Machine Learning Workflow .....	9
Figure 5 : Deep Learning Workflow .....	10
Figure 6 : Application Android Face Detection -AI .....	11
Figure 7 : Application Android Luxand Face Recognition.....	12
Figure 8 : Application Android Face Recognition Demo with OpenCV Manager.....	12
Figure 9 : Open Computer Vision .....	13
Figure 10 : TensorFlow .....	14
Figure 11 : Keras .....	15
Figure 12 : Architecture des couches liant TensorFlow et Keras .....	15
Figure 13 : YOLO Real-Time Object Detection.....	16
Figure 14 : Test du modèle pré-entraîné de YOLO.....	16
Figure 15 : Firebase.....	17
Figure 16 : Luxand.....	18
Figure 17 : Microsoft Azure .....	18
Figure 18 : Google Cloud .....	19
Figure 19 : Affectiva .....	20
Figure 20 : fichier local.properties (source StackOverflow) .....	25

### *Sites web :*

<https://fr.wikipedia.org/wiki>

<https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>

<https://archive.ics.uci.edu/ml/datasets/Iris>

<https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>

<https://pjreddie.com/darknet/yolo/>

<https://play.google.com/store/apps/details?id=demo.recgonition.face.facerecognitiondemo>

<https://play.google.com/store/apps/details?id=com.luxand.facerecognition>

<https://play.google.com/store/apps/details?id=com.rebel.user.findfaces>

<https://www.tensorflow.org/>

<https://github.com/tensorflow/tensorflow>

<http://codeinpython.com/tutorials/deep-learning-tensorflow-keras-pytorch/>

<https://github.com/keras-team/keras/tree/master/keras>

<https://keras.io/>

<https://firebase.google.com/products/ml-kit/>

<https://www.luxand.com/facesdk/>

<https://azure.microsoft.com/en-us/services/cognitive-services/face/>

<https://cloud.google.com/vision/>

<https://cloud.google.com/video-intelligence/>

<https://www.affectiva.com/product/emotion-sdk/>

<https://www.youtube.com/watch?v=LhVY0-S7cAM>