



MakeMeLaugh

Interaction multimodale et affective

15.06.2017

Enzo Poggio

Djavan Sergent

Université de Genève

<https://github.com/sergentd/IMA>

Cahier des charges

Projet initial

Le projet initial consistait à proposer une application ayant pour but de faire rire son utilisateur le plus vite et le plus efficacement possible, ou du moins de provoquer des émotions positives chez l'utilisateur. L'application devait prendre en compte les réactions de ses utilisateurs afin d'améliorer continuellement ses performances (temps moyen pour faire rire une personne, pourcentage d'utilisateurs ayant ri, etc.). Celle-ci devait également, dans l'idéal, faire une optimisation entre le "*mood*" de l'utilisateur et les vidéos les plus appropriées pour passer de cet état à celui du rire. La détection du rire et de l'état d'une personne sont donc des éléments centraux du projet tel que présenté initialement.

Le projet devait s'appuyer sur plusieurs modules :

- Le module de visualisation de vidéos
- Le module de choix de vidéos
- Le module d'analyse en temps réel des émotions
- Le module d'apprentissage

Rectification

L'analyse en temps réel par l'utilisation d'OpenFace présente certaines contraintes et difficultés qui rendent l'application initiale trop complexe à mettre en oeuvre dans le temps imparti. Nous avons donc choisi de réorienter le projet afin de nous focaliser sur le module d'apprentissage qui permettrait par la suite d'effectuer un choix de vidéos efficace. Il s'agit donc de mettre en place un modèle d'analyse préliminaire qui permet la mise en oeuvre d'une application telle que décrite dans notre premier cahier des charges.

Notre projet consiste donc à comprendre comment analyser au mieux le visage d'un utilisateur afin d'extraire les informations permettant de classifier une vidéo comme étant drôle ou non et de comparer cette classification avec celle que l'utilisateur aura faite lui-même. Cette classification est indispensable à un système qui propose de faire rire les utilisateurs et il convient donc d'y porter une attention particulière pour obtenir des résultats satisfaisants.

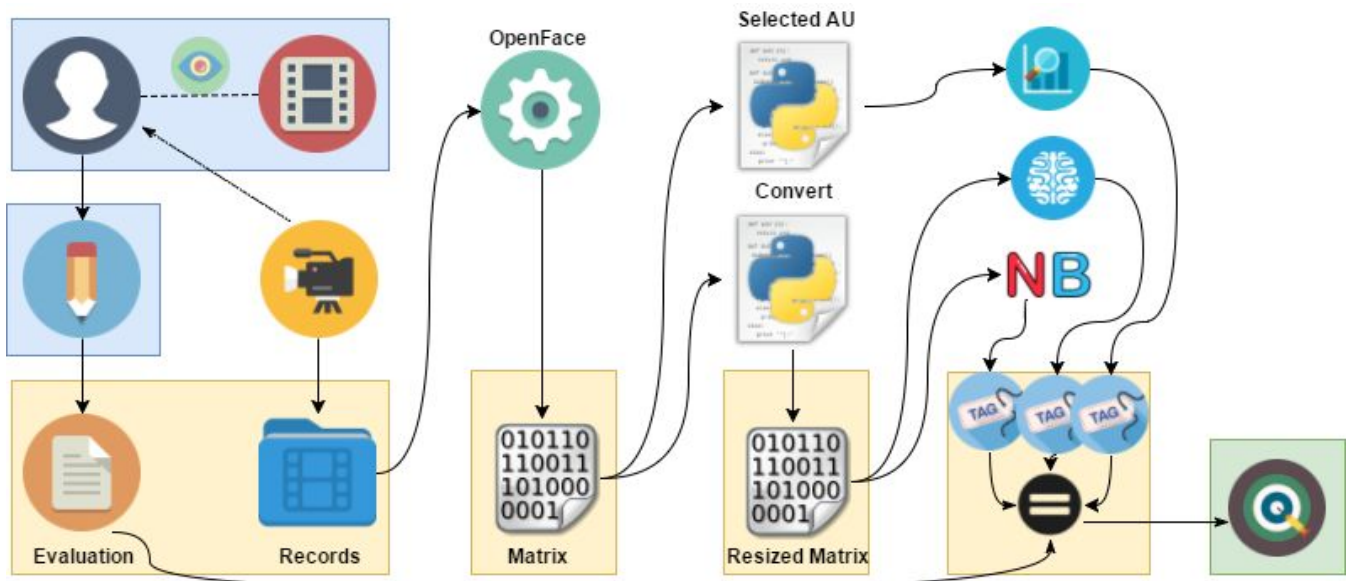
Notre application est donc composée des modules suivants (à droite : effectué par) :

- | | |
|----------------------------------------------|-----------------------------|
| • Visualisation de vidéos | Djavan Sergent |
| • Enregistrement du visage de l'utilisateur | Djavan Sergent |
| • Evaluation d'une vidéo par l'utilisateur | Djavan Sergent |
| • Extraction des Actions Units par OpenFace | Enzo Poggio |
| • Opération de conversion des matrices | Enzo Poggio |
| • Classification basée sur les Actions Units | Djavan Sergent |
| • Classification Naïve Bayes | Enzo Poggio |
| • Classification Neural Network | Enzo Poggio |
| • Évaluation des classifieurs | Enzo Poggio, Djavan Sergent |

- Récupération des données

Enzo Poggio

Schéma général



Blocs bleus : Tâches de l'utilisateur
 Blocs oranges : Output à l'étape courante
 Bloc vert : Evaluation du système

Récupération initiale

1. Un utilisateur regarde une vidéo. Le visage de l'utilisateur est enregistré durant cette tâche.
2. A la fin de la vidéo, l'utilisateur évalue si celle-ci était drôle ou non.
3. Il évalue ensuite son "degré d'expression facial" (c.f plus bas), indépendamment du fait que la vidéo soit drôle ou non.
4. Après l'évaluation, on propose une nouvelle vidéo à l'utilisateur et il refait les mêmes étapes. Il y a 35 vidéos ainsi que 2 vidéos de familiarisation dans le système.

L'utilisateur va donc interagir avec le système par le biais de sa souris (évaluation) et de la caméra (enregistrements pour l'interprétation des émotions).

Traitements des enregistrements (asynchrone)

1. Chaque enregistrement est analysé par OpenFace afin d'extraire les caractéristiques de la vidéo sous la forme d'actions units (AUs). En output nous avons des matrices d'AUs : présence, intensité, fiabilité.
2. On analyse ensuite selon 3 méthodes différentes ces matrices afin d'obtenir un tag:
 - a. La première méthode utilise une analyse basique de présence simultanée de deux AUs avec les matrices issues d'OpenFace sans redimensionnement.
 - b. La deuxième méthode utilise un Naive Bayes avec des matrices redimensionnées, sans sélection d'AUs.
 - c. La troisième méthode utilise un réseau neuronal avec des matrices redimensionnées, sans sélection d'AUs.
3. Le tag est comparé avec l'évaluation de l'utilisateur.

4. Le système entier est évalué sur sa précision.

Hypothèses fondamentales

1 : Une vidéo drôle fait exprimer des AUs liées à la joie à celui qui la regarde

Dans cette première hypothèse, nous cherchons à valider que certaines vidéos font réagir l'utilisateur de façon positive, ce qui influence sa classification, et qu'il est possible de récupérer cette information. Nous nous intéressons donc particulièrement à récupérer les AUs relatives à la joie (6 et 12). Bien que cette hypothèse puisse paraître évidente, il est nécessaire de la valider sans quoi toute l'application ne peut pas fonctionner sur la base de l'analyse du visage.

2 : L'analyse du visage seule permet de savoir si une vidéo est drôle ou non

Dans cette deuxième hypothèse, nous voulons valider que les informations obtenues via OpenFace permettent une meilleure classification qu'une classification aléatoire. On sait déjà que l'analyse des mouvements du visage permettent de détecter certaines émotions. Les AUs décomposent ces mouvements (contractions et décontraction). Nous nous intéressons ici à valider que la mise en relation de ces AUs permet bien de détecter une émotion positive d'une personne et que cela est suffisant pour obtenir des résultats fiables sans croiser ces informations avec d'autres (son par exemple).

Récupération de données

Dans notre système, la récupération des données est effectuée de façon totalement asynchrone de leur évaluation. Nous ne fonctionnons pas en temps réel.

Recherches et développement interface

La visualisation de vidéos est au coeur de notre système. Une bonne partie du début du projet a été investie dans la recherche d'une méthode de visualisation efficace. Nous avons exploré de multiples pistes avant de trouver un moyen élégant d'effectuer cette tâche.

pyMedia

Nous voulions initialement utiliser une librairie qui propose un lecteur de vidéos classique en Python. Cette idée est vite devenue problématique puisque les librairies en question utilisent une version de Python antérieure à celle que nous utilisons pour notre projet. Nous avons rencontré de nombreux problèmes de compatibilité, elles sont donc impossibles à utiliser en l'état et il n'existe pas de version plus récente.

OpenCV

Nous devons en parallèle de l'affichage d'une vidéo enregistrer le visage de l'utilisateur, ce que nous avons fait au moyen d'OpenCV. Nous nous sommes alors demandé si nous ne pouvions pas utiliser OpenCV pour afficher nos vidéos également. Il nous restait encore à lire le son de la vidéo puisque OpenCV ne prend pas du tout en charge cet aspect.

Nous avons donc mis en place le module de lecture de médias audio de pyGame. Cependant, ce module nécessite d'avoir à sa disposition des fichiers audio et non pas des fichiers vidéos, ce qui a introduit un pré traitement sur toutes les vidéos pour en extraire le son avec le logiciel ffmpeg. Chaque changement de vidéo à montrer aux utilisateurs implique donc de devoir refaire ce pré traitement.

Bien que cette solution soit fonctionnelle du point de vue de la lecture de vidéos et du son, nous avons par contre rencontré de gros problèmes de synchronisation. Il était strictement impossible de réussir à synchroniser ces deux éléments, ce qui rend cette solution inutilisable.

iPython

Nous avons réorienté la recherche d'une solution vers la puissance des navigateurs web, au travers d'un notebook, qui permettent une excellente gestion de ces médias. Il était ainsi facile de visualiser une vidéo et d'avoir le son synchronisé. Nous n'avons par contre par trouvé le moyen de détecter la fin de la vidéo de façon automatique afin d'arrêter l'enregistrement parallèle. Cette solution est donc totalement incompatible avec notre projet.

VLC (solution retenue)

Notre solution finale consiste en l'utilisation d'un programme tiers pour la lecture de vidéos. Nous avons choisi VLC pour son aspect multi-plateforme et ses nombreuses options de lecture. Pour chaque vidéo, nous démarrons le lecteur en plein écran et en mode "dummy interface", ce qui implique qu'aucun bouton de contrôle n'est disponible (pause, stop etc.). On demande également à VLC de se fermer directement après la lecture de la vidéo, ce qui permet à notre application de reprendre son cours automatiquement.

Cette solution présente les avantages suivants :

- Détection implicite de la fin d'une vidéo (le lecteur se ferme automatiquement)
- Pas de boutons ou autre contenu sur le lecteur
- Vidéo et audio synchronisés et unifiés
- Pas de pré traitement sur les vidéos pour en extraire le son avec un logiciel tiers, donc possibilité de modifier notre liste de lecture sans problème
- Facile d'utilisation
- Si l'utilisateur met en pause la vidéo, cela ne perturbe pas la captation qui continue jusqu'à la fin de la vidéo.

Capture et visualisation de vidéos

Notre application utilise OpenCV pour la captation de la vidéo du visage de l'utilisateur et VLC pour afficher les vidéos présélectionnées. La lecture et l'enregistrement se font en parallèle.

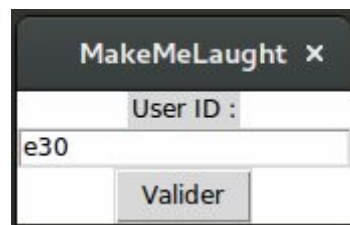
Nous utilisons les librairies et modules suivants :

- subprocess : permet de lancer un programme tiers depuis Python
- threading : permet la parallélisation des tâches
- cv2 : OpenCV

Interfaces interactives finales

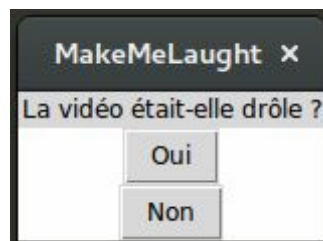
Identifiant d'utilisateur

Chaque personne ayant participé à l'expérience s'est vu assigner un identifiant unique. Cet identifiant ne permet pas de retrouver la personne ayant participé à l'expérience mais permet en revanche de différencier les captations. La fenêtre présente un champ de texte non restrictif (aucune vérification de donnée effectuée, l'identifiant importe peu tant qu'il est unique).

A screenshot of a software window titled "MakeMeLaught" with a close button (X) in the top right corner. Inside the window, there is a label "User ID :" followed by a text input field containing the text "e30". Below the input field is a button labeled "Valider".

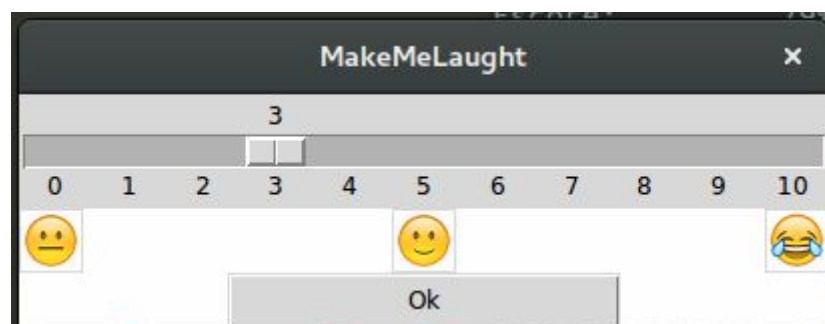
Evaluation de la vidéo

Pour chaque vidéo que l'utilisateur a regardé, il doit évaluer s'il considère celle-ci comme étant drôle ou non. La fenêtre ne présente que deux boutons qui représentent la classification par l'utilisateur.

A screenshot of a software window titled "MakeMeLaught" with a close button (X) in the top right corner. The window displays the question "La vidéo était-elle drôle ?". Below the question are two buttons: "Oui" (Yes) and "Non" (No).

Évaluation du degré d'expression faciale

Le degré d'expression faciale nous indique à quel point l'utilisateur pense avoir réagi par rapport au contenu de la vidéo. Cette information est totalement indépendante du fait qu'une vidéo soit drôle ou non mais permet plutôt d'évaluer à quel point elle est ou n'est pas drôle. Par exemple si l'utilisateur a exprimé du dégoût ou de la peur et non pas de la joie, il devra tout de même évaluer son degré d'expression faciale. La fenêtre propose un slider qui va de 0 à 10 et un bouton qui permet de valider son choix.

A screenshot of a software window titled "MakeMeLaught" with a close button (X) in the top right corner. The window features a horizontal slider scale from 0 to 10. Above the scale, the number "3" is displayed, and a slider handle is positioned at the value 3. Below the scale, there are three emoji icons: a neutral face at 0, a smiling face at 5, and a laughing face with tears at 10. At the bottom of the window is a button labeled "Ok".

Protocole expérimental

Notre protocole expérimental est composé de 3 parties : un questionnaire, une petite familiarisation avec le système et l'accomplissement de la tâche principale.

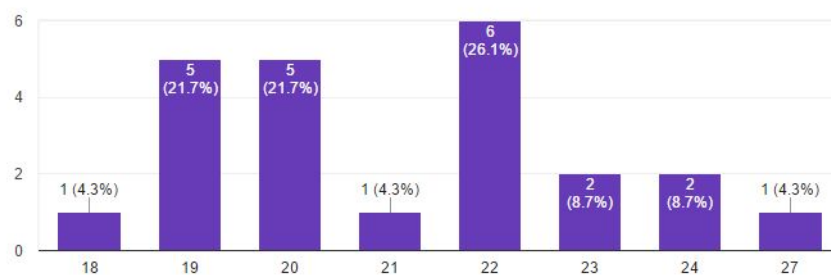
Questionnaire

Notre questionnaire se présente sous la forme d'un Google Form. Dans ce questionnaire, nous demandons à l'utilisateur de décrire certaines de ses caractéristiques (âge, sexe, état émotionnel). A ce stade du projet, ces données ne sont pas encore utilisées mais pourraient mener à de nouvelles analyses de résultats par exemple de savoir si ce sont les mêmes vidéos qui font rire à tout âge ou si la notion d'humour évolue avec le temps.

L'utilisateur donne son consentement explicite pour l'utilisation de ces informations à des fins de recherche.

Quel est votre age ?

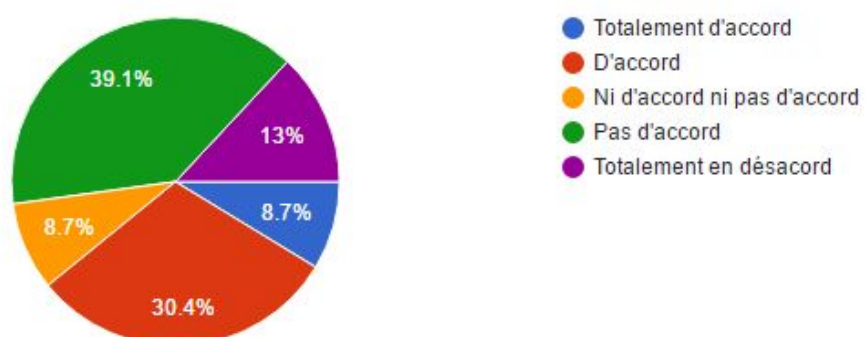
23 responses



État Émotionnel

Je suis stressé en ce moment

23 responses



Familiarisation

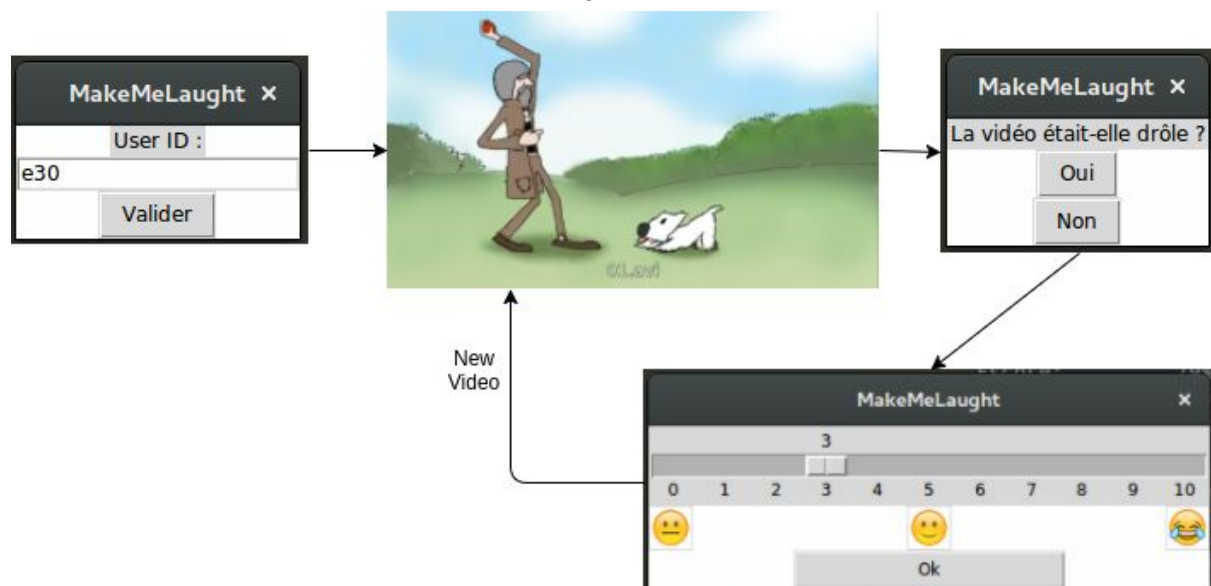
Afin que l'utilisateur se familiarise avec le système, il doit faire deux fois le processus complet d'évaluation d'une vidéo (visionnage, classification, évaluation du degré d'expression faciale). C'est à ce moment que nous rappelons également à l'utilisateur qu'il ne doit pas cacher son visage avec sa main et comment évaluer son degré d'expression faciale. Nous n'indiquons pas clairement à l'utilisateur qu'il sera filmé pendant l'expérience. Lors de cette phase, tous les participants à l'expérience visionnent les deux mêmes vidéos dans le même ordre.

Visionnage

Après avoir saisi son identifiant et s'être familiarisé avec le système, la tâche de l'utilisateur consiste à :

1. Regarder 35 vidéos d'une durée de plus ou moins 10 secondes
2. Pour chacune d'elles et juste après l'avoir visionnée :
 - a. Classifier la vidéo (drôle ou pas)
 - b. Évaluer son degré d'expression faciale

Schématiquement, le processus de visionnage est le suivant :



La durée de l'expérience est comprise entre 7 et 10 minutes, dépendamment de la rapidité de l'utilisateur à remplir la deuxième étape. Chaque participant visionne les vidéos dans un ordre différent afin de limiter l'impact résiduel d'une vidéo drôle sur le visage de l'utilisateur. Si tous les utilisateurs visionnent les vidéos dans le même ordre, les résultats d'une vidéo pourraient être directement impactés par la vidéo précédente. En mélangeant l'ordre, on diminue l'impact de ce risque sans pour autant l'atténuer complètement. La tâche d'évaluation de la vidéo tend également à ce que l'expression du visage de l'utilisateur revienne à un état neutre.

Analyses préliminaires des données

Hypothèses

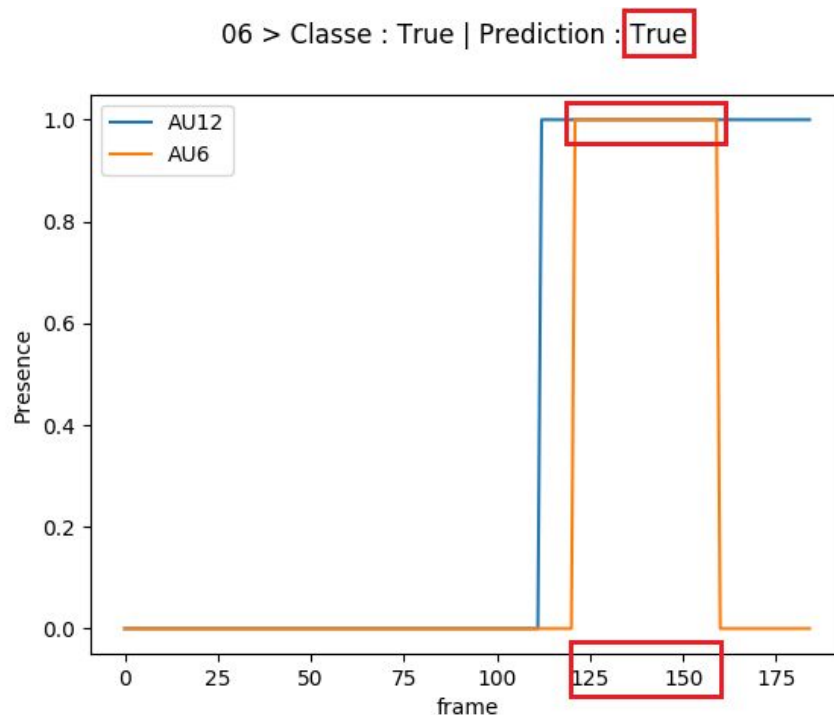
On cherche ici à valider nos hypothèses fondamentales 1 et 2.

Ensemble de données

Pour chaque matrice générée par OpenFace de chaque enregistrement de chaque utilisateur, nous prenons en entrée la présence/absence de l'AU6 et de l'AU12 sur chaque frame de l'enregistrement :

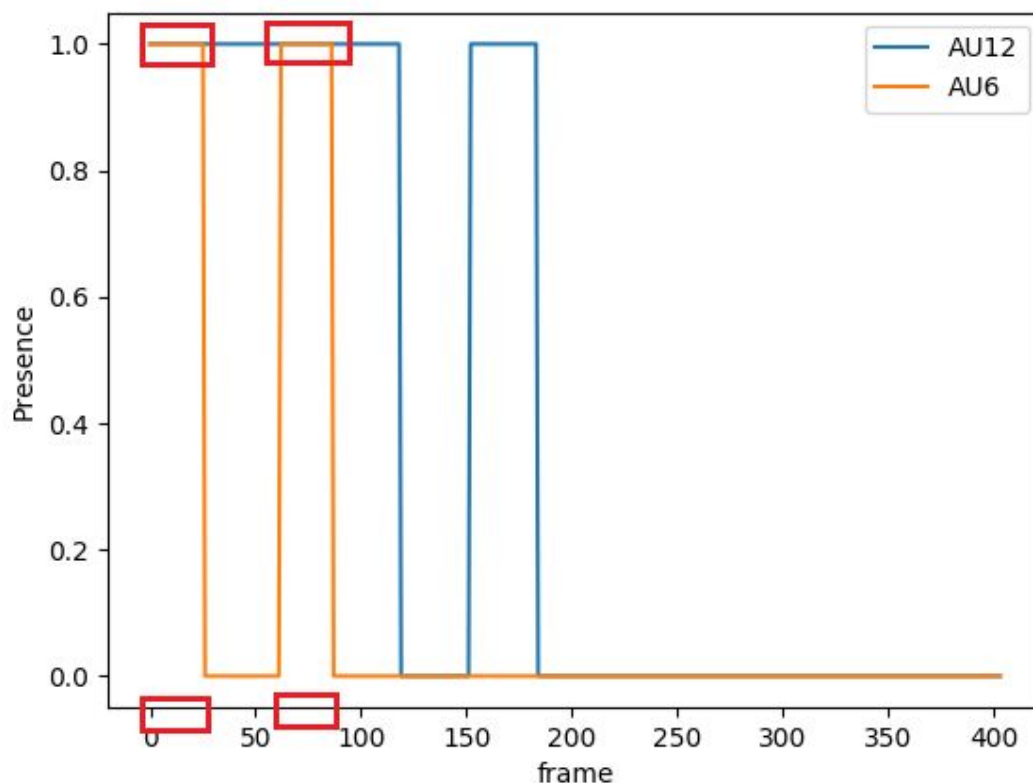
| Frame | AU6_c | AU12_c |
|-------|-------|--------|
| 1 | 1 | 1 |
| 2 | 0 | 1 |
| ... | ... | ... |

On ne tient ni compte de l'intensité de l'AU ni de la fiabilité de la prédiction d'OpenFace. Nous comptons le nombre d'occurrences simultanées de ces AUs sur une même frame et comparons ça avec le nombre total de frames. Si on constate que ce nombre d'occurrences est plus grand ou égal à 20% du nombre total de frames (valeur empirique pour des vidéos d'une durée d'environ 10 secondes) on classe la vidéo comme étant drôle.



Exemple 1 : présence simultanée suffisante, catégorie drôle

27 > Classe : False | Prediction : **False**



Exemple 2 : présence simultanée insuffisante, catégorie non drôle

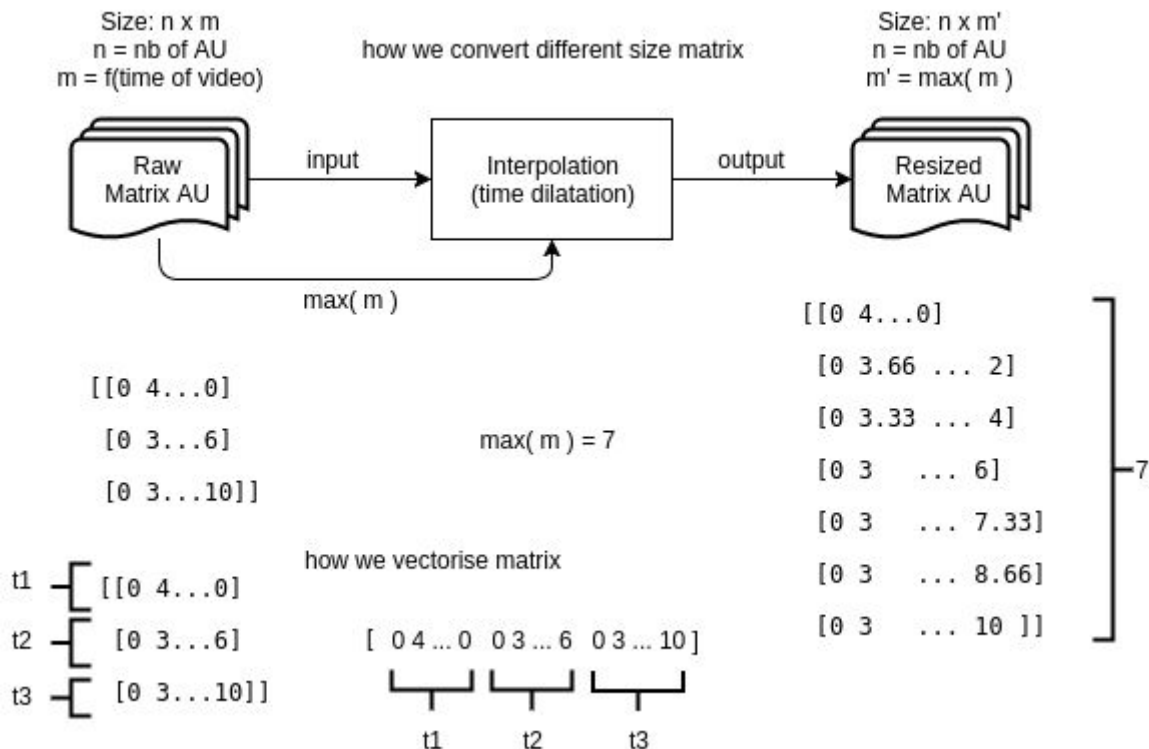
Le résultat d'une analyse est uniquement défini par l'enregistrement et n'est pas influencé par d'autres enregistrements ou classifications antérieures

Résultats de l'analyse préliminaire

Sur notre corpus complet d'enregistrements, notre prédiction se révèle juste dans **78.43%** des cas, soit nettement supérieure à une classification aléatoire. Ce résultat confirme que baser la prédiction de nos classes sur la base d'AUs relatives à la joie est pertinent.

Dans notre cas, l'analyse est volontairement restée basique (deux AUs de poids identique) et est réellement spécifique à notre corpus de vidéo puisque celles-ci sont relativement courtes. Une telle approche devrait subir d'importants ajustements dans un cas d'application sur des vidéos plus longues pour lesquelles une valeur de 20% de présence simultanée peut être inappropriée.

Preprocessing des données d'entrée



Afin d'uniformiser nos données, nous devons les pré-traiter. Pour ce faire, nous avons procédé à une "dilatation du temps". Par ces termes, nous voulons exprimer le fait que nous avons ajusté la taille toutes les matrices d'AUs obtenues avec OpenFace, par rapport à la plus grande. Pour ce faire, nous avons fait une interpolation unidimensionnelle afin d'agrandir la dimension m qui représente chaque mesure d'OpenFace dans le temps. On obtient alors des matrices de taille $n \times m'$ (n étant le nombre d'AUs et m' le nombre maximal de dimensions des matrices d'origine). Ensuite, afin de pouvoir manipuler facilement ces matrices, nous les avons aplaties sur une dimension en concaténant les différents vecteurs de la dimension m' .

Modèles utilisés de machine learning:

Aléatoire

Afin d'avoir un modèle de comparaison, nous avons créé un modèle témoin aléatoire. Celui-ci attribue pour chaque matrices un label "funny" ou "not funny" (par un booléen) aléatoirement et un nombre aléatoire entre 0 et 10 pour le degré d'expression faciale.

Nous n'exposerons pas les résultats de ce modèle ici, car, comme vous avez pu le deviner, ces résultats sont inconsistants. Mais ils oscillent entre les extrêmes sans atteindre quelque chose de cohérent. Ce modèle est juste un point de comparaison, si les modèles suivants n'ont pas des résultats invariants.

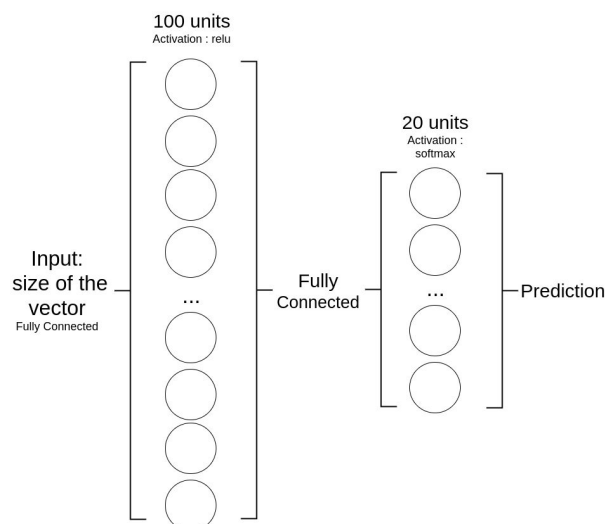
Naive bayes

Un premier modèle statistique simple que nous avons réalisé est un modèles *naive bayes*. Ce type de modèle aide à faire une première analyse d'un corpus de données. Nous avons utilisé la librairie scikit-learn pour le réaliser. Ce modèle utilise la probabilité a priori de faire partie d'une classe $p(C)$ et la probabilité d'un trait A , sachant une classe particulière donnant $P(A|C)$ (ici étant le produit de toutes les probabilités qu'un AUs signifie une classe particulière). Grâce à une fonction de maximisation, on peut obtenir une classification à partir de la liste des produits de chaque combinaison possible trait/classe.

$$P(C | A) = \frac{P(A | C)P(C)}{P(A)}$$

Neural network

Notre troisième modèle est un réseau neuronal propulsé par Keras dans un environnement TensorFlow. Celui-ci n'a pas de couche cachée. Il a juste une couche d'entrée de 100 unités, s'activant grâce à *relu* et une couche de sortie de 20 unités, s'activant grâce à une fonction *softmax*. Le nombre de couches d'unités et les activations ont été choisis de manière totalement arbitraire. En effet c'est en faisant plusieurs petits tests que nous avons remarqué que le réseau donnait de bons résultats avec cette configuration-là.



Configurations

Nous avons fait plusieurs configurations au niveau de l'entraînement du modèle. Cela nous a permis de mieux comprendre comment réagissait notre modèle. Nous verrons donc des résultats pour 5, 50 et 150 périodes d'entraînement. Par contre, les mesures proviendront

toujours du modèle entraîné pendant 50 périodes. Certains retours critiques utiliseront les données des autres modèles entraînés sur 5 et 150 périodes.

Hypothèses

Dans cette section, nous allons présenter nos hypothèses et les ensembles de données relatifs, exposer et interpréter nos résultats pour chaque modèle, puis faire un retour critique sur les réseaux neuronaux et leur utilisation.

Test des modèles

Avant de se lancer dans la vérification d'hypothèses complexes, nous avons essayé de voir si nos modèles étaient cohérents et s'ils donnaient des résultats allant dans le sens attendu. Ainsi pour tester nos modèles, nous les avons sommés de réaliser la tâche de reconnaître subjectivement le rire chez une personne en particulier.

Beaucoup de conventions de notation, de mesure ou de lecture des données sont expliquées dans cette sous section qui seront utiles pour les deux sections suivantes, et qui ne seront certainement pas répétées.

Ensemble de données

| Id Usr \ Id Vid | Vid 01 | Vid 02 | . | . | . | . | . | Vid 10 | . | . | . | . | Vid 20 | . | . | . | . | Vid 34 | Vid 35 |
|-----------------------|-----------|-----------|---|---|---|---|---|-----------|---|---|---|---|-----------|---|---|---|---|-----------|-----------|
| Usr XX | mt | mt | . | . | . | . | . | mt | . | . | . | . | mt | . | . | . | . | mt | mt |

Dans ce tableau sont représentées les matrices (mt) d'un utilisateur XX. Les matrices en jaune représentent l'ensemble de test et les matrices en blanc l'ensemble d'entraînement.

Pour chaque utilisateur, nous avons pris 10 matrices pour l'ensemble de test et 25 pour l'ensemble d'entraînement. Soit quand on ré-additionne le tout, 200 matrices pour l'ensemble de test et 500 pour l'ensemble d'entraînement pour 20 entraînements (un par utilisateur).

Résultats

Pour chaque modèle nous exposons la table de confusion en fonction des labels que celui-ci s'est vu attribué et les différentes mesures liées.

Naive bayes

Pour Naive bayes, nous avons utilisé nos deux labels. Les résultats de la classification d'expression faciale n'étant pas très probants, ceux-ci n'ont pas été répétés pour les réseaux neuronaux. En outre, on voit clairement se démarquer une diagonale ou les valeurs les plus grandes se retrouvent. Ceci montre que le modèle classe certes en faisant des

erreurs, mais il n'est jamais très loin de la solution. Nous avons donc créé une mesure le pourcentage d'erreur qui est la moyenne de la valeur absolue des différences entre le label et la prédiction. Plus cette mesure est basse, plus le modèle peut être considéré comme efficient.

| C:Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | All |
|-------------|----|----|----|----|---|----|----|----|---|---|----|-----|
| 0 | 26 | 12 | 3 | 0 | 3 | 5 | 0 | 0 | 0 | 0 | 0 | 49 |
| 1 | 9 | 4 | 1 | 3 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 20 |
| 2 | 3 | 6 | 2 | 5 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 19 |
| 3 | 1 | 3 | 4 | 2 | 1 | 5 | 3 | 4 | 1 | 1 | 1 | 26 |
| 4 | 0 | 4 | 4 | 1 | 1 | 3 | 2 | 2 | 0 | 0 | 0 | 17 |
| 5 | 1 | 2 | 3 | 1 | 1 | 2 | 2 | 1 | 0 | 0 | 2 | 15 |
| 6 | 0 | 0 | 1 | 1 | 0 | 7 | 5 | 5 | 0 | 1 | 0 | 20 |
| 7 | 1 | 0 | 1 | 2 | 0 | 3 | 5 | 2 | 0 | 3 | 0 | 17 |
| 8 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 2 | 2 | 1 | 0 | 9 |
| 9 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 1 | 1 | 6 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| All | 41 | 31 | 19 | 15 | 7 | 30 | 20 | 19 | 5 | 8 | 5 | 200 |

Error percent: 17%

Le modèle est plutôt cohérent. Il n'a fait que 17% d'erreurs. En même temps, le degré d'expression faciale est une donnée très difficile à reconnaître. En effet, premièrement, il provient d'une donnée totalement subjective d'auto-évaluation (ce qui peut être difficile à estimer pour l'utilisateur). Deuxièmement, l'ensemble d'entraînement est trop petit et pas assez homogène. Et troisièmement, notre algorithme naïve bayes classifie par label et ne recalcule pas le degré d'expression faciale comme un score. Cela a pour effet que si à l'entraînement il n'a jamais vu un label de test, il ne pourra pas le deviner.

| C:Predicted | False | True | All | Accuracy: 80% |
|-------------|-------|------|-----|----------------|
| False | 76 | 28 | 104 | Fscore: 80% |
| True | 13 | 83 | 96 | Precision: 75% |
| All | 89 | 111 | 200 | Recall: 86% |

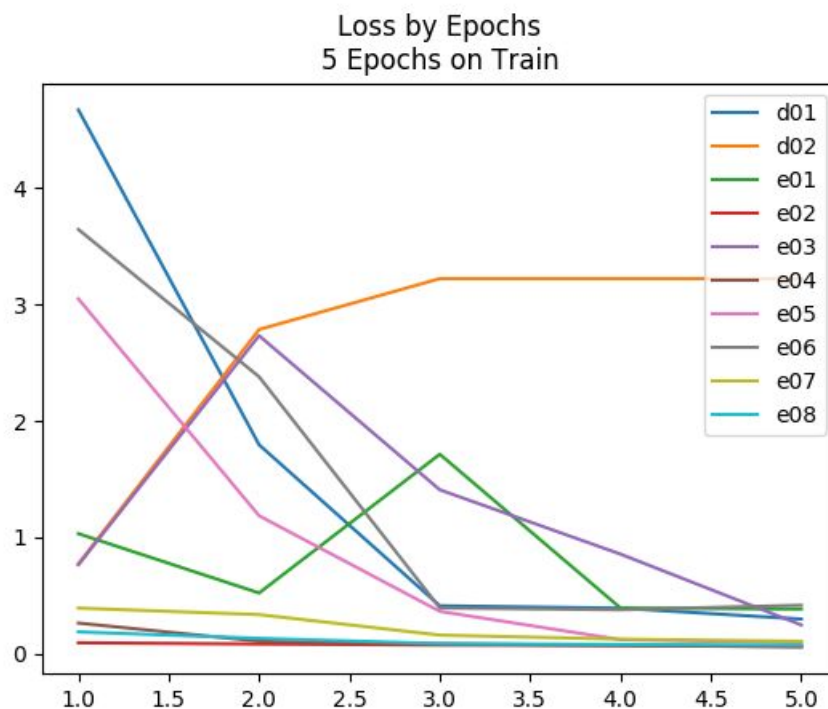
La précision et le rappel sont exprimé en fonction du label True pour toutes les mesures de cette section.

Nous voyons une forte exactitude, cependant comme nous l'avons vu précédemment dans les "Analyses préliminaires des données", il est assez facile d'avoir de bons résultats en se basant uniquement sur deux AUs. Donc cela est un succès à relativiser.

Neural Network

| C:Predicted | False | True | All | Accuracy: 83% |
|-------------|-------|------|-----|----------------|
| False | 87 | 17 | 104 | Fscore: 82% |
| True | 17 | 79 | 96 | Precision: 82% |
| All | 104 | 96 | 200 | Recall: 82% |

Sans grande surprise, le réseau neuronal dépasse de peu naive bayes. La précision et le rappel s'uniformisent au profit de l'exactitude.



Pour que les graphiques restent visibles et compréhensibles, nous n'afficherons que les dix premières courbes pour chaque ensemble de données.

Le système n'est-il pas sur-entraîné pour rien, sachant que la majorité des descentes de gradient se fait sur les cinq premières périodes!

Hypothèse 1

Ici nous tentons de reconnaître le rire de manière universelle et ainsi de déterminer si une vidéo est drôle ou non pour un utilisateur particulier.

Ensemble de données

| Id Usr \ Id Vid | Vid 01 | Vid 02 | Vid 03 | ... | Vid 35 |
|-----------------|------------|------------|------------|-----|------------|
| Usr 01 | mt[01][01] | mt[01][02] | mt[01][03] | ... | mt[01][35] |
| Usr 02 | mt[02][01] | mt[02][02] | mt[02][03] | ... | mt[02][35] |
| Usr 03 | mt[03][01] | mt[03][02] | mt[03][03] | ... | mt[03][35] |
| ... | ... | ... | ... | ... | ... |
| Usr 20 | mt[20][01] | mt[20][02] | mt[20][03] | ... | mt[20][35] |

Pour chaque utilisateur, ses matrices ont fait l'objet d'un test par rapport aux matrices de tous les autres utilisateurs. Soit pour chacun des vingt utilisateurs, on a fait un entraînement sur 665 matrices pour classifier les 35 matrices de l'utilisateur concerné.

Résultats

Naive bayes

| C:Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | All |
|-------------|-----|----|----|-----|----|----|----|----|----|----|----|-----|
| 0 | 75 | 19 | 7 | 16 | 4 | 4 | 3 | 0 | 0 | 0 | 0 | 128 |
| 1 | 22 | 10 | 9 | 14 | 3 | 5 | 1 | 4 | 0 | 0 | 0 | 68 |
| 2 | 16 | 14 | 4 | 10 | 5 | 5 | 2 | 2 | 2 | 0 | 0 | 60 |
| 3 | 17 | 16 | 4 | 15 | 2 | 6 | 4 | 4 | 1 | 4 | 1 | 74 |
| 4 | 11 | 8 | 2 | 9 | 5 | 9 | 7 | 3 | 4 | 0 | 0 | 58 |
| 5 | 8 | 6 | 4 | 16 | 3 | 6 | 9 | 12 | 4 | 7 | 0 | 75 |
| 6 | 1 | 6 | 1 | 13 | 4 | 12 | 16 | 8 | 7 | 11 | 2 | 81 |
| 7 | 1 | 5 | 1 | 7 | 0 | 6 | 9 | 16 | 5 | 8 | 1 | 59 |
| 8 | 0 | 0 | 2 | 2 | 0 | 2 | 8 | 11 | 6 | 13 | 1 | 45 |
| 9 | 1 | 0 | 1 | 2 | 0 | 2 | 5 | 8 | 2 | 10 | 1 | 32 |
| 10 | 0 | 0 | 0 | 3 | 0 | 2 | 0 | 4 | 3 | 8 | 0 | 20 |
| All | 152 | 84 | 35 | 107 | 26 | 59 | 64 | 72 | 34 | 61 | 6 | 700 |

Error percent: 19%

Le pourcentage d'erreur n'est pas très haut mais l'exactitude non plus. La diagonale se démarque toujours ce qui veut quand même dire que le modèle est cohérent.

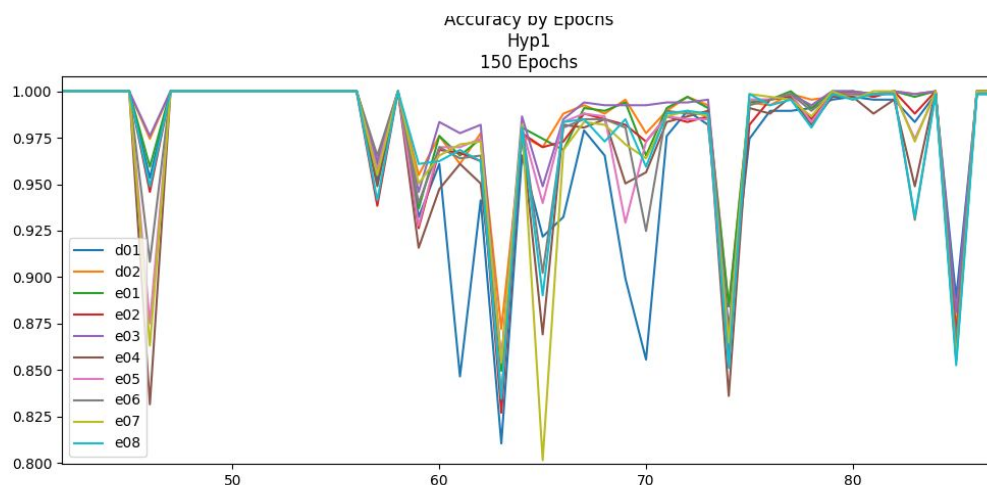
| C:Predicted | False | True | All | |
|-------------|-------|------|-----|----------------|
| False | 257 | 51 | 308 | Accuracy: 76% |
| True | 118 | 274 | 392 | Fscore: 76% |
| All | 375 | 325 | 700 | Precision: 84% |
| | | | | Recall: 70% |

On arrive à des résultats plutôt bons avec une forte précision. Cependant cette tâche reste très difficile. Car on n'a jamais vu comment l'utilisateur testé riait auparavant et on doit l'évaluer par rapport aux autres. Donc si l'utilisateur ne rit pas de manière conventionnelle, le système se trompera.

Neural Network

| C:Predicted | False | True | All | |
|-------------|-------|------|-----|----------------|
| False | 218 | 90 | 308 | Accuracy: 74% |
| True | 92 | 300 | 392 | Fscore: 77% |
| All | 310 | 390 | 700 | Precision: 77% |
| | | | | Recall: 77% |

Ce que l'on n'explique pas, par contre, c'est que les résultats du réseau neuronal sont moins bons que les résultats de naive bayes.



Pourtant, pendant l'entraînement, il n'y avait pas de "zone de turbulences" dans l'exactitude pour 50 périodes d'entraînement. Mais peut-être que la turbulences entre 40 et 50 biaise le modèle.

Hypothèse 2

Ici nous tentons d'identifier une vidéo comme drôle ou non sur la base des réactions des utilisateurs voyant ces vidéos.

Ensemble de données

| Id Usr \ Id Vid | Vid 01 | Vid 02 | Vid 03 | ... | Vid 35 |
|-----------------|------------|------------|------------|-----|------------|
| Usr 01 | mt[01][01] | mt[01][02] | mt[01][03] | ... | mt[01][35] |
| Usr 02 | mt[02][01] | mt[02][02] | mt[02][03] | ... | mt[02][35] |
| Usr 03 | mt[03][01] | mt[03][02] | mt[03][03] | ... | mt[03][35] |
| ... | ... | ... | ... | ... | ... |
| Usr 20 | mt[20][01] | mt[20][02] | mt[20][03] | ... | mt[20][35] |

Pour chaque vidéo, leurs matrices ont fait l'objet d'un test par rapport aux matrices de tout les autres vidéos. Soit pour chacune des 35 vidéos, on a fait un entraînement sur 680 matrices pour classifier les 20 matrices de la vidéo concernée.

Résultats

Naive bayes

| C:Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | All |
|-------------|-----|----|----|-----|----|----|----|----|----|----|----|-----|
| 0 | 76 | 24 | 3 | 11 | 9 | 2 | 3 | 0 | 0 | 0 | 0 | 128 |
| 1 | 22 | 14 | 8 | 13 | 4 | 3 | 0 | 4 | 0 | 0 | 0 | 68 |
| 2 | 13 | 18 | 1 | 16 | 4 | 4 | 1 | 1 | 2 | 0 | 0 | 60 |
| 3 | 17 | 13 | 4 | 16 | 3 | 10 | 4 | 2 | 1 | 3 | 1 | 74 |
| 4 | 11 | 8 | 4 | 10 | 3 | 11 | 6 | 2 | 3 | 0 | 0 | 58 |
| 5 | 7 | 5 | 2 | 18 | 5 | 4 | 14 | 9 | 4 | 7 | 0 | 75 |
| 6 | 2 | 3 | 2 | 15 | 3 | 10 | 17 | 6 | 8 | 13 | 2 | 81 |
| 7 | 1 | 1 | 1 | 11 | 0 | 8 | 7 | 14 | 7 | 7 | 2 | 59 |
| 8 | 0 | 0 | 2 | 2 | 1 | 1 | 7 | 10 | 7 | 14 | 1 | 45 |
| 9 | 1 | 0 | 1 | 3 | 0 | 0 | 5 | 4 | 7 | 10 | 1 | 32 |
| 10 | 0 | 0 | 0 | 3 | 0 | 2 | 0 | 5 | 3 | 6 | 1 | 20 |
| All | 150 | 86 | 28 | 118 | 32 | 55 | 64 | 57 | 42 | 60 | 8 | 700 |

Error percent: 18%

La matrice de confusion pour les degrés d'expression faciale ne nous montre rien de plus particulier que les deux précédentes, à part peut-être le fait que l'on classifie plutôt bien les matrices avec un degré d'expression faciale faible.

| C:Predicted | False | True | All | |
|-------------|-------|------|-----|----------------|
| False | 260 | 48 | 308 | Accuracy: 77% |
| True | 110 | 282 | 392 | Fscore: 78% |
| All | 370 | 330 | 700 | Precision: 85% |
| | | | | Recall: 72% |

Le modèle naive bayes donne des résultats proches des analyses préliminaires, ce qui est normal. Cette tâche est celle qui se rapproche le plus de cette analyse.

Neural Network

| C:Predicted | False | True | All | |
|-------------|-------|------|-----|----------------|
| False | 244 | 64 | 308 | Accuracy: 83% |
| True | 56 | 336 | 392 | Fscore: 85% |
| All | 300 | 400 | 700 | Precision: 84% |
| | | | | Recall: 86% |

En ce qui concerne le réseau neuronal, il donne de très bons résultats et avance de 6 points par rapport à Naive bayes. Par contre, on peut se demander si un gros entraînement est nécessaire, car avec cinq périodes, on arrive au même résultat qu'avec 50 périodes. Et avec 150 périodes, on ne gagne que 1 point d'exactitude (84%).

Conclusion

Enzo

Ce travail nous montre typiquement qu'il est difficile de cerner une émotion, que ce soit dans les interactions d'un utilisateur avec une interface, ou dans l'extraction de celle-ci à partir des données. Nous sommes arrivés à des résultats intéressants, mais il est difficile d'outrepasser ceux-ci. Nos hypothèses ne sont donc pas vérifiées à 100%. Par ailleurs, il aurait été nécessaire d'avoir plus de temps pour discuter de la méthodologie à adopter. Ainsi, de ce projet, il reste un corpus bien constitué, qui permet de faire de multiples hypothèses et comparaisons. Il demanderait à être complété par d'autres interactions multimodales comme le son de l'utilisateur pendant l'expérience par exemple (ce que nous n'avons pas fait pour deux raisons: la première, nous n'y avons pensé que trop tard; la deuxième, il faudrait une espace d'expérimentation très calme et dédié à cette expérience).

Djavan

Bien que nous n'ayons pas pu valider complètement chacune de nos hypothèses, les résultats obtenus dans le cadre de l'analyse préliminaire nous montre que sur la base de peu d'informations il est possible d'avoir une classification juste dans près de 80% des cas. Bien que les méthodes d'apprentissage n'aient pas apporté d'amélioration très significatives, elles tendent à prouver qu'un modèle bien conçu serait apte à donner des résultats nettement supérieurs. Enfin, bien qu'elles offrent de nombreuses possibilités, les informations relatives à l'utilisateur (âge, sexe, humeur) étaient bien trop complexes à interpréter et à intégrer à notre logique de prédiction.