

## ► AM2. Visage Humain : Reconnaissance, Interaction, Animation

Objectif : Classification et reconnaissance des émotions sur un visage humain en utilisant un réseau de neurones

### Table des matières

1. Présentation du sujet.....	2
2. Mise en place de la solution.....	2
Démarrage du projet :.....	2
Utilisation du réseau de neurones :.....	3
Adaptation du réseau de neurones a notre objectif :.....	3
Utilisation des résultats :.....	4
3. Les problèmes rencontrés.....	5
4. Conclusion.....	5

Auteurs :

Nour MEDJEDEL 11411470  
Louis POITEVIN 11410541  
Thibault DUMENIL 11306146



## 1. Présentation du sujet

Le domaine de travail choisi étant "Visage Humain : Reconnaissance, Interaction, Animation", nous avons décidé de nous diriger vers la partie reconnaissance de ce thème. Dans un premier temps nous voulions travailler sur un programme de reconnaissance facial, c'est à dire soumettre une photo de quelqu'un a notre programme et qu'il soit capable de nous dire le nom de cette personne, à partir d'une base de données. Après quelques semaines sur ce sujet nous nous sommes rendu compte que nos outils été bien trop imprécis pour ce genre de reconnaissance et qu'il nous faudrait une énorme base de données de photo pour que le programme puisse reconnaître une personne avec un bon taux de réussite. Nous nous sommes donc mis d'accord pour changer légèrement de but et pour travailler sur l'objectif suivant : Reconnaissance et classification des émotions sur un visage humain. Notre but été de présenter un programme capable de reconnaître les émotions présentes sur un visage humain via une photo, vidéo ou un flux webcam. Les émotions que nous voulions reconnaître été aux nombres de 6 : la colère, le dégoût, la peur, la joie, la tristesse et la surprise. Finalement, et toujours dans un but d'obtenir des résultats précis, nous avons convenue de reconnaître seulement deux émotions, la joie et la colère, et de généraliser cela a une émotion positive (la joie) et une négative (la colère).

## 2. Mise en place de la solution

### Démarrage du projet :

Les premières semaines de ce projet nous nous sommes donc attelés à apprendre à se servir de la librairie OpenCV et toute ces composantes (dlib, imutils, numpy ...). Après cette étape nous avons tous les outils pour utiliser les 68 points caractéristiques d'un visage que dlib extrait à partir d'une photo, vidéo ou de la webcam. De plus, notre professeur référent nous a fourni un dossier contenant des vidéos en gros plan de personnes diverses effectuant nos 6 émotions précédemment évoquées. Nous avons convertie toutes ces vidéos en photo, en extrayant seulement une image de l'émotion réalisée, via un programme annexe utilisant OpenCV, et cela nous a servi notre base de données. Avec cela nous avons commencé à mettre en place une analyse des points caractéristiques, via des calculs de distances pour essayer de les classifier manuellement. En effet notre but à ce moment-là fût de trouver des distances assez représentatives pour déterminer une émotion. Par exemple, nous avons déterminé des distances, comme la largeur de la bouche, la

longueur du nez, la distance sourcil/œil... et en les comparant à des valeurs moyennes de ces distances pour chaque émotions, nous pensions pouvoir déterminer ces dernières. Mais cette méthode de calcul était trop imprécise. Après quelques essais infructueux, nous nous sommes rendu compte que ces techniques ne donneraient jamais des résultats assez précis pour être utilisés.

### Utilisation du réseau de neurones :

En concertation avec notre professeur référent, nous avons donc décidé d'utiliser un réseau de neurones pour classer les photos selon l'émotion en présence. Il a donc fallu quelques semaines pour que l'on comprenne ce qu'est un réseau de neurones et comment ce dernier fonctionne. En effet, le principe du réseau de neurones nous était complètement inconnu. Pour faire une explication courte, le réseau de neurones apparaît comme une analogie des neurones en biologie, à la différence que les neurones en informatique sont des petites unités de calcul. Chaque réseau possède un certain nombre de couches de neurones et les données en entrées passent à travers chaque couche du système. Le but du réseau est de classer les données d'entrées, comme dans un graphe. Ainsi chaque donnée est associée à une valeur dans ce graphe (dans notre exemple, chaque paquet de données est associé à une émotion). Ainsi durant sa phase d'apprentissage, le réseau va classer ces données en fonction de leur valeur et retenir l'association faite avec ces données. Différents éléments de cette phase sont paramétrables, comme le nombre de couche, le nombre de neurones sur chaque couche ou encore la fonction de calcul de chaque neurone. Une fois cette phase d'apprentissage terminée, le système va se tester sur des nouvelles données et il va prédire, d'après ce qu'il sait, à quoi sont associées ces données (dans notre cas, une émotion). Voilà grossièrement comment fonctionne un réseau de neurones. De plus, il a aussi fallu le mettre en place au niveau de la programmation. Pour cela nous avons utilisé la librairie Keras, qui est sur une surcouche de la librairie TensorFlow, ce qui nous a donc un peu facilité son utilisation. Pour le paramétrage de notre réseau, nous avons utilisé un modèle simple. Ce dernier tirait des nombres au hasard qui étaient soit au-dessus, soit en dessous d'une courbe de la fonction cosinus. Après sa phase d'apprentissage, le système tirait une nouvelle valeur et le réseau de neurones était capable de dire si ce chiffre était au-dessus ou en dessous de la courbe cosinus avec un taux de réussite d'environ 98%.

### Adaptation du réseau de neurones à notre objectif :

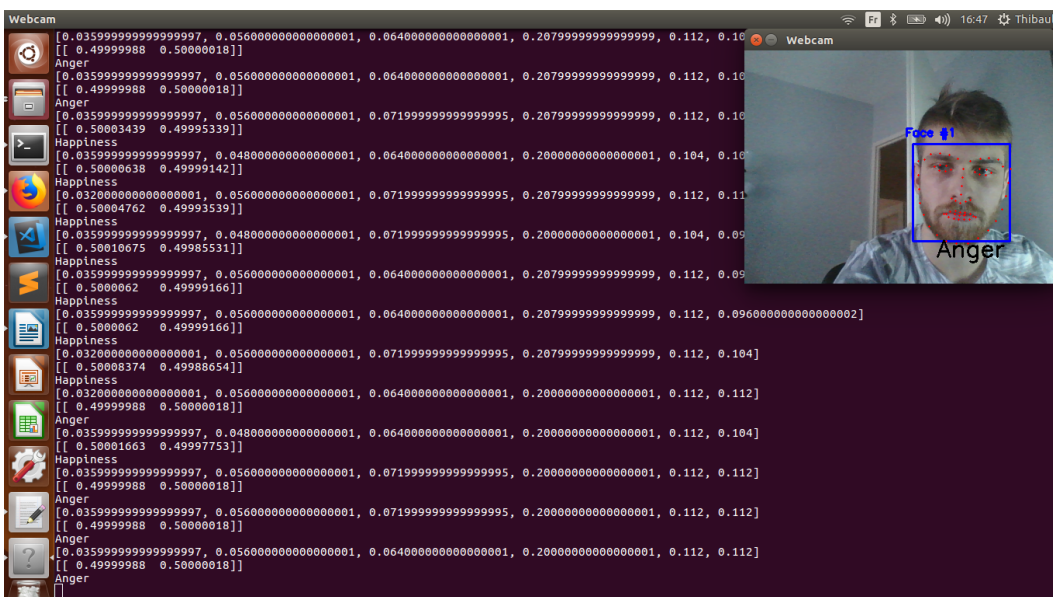
Nous avons alors adapté ce modèle à notre problème. En effet, il fallait modifier les données en entrée car au lieu de ne recevoir qu'un seul chiffre, le réseau recevait un paquet de 68 points, et les valeurs en sortie. Les premiers résultats n'étaient pas du tout concluants. Notre réseau avait en moyenne un taux de réussite aux alentours de 15%. Etant donné que nous voulions identifier une émotion parmi 6, les résultats revenaient quasiment à un résultat aléatoire. De là, nous avons utilisé plusieurs pistes d'amélioration.

Nous avons, par exemple, expérimenté un méta-réseau qui testait plusieurs configurations de réseau de neurones et gardait celui qui avait le meilleur taux de réussite à la fin. Malheureusement, ce système était très lourd et n'a pas donné de résultat concluant. En effet, le programme a tourné durant plus de 48h non-stop et en testant un peu mieux la configuration renvoyée par ce dernier, les résultats n'étaient pas satisfaisants. C'est à ce moment-là que nous avons réduit les émotions à seulement deux pour essayer d'obtenir des résultats utilisables. De plus, nous avons arrêté de donner 68 points brut en entrée de réseau. Après plusieurs modifications, nous utilisons 6 distances (largeur et hauteur de l'œil droit, largeur et hauteur des lèvres, distance entre l'œil droit et le sourcil droit et la distance entre le haut des lèvres et le nez). Toute ces distances sont divisé par la largeur du visage, pour ainsi obtenir un ratio cohérent peu importe la distance entre le visage et l'objectif. Et enfin les résultats obtenus sont utilisables.

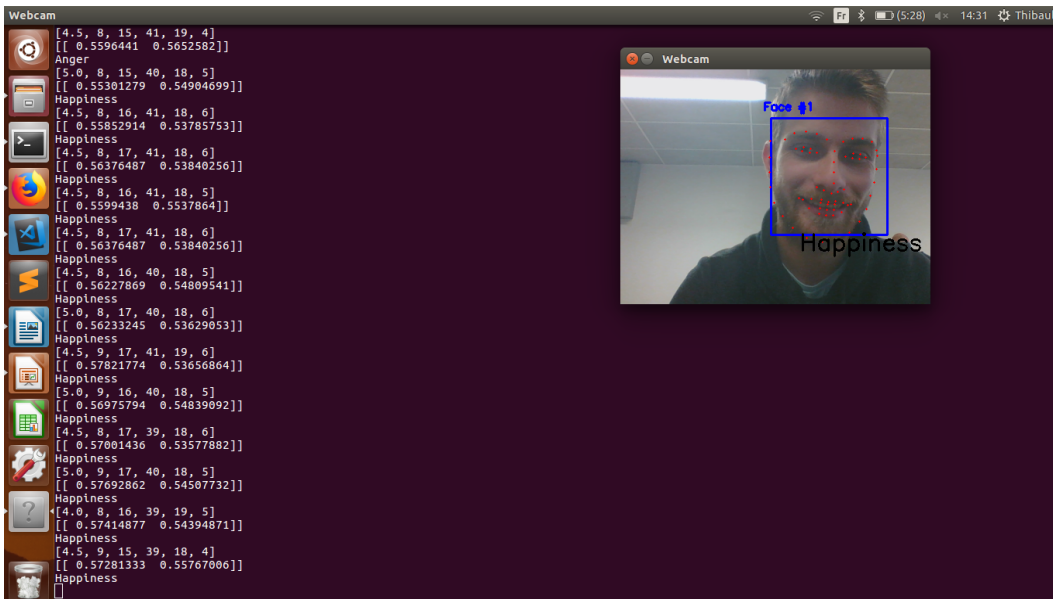
### Utilisation des résultats :

La dernière étape de notre projet était alors de mettre en relation les résultats du réseau de neurones et les données venant d'une nouvelle image afin qu'il puisse se tester dessus et ainsi prédire l'émotion présente sur l'image. Nous avons donc mis en place un système de sauvegarde pour le réseau afin de ne plus reproduire la phase d'apprentissage avant chaque test sur une nouvelle image. Ainsi, nous avons construit une fonction qui ne fait que récupérer l'apprentissage d'un réseau de neurones précédemment entraîné pour l'utiliser sur une nouvelle image. De même la nouvelle image est extraite d'un flux webcam, c'est à dire que dès que le programme reconnaît un visage à la webcam, il va tester toutes les images de ce flux pour interpréter la position des points en une émotion.

### Émotion négative :



## Émotion positive :



## 3. Les problèmes rencontrés

Le premier problème auquel nous avons fait face a été celui du langage et des librairies. En effet, nous avons décidé de faire écrire nos programmes en python, un langage totalement nouveau pour nous. Néanmoins, c'est un langage extrêmement simple à prendre en main et très proche du C++, ce qui nous a facilité notre apprentissage.

De plus, comme mentionné dans la présentation du projet, nous étions d'abord partis sur un programme de reconnaissance facial que nous avons abandonné au bout de quelques semaines. Nous nous sommes rendu compte que les outils dont nous disposions n'étaient pas suffisamment précis. En effet, là où OpenCV nous propose 68 points caractéristiques sur un visage, le système FaceID du dernier iPhone (iPhone X) utilise 30 000 points différents avec un système de scanner à lumière structurée, pouvant modéliser la surface d'un visage en 3D, et même ce système n'est pas infallible. Nous avons quand même tenté d'écrire ce programme et au moment des tests, les résultats étaient beaucoup trop aléatoires. Et c'est au moment d'évoquer l'utilisation du réseau de neurones que nous nous sommes rendu compte qu'il fallait une base de données considérable de photos pour que ce dernier puisse apprendre correctement. Dans le cas où on souhaiterait, par exemple, identifier les personnes présentes avec nous dans une salle de TP, il faudrait prendre au moins une dizaine de photos de chaque personne, les modifier pour en obtenir une centaine

et ensuite faire apprendre tous cela au réseau de neurones. Nous avons donc changé d'objectif pour partir sur la reconnaissance d'émotion.

Après cela, le principal problème a été la prise en main du réseau de neurones, en effet comprendre son fonctionnement nous a pris énormément de temps. Le temps passé à tester différentes configurations était très frustrant. En effet, ce n'est pas un problème purement informatique comme une incohérence ou des fautes dans le code, mais plutôt un problème de compréhension du système. Ainsi, il faut réfléchir différemment à une solution, pas forcément en modifiant du code ce qui donne l'impression d'une perte de temps.

## **4. Conclusion**

En bilan, nous pouvons dire que le programme que nous voulions écrire fonctionne et fait ce que nous voulions de lui. Cependant, en comparaison de ce que nous avions en tête à la naissance de ce projet, il a été drastiquement modifié dans la mesure où plusieurs fonctionnalités ont été abandonnées. En effet, la principale modification, s'il nous restait plus de temps, serait le fait de revenir à six émotions et non plus à seulement deux.

De façon plus générale, le principe de travail sur ce projet, en équipe et sans barrières est très intéressant et formateur. En effet, nous avons dû apprendre à utiliser le langage Python et les différentes bibliothèques que nous avons maîtrisé très vite. De plus, le fait d'être en groupe sur un projet qui dure un certain temps nous a permis de travailler dans des conditions similaires à celle du monde de l'entreprise.