Système de Reconnaissance faciale

Traitement du signal et des images

Fadil boodoo & Adrien Leroy-Lechat - 12 April 2019

Table de matière

Introduction

Différents méthode de reconnaissance faciale

I. La méthode LBPH

Données d'entrée

Calcul du LBP de l'image

Extraction des l'histogrammes

Reconnaissance de visage

Données en sortie

II. Implementation du code

III. Résultats

Constitution de la base de donnée

1er test

2nd test

Conclusion

Bibliographie

Introduction

Nous cherchons à implementer un système capable de reconnaitre le visage d'une personne à partir d'une base de donnée contenant déjà les images des personnes. Ce sytème est pensé pour être couplé à un régulateur de flux de personne (porte d'entrée, portique, checkpoint,..) afin de contrôler de manière automatique par reconnaissance faciale les personnes autorisées à passer.

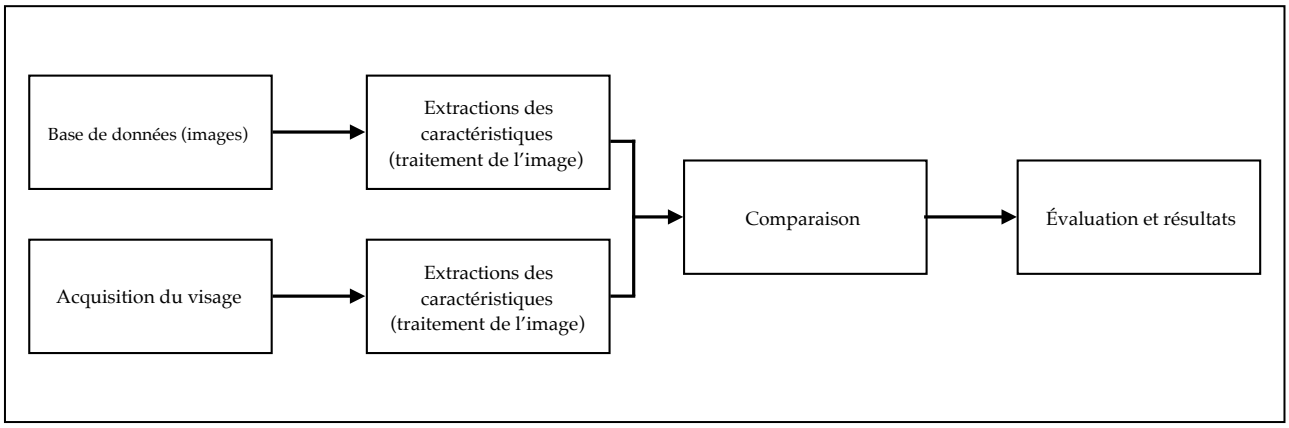

Figure 1 : Schéma fonctionnel du système

Différents méthode de reconnaissance faciale

Pour répondre à ce besoin, il existe plusieurs méthodes divisibles en deux catégories: la reconnaissance 3D, et la reconnaissance 2D.

La reconnaissance 2D est elle-même divisible en deux sous-catégories : les algorithmes qui créer une image géométrique de l'utilisateur en fonction de différents paramètres comme la taille d'éléments du visage, forme et distance entre eux. Et les algorithmes qui encode numériquement l'image, en utilisant par exemple les algorithmes de fourrier, les Eigenfaces, les Fisherfaces, le Scale Invariant Features Transform (SIFT), le Speed Up Robust Features (SURF), ou encore le Local Binary Patternes Histograms (LBPH).

Nous avons choisi de répondre à ce besoin en utilisant la méthode du Local Binary Patternes Histograms pour sa facilité d'implémentation, tout en conservant une bonne efficacité dans la reconnaissance faciale.

La méthode LBPH

La méthode LBPH se repose principalement sur le Local Binary Pattern (LBP). C'est une méthode de traitement d'image qui permet de prendre en compte la texture de l'image en considérant pour un chaque pixel un seuillage binaire entre le pixel et ses voisins.

Il a était décrit pour la première fois en 1994. Il est depuis considéré comme un puissant outils pour la reconnaissance de texture en traitement d'image. Il a était par le suite découvert que associé à un histogramme, la méthode LPB devenait bien plus performant dans la reconnaissance d'image. Avec ce dernier, nous pouvons representer les images comme de simple vecteur.

Données d'entrée

Comme donnée d'entrée, nous allons prendre des images des personnes autorisées à entrées. Ces images sont stockées dans un dossier. Nous avons aussi besoin de l'image de la personne qui essaie de pénétrer.

Calcul du LBP de l'image

Une fois l'image acquise, nous allons créer une nouvelle image qui décrit mieux l'image originale, en faisant apparaitre les caractéristiques faciales. Pour faire ceci, l'algorithme applique sont calcul par fenêtre (matrice), en se basant sur les paramètres de **rayon**, et de **voisinage**.

Nous supposons l'image d'origine est en niveaux de gris.
La fenêtre est une matrice dont la taille depends du rayon, et du voisinage.

Le **rayon** représente le rayon autour du pixel centrale, dont lequel les pixels sont pris en compte le calcul du LBP. Plus le rayon est élevé, plus les pixels pris en compte pour le calcul seront éloignés. Il est généralement mis à 1.

Le **voisinage** représente le nombre de pixel pris en compte pour le calcul du LBP. Plus ce nombre est élevé, plus le coût de calcul devient élevé. Il est généralement fixé à 8.
A noter que si la valeur du nombre de voisinage ne permet pas de couvrir tous les pixels se situant dans le disque couvert par le rayon, alors ce sont les pixels les plus éloignés qui sont pris en compte en premier.

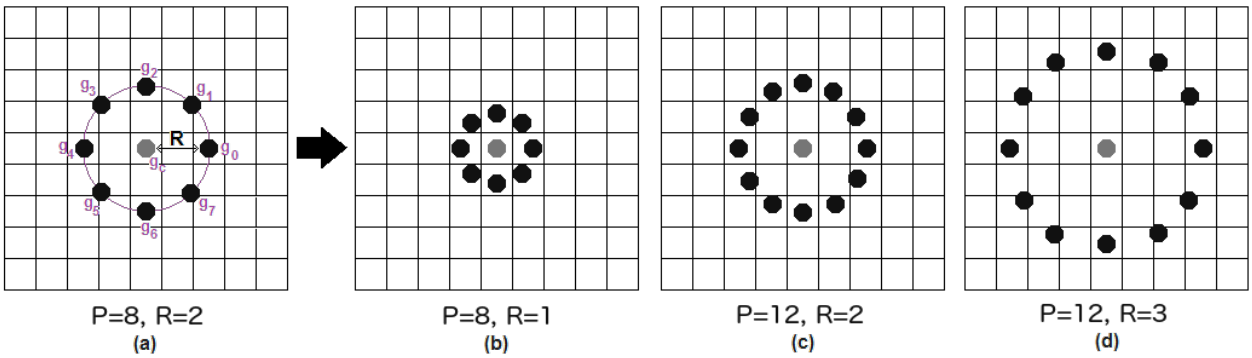

Figure 2 : Influence des paramètres rayon et voisinage dans la définition de la taille de la fenêtre

Une fois la fenêtre (taille de la matrice) fixé, on applique cette fenêtre pour un pixel de l'image. La valeur centrale de la fenêtre sera la valeur du pixel et servira de seuil. Si la valeur du pixel voisin est inférieures à la valeur du seuil, on mets 0 dans la fenêtre pour ce pixel voisin. Sinon on mets 1. Et on fais ceci pour tous les voisins du pixel considérée.
On obtient alors une matrice avec au centre le seuil, et autour des valeurs binaires.
On "lit" la valeur binaire obtenue (nous pouvons la lire de haut en bas, ou dans le sens horaire, mais la manière de le lire importe peu), et on on transforme ce nombre binaire an nombre decimal.
On stocke ce nombre decimal dans une nouvelle matrice. Ce nombre decimal correspond à la transformation LBP du pixel considéré.

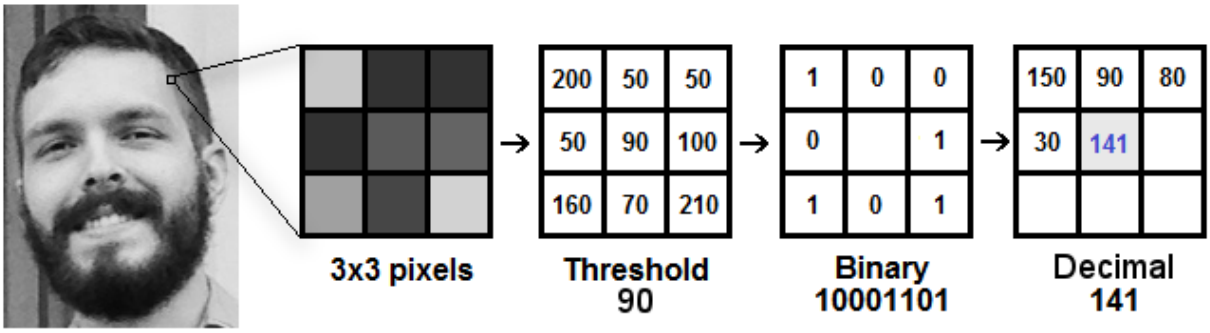

Figure 3 : Processus du calcul LBP d'un pixel de l'image. Sur cette exemple, la valeur du pixel et par ailleurs du seuil est à 90, ce qui donne pour nombre binaire $(10001101)_2$. Ce qui fait 141 en décimal, cette dernière est la valeur LPB du pixel.

On répète ce processus pour chaque pixel de l'image. On obtient ainsi la matrice LPB de l'image. Cette nouvelle matrice LPB est bien plus représentatif des caractéristiques que l'image originale.
A noter que les valeurs de la matrice LPB changent en fonction des paramètres (rayon, voisinage) de la fenêtre.

Extraction des l'histogrammes

Une fois la matrice LBP de l'image obtenue. Nous calculer son histogrammes. Mais cela ne se fait pas en une fois. En effet, il est important de considérer l'image par region, pour prendre les caractéristiques macroscopique de l'image (bouche, oeil, nez, ..). Pour ce faire, nous allons faire un calcul d'histogramme par région.
Il faut tout d'abord créer un matrice qui servira de grillage pour delimiter la matrice LBP de l'image, en region. Cette matrice grillage depends de sa taille. Elle a donc comme paramètre sa **longueur x**, et sa **largeur y**.
La **longueur x** correspond donc nombre de pixel en vertical pris en compte pour le calcul de l'histogramme régional. Il est généralement mis à 8.
La **longueur y** correspond au nombre de pixel en horizontale pris en compte pour le calcul de l'histogramme régional. Il est généralement mis à 8.
On applique le grillage sur la matrice LBP de l'image, et on calcul l'histogramme du grillage.

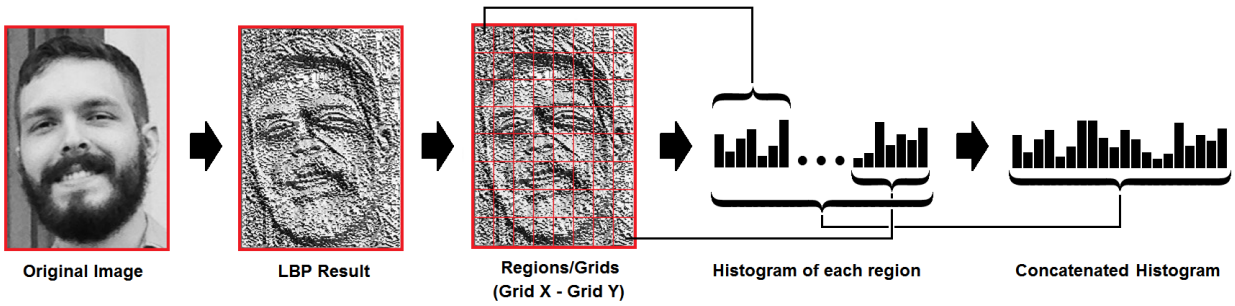

Figure 4 : Processus du calcul de l'histogramme. On divise la matrice LBP en sous-region. On calcul l'histogramme de chaque region indépendamment, pour ensuite les concaténer en un seul histogramme.

On répète ce processus da manière à couvrir toute la matrice LBP de l'image. Ensuite, on fait une concaténation de tous ces histogrammes régionaux, en un seul histogramme.
Cet histogramme est l'histogramme LBP de l'image.

Reconnaissance de visage

On considère a cette étape que l'algorithme à deja calcule l'histogramme LBP de toutes les images contenues dans la base de donnée. On veut à present lui presenter une nouvelle image et lui demander si il reconnait se visage.
Pour ce faire il faudra tout d'abord qu'il calcule l'histogramme LBP de cette nouvelle image. Ensuite, il faudra qu'il calcule la distance euclidienne entre l'histogramme LBP de cette nouvelle image, et l'histogramme LBP d'un images de la base de donnée.
La distance euclidienne se calcul à travers cette formule :

$$distance\ eucledienne = \sqrt{\sum_{i=1}^{n} (histogrammeLBP1_i - histogrammeLBP2_i)^2}$$

On répète ce processus pour chacune des images de la base des données.
Ensuite on compare les distances euclidiennes. Plus la distance est petite, plus l'algorithme le considère les deux images comme étant proche.
On peut donc définir un seuil de confiance en dessous du quel on estime que les deux images correspondent au même visage.

Données en sortie

Si la plus petite distance euclidienne est en dessous du seuil de confiance, alors l'algorithme considérera que c'est une personne qu'il connait, et retournera une message positif. Dans le cas contraire, c'est un message négatif qui sera retourné.

Implémentation du code

Le code s'implémente

Résultas

Constitution de la base de donnée

Il existe plusieurs base de données en libre de droit contenant des images de visages permettant de tester les algorithme de reconnaissance faciale.
Il existe par exemple celui de NIST contenant les images faciales de 1573 personnes, ou bien la PIE database contenant 41369 images de 68 personnes.

Pour notre test, nous avons utilisé la base de donnée de l'université de Yales. C'est une base de donnée contenant 11 images de 15 personnes dans différents expression : neutre, content, triste, endormie, etc. Les images sont en niveaux de gris. Les photos ont était prises avec la même luminosité, et sur le même fond blanc. Les images sont tous de résolution identiques (320×243) en format GIF.
Cette base de donnée est libre de droit, et téléchargeable à cette adresse : http://www.face-rec.org/databases/

1er test

Pour le premier test, nous avons utilisé entrée dans la base de donnée les 15 image des personnes avec une expression neutre.
Ensuite, nous avons demandé de reconnaître une image d'une personne avec l'expression content, et ceci pour chaque personne. Nous avons relevé la distance euclidienne minimum, et si cela correspondait bien à la personne en question.

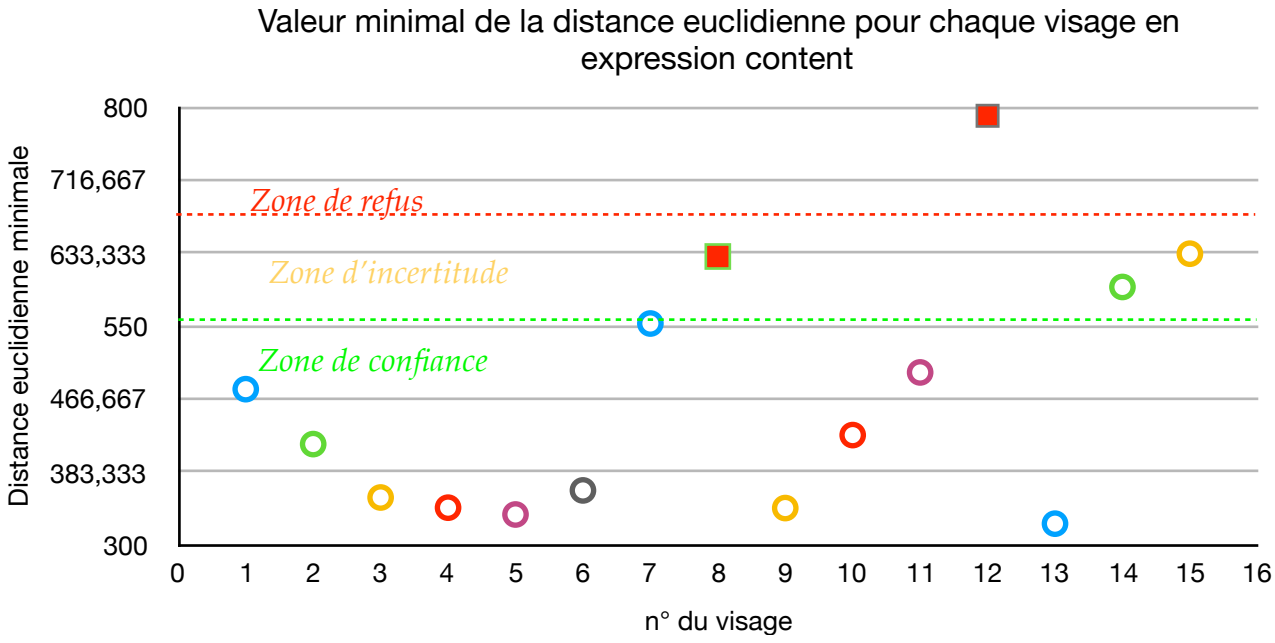

Figure 5 : graphique montrant la distance minimal euclidienne pour chaque visage en expression content. Les carrées rouges indique que la distance minimal trouvé par l'algorithme ne correspond pas à l'image de l'expression neutre de la même personne.

On remarque avec la figure 5 que les l'algorithme s'est trompé sur 2 visages pour 15 visages présentées. Ce qui fait un Succès de de 86,7%. L'algorithme s'est trompé pour le visage n°8 et le visage n°12 avec pour valeur respective de la distance euclidienne 629,35 et 790,09.
Par contre il a trouvé correctement pour le visage n°15, malgré une valeur euclidienne à 632,56.
Ce qui nous amène à créer trois zones. Une zones de confiance, où nous pouvons conclure que c'est bien le même visage, et zone d'incertitude ou nous ne pouvons lever l'indétermination , et zone de refus et nous pouvons refuser la personne.

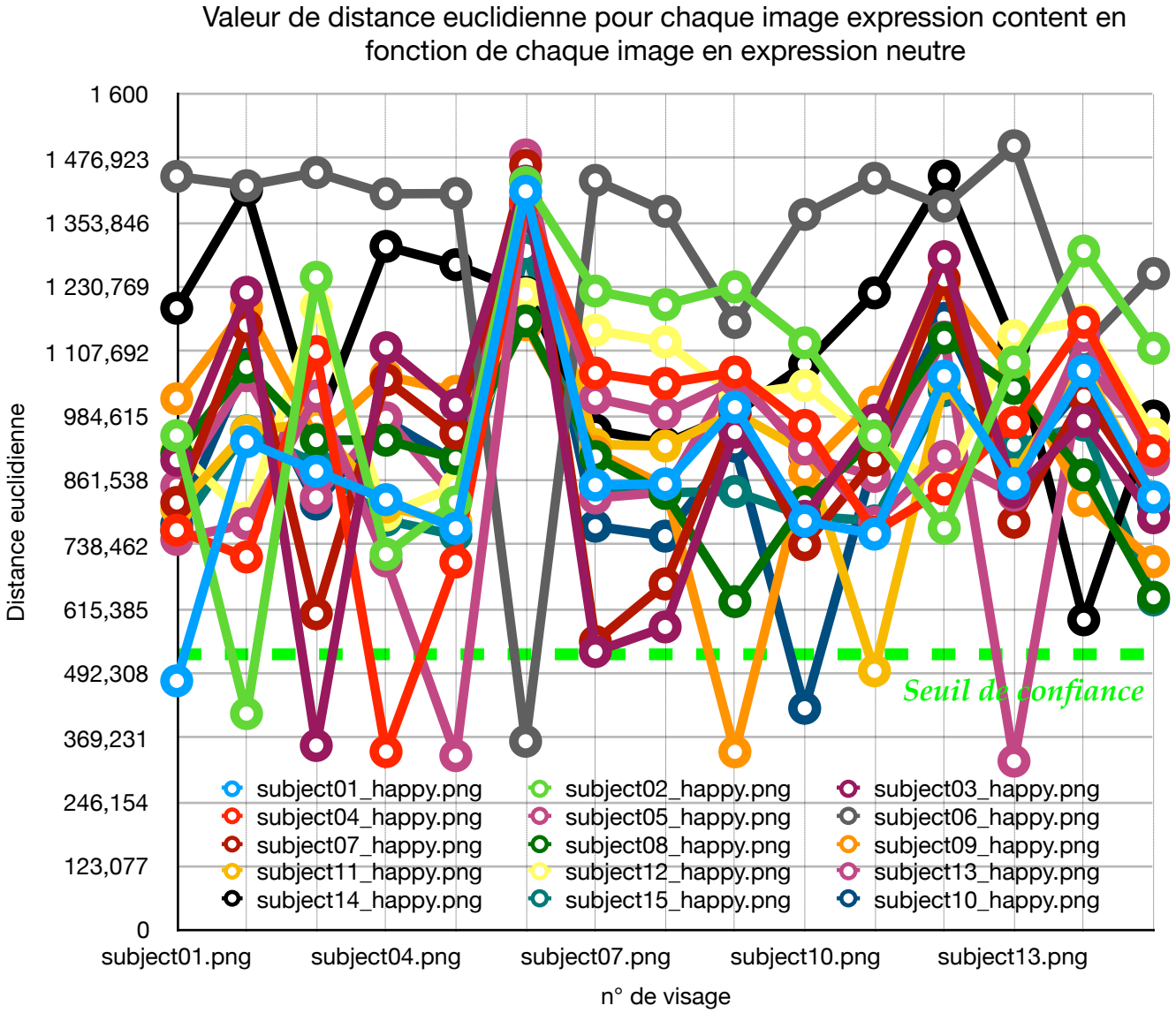

Figure 6 : graphique montrant la distance minimal euclidienne pour chaque visage en expression content, en fonction de chaque image en expression neutre.

Dans cette figure, on remarque que tous les points situé en dessous de 550 indique la bonne personnes. Nous pouvons donc prendre la valeur de **550 comme seuil de confiance**.

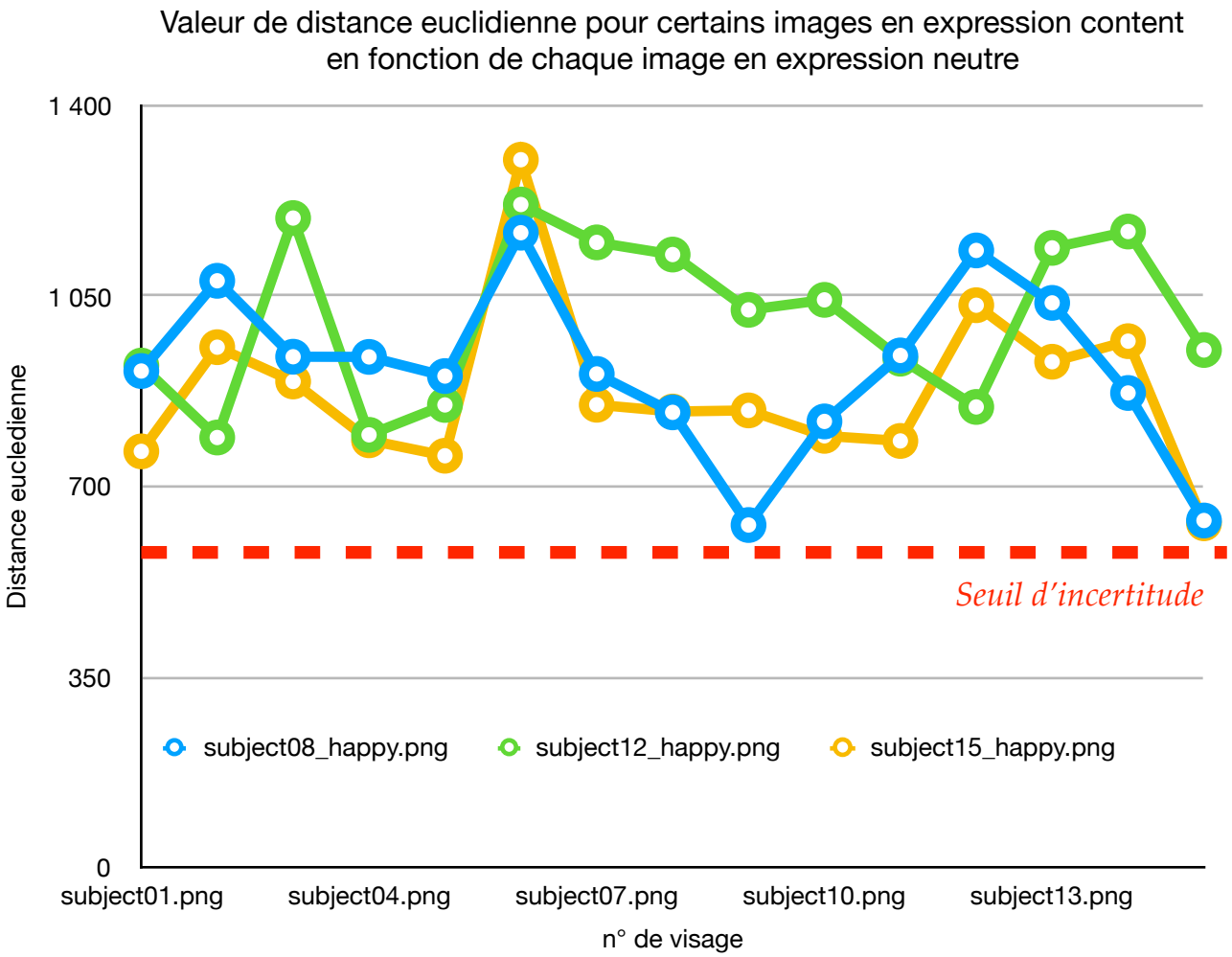

Figure 7 : graphique montrant la distance minimal euclidienne pour les visages 8, 12, et 15 en expression content, en fonction de chaque image en expression neutre.

Dans le figure 7, nous avons tracer plus les valeurs de la distance euclidienne pour les image qui ne donne pas un résultats correcte, à savoir le visage n° 8 et 12. Nous avons tracer celui de n°15 qui malgré qu'il donne un résultat correcte, reste très haut en valeur.

On remarque que les courbes reste très hautes et ne descende pas en dessous de 629.

Nous pouvons donc placer notre **seuil d'incertitude à 600**.

2ème test

Pour ce second test, nous avons entrée dans la base de donnée 10 expressions des 15 personnes.
Nous avons ensuite, demandé à l'algorithme de reconnaître la personne avec l'expression restante. Nous avons relevé les 10 plus petites distances euclidiennes, et si cela correspondait bien à la personne en question.

Conclusion

L'efficacité de l'algorithme LBPH est assez impressionnant compte tenu de la simplicité du calcul. Il est aussi pas très gourmand en ressource informatique, et assez robuste.

Néanmoins, plusieurs axes d'améliorations sont envisageables, notamment le fait que pour avoir des résultats significatives, il faut que les photos soit prises dans le même environnement. La définition plus fine des paramètres de rayon, et voisinage lors du calcul du paramètrage de la fenêtre pour le calcul de la LPB de l'image, ou bien du la longueur x, et la largeur y pour le paramaètrage du grillage pour la calcul de l'histogramme regional, pourrait améliorer considérablement les résultats, même pour des photos prises dans des environnements différents.

Enfin, on remarque que l'algorithme se comporte de manière singulière pour les images qui ne donne pas un résultats correcte, et ceci quelque soit l'image avec laquelle elle comparé. Ce qui nous laisse pensé que la definition des zones de confiance, d'incertitude, et de refus, serait probablement amélioré, en couplant l'analyse des valeurs euclidiennes avec une machine learning. Et non par définition d'un seuil.

La conception de ce système nous a permis de comprendre les différents algorithmes existantes, ainsi que les différents possibilités qu'ils offrait. Il nous aussi permis de comprendre l'importance du traitement de l'image dans la reconnaissance faciale, et les axes d'amélioration possible dans ce domaine de recherche.

Bibliographie

- Page Wikipedia : https://en.wikipedia.org/wiki/Local_binary_patterns

- Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. “Face description with local binary patterns: Application to face recognition.” IEEE transactions on pattern analysis and machine intelligence 28.12 (2006): 2037–2041.

- Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.” IEEE Transactions on pattern analysis and machine intelligence 24.7 (2002): 971–987.

- Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. “Face recognition with local binary patterns.” Computer vision-eccv 2004 (2004): 469–481.

- LBPH OpenCV: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms

- Local Binary Patterns: http://www.scholarpedia.org/article/Local_Binary_Patterns