



DETECTION D'UNE PHOTO PRESENTEE DEVANT UN SYSTEME DE RECONNAISSANCE FACIALE EN TEMPS REEL VIA L'APPROCHE ALGORITHMIQUE

Par

1. Ass. BINYONGO KIBANGALA Omar

Domaine de Recherche : Informatique/Vision par Ordinateur

Résumé : *La présente étude propose une piste de solution au problème que connaît ce dernier temps un système de reconnaissance faciale en informatique. Nombreux de ce système sont trompés par une simple photo devant la caméra. C'est pourquoi, en vu de résoudre ce problème, dans cette étude, nous avons développé une approche algorithmique basée sur la vérification du dynamisme des expressions faciales de la face détectée devant la caméra. L'objectif que nous avons poursuivi dans cette étude était bel et bien celui de mettre en place un algorithme qui avait permis à un système de reconnaissance faciale de différencier une personne réelle à une photo devant la caméra. Pour l'atteindre, nous avons exploité la méthode analytique couplée à la technique documentaire.*

Mots-clés : **Système de reconnaissance faciale, Photo, Algorithme**

Abstract : This study proposes a possible solution to the problem recently experienced by a computerized facial recognition system. Many of this system are fooled by a simple photo in front of the camera. This is why, in view of solving this problem, in this study, we have developed an algorithmic approach based on the verification of the dynamism of the facial expressions of the face detected in front of the camera. The objective that we pursued in this study was indeed that of setting up an algorithm which had enabled a facial recognition system to differentiate a real person from a photo in front of the camera. To achieve this, we used the analytical method coupled with the documentary technique.

Introduction

De nos jours on parle de plus en plus de l'insécurité dans divers secteurs ainsi que des moyens informatiques à mettre en œuvre pour contrer cette tendance : le contrôle d'accès aux ordinateurs, le commerce, les opérations bancaires basées sur l'identification du demandeur, etc.

Il existe traditionnellement deux méthodes d'identifier un individu. La première est basée sur une connaissance à priori "knowledge-based" de la personne telle que, par exemple, la connaissance de son code PIN qui permet d'activer un téléphone portable. La seconde méthode est basée sur la possession d'un objet "token-based" qui peut être une pièce d'identité, une clef, un badge, etc.

Ces deux modes d'identification peuvent être utilisés de manière complémentaire afin d'obtenir une sécurité accrue comme dans le cas de la carte bleue. Cependant, elles ont chacune leurs faiblesses. Dans le premier cas, le mot de passe ou le code PIN peut être oublié par son utilisateur ou bien deviné par une autre personne. De plus une personne sur quatre seulement fait l'effort d'appliquer les consignes de sécurité avant de retirer de l'argent (regarder derrière soi, cacher le clavier avec sa main lors de la saisie du code secret, etc.). Dans le second cas, le badge (ou la pièce d'identité ou la clef) peut être perdu ou volé. Les caractéristiques biométriques sont une solution alternative aux deux modes d'identification précédents.

L'avantage de ces caractéristiques biométriques est d'être universelles et personnel, c'est-à-dire présentes chez toutes les personnes à identifier de manière unique, ce qui explique que deux personnes ne peuvent posséder exactement la même caractéristique⁽¹⁾. Elles sont aussi permanentes ce qui signifie qu'elles ne varient pas au cours du temps. Hélas malheureusement, ces technologies sont aussi soumises à des attaques dans le but de les tromper.

Cependant, en 2011 par exemple, lorsque l'entreprise GOOGLE avait lancé le système d'exploitation androïde 4.0 doté d'un système de déverrouillage par reconnaissance faciale appelé « Face Unlock »⁽²⁾, il y avait des vidéos qui avaient prouvées qu'il était possible de tromper le système Face Unlock en présentant juste une photo devant la caméra. Le même

⁽¹⁾ SOUHILA GUERFI ABABSA, *Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D*, Thèse de doctorat en Sciences de l'ingénieur, Université Evry Val d'Essone, 2008, p.15

⁽²⁾ <https://arstechnica.com/gadgets/2017/03/video-shows-galaxy-s8-face-recognition-can-be-defeated-with-a-picture/>, consulté le 21/04/2017 à 14h45.

problème était constaté chez SAMSUNG lors du lancement sur le marché de test du téléphone GALAXY S8 en avril 2017, dotée d'un système de déverrouillage par reconnaissance faciale et qui avait subi le même problème (Huffingtonpost.fr, 2017)(3).

De ce qui précède notre problématique s'articule autour de la question centrale ci-après comme fil conducteur de cette recherche. La mise en place d'un algorithme basé sur la variation des mouvements des yeux ne serait-elle pas un outil d'aide au système informatique trompé par une simple photo devant la caméra?

La réponse à priori à cette interrogation que nous formulons sous forme d'hypothèse est que : « une personne normale ne peut pas faire un certain moment sans émouvoir les mouvements d'ouverture des yeux, nous postulons que la mise en place d'un algorithme basé sur la variation des mouvements des yeux serait un outil qui permettrait à un système de reconnaissance faciale de différencier une personne en temps réel et une photo ».

La vérification de cette hypothèse nous recommande de charpenter notre communication en quatre points.

I. Cadres Théorique et Méthodologique de l'Etude

1.1. Analyse des expressions du visage : état de l'art

Les travaux fondamentaux dans le domaine de l'analyse des expressions du visage sont dus principalement à Charles Darwin, Guillaume Duchenne De Boulogne au XIXème siècle et plus récemment Paul Ekman.

Au XIXème siècle, Guillaume Duchenne de Boulogne est le premier à localiser individuellement les différents muscles faciaux par activation électrique. Il est un des premiers à livrer à la communauté scientifique un ensemble de photographies montrant l'activation des différents muscles faciaux.

⁽³⁾ http://www.huffingtonpost.fr/2017/04/01/la-video-qui-fait-du-mal-a-la-reconnaissance-faciale-du-galaxy-s8_a_22021404/, consulté le 21/04/2017 à 14h00.



Figure N°1 : Exemples de photographies de Duchenne de Boulogne.

Charles Darwin, est le premier à traiter de l'universalité des expressions du visage et à proposer une théorie évolutionniste sur la formation des expressions. L'argument principal est que les expressions des enfants et des nouveaux nés existent aussi chez les adultes. D'après lui, l'expression des émotions est un processus nécessaire à la survie. Ainsi, les expressions non verbales sont aussi importantes que les interactions verbales dans le processus de communication humain. ⁽⁴⁾

Paul Ekman, psychologue, s'intéresse à partir du milieu des années 1960 aux expressions et émotions humaines. Il met en évidence l'universalité de certaines émotions et développe un outil de codification des expressions du visage largement utilisé aujourd'hui. Il s'intéresse désormais à l'analyse des expressions de manière informatique.⁽⁵⁾

Dans la présente étude, nous nous intéressons aux mouvements des yeux de manière à vérifier le temps réel d'une personne devant la caméra d'une part et de l'autre nous analysons l'une des phases de la reconnaissance faciale qui nous préoccupe à savoir de la phase de détection de visage.

II. Analyse du Mouvement des yeux et Détection de Visage en Informatique

2.1. Analyse du mouvement des yeux

⁽⁴⁾ Hugo MERCIER, *Analyse automatique des expressions du visage : Application à la langue des singes*, D.E.A Informatique de l'Image et du Langage, Université Paul Sabatier - Toulouse III, (2002-2003), p.16.

⁽⁵⁾ Idem

Les analyses qui seront faites à ce niveau, sont basées sur une approche géométrique de reconnaissance faciale. La librairie Luxand FaceSDK faisant partie des librairies de reconnaissance faciale par approche géométrique nous servira d'exemple. Cependant, nous allons présenter un œil sur image dans son aspect naturel et l'autre œil avec des points caractéristiques détectés avec Luxand FaceSDK.



Figure N°2 : œil normal sur l'œil



Figure N°3 : caractéristiques détectés

Dans le présent article, nous ne traitons pas des algorithmes de détection des points caractéristiques. Cependant, nous supposons qu'un algorithme est déjà en place pour détecter les caractéristiques et fonctionne bien. Dans l'exemple de la figure N°3, nous nous basons sur les positions des points ci-après : {25, 39, 32, 40, 26, 42, 31, 41}. Ce sont ces points qui définissent le contour de l'œil.

Dans le processus d'extraction des caractéristiques de la face, une boucle sera entrain de balayer le visage et récupérer ainsi les informations sur la position des points que nous avons énumérés ci-haut. Ainsi donc le taux d'ouverture de l'œil sera déterminé sur base des positions de ces variables.

2.2. Détection de Visage

Après avoir capturé l'image contenant un visage, la deuxième étape consiste à l'extraire de l'image. Cela peut se faire par détection de la couleur de la peau, ou par des méthodes détectant les différentes caractéristiques du visage par des descripteurs locaux. Cette étape est autant plus délicate pareillement que l'image acquise contient plusieurs objets de visage ou un fond non uniforme qui crée une texture perturbant la bonne segmentation du visage. Elle est dépendante de la qualité des images acquise.

Après la segmentation du visage, on peut filtrer ou améliorer la qualité par des prétraitements qui sont appliqués au visage extrait. Selon un

algorithme d'extraction des caractéristiques utilisée, la manière d'extraire les données sur le visage se diffère. Prenons l'exemple de l'extraction du visage avec luxand FaceSDK qui cherche à capturer 70 points caractéristiques du visage dans la figure ci-dessous.

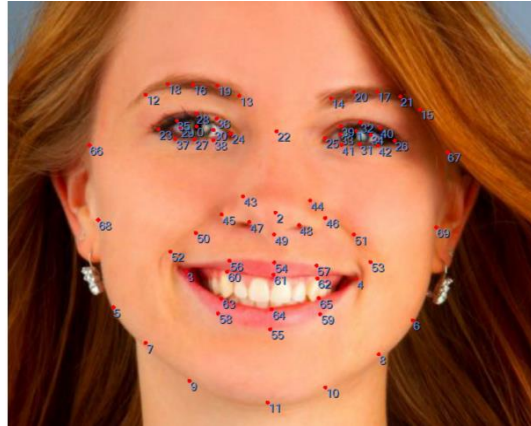


Figure N°4 : Les 70 points caractéristiques détectés par Luxand FaceSDK. ⁽⁶⁾

Signalons que chaque point de caractéristique est stocké dans un tableau mais aussi avec sa position de localisation sur le visage. C'est sur ces positions où nous allons nous baser pour détecter la variabilité des mouvements.

III. Développement de l'Algorithme

En ayant déjà des connaissances sur l'analyse des mouvements que nous allons vérifier dans ce travail, notre algorithme consistera alors de stocker les anciennes informations sur la localisation des caractéristiques du contour de l'œil (ce qu'on appelle taux d'ouverture des yeux) pour les comparer avec les nouvelles informations détectées après une incrémentation de la boucle.

a) Variables utilisées

Dans cet algorithme nous sommes en train d'utiliser les variables ci-après :

- a) **Ve** : c'est une variable qui va stocker le taux d'ouverture des yeux à une itération donnée de la boucle. Ainsi donc, nous pouvons formaliser *Ve* de la manière ci-après :

$$Ve = \{P_{25}; P_{39}; P_{32}; P_{40}; P_{26}; P_{42}; P_{31}; P_{41}\} \text{ Où } P = \text{position}$$

⁽⁶⁾ Luxand FaceSDK 6.2 ,*Face Detection and Recognition Library : Developer's guide*, Luxand 2016,p.50.

À ce niveau, une fonction sera appliquée à cette variable afin de détecter le taux en termes de pourcentage. Dans le cas de la librairie Luxand FaceSDK par exemple, on utilisera la fonction FSDK_GetTrackerFacialAttribute.

- b) **tmpVe** : cette variable stockera temporairement la valeur de la variable *Ve* de l'itération n-1.
- c) **I** : cette variable stockera le flux vidéo qui sera envoyé par la caméra.

Une fois que nous aurons déjà la valeur de *Ve* et *tmpVe*, la différence entre ces deux valeurs nous donnera l'information sur le mouvement observé au niveau des yeux. En se basant sur notre hypothèse de départ, une fois que cette différence donne 0, cela signifie qu'il n'y a pas eu variation. Une situation qui poussera l'algorithme de décider que c'est une photo qu'on présente devant la caméra.

3.1. Construction de l'algorithme

Voilà en synthèse, la littérature présentée ci-haut. Notre algorithme se résume de la manière suivante :

Code de l'algorithme

```

1  VARIABLES
2    Ve EST_DU_TYPE NOMBRE
3    tmpVe EST_DU_TYPE NOMBRE
4    I EST_DU_TYPE NOMBRE
5    face EST_DU_TYPE NOMBRE
6  DEBUT_ALGORITHME
7    Ve PREND_LA_VALEUR 0
8    tmpVe PREND_LA_VALEUR 0
9    I PREND_LA_VALEUR FluxVideo
10   TANT_QUE (detecterFace(I)) FAIRE
11     DEBUT_TANT_QUE
12       face PREND_LA_VALEUR detecterFace(I)
13       tmpVe PREND_LA_VALEUR Ve
14       Ve PREND_LA_VALEUR TauxOuvertureYeux(face)
15       SI (Ve-tmpVe==0) ALORS
16         DEBUT_SI
17           AFFICHER "Visage suspecté comme une photo"
18         FIN_SI
19     FIN_TANT_QUE
20 FIN_ALGORITHME

```

Source : Notre propre confection

3.2. Mesure de la Qualité de l'Algorithme Développé

Avant de procéder à la mesure de la qualité de notre algorithme, nous allons donner des exemples de mesure des qualités des logiciels et montrer les mesures qui seront utilisées pour notre cas.

Exemples des métriques logiciels :⁽⁷⁾

1) Métriques de taille

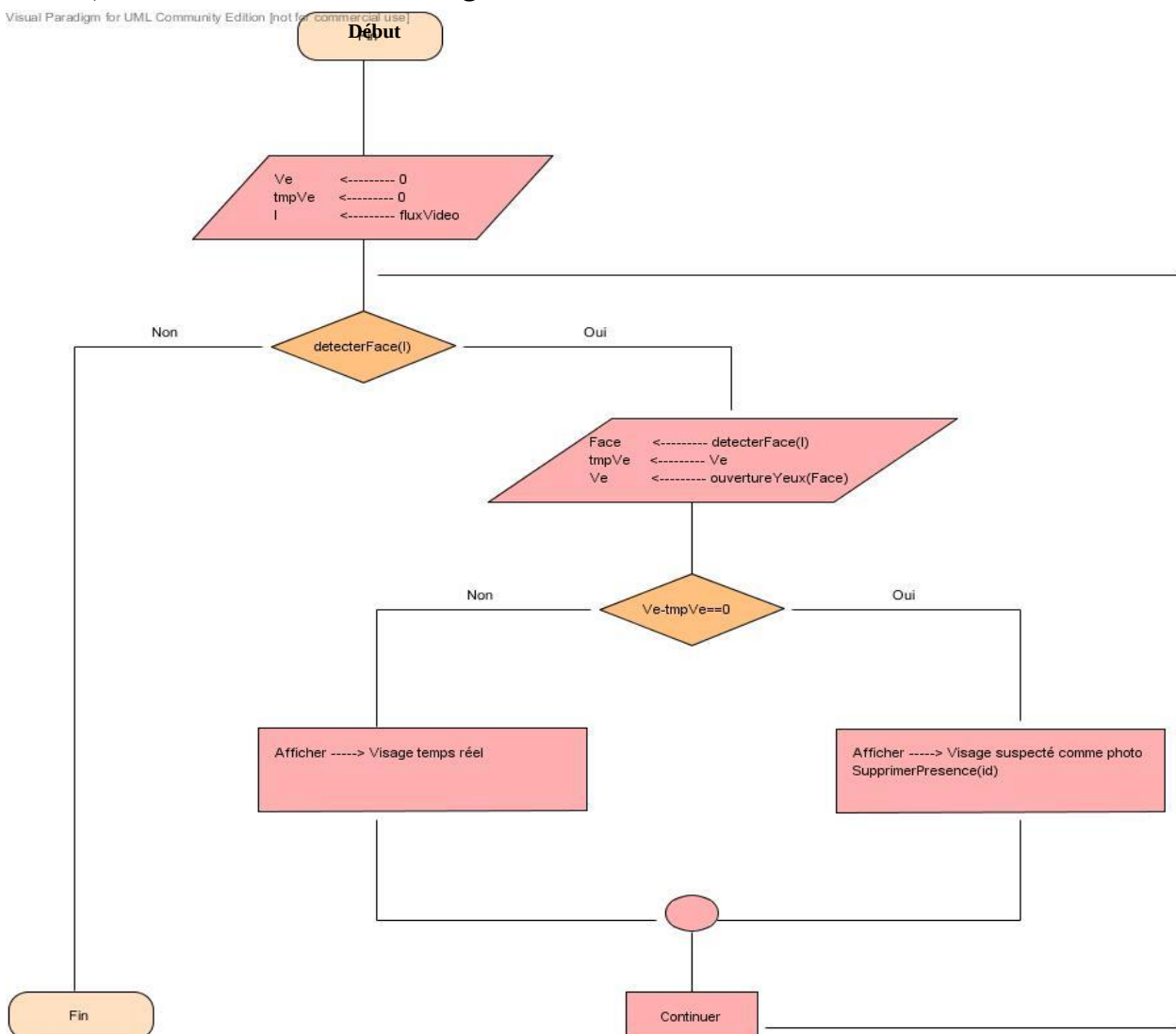
- a) plus la taille est grande, plus la complexité est haute

⁽⁷⁾ HAO WANG, *Les métriques appliquées dans la construction de logiciel, Mémoire de Maitrise en Informatique*, Université du Québec Montréal, Mai 2007, p.14.

- b) e.g. nombre de lignes de code, nombre de paramètres de méthode
- 2) Métriques de complexité
 - a) indiquent les méthodes qui sont difficile de comprendre
 - b) e.g. complexité cyclomatique
- 3) Métriques de couplage
 - a) indiquent les éléments qui ont trop d'interdépendances avec d'autres
- 4) Métriques de cohésion
- 5) Métriques de similarité
 - a) peuvent indiquer le code dupliqué

Afin de tester notre algorithme, nous avons utilisé la métrique de complexité notamment la complexité cyclomatique.

a) Construction de l'ordinogramme



Source : Notre propre confection

L'ordinogramme ci-dessus a été testé dans le cadre de détecter une photo devant la caméra dans une application de gestion des présences des agents. Raison pour laquelle il y a usage de la fonction *supprimerPresence*.

b) Calcul de la complexité cyclomatique

1) complexité cyclomatique =

- a) nombre de chemins indépendants dans le graphe de flux de commande (*control flow graph*) d'un programme ;
- b) limite supérieure pour le nombre de tests nécessaires pour s'assurer que toutes les instructions ont été exécutées au moins une fois ;
- c) Définition formelle (basée sur la théorie de graphe)

$CC(G)$ = nombre de régions dans le graphe G

= nombre de arcs - nombre de nœuds + 2

Pour déterminer le nombre des jets de données, nous utilisons la formule suivante :

$$CC(G) = (\text{nombre d'arcs} - \text{nombre de noeuds}) + 2$$

Nombre d'arcs = 11

Nombre de nœuds = 10, alors :

$$CC(G) = (11 - 10) + 2, CC(G) = 3$$

C'est-à-dire que l'algorithme fera 3 jets de données pour parcourir toutes les instructions au moins une seule fois.

c) Calcul de l'ordre de la complexité ($O(g)$)

Soit $f(n)$ le temps de calcul de notre algorithme, les opérations élémentaires comme affectation, calcul, comparaison et appelle d'une fonction seront égal à 1, soit $O(1)$

Ainsi donc, notre $f(n)$ peut se présenté de la manière suivante :

$$f(n) = 1 + 1 + 1 + \sum_{i=0}^n 1 + 2 + 1 + 2 + 2$$

Dans notre $f(n)$:

- ✓ Les 3 premières constantes correspondent aux trois opérations d'initialisation dans notre algorithme ;
- ✓ L'autre partie de $f(n)$ illustre la valeur asymptotique n . C'est-à-dire une répétition n fois des 8 opérations élémentaires ;
- ✓ Après le symbole de la somme, le 1 montre la comparaison de la boucle et le 2 illustre l'appel d'une fonction ainsi que l'affectation ainsi de suite ;
- ✓ Il est à noter que lors de la comparaison (ligne N°15 de notre algorithme), un calcul s'est effectué avant de comparer ; ce qui fait à ce qu'il y ait 2 coûts en terme de temps.

Cependant, notre $f(n)$ peut être écrit de la manière suivante :

$$f(n) = 3 + 8n$$

Ce qui nous amènes à dire que notre algorithme est de l'ordre de complexité $O(n)$, soit une complexité linéaire telle que définie par Stéphane Grandcolas dans son ouvrage intitulé « Complexité des algorithmes ». Autrement dit, nous pouvons dire que f est dominé par g avec la notation $f = O(g)$.

Bien plus, il semble être important de prouver dans notre cas que $f(n) = O(n)$; pour y arriver, nous devons vérifier l'expression suivante :

$$\exists n_0, \exists c > 0 \mid \forall n \geq n_0, \quad f(n) \leq c.n$$

En utilisant les mêmes coefficients de $f(n)$, nous aurons ce qui suit :

$$\left. \begin{array}{l} \checkmark \quad 3 \leq n \\ \checkmark \quad 8n \leq 8n \end{array} \right\} \quad f(n) \leq 9n$$

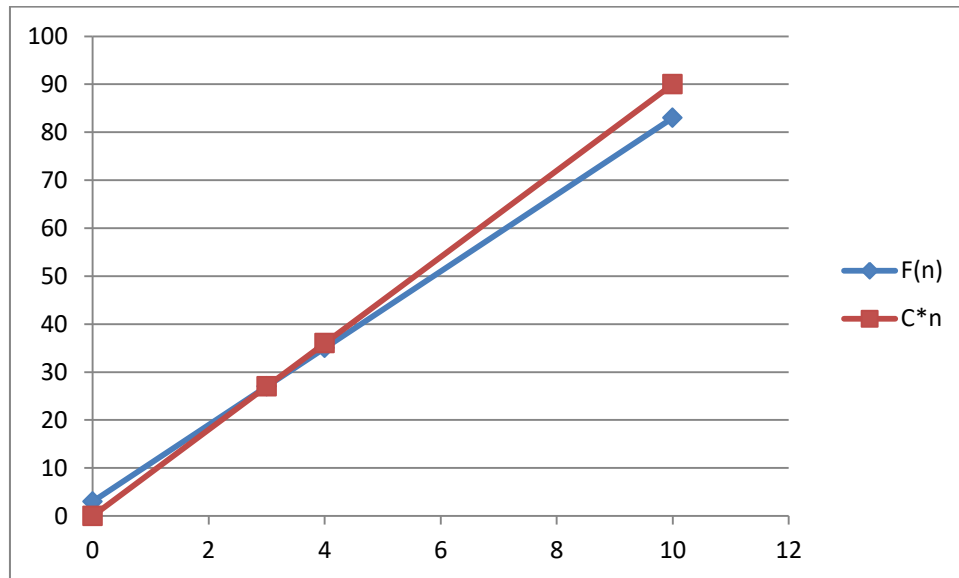
NB :

- 9 est déjà notre c trouvé à partir de la somme de $8+1$;
- Pour le premier cas, la comparaison ne sera vraie qu'à partir de $n=3$;
- Pour le second cas, la comparaison est vraie peut-importe la valeur de n qui doit être supérieur ou égal à 0 bien sûr.

Cependant, nous pouvons dire que le polynôme $f(n) = 3 + 8n \leq 9n$ sera vrai $\forall n \geq 3$ ce qui nous donne le couple $(3, 9)$ avec $n_0=3$ et $c=9$. Nous pouvons vérifier cela à partir de l'ensemble des solutions ci-après :

- ✓ $N=0 \rightarrow f(n)=3$ et $c.n=0$;
- ✓ $N=3 \rightarrow f(n)=27$ et $c.n=27$;
- ✓ $N=4 \rightarrow f(n)=35$ et $c.n=36$;
- ✓ $N=10 \rightarrow f(n)=83$ et $c.n=90$.

Ceci peut être représenté graphiquement de la manière suivante :



Source : Notre confection sous Excel

En conclusion, nous avons trouvés une limite supérieure asymptotique de n qui varie en fonction de sa valeur. Comme vous pouvez le remarquer sur le graphique, lorsque n est inférieur à 3, $f(n)$ est toujours supérieur à $c.n$ par contre c'est l'inverse ; une situation qui prouve que $f(n)=O(n)$.

IV. Discussion des Résultats

Nous venons donc de réaliser un algorithme qui désormais, une fois implémenté dans un système de reconnaissance faciale, est à mesure de vérifier le dynamisme du taux d'ouverture des yeux du visage présenté devant la caméra.

Par rapport aux problèmes qu'avaient connus GOOGLE et SAMSUNG, cet algorithme peut venir en aide comme solution mais néanmoins il n'est pas du tout parfait. Nous soulevons cette notion étant donné que l'algorithme que nous avons développé ne prend pas en charge le contrôle des vidéos. Imaginez par exemple si une personne plaque une vidéo

de son smart phone devant la caméra pour chercher à tromper cette dernière. La suite est que le système sera trompé, car l'algorithme développé ne prend pas en charge cet aspect. Donc, l'idéal est d'améliorer l'algorithme afin de détecter ce genre de menace.

Conclusion

Le renforcement de la sécurité des systèmes de reconnaissances faciales était l'objet de la présente étude sachant que nous nous sommes attardé au problème consistant à tromper un système de reconnaissance faciale à temps réel, en présentant juste une simple photo devant la caméra qui envoie des signaux à ce dernier. Notre objectif était de proposer dans ce cadre, une approche algorithmique qui différencierait une personne réelle à une photo devant la caméra.

En nous tablant sur l'hypothèse selon laquelle une personne normale ne peut pas rester un moment donné sans manifester des mouvements d'expressions faciales ; dans cette étude, l'analyse du dynamisme des mouvements des yeux en termes de taux était notre piste de solution pour éviter ce problème.

Tout compte fait, nous avons élaboré un algorithme qui vérifie dans le flux vidéo si le taux d'ouverture des yeux varie ou pas. C'est à partir de cette information que le système prendra une décision de savoir s'il s'agit d'une photo devant la caméra.

A nos lecteurs, nous rappelons que ce travail n'est pas un modèle consubstantiel et parfait.

Références Bibliographiques

- HAO WANG, *Les métriques appliquées dans la construction de logiciel*, Mémoire de Maîtrise en Informatique, Université du Québec Montréal, Mai 2007
- Hugo MERCIER, *Analyse automatique des expressions du visage : Application à la langue des singes*, D.E.A Informatique de l'Image et du Langage, Université Paul Sabatier – Toulouse III , (2002-2003),
- Luxand FaceSDK 6.2 ,*Face Detection and Recognition Library : Developer's guide*, Luxand 2016,
- SOUHILA GUERFI ABABSA, *Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D*, Thèse de doctorat en Sciences de l'ingénieur, Université Evry Val d'Essone, 2008 ;
- Stéphane Grandcolas, *Complexité des algorithmes*, Université de Namur
- <https://arstechnica.com/gadgets/2017/03/video-shows-galaxy-s8-face-recognition-can-be-defeated-with-a-picture/> consulté le 21/04/2021 à 14h45 ;
- http://www.huffingtonpost.fr/2017/04/01/la-video-qui-fait-du-mal-a-la-reconnaissance-faciale-du-galaxy-s8_a_22021404/ consulté le 21/04/2021 à 16h45.