Revue Internationale du Chercheur
ISSN: 2726-5889
Volume 3 : Numéro 4

Une approche algorithmique pour détecter une photo présentée devant un système de reconnaissance faciale en temps réel

An algorithmic approach to detect a photo presented to a face recognition system in real time

MURHULA KABI Grâce
Enseignant chercheur en informatique de Gestion
Institut Supérieur de Commerce de Bukavu
murhulakabi@gmail.com

BAKENGE BUGOYE Cirhuza
Enseignant chercheur en informatique de Gestion
Institut Supérieur Pédagogique de Kaziba
anicetbakenge@gmail.com

BYAMUNGU BARHACIKUBAGIRWA
Enseignant chercheur en informatique de Gestion
Institut Supérieur d'Agroforesterie et de Gestion de l'environnement
byamungujustinmasu@gmail.com

KASAMBI BALIMWENGU
Enseignant chercheur en informatique de Gestion
Institut Supérieur Pédagogique de Bukavu
ballykasambi@gmail.com

RUJULIKA MUHOZA Alain
Enseignant chercheur en informatique de Gestion
Institut Supérieur Pédagogique d'UVIRA
rujalikamuhoza@gmail.com

RUSAGARA BABONE MBONGO
Enseignant chercheur en Mathématique
Institut Supérieur Pédagogique de KAZIBA
rusagaraba@gmail.com

Date de soumission : 09/10/2022

Date d'acceptation : 24/12/2022

Pour citer cet article :

NASRI C. (2022) « Le monde de l'animation socioculturelle : l'accueil de la méthode de gestion de conflits BETZAVTA en Europe », Revue Internationale du Chercheur « Volume 3 : Numéro 4 » pp : 523 – 538

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 présentation d'une photo devant la caméra. Dans ce sens, afin de résoudre ce problème, nous avons développés une approche algorithmique basée sur la vérification du dynamisme des expressions faciales de la face détectée devant la caméra. Désormais, le système de reconnaissance faciale sera en train de vérifier ces expressions faciales afin de prendre une décision s'il s'agit d'une photo ou pas. L'algorithme consiste à stocker les 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 (une nouvelle séquence de capture de la caméra). Cependant, l'objectif que nous poursuivons dans ce travail, est de mettre en place un algorithme qui permettra à un système de reconnaissance faciale de distinguer une personne réelle et une photo devant la caméra. Pour l'atteindre, nous avons exploité la méthode analytique couplée de la technique documentaire.

Mots clés : Système ; Reconnaissance faciale ; Algorithme ; Temps réel

Abstract

This study proposes a solution to the problem that a facial recognition system in computer science has been experiencing lately. Many of these systems are deceived by a simple presentation of a photo in front of the camera. In this sense, in order to solve this problem, we have developed an algorithmic approach based on the verification of the dynamism of facial expressions of the face detected in front of the camera. From now on, the facial recognition system will be checking these facial expressions in order to make a decision if it is a photo or not. The algorithm consists of storing the information about the location of the eye features (the so-called eye opening rate) to compare it with the new information detected after a loop increment (a new camera capture sequence). However, the goal of this work is to implement an algorithm that will allow a facial recognition system to distinguish between a real person and a photo in front of the camera. To achieve this goal, we have used the analytical method coupled with the documentary technique.

Keywords : System ; Facial recognition ; Algorithm; Real time

Introduction

Ce dernier temps, on parle de plus en plus de l'insécurité dans différents secteurs ainsi que des moyens informatiques à mettre en œuvre pour contrer cette prédisposition. Parmi les secteurs les plus touchés, nous pouvons citer, le contrôle d'accès aux ordinateurs, le commerce, les opérations bancaires basées sur l'identification du demandeur, etc. Il existe classiquement 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 (Grâce, 2020).

Ces deux modes d'identification peuvent être employés de manière complémentaire afin d'obtenir une sécurité accrue comme dans le cas de la carte bleue. Bien plus, 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. Par ailleurs, il a été prouvé qu'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.) (ABABSA, 2008). Dans le second cas, le badge (ou la pièce d'identité ou la clef) peut être perdu ou volé. Pour pallier à ce problème, les caractéristiques biométriques sont une solution alternative aux deux modes d'identification précédents.

L'avantage que présente ces caractéristiques biométriques est celui d'être universel et personnel. Cela veut dire qu'elles sont présentes chez toutes les personnes à identifier de manière unique, ce qui explique que deux personnes ne peuvent avoir exactement les mêmes caractéristiques (KABI, 2019). Elles sont aussi permanentes ce qui signifie qu'elles ne varient pas même si les années passent. 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 », il était facile de tromper ce système de déverrouillage en présentant simplement une photo devant la caméra (Zehil, 2011). Signalons que le problème de méthode de déverrouillage reste encore un défi au sein de l'entreprise Alphabet (Kaufmann, 2022). Le même problème était constaté chez SAMSUNG lors que l'entreprise avait lancé sur le marché

de test GALAXY S8 en avril 2017, un appareil doté d'un système de déverrouillage par reconnaissance faciale et qui avait connu les mêmes difficultés (Randroid, 2017).

De ce qui précède, notre problématique s'articule autour de la question centrale ci-après : « 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 cette problématique nous recommande de charpenter notre travail en 3 principaux points. Le premier point portera sur la revue de la littérature et cadre méthodologique de notre étude, ensuite suivra le développement de notre algorithme pour finir avec la discussion des résultats.

1. Revue de la littérature et cadre méthodologique

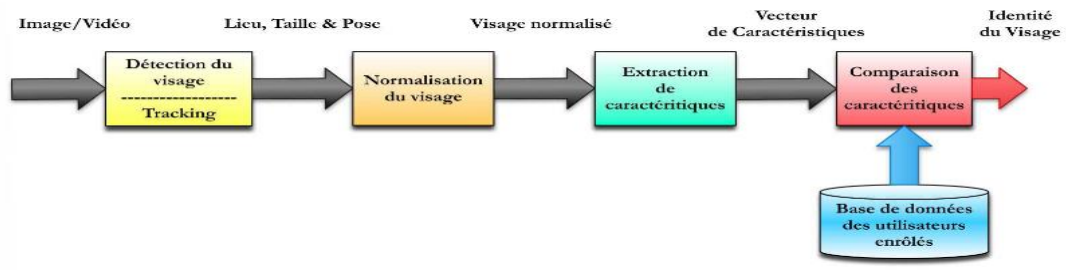
1.1. Revue de la littérature

1.1.1. Revue théorique de la littérature

La reconnaissance faciale est une technique qui permet à partir des traits de visage d'authentifier ou d'identifier une personne (X. Tan, 2006). En parlant de l'authentification, il s'agit de vérifier qu'une personne est bien celle qu'elle prétend être (dans le cadre d'un contrôle d'accès par exemple). De l'autre côté, identifier une personne est une manière de retrouver cette dernière au sein d'un groupe d'individus, dans un lieu, une image ou une base de données (Lazzouni, 2022).

Un système de reconnaissance faciale est composé par plusieurs parties qui sont à leur tour regroupés en deux phases : la phase d'apprentissage et la phase de reconnaissance tel qu'illustré par la figure N°1.

Figure 1 : Présentation d'un système de reconnaissance faciale



Source : (BASEMA, 2010)

Cependant, ce système est beaucoup plus attaqué dans sa partie reconnaissance car les personnes cherchent à se faire passer pour des inconnues d'un côté, et pour des fraudeurs de l'autre côté (Faresse, 2022).

Notre étude s'inscrit dans le cadre de la recherche d'une solution aux attaques des systèmes informatiques basés sur la reconnaissance faciale. Loin de nous l'idée de prétendre être le premier à travailler sur ce sujet. nous avons exploités différents travaux scientifiques qui nous ont beaucoup plus inspirés dans nos analyses.

1.1.2. Synthèse des travaux

Dans son travail de mémoire publié sur le site web grand lac Online, Monsieur Grâce KABI a travaillé sur la réalisation d'une application de gestion des présences des agents d'une institution d'enseignement supérieur et Universitaire. Il est abouti à un résultat qui a permis d'élaguer le système de fraude qui permettait à un agent de signer la présence pour un autre (KABI, 2019).

La reconnaissance faciale est aussi exploitée dans la gestion des émotions comme l'a démontré J. Maire dans son article qui portait sur la reconnaissance des émotions faciales des enfants atteint de troubles déficit de l'attention. Les résultats ont montré que les enfants atteints de TDA/H possèdent un déficit de REF par rapport aux enfants qui ne présentent pas les signes de TDA/H. Ils ont plus de difficultés à reconnaître la tristesse, le dégoût et la joie (J. Maire, 2020).

Ces systèmes de reconnaissances faciales développés par les auteurs précités sont exposés à des failles de sécurités surtout quand on présente une photo devant la caméra pour être analyser. A la recherche d'une solution à ce problème, nous avons exploité aussi les travaux dans le domaine des expressions du visage.

Nous sommes au 19ème siècle, Monsieur Guillaume Duchenne réussi à localiser individuellement les différents muscles faciaux par activation électrique tel qu'illustré par la figure N°2 (MERCIER, 2007).

Figure 2 : les muscles faciaux selon Guillaume Duchenne

Source : (MERCIER, 2007)

Dans sa thèse de doctorat, Monsieur Hugo Mercier montre aussi qu'il est possible d'analyser les expressions faciales d'une personne qui se présente devant une caméra de reconnaissance faciale. Au regard de ces recherches, nous avons compris qu'il serait possible de renforcer le système de reconnaissance par la vérification des différentes expressions faciales. Cela nous amène à dire qu'une personne normale ne pourrait pas faire un certain moment sans présenter des mouvements d'ouverture des yeux.

Cependant, nous avons formulés notre hypothèse de la manière suivante : « 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 ».

1.2. Cadre méthodologique

Notre méthodologie est axée sur l'analyse des mouvements des yeux et quelques techniques de détection faciale avec la librairie Luxand Face SDK.

1.2.1. Analyse des mouvements des yeux

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 dans son aspect naturel (figure N°3) et l'autre œil avec des points caractéristiques détectées via Luxand FaceSDK.

Figure 3 : Œil normal

Source : (Luxand, 2022)

Figure 4 : Caractéristiques détectés sur l'œil

Source : (Luxand, 2022)

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°4, 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.

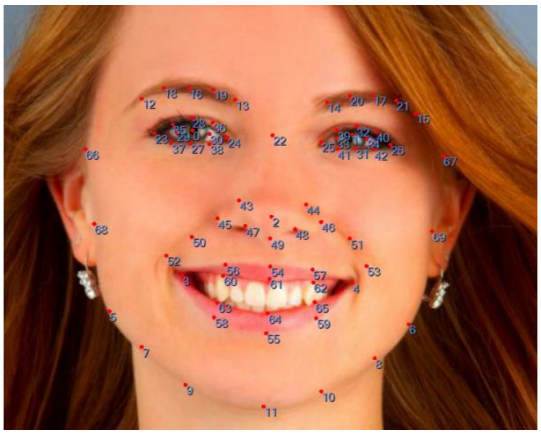
Dans le processus d'extraction des caractéristiques de la face, une boucle sera entrain de balayer le visage et récupérer ainsi les informations sur la position des points que nous avons énumérés ci-haut. Ainsi donc, le taux d'ouverture de l'œil sera déterminé sur base des positions de ces variables.

1.2.2. Processus de détection d'un visage

Une fois la capture de l'image contenant un visage se termine, la deuxième phase consiste à extraire ce dernier de l'image. Cela peut se faire par détection de la couleur de la peau, ou soit par des méthodes détectant les différentes caractéristiques du visage via des descripteurs locaux. Cette étape est très délicate selon que l'image acquise contient plusieurs objets de visage ou un fond non uniforme qui crée une texture perturbant la bonne segmentation du visage. Nous pouvons ainsi dire qu'elle est dépendante de la qualité des images acquises.

Lorsque le visage sera segmenté, 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. La figure N°5 présente un exemple d'extraction du visage avec luxand FaceSDK qui capture 70 points caractéristiques.

Figure 5 **: les 70 points caractéristiques détectés par Luxand FaceSDK**



Source : (Luxand, 2022)

Signalons que chaque point caractéristique est stocké dans un tableau avec une position de localisation sur le visage. Nous allons nous intéresser sur ces différentes positions pour détecter la variabilité des ces points.

2. Développement de notre algorithme

Ayant déjà des connaissances sur l'analyse des mouvements que nous allons vérifier dans ce travail, notre algorithme consistera à 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.

2.1. Variables utilisées

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

- **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 :
  $Ve = \{P_{25}; P_{39}; P_{32}; P_{40}; P_{26}; P_{42}; P_{31}; P_{41}\}$ Où P= position
  À ce niveau, une fonction sera appliquée sur cette variable afin de détecter le taux en termes de pourcentage. Dans le cas de la librairie Luxand FaceSDK, on utilisera la fonction *FSDK_GetTrackerFacialAttribute*.
- **tmpVe** : cette variable stockera temporairement la valeur de la variable ***Ve*** de l'itération *n-1*.
- **I** : cette variable stockera le flux vidéo qui sera envoyé par la caméra.

Une fois 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, si 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.

2.2. Construction de l'algorithme

A l'aide de l'outil Algo Box, nous avons réalisé un algorithme qui se présente de la manière suivante :

Figure 6 : Présentation de l'algorithme

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

2.3. Mesure de la qualité de l'algorithme développé

Avant de mesurer la qualité de notre algorithme, abordons d'abord quelques exemples de mesure des qualités des logiciels pour finir par montrer les mesures qui seront utilisées dans notre cas d'étude.

Selon WANG, il existe les métriques logiciels ci-après : (WANG, 2007)

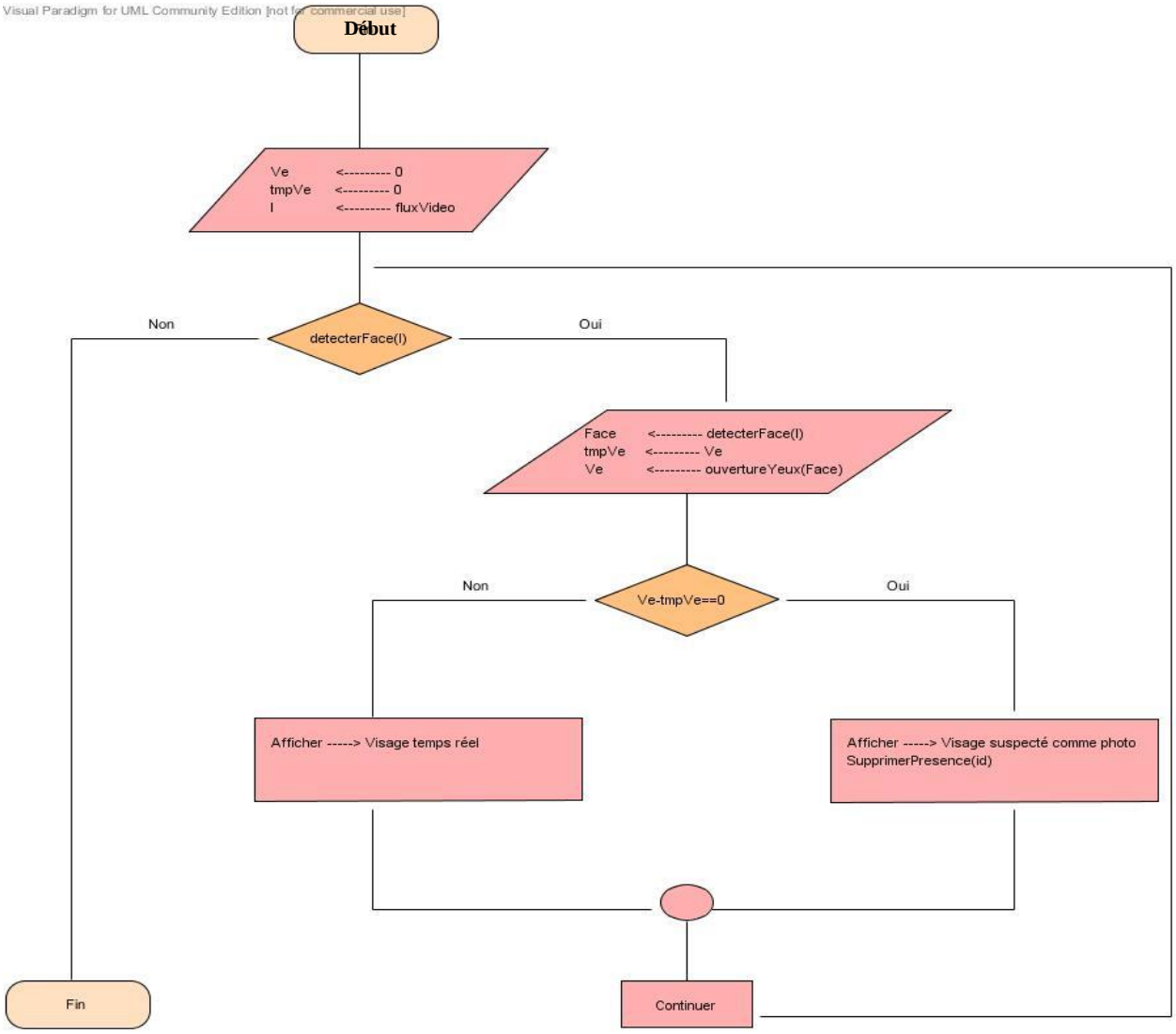
- Métriques de taille,

- Métriques de complexité,
- Métriques de couplage,
- Métriques de cohésion,
- Métriques de similarité.

Pour tester notre algorithme, nous avons utilisé la métrique de complexité, plus précisément, la complexité cyclomatique.

2.3.1. Construction de l'ordinogramme

Figure 7 : Présentation de l'ordinogramme



Source : Notre propre confection

L'ordinogramme ci-dessus (figure N°7) 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 ().***

2.3.2. Calcul de la complexité cyclomatique

Afin de déterminer le nombre des jets de données de notre algorithme, nous utiliserons la formule suivante :

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

Dans notre algorithme, *nombre d'arcs* = 11 et *nombre de nœuds* = 10, dans ce cas, nous auront :

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

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

2.3.3. 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 présenter 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 écrite de la manière suivante :

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

Ceci 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 » (Grandcolas, 2020). Autrement, 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 :

$$\exists_{n0}, \exists_c > 0 \mid \forall n \geq n0, \qquad f(n) \leq c.n$$

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

- ✓ $3 \leq n$
- ✓ $8n \leq 8n$

$\Big\}$ $f(n) \leq 9n$

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.

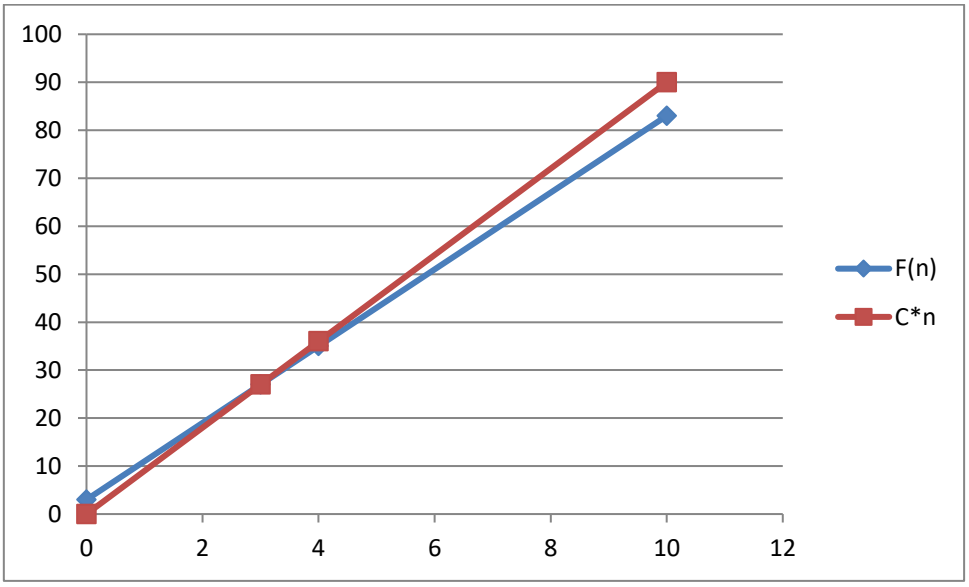
Cependant, nous pouvons dire que le polynôme $f(n) = 3 + 8n \leq 9n$ sera vrai $\forall_n \geq 3$ ce qui nous donne le couple (3, 9) avec n0=3 et c=9. Nous pouvons vérifier cela à partir d'un ensemble des solutions ci-après :

- ❖ N=0 ➔ f(n)=3 et c.n=0 ;
- ❖ N=3 ➔ f(n)=27 et c.n=27 ;
- ❖ N=4 ➔ f(n)=35 et c.n=36 ;

- N=10 ➔ f(n)=83 et c.n=90.

Ceci peut être représenté graphiquement de la manière suivante :

Figure 8 : représentation graphique des valeurs du polynome



Source : nos confections 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).

3. 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.

Il est possible d'affirmer que cette solution apporte une solution qu'avait connu Grâce KABI dans son mémoire qui portait sur la gestion des présences par reconnaissance faciale. L'application qu'il avait réalisée était capable d'enregistrer la présence d'un agent une fois ce dernier se présente devant la caméra. Le problème résidait au niveau où un agent pouvait passer à son collègue une photo pour la présenter devant le système de reconnaissance faciale.

Malheureusement, son système n'était pas capable de différencier les deux éléments devant la caméra.

Le résultat obtenu grâce à notre approche algorithmique prouve à suffisance qu'il est possible de renforcer le système de reconnaissance en y associant les variables qui stockent les mouvements des expressions faciales.

Conclusion

Le renforcement de la sécurité des systèmes de reconnaissances faciales était l'objet de la présente étude. Nous nous sommes intéressés 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 et 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. Nous avons ainsi 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 la décision de savoir s'il s'agit d'une photo devant la caméra.

Nonobstant ce résultat, 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. En terme de limite et perspectives de notre recherche, l'idéal est d'améliorer l'algorithme afin de détecter ce genre de menace vidéo.

A nos lecteurs, nous rappelons que ce travail n'est pas un modèle consubstantiel et parfait. C'est pourquoi, nous restons ouverts à toute critique et nous sommes prêts à recevoir toutes les recommandations et remarques tendant à progresser davantage cette étude.

BIBLIOGRAPHIE

1. ABABSA, S. G., (2008). *Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D.* [Online]
   Available at: https://fr.slideshare.net/GataHipogata/jecroij
   [Accessed 04 06 2022].

2. AMADEO, R., (2017). *video-shows-galaxy-s8-face-recognition-can-be-defeated-with-a-picture/.* [Online]
   Available at: https://arstechnica.com/gadgets/2017/03/video-shows-galaxy-s8-face-recognition-can-be-defeated-with-a-picture/
   [Accessed 19 05 2022].

3. BASEMA, S. K., (2010). *Identification des personnes par reconnaissance de visage pour la sécurité d'une institution bancaire,* BUKAVU: ISP BUKAVU.

4. Faresse, M., (2022). *Les méthodes d'usurpation de la reconnaissance faciale les plus courantes et comment les éviter.* [Online]
   Available at: https://blog.dormakaba.com/fr/les-methodes-dusurpation-de-la-reconnaissance-faciale-les-plus-courantes/
   [Accessed 2022 12 01].

5. Grâce, M. K., (2016). *Conception et réalisation d'une application de gestion des présences par reconnaissance faciale* . [Online]
   Available at: https://docplayer.fr/109440988-Conception-et-realisation-d-une-application-de-reconnaissance-faciale-pour-la-gestion-des-presences-cas-des-agents-de-l-isp-bukavu.html
   [Accessed 06 05 2022].

6. Grandcolas, S., (2020). *Compléxité des algorithmes,* Belgique: Université de NAMUR.

7. J. Maire & G. Mark., (2020). La reconnaissance des émotions faciales des enfants atteints de trouble déficit de l'attention/hyperactivité. *Cambridge University Press*, 16 Avril, p. https://doi.org/10.1016/j.eurpsy.2013.09.171.

8. Kaufmann, J., (2022). *fix-android-security-flaw-that-lets-anyone-unlock-your-phone.* [Online]
   Available at: https://crast.net/204418/fix-android-security-flaw-that-lets-anyone-unlock-your-phone/
   [Accessed 15 10 2022].

9. Lazzouni, N., (2022). *Surveillance, reconnaissance faciale... Ce qui se passe en France est effrayant.* [Online]
   Available at: https://www.lemediatv.fr/emissions/2022/surveillance-reconnaissance-faciale-

ce-qui-se-passe-en-france-est-effrayant-M5hYSvU2TE2bKUSjEUbh-A
[Accessed 11 30 2022].

10. Luxand, (2022). *Detect and Recognize Faces with Luxand FaceSDK.* [Online]
Available at: https://www.luxand.com/facesdk/
[Accessed 10 11 2022].

11. MERCIER, H., (2007). *Analyse automatique des expressions du visage : Application à la langue des singes.* Toulouse: Université Paul Sabatier.

12. Randroid, (2017). *la video qui fait du mal à la reconnaissance faciale du galaxy s8.* [Online]
Available at: https://www.frandroid.com/marques/samsung/417072_samsung-galaxy-s8-la-reconnaissance-faciale-plus-rapide-que-le-scanner-diris
[Accessed 20 05 2022].

13. WANG, H., (2007). *Les métriques appliquées dans la construction de logiciel, Mémoire de Maitrise en Informatique.* Québec Montréal: s.n.

14. X. Tan & Ali., (2006). Face recognition from a single image per person: A survey. *Science Direct,* Volume 39, p. DOI doi:10.1016/j.patcog.2006.03.013.

15. Zehil, D., (2011). *android-4-0-fail-face-unlock.* [Online]
Available at: http://www.lgeek.info/2011/11/android-4-0-fail-face-unlock/
[Accessed 05 10 2022].