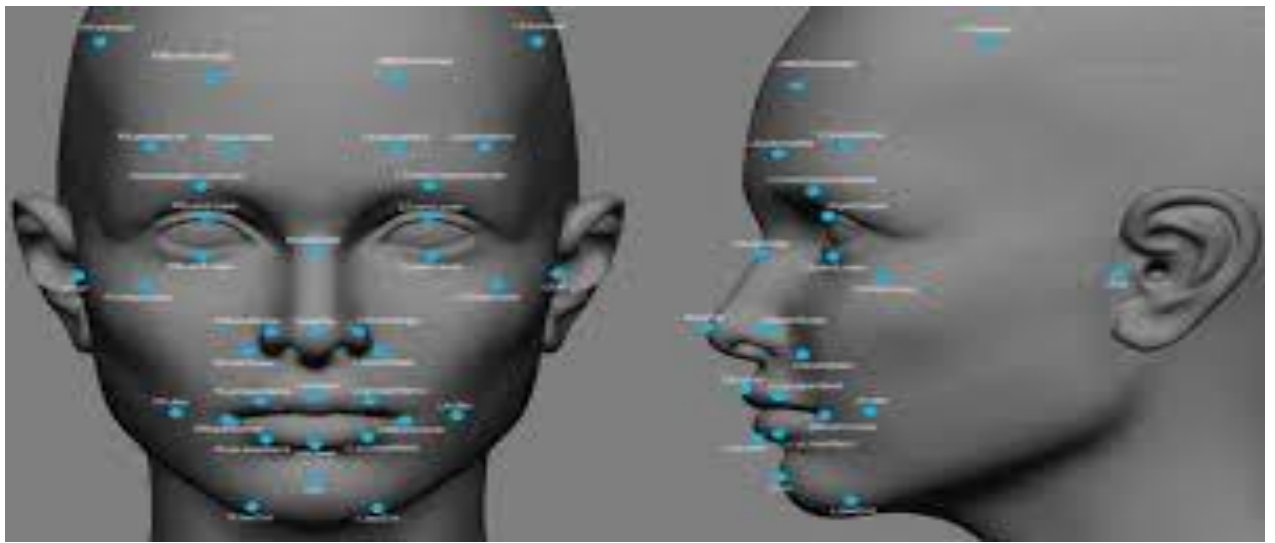


Conception et réalisation d'une application de reconnaissance faciale



Sommaire

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I : Etat de l'art

I.1 Introduction	2
I.2 La reconnaissance de visage	2
I.2.1 Approches globales.....	3
I.2.2 Méthodes locales	3
I.2.3 Méthodes hybrides	3
I.3 Principales difficultés de la reconnaissance de visage.....	3
I.4 Conclusion	4

Chapitre II : Plateforme de développement et contrainte

II.1 Introduction	5
II.2 Environnement de développement	5
II.3 Présentation d'Android Studio	5
II.4 Présentation de la SDK	5
II.5 Traitement d'image	5
II.5.1 Présentation d'OpenCV	5
II.6 Importation d'OpenCV sur Android studio	6
II.7 Contraintes de la reconnaissance	6
II.8 Conclusion	7

Chapitre III : Choix arrêtés

III.1. Introduction	7
III.2. Les LBP appliqués à la reconnaissance faciale	7
III.3. Prétraitement	7
III.4. Descripteur d'image	7
III.4.1. Histogramme.....	8
III.5. Plus proche voisin	9
III.6. Conclusion.....	9
Conclusion générale	10
Bibliographie	11

Liste des figures

Liste des figures

Figure 1 : Classification des algorithmes principaux utilisés en reconnaissance faciale....	3
Figure 2 : Construction du motif 00101010	3
Figure 3 : Extension de l'opérateur LBP basique. Les entourages (8,1), (16,2), (8,2) sont représentés	4
Figure 4 : Grille pour l'échantillonnage (grille « a » 3x3 et grille « b » 2x2).....	4

CHAPITRE

Etat de l'art

I.1 Introduction

La biométrie est devenue de plus en plus utilisée dans les domaines de sécurité et de contrôle. Et parmi toutes les technologies biométriques qui existent, la reconnaissance des visages est l'une des technologies les plus utilisées et les plus adaptées.

Nous allons voir dans ce chapitre les systèmes de reconnaissance faciale, leurs approches et les problèmes liés à leur application.

I.2. La reconnaissance de visage:

La méthode de reconnaissance du visage s'avère plus avantageuse, car c'est une méthode non intrusive, et les capteurs utilisés sont peu coûteux (une simple caméra) contrairement à l'empreinte digitale et à l'iris où le sujet devra être très proche du capteur et devra coopérer pour l'acquisition de l'image sans oublier le coût de l'équipement nécessaire pour l'acquisition. [3]

Les systèmes de reconnaissance faciale peuvent opérer dans les deux modes suivants : authentification ou identification.

L'identification permet d'établir l'identité d'une personne à partir d'une base de données. [1]

Dans la vérification ou l'authentification le système valide l'identité d'une personne en comparant les données biométriques saisies avec le modèle biométrique de cette personne stockée dans la base de données du système. [1]

Les méthodes de reconnaissance de visages peuvent être classées en trois grandes approches. Une approche globale dans laquelle on analyse le visage dans son entier, une approche locale basée sur un modèle, dans laquelle le système essaie de détecter, regrouper et reconnaître les différents éléments constitutifs du visage tel que le nez, les yeux et la bouche. Enfin, il existe des méthodes hybrides qui combinent les deux approches précédentes.

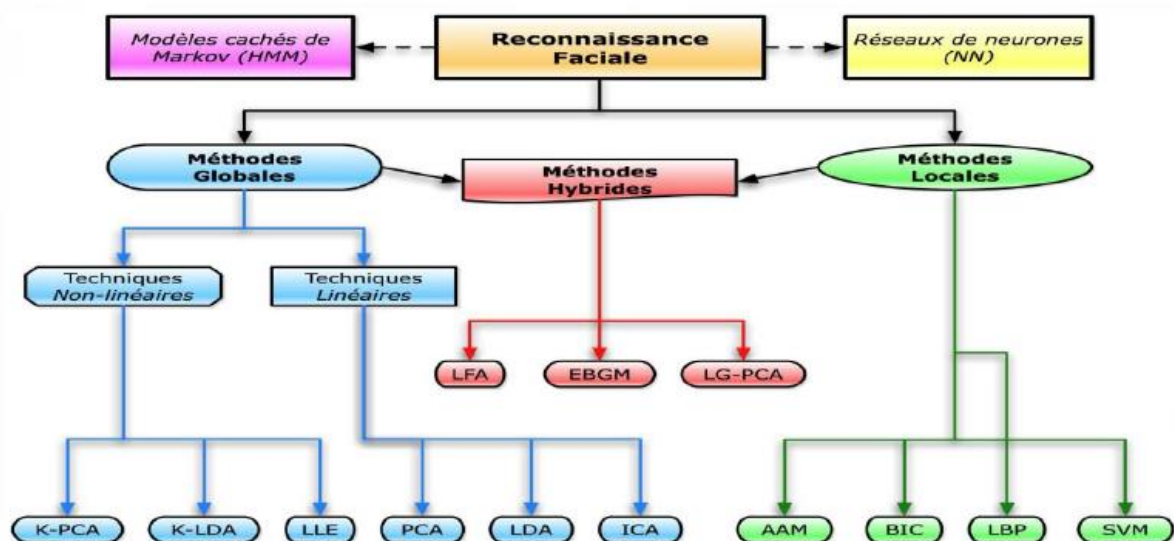


Figure 1: Classification des algorithmes principaux utilisés en reconnaissance faciale. [13]

I.2.1. Approches globales

Le principe de ces approches est d'utiliser toute la surface du visage comme source d'information sans tenir compte des caractéristiques locales comme les yeux, la bouche, et... Ils sont relativement rapides à mettre en œuvre, mais sont sensibles aux variations d'illumination, de pose et d'expression faciale [5]. Parmi les approches les plus importantes réunies au sein de cette classe on trouve:

- L'Analyse en Composantes Principales (PCA ou Eigen Faces).
- L'Analyse Discriminante Linéaire (LDA).
- Machine à Vecteurs de Support (SVM).
- Les Réseaux de Neurones (RNA).
- Mélange de Gaussiennes (GMM).
- Modèle Surfactive du Visage (3D).
- L'approche statistique et probabiliste.

I.2.2. Méthodes locales

Le principe de ses approches consiste à extraire les caractéristiques locales de visage comme les yeux, le nez et la bouche, puis utilisent leur géométrie et/ou l'apparence comme donnée d'entrée du classificateur, l'énergie sera accordée aux petits détails locaux évitant le bruit engendré par les cheveux, les lunettes, les chapeaux, la barbe, etc... Mais leur difficulté se présente lorsqu'il s'agit de prendre en considération plusieurs vues du visage ainsi que le manque de précision dans la phase "extraction" des points constituent leur inconvénient majeur [7].

Parmi ces approches on peut citer :

- Modèles de Markov Cachés (Hidden Markov Models (HMM)).
- L'algorithme Elastic Bunch Graph Matching (EBGM).
- Eigen Object (EO).
- LBP (Local Binary Patterns)

I.2.3. Méthodes hybrides

Les Méthodes hybrides combinent les deux méthodes précédentes pour une meilleure caractérisation des images de visages. Les caractéristiques locales et les caractéristiques globales ont des propriétés différentes. On peut essayer d'exploiter leur complémentarité pour améliorer la classification [6].

I.3. Principales difficultés de la reconnaissance de visage

La construction d'un système de reconnaissance faciale constitue un sérieux défi lorsque les conditions d'acquisition des images sont très variables, Il existe deux types de variations associées aux images de visages : inter et intra sujet. La variation inter-sujet est limitée à cause de la ressemblance physique entre les personnes. Par contre la variation intra-sujet est plus vaste.

Elle peut être attribuée à plusieurs facteurs que nous analysons ci-dessous:

- Les changements d'éclairage influencent l'apparition d'un visage, même si la pose de la face est fixée.
- La variation de pose est considérée comme un problème majeur pour les systèmes de reconnaissance faciale, Quand le visage est de profil dans le plan image, il peut être normalisé en détectant au moins deux traits faciaux, Cependant, lorsque la rotation est supérieure à 30° , la normalisation géométrique n'est plus possible
- Les expressions faciales son localisées principalement sur la partie inférieure du visage. L'information faciale se situant dans la partie supérieure du visage reste quasi invariable. Etant donné que l'expression faciale modifie l'aspect du visage, elle entraine forcément une diminution du taux de reconnaissance [4].
- À long terme un visage change en raison du vieillissement, à une coiffure ou selon maquillage ou accessoires.

I.4 Conclusion

Dans se chapitre nous avons fait une présentation de la reconnaissance automatique de visage et des différentes difficultés inhérentes notamment l'invariance de l'illumination, pose et invariance faciale.

Dans le chapitre suivant nous ferons une présentation des différents outils nécessaires pour le développement d'une application de reconnaissance faciale sur une plateforme mobile.

CHAPITRE

Environnement de développement et contrainte

II.1 Introduction

Avant tout développement informatique il est indispensable de bien choisir des outils de développement adapté au projet souhaité.

Et pour mener à bien ce projet on a opté pour les outils de développement suivants :

II.2 Environnement de développement

Pour le choix de l'environnement nous avons opté pour Android Studio.

II.3 Présentation d'Android Studio

Android Studio est l'environnement de développement que Google propose à ses développeurs pour créer des applications Android. Présenté à la Google I/O en 2013, Android Studio propose un environnement de travail complet pour développer des applications sous Android. Android studio a un assistant permettant de créer rapidement un projet pour tous les périphériques Android. L'assistant va se charger de récupérer le bon SDK Android, de paramétrer l'environnement de développement et de créer un émulateur pour tester les applications. [9]

Le système Android est basé sur un fork du noyau de Linux. Ce dernier a été modifié pour être plus adapté aux terminaux mobiles ayant peu de puissance de calcul, de mémoire et de batterie.

Les applications sont écrites en Java, et fonctionnent au sein d'une machine virtuelle Dalvik. [10]

II.4 Présentation de la SDK

Software Development Kit est un ensemble d'outils logiciels permettant de faciliter le développement d'un logiciel sur une plateforme donnée (par exemple, il existe un SDK pour Android, un pour iOS, etc.)

Le SDK d'Android est multi-plateforme, permettant de tester son application sur plusieurs versions différentes d'Android, plusieurs tailles d'écran, etc. et ce, même si on n'a pas d'appareil physique. [11]

III.5 Traitement d'image

Pour la partie traitement d'image du projet nous allons utiliser la librairie de traitement d'image openCV.

III.5.1 Présentation d'OpenCV

OpenCV(Open Computer Vision) est une bibliothèque graphique libre(open source), initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage assure le support de cette bibliothèque depuis 2008.

Chapitre II : Plateformes de développement & contrainte

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

Elle propose la plupart des opérations classiques en traitement des images :

- lecture, écriture et affichage d'une image.
- calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs.
- lissage, filtrage.
- seuillage d'image (méthode d'Otsu, seuillage adaptatif).
- Segmentation.
- morphologie mathématique. [12]

II.6 Importation d'OpenCV sur Android studio

Pour se faire nous avons suivi les étapes suivantes :

1. Ouverture de l'IDE Android studio.
2. Création d'un nouveau projet.
3. Importez l'OpenCV pour le module SDK Android dans Android Studio.
4. Configuration de la librairie OpenCVLibrary300>build.gradle.
5. Association d'OpenCV pour les bibliothèques d'Android.

A cette étape le projet est enfin prêt pour des futurs développements sur OpenCV.

II.7 Contraintes de la reconnaissance

Le projet sera réalisé sur une plateforme légère « tablette Asus » fonctionnant sous Android, qui dispose d'un appareil photo d'une résolution d'image de 5 MPixels.

Pour notre premier prototype, l'application doit faire face à plusieurs contraintes :

- Variation de lumière : l'apparence du visage change en fonction de l'illumination de la scène lors de la prise d'image, les variations de lumière rendent la tâche de reconnaissance faciale très difficile.
- Variation de pose : la variation de pose est considérée comme un problème majeur lors de la reconnaissance d'image.
- Bon cadrage : l'effet produit par la photo est différent selon la façon dont on l'a cadré.
- Zoom : Lors de la prise d'image si le visage est trop loin ou trop près de la caméra cela engendrera une mauvaise reconnaissance de la géométrie du visage

II.8 Conclusion

Dans ce chapitre, nous avons mis l'accent sur les différents outils de développement nécessaires pour le développement d'une application de reconnaissance faciale sur une plateforme mobile en utilisant l'IDE Android studio et les contraintes liées à son application.

Dans le chapitre qui suit nous allons traiter la méthode que nous avons choisi pour réaliser ce projet en l'occurrence LBP (Local Binary Pattern) cette méthode a été choisie pour sa robustesse face aux variations de gris et sa rapidité à générer.

CHAPITRE

III

Choix arrêtés

III.1 Introduction

La plupart des algorithmes de reconnaissance de visage sont conçus pour fonctionner mieux avec des images de visage bien alignés , bien éclairée, et des poses frontales , cependant, il n'est pas possible de répondre à toutes ces conditions.

En effet, il y'a eu de nombreux efforts récents pour développer des algorithmes de reconnaissance faciale qui fonctionnent bien avec toutes ses contraintes. Dans ce contexte, Les algorithmes qui utilisent des descripteurs d'apparence locale sont plus robustes contre l'occlusion, les variations d'expressions, les variations de poses et les échantillons de petites tailles comparées aux algorithmes globaux traditionnels.

Dans ce chapitre, nous allons nous concentrer sur l'approche locale LBP (local Binary Pattern), car elle est a la fois bonne, robuste face aux variations de gris et rapide à générer.

III.2 Les LBP (Local Binary Patterns) appliqués à la reconnaissance faciale

L'identification se fait en 3 étapes :

1. Chaque image subit un prétraitement destiné a atténué les effets des variations de luminosité.
2. Un descripteur est calculé pour chaque image.
3. L'identité prédite pour chaque image de la base de test est l'identité du plus proche voisin parmi les visages de la base d'apprentissage.

III.3 Prétraitement

Augmenter les contrastes des régions sombres et atténuer celui les régions lumineuses, supprimer le bruit et les gradients d'illumination et normaliser le contraste.

III.4 Descripteur d'image

Le descripteur LBP (Local Binary Pattern) permet la classification de textures. Il est calculé sur une région de 3×3 pixels.

$$\text{LBP}(x_c, y_c) = \sum_{n=0}^7 2^n s(i_n - i_c)$$

avec $s(u) = 1$ si $u > 0$ et 0 sinon, (X_c, Y_c) les coordonnées du point ou on calcul le descripteur, i_c la valeur de ce point et les i_n parcourant le voisinage comme indiqué sur la figure.



Figure 2 : Construction du motif 00101010.

L'opérateur LBP prend comme entrée un carré de 9 pixels. Un seuil est appliqué tel que tous les pixels périphériques dont la valeur est supérieure à la valeur du pixel central se voient attribuer la valeur 1 tandis que les autres reçoivent la valeur 0. La valeur LBP finalement obtenue est le nombre binaire se lisant dans le sens des aiguilles d'une montre autour du pixel central. Ensuite cet opérateur simple a été étendu pour rester fiable à différentes échelles. Ainsi, P points décrivent le nombre binaire et ceux-ci sont distribués le long d'un cercle de rayon R . Cet entourage sera noté (P, R) . Comme ces P points ne tombent pas nécessairement au centre d'un pixel de l'image, leurs valeurs sont obtenues par interpolation bilinéaire. Voir la figure suivante.

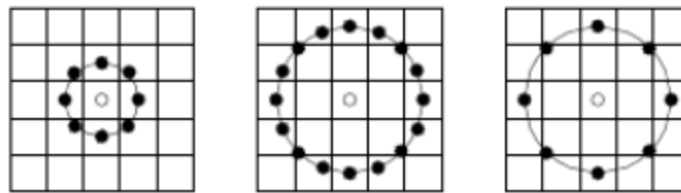


Figure 3 : Extension de l'opérateur LBP basique. Les entourages $(8,1)$, $(16,2)$, $(8,2)$ sont représentés.

De plus, on définit un motif uniforme comme étant un motif possédant exactement 0 ou 2 transitions 01 ou 10 (par exemple 10000001 ou 00011000, mais pas 00101010) [7] [8].

III.4.1 Histogramme

On calcule ensuite le LBP de chacun des pixels. Plutôt que de décrire l'image par la suite des motifs LBP, on va réduire la quantité de données en construisant un histogramme de dimension 59.

Chacune des 58 premières dimensions contiendra le nombre d'occurrences de l'un des motifs uniformes. La dernière contiendra le nombre d'occurrence de tous les motifs non uniformes, ce regroupement permettant de réduire la dimension sans perdre trop d'information utile.

Chaque image est découpée selon une grille régulière. L'histogramme est calculé sur chacune des cellules.

L'image est décrite par la concaténation de tous les histogrammes.

III.5 Plus proche voisin

La mesure de similarité est effectuée à l'aide d'une distance X^2 entre deux histogrammes :

$$\chi^2(x, y) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}$$

La recherche d'une image dans la base se fait simplement en calculant les distances entre l'image de test et toutes les images de la base et en choisissant la plus proche.

1	2	3
4	5	6
7	8	9

1	2
3	4

Figure 4 : Grille pour l'échantillonnage (grille « a » 3x3 et grille « b » 2x2).

III.6 Conclusion

Ce chapitre a été consacré à la présentation de la méthode de reconnaissance faciale LBP qui est basée sur l'utilisation du descripteur. Cette technique permet de diminuer le nombre de composantes tout en gardant l'information caractérisant l'objet à analyser sans perte d'informations significatives.

Conclusion

A travers ce projet nous avons fait la conception d'une application de reconnaissance faciale capable, de reconnaître les visages, celle-ci doit répondre à des exigences de rapidité et de robustesse des résultats, en utilisant une approche locale LBP (local Binary Pattern).

Dans le 1er chapitre on a fait une présentation de la reconnaissance faciale, les problèmes liés à son application ainsi que les différentes approches nécessaires à la réalisation d'un projet de reconnaissance facial. Le second chapitre est consacré à la présentation des outils qu'on utilisera dans la partie réalisation de notre projet ainsi que les contraintes quant au dernier chapitre il comporte les choix arrêtés.

Ce projet est un premier prototype de travail à réaliser avant la fin du 2eme semestre. Lors de la réalisation de ce projet nous serons confrontés à plusieurs problèmes parmi eux :

- L'utilisation de la librairie Opencv sur Android Studio.
- La détection de visage dans une image.
- La séparation du visage dans le paysage.

Bibliographie:

- [1] “A pratical Guide to Biometric Security Technology” Simon Liu and Mark Silverman.
- [2] “Face recognition: A literature survey” W. Zhao, R. Chellappa, P.J.Phillips, et A.Rosenfeld
- [3] “Biométrie et Authentification” *Djamila Mahmoudi. PhD, collaboratrice au département Corporate Information and Technology de Swisscom AG*
- [4] “Mixture model for face-color modeling and segmentation, Hayet Greenspan, Jacob Goldberger, Itay Eshet.
- [5] <http://www.sciencedirect.com/science/article/pii/S1077314212000318>
- [6] “Local Binary Patterns applied to Face Detection and Recognition” Laura Sánchez López
- [7] “Face Detection using LBP features” Jo Chang-yeon
- [8] “Face Recognition using Local Binary Patterns (LBP)” Abdur Rahim, Najmul Hossain, Tanzillah Wahid et Shafiul Azam
- [9] http://www.frandroid.com/android/developpement/141938_android-studio-face-a-eclipse-la-comparaison
- [10] <http://igm.univ-mlv.fr/~dr/XPOSE2011/SDKAndroid/android.html>
- [11] <https://openclassrooms.com/courses/creez-des-applications-pour-android/installation-et-configuration-des-outils>
- [12] <https://fr.wikipedia.org/wiki/OpenCV>
- [13] “Identification des personnes par reconnaissance de visage pour la sécurité d’une institution bancaire” Serge KOMANDA BASEMA.