

# BE - LBP-BASED FACE RECOGNITION

Capteurs et traitement d'images

**EMELINE GOT - ALEXANDRE JOYEUX**



ÉCOLE  
**CENTRALE** LYON

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Algorithme</b>	<b>4</b>
2.1	Reconnaissance LBP . . . . .	4
2.2	Distance Chi square . . . . .	4
<b>3</b>	<b>Expérimentation</b>	<b>5</b>
3.1	Images non divisées . . . . .	5
3.2	Images 3x3 . . . . .	5
3.3	Images 5x5 . . . . .	5
3.4	Images 7x7 . . . . .	6
3.5	Bilan des résultats . . . . .	6
<b>4</b>	<b>Courbe ROC</b>	<b>6</b>

# 1 Introduction

Dans ce bureau d'étude, nous cherchons à réaliser une méthode de reconnaissance des visages dans une banque d'image. Pour cela, nous avons à notre disposition 10 images de 40 personnes différentes. À l'aide de manipulation des images, de calcul de distance Chi Square, de reconnaissance par LBP et de validation croisée, nous avons mis en place deux types d'algorithmes :

- Un algorithme basé sur l'ensemble de l'image,
- Un algorithme s'appliquant sur des images divisées en 7x7, 5x5 ou 3x3 régions.

Dans un premier temps, nous allons présenter le principe de l'algorithme utilisé. Et ensuite nous présenterons les résultats obtenus.

## 2 Algorithme

### 2.1 Reconnaissance LBP

Le principe du LBP (Local Binary Patterns) est de considérer chaque pixel de l'image et ses 8 voisins respectifs. On compare la valeur du pixel central à ses voisins. Si la valeur est inférieure du pixel voisin est inférieure à celle du pixel central, alors on ne le prend pas en considération. Si par contre elle est supérieure, on fait la somme en binaire de ce pixel.

Cela est donné par la formule suivante :

$$LBP(x_c, y_c) = \sum_{p=0}^7 S(g_p - g_c) \times 2^p, \quad S(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

À partir des valeurs ainsi attribuées à chacun des pixels, on récupère l'histogramme associé. C'est-à-dire que l'on va compter le nombre de pixels valant chacune des valeurs entre 0 et 255 et les regrouper dans une liste.

Cette étape est effectuée par *LBP1.m* pour le calcul du LBP label de chaque pixel ainsi que *LBP\_hist.m* et *LBP\_hist\_n.m* pour l'obtention des histogrammes correspondants.

### 2.2 Distance Chi square

Afin de déterminer des coefficients de ressemblance entre les images, nous utilisons le calcul de distance Chi Square. Ce calcul permet de mesurer une distance entre deux images. La ressemblance sera effective en minimisant la valeur de cette distance.

Pour deux images simples, le calcul est donné par la formule suivante :

$$\chi^2(\alpha, \beta) = \sum_i \frac{(\alpha_i - \beta_i)^2}{\alpha_i + \beta_i}$$

Où  $\alpha$  et  $\beta$  représentent les valeurs des histogrammes de deux images.

Pour des images divisées en plusieurs régions et où l'histogramme LBP est calculé sur chacune des régions, nous pouvons effectuer le même calcul en ajoutant des poids sur les régions. C'est-à-dire que nous pouvons déterminer des régions qui sont plus susceptibles de participer à la ressemblance entre deux images. Dans le cas d'une image 7x7 par exemple, nous pouvons éliminer les pixels correspondants au fond de l'image pour se concentrer sur le visage de la personne.

Le calcul est donné par la formule suivante :

$$\chi^2(\alpha, \beta) = \sum_{j,i} w_j \frac{(\alpha_{i,j} - \beta_{i,j})^2}{\alpha_{i,j} + \beta_{i,j}}$$

Où  $w$  représente la valeur du poids.

Cette étape est effectuée par *Chi\_square.m* pour les images non divisées et *Chi\_square\_n.m* pour les images divisées.

### 3 Expérimentation

Pour tester les algorithmes obtenus précédemment, nous avons à disposition une banque de 400 images. Nous procédons par validation croisée. C'est-à-dire que nous sélectionnons dans la base de données 4 images de chaque sujet pour créer une base de visages de référence. Nous allons ensuite effectuer 10 tests avec des images aléatoires et tenter de reconnaître la personne associée à chaque image. Pour cela, nous cherchons l'image de référence avec laquelle l'image testée a la distance Chi Square minimale. Nous notons à chaque essai le taux correspondant au nombre de succès.

#### 3.1 Images non divisées

Dans le cas d'images non divisées, nous obtenons les résultats suivants pour les 10 essais : [0, 8500; 0, 8500; 0, 8583; 0, 8375; 0, 8541; 0, 8750; 0, 8541; 0, 8791; 0, 8791; 0, 8708].

Cela donne une moyenne de 86% de bonne reconnaissance des visages. Cela représente un bon taux de reconnaissance. Nous allons tenter de l'améliorer en divisant les images en plusieurs régions.

#### 3.2 Images 3x3

Dans le cas d'images divisées en 3x3 sections, nous avons les résultats suivants : [0, 9083; 0, 9166; 0, 9125; 0, 9208; 0, 8916; 0, 9166; 0, 8958; 0, 9083; 0, 9208; 0, 9083].

Cela donne une moyenne de 91% de bonne reconnaissance des visages. Nous avons amélioré le résultat à l'aide de la division des images et du poids ajouté sur les zones des images.

#### 3.3 Images 5x5

Dans le cas d'images divisées en 5x5, nous avons les résultats suivants :

[0, 8791; 0, 8666; 0, 9291; 0, 8833; 0, 8958; 0, 9083; 0, 9041; 0, 8875; 0, 9250; 0, 8875].

Ce qui fait une moyenne de 90%. Nous trouvons un résultat légèrement inférieur au précédent. Il faudrait peut-être modifier les poids attribués aux zones de l'images pour avoir une reconnaissance plus performante.

### 3.4 Images 7x7

Dans le cas d'images divisées en 5x5, nous avons les résultats suivants :

[0, 8375; 0, 8208; 0, 8542; 0, 8500; 0, 7917; 0, 8417; 0, 8167; 0, 8583; 0, 8833; 0, 8375].

Cela donne une moyenne de 84% de bonne reconnaissance des visages. On obtient un résultat inférieur au résultat obtenu pour une image non divisée. Ce n'est pas intéressant de diviser autant l'image pour la reconnaissance.

### 3.5 Bilan des résultats

Nous avons obtenu des résultats globalement satisfaisant pour la reconnaissance des images avec 90% de visages reconnus dans les meilleurs cas. Nous avons pu observer que la division des images pouvait permettre d'augmenter sensiblement le taux de reconnaissance car nous pouvons attribuer des poids plus importants à des zones clés de l'image comme la bouche ou les yeux. Néanmoins, cette méthode a des limites car si nous divisons trop l'image, il n'y a plus d'amélioration du processus et les résultats sont au contraire moins bons que si nous considérons l'image non divisée. Il faut donc trouver le bon taux de division et les bons poids à attribuer pour avoir une reconnaissance optimale.

## 4 Courbe ROC

Nous n'avons pas réussi à comprendre comment déterminer les taux de faux positifs pour pouvoir obtenir les valeurs des axes de la courbe ROC.