

ÉLECTIF : Capteurs et traitement d'images

BE3

LBP-based Face Recognition

Valentin Vié

Table des matières :

<u>I. INTRODUCTION</u>	<u>1</u>
<u>II. EXPLICATION DU CODE</u>	<u>1</u>
<u>III. RESULTATS</u>	<u>2</u>
A. Pas de régions, image entières.....	2
B. Régions 3x3.....	3
C. Régions 5x5.....	4
D. Régions 7x7.....	5
<u>CONCLUSION</u>	<u>6</u>

Introduction

Dans ce bureau d'étude, on se propose d'implémenter une méthode pour reconnaître des visages de manière efficace. On se base sur la méthode LBP pour traiter les images. On explicitera plusieurs variantes ainsi que leurs résultats.

I. Explication du code

On compare ici le taux de reconnaissance d'ordre 1. C'est à dire que l'on valide la reconnaissance si le visage de référence qui possède la plus petite distance avec le visage inconnu appartient à la même personne.

$$Recognition\ Rate = \frac{N\ recognized}{N\ tested}$$

Pour un visage donné, la première chose à faire est de calculer son masque LBP (défini dans le sujet). C'est le but de la fonction LBP_mask. Elle se sert de la méthode neighbour qui renvoie une liste de voisin du pixel (i,j) dans l'ordre de parcours défini par le sujet.

Ensuite on calcule l'histogramme de la matrice obtenue après la transformation LBP. On propose ici plusieurs variantes, soit on calcule l'histogramme sur toute la matrice (méthode compute_hist) soit on divise notre image en régions de n*n pixels et on calcule l'histogramme sur chaque région. Cela donne finalement une matrice d'histogramme : compute_hist_n.

Par la suite, il faut mesurer la distance entre chaque histogramme calculé. Cette distance est donnée par chi_single_hist si on ne choisit pas de diviser notre image en régions ou par chi_multiple_hist si on choisit de la diviser en régions. On applique des poids différents à chaque région dans le calcul de la distance. En effet, on accorde trois fois plus d'importance au yeux qu'aux autres régions ainsi que deux fois plus pour la bouche et le nez.

On choisit ensuite 4 visages de référence au hasard à l'aide de la fonction alea_liste. On choisit ensuite de les stocker dans deux matrices : T et R (respectivement les visages « à tester » et les visages de référence) grâce à la fonction reference_single_list ou reference_list.

Les fichiers pour l'exécution du programme sont les suivants : BE2_main_#* calcule le taux de reconnaissance du premier ordre avec #* régions. RocCurve_#** calcule la courbe ROC avec #** régions.

Pour plus d'information concernant le code, voir les commentaires.

II. Résultats

A. Pas de régions, image entières

Ici, on ne divise pas l'image en régions, on a un seul histogramme et il s'agit d'interpréter les distances selon la loi du χ^2 entre les différents visages.

On sélectionne toujours 4 images aux hasard comme références. On calcule le taux de réussite sur les autres images 10 fois en changeant de référence. Ceci nous donne les moyennes suivantes : 0,8333 - 0,9250 - 0,9042 - 0,8292 - 0,8250 - 0,8167 - 0,8792 - 0, - 0,8417 - 0,8583.

Donc en moyenne, on a un taux de reconnaissance de 85,71%, avec un écart type de $\sigma = 0,0358$. Pour un FAR de 0,01 on a un taux de vérification de 49%.

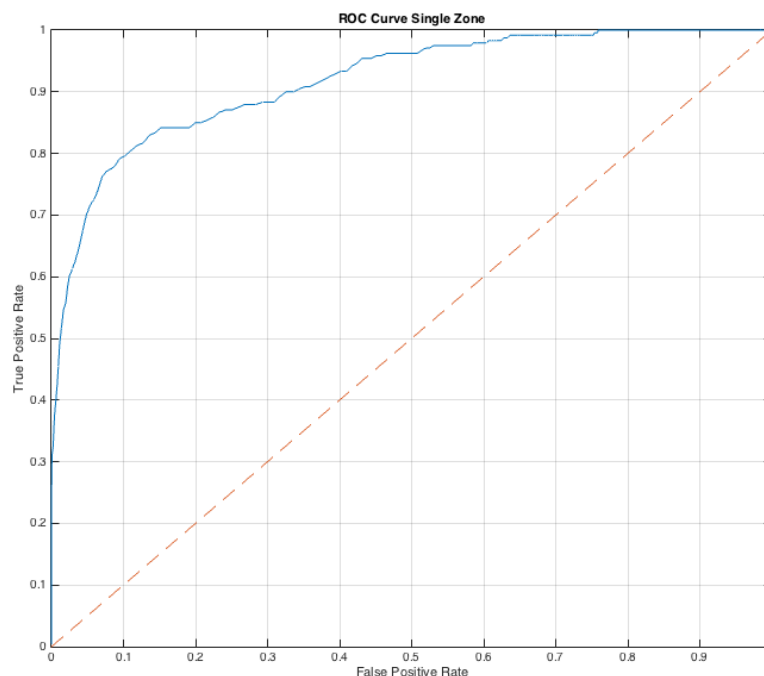


Figure 1 - Courbe ROC sans région

B. Régions 3x3

On divise ici le visage en 3 régions comme sur la figure ci dessous. Les régions sont de taille 31x31 pixels.

On sélectionne toujours 4 images aux hasard comme références. On calcule le taux de réussite sur les autres images 10 fois en changeant de référence. Ceci nous donne les moyennes suivantes : 0,8750 - 0,9542 - 0,9208 - 0,9250 - 0,8917 - 0,9250 - 0,9250 - 0,9500 - 0,9500 - 0,9042.

Donc en moyenne, on a un taux de reconnaissance de 92,21%, avec un écart type de $\sigma \equiv 0,0260$. Pour un FAR de 0,01 on a un taux de vérification de 70%.

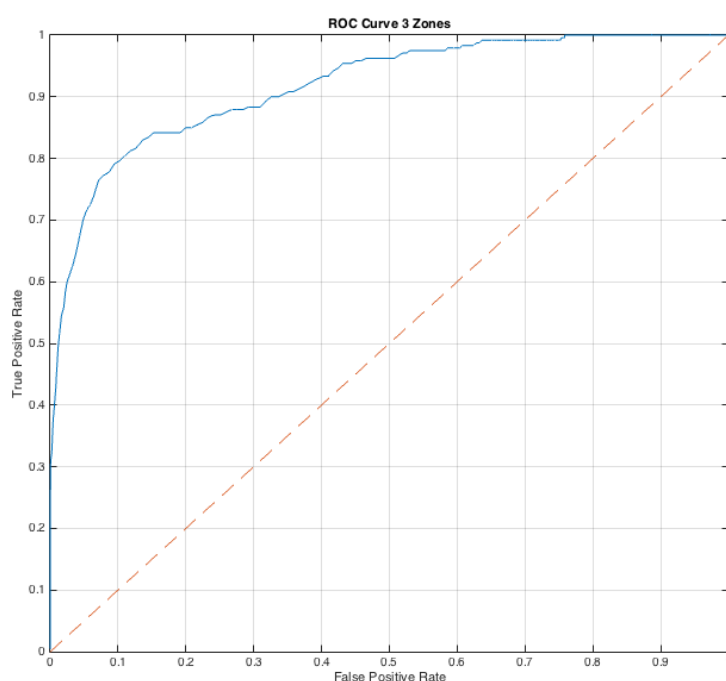


Figure 2 - Courbe ROC 3x3 régions

C. Régions 5x5

On divise ici le visage en 5 régions comme sur la figure ci dessous. Les régions sont de taille 18x18 pixels.

On sélectionne toujours 4 images aux hasard comme références. On calcule le taux de réussite sur les autres images 10 fois en changeant de référence. Ceci nous donne les moyennes suivantes : 0,9250 - 0,9667 - 0,9458 - 0,8958 - 0,9125 - 0,9292 - 0,8875 - 0,9292 - 0,8917 - 0,9375.

Donc en moyenne, on a un taux de reconnaissance de 92,23%, avec un écart type de $\sigma \equiv 0,0254$. Pour un FAR de 0,01 on a un taux de vérification de 73%.

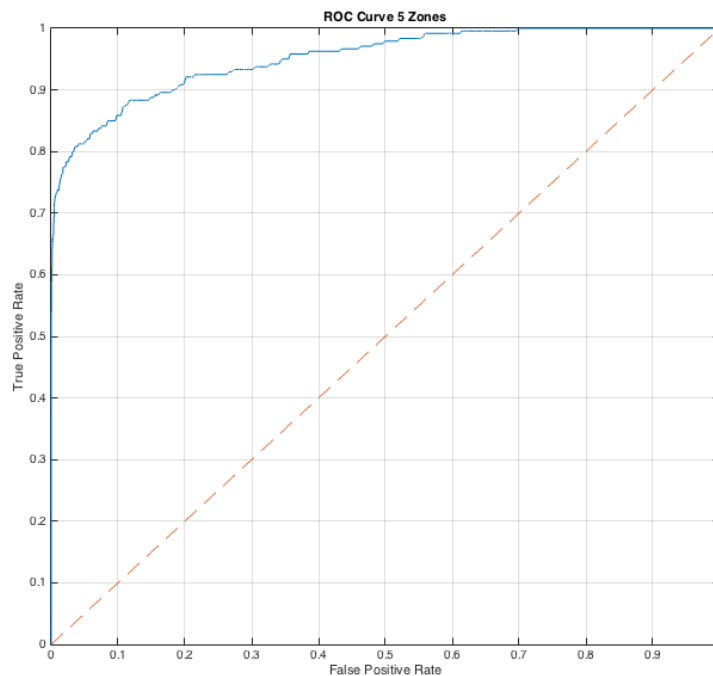


Figure 3 - Courbe ROC 5x5 régions

D. Régions 7x7

On divise ici le visage en 7 régions comme sur la figure ci dessous. Les régions sont de taille 13x13 pixels.

On sélectionne toujours 4 images aux hasard comme références. On calcule le taux de réussite sur les autres images 10 fois en changeant de référence. Ceci nous donne les moyennes suivantes : 0,8875 - 0,8750 - 0,9000 - 0,9125 - 0,9083 - 0,8958 - 0,8542 - 0,8708 - 0,8667 - 0,8750,

Donc en moyenne, on a un taux de reconnaissance de 88,46% avec un écart type de $\sigma \equiv 0,0193$. Pour un FAR de 0,01 on a un taux de vérification de 68%.

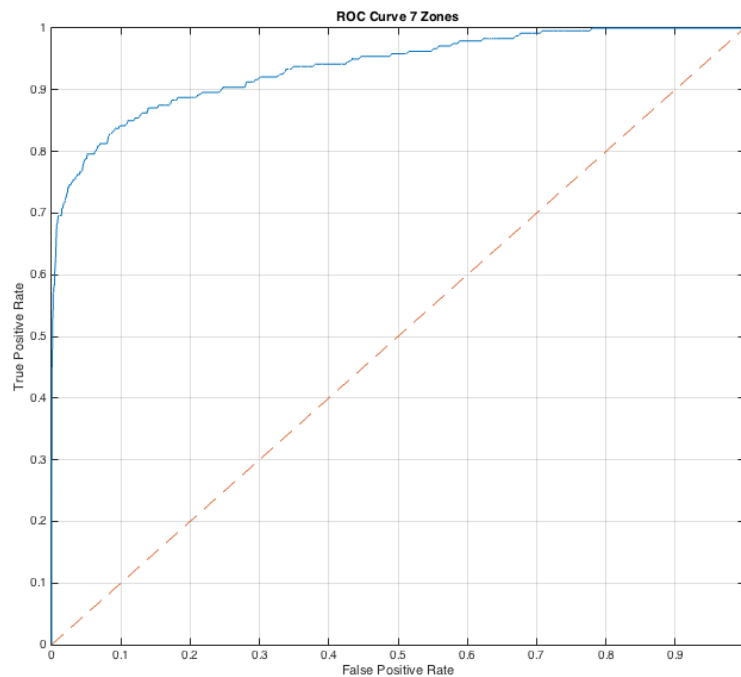


Figure 4 - Courbe ROC 7x7 régions

Conclusion

En conclusion la méthode se révèle assez peu perturbée en changeant d'échantillons. De plus le taux de reconnaissance de 90-95%% est vraiment bon. On est alors presque sûr de trouver la personne attendue si l'on prend les premières correspondances les plus proches. On remarque que si l'on divise l'image par régions, on améliore le résultat. En revanche, si l'on augmente trop le nombre de régions, on est trop dans le détail et l'on diminue le taux de reconnaissance. Il faut alors trouver le juste équilibre entre la division pour accentuer le poids les zones intéressantes et l'importance que l'on accorde aux détails.

D'après les courbes ROC se basant sur les quatre premières images comme références, on trouve que c'est en divisant en 3x3 ou 5x5 que l'on obtient les meilleurs résultats.