CONTRIBUTION A L'ETUDE DE L'APPRENTISSAGE D'ENSEMBLE AVEC DES SOUS MODELES HYBRIDES CNN_SIFT ET CNN_D SIFT APPLIQUEE A LA RECONNAISSANCE DE L'EXPRESSION FACIALE

RABARIJAONA E.S[1], RANDRIAMITANTSOA P.A[2]

Mention Télécommunication

Ecole Supérieur Polytechnique d'Antananarivo (ESPA)

Université d'Antananarivo

[1]eliisorabary@gmail.com, [2]rpauguste@gmail.com

Résumé

L'extraction des composants efficaces pour la reconnaissance de l'expression faciale est importante pour un système d'interaction homme-machine réussi. Néanmoins, reconnaître l'expression du visage reste une tâche difficile. Ce mémoire décrit une nouvelle approche de la tâche de reconnaissance des expressions faciales qui favorise l'obtention d'une bonne précision avec peu de données de formation. La méthode proposée est motivée par le succès des réseaux de neurones convolutif sur le problème de classification d'image. Et puisque SIFT et Dense SIFT ne nécessite pas de données d'entraînement approfondies pour générer des fonctionnalités utiles. Nous avons fusionné les deux méthodes pour former des modèles hybrides. De plus, les modèles hybrides sont combinés avec deux autres modèles CNN. Le modèle combiné est testé sur un ensemble de test de Fer2013. Les résultats démontrent la supériorité de CNN_SIFT sur CNN_D SIFT et les CNN conventionnels. La précision a même augmenté lorsque tous les modèles sont combinés, ce qui génère des résultats de pointe où elle a atteint une précision de 72.89 %.

Mots clés : reconnaissance des expressions faciales, CNN, dense SIFT, SIFT

Abstract

Extracting effective components for facial expression recognition is important for a successful human-machine interaction system. Nevertheless, recognizing facial expression remains a difficult task. This thesis describes a new approach to the task of facial expression recognition that promotes good accuracy with little training data. The proposed method is motivated by the success of convolutional neural networks on the image classification problem. And since SIFT and Fast SIFT do not require extensive training data to generate useful features. We merged them with both methods to form hybrid models. In addition, the hybrid models are combined with two other CNN models. The combined model is tested on a test set of Fer2013. The results demonstrate the superiority of CNN_SIFT over CNN_D SIFT and

conventional CNNs. Accuracy even increased when all models were combined, generating peak results where it reached an accuracy of 72.89%.

Keywords: facial expression recognition, CNN, dense SIFT, SIFT

1. Réseaux de neurones convolutifs

1.1 Entrainement de CNN

1.1.1 Propagation en avant

La sortie du neurone à la ligne x, à la colonne y, à la $l-$ème couche de convolution et au $k-$ème noyau est :

$$O_{x,y}^{(l,k)} = ReL\,U\left(\sum_{t=0}^{f-1}\sum_{r=0}^{k_h}\sum_{c=0}^{k_w} W_{(r,c)}^{(k,t)}\, O_{(x+r,x+c)}^{(l-1,t)} + b^{(l,k)}\right) \quad (01)$$

Où f est le nombre de noyaux de convolution.

Et la sortie du $i-$ème neurone de la dernière couche MLP est :

$$O_{(l,i)} = ReLU(\sum_{j=0}^{H} O_{(l-1,j)} W_{(i,j)}^{l} + b^{(l,i)}) \quad (02)$$

Avec H, le nombre de neurones de la couche précédente.

1.1.2 Rétropropagation

Pour un échantillon de donnée de taille P, le MSE est donné par :

$$E = \frac{1}{2}\sum_{p=1}^{P}(t_p - a_p^L)^2 \quad (03)$$

Avec a_p^L, les sorties prédites par le modèle, et t_p, les valeurs réelles correspondantes.

L'apprentissage peut être achevée en ajustant les poids de sorte que A^L soit de plus en plus proche de T.

Le gradient pour chaque poids des couches à convolution peut être obtenu par la formule suivante :

$$\delta_{x,y}^{l} = \delta_{x,y}^{l+1} * rot_{180°}(w_{x,y}^{l+1}) f'(o_{x,y}^{l}) \quad (04)$$

Avec

$$\delta_{x,y}^{l} = \frac{\partial E}{\partial o_{x,y}^{l}} \quad (05)$$

1.2 Avantages de CNN

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation. C'est le principal avantage du CNN par rapport au MLP.

Comparés à d'autres algorithmes de classification de l'image, les réseaux de neurones convolutifs utilisent relativement peu de prétraitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres, ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est un atout majeur des CNN.

2. Scale-Invariant Feature Transform

La méthode des SIFT (scale-invariant feature transform ou transformation de caractéristiques visuelles invariantes à l'échelle), est une méthode permettant de transformer une image en ensemble de vecteurs de caractéristiques qui sont invariantes par transformations géométriques usuelles (homothétie, rotation).

2.1 L'espace des échelles

L'espace des échelles est un espace discret dans lequel on affecte à chaque pixel, en plus de ses coordonnées cartésiennes (x, y), une troisième composante σ, qui représente le facteur d'échelle. Pour cela, on effectue une convolution classique entre l'image de départ I et une gaussienne qui prend en argument x, y et σ :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (06)$$

Où :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (07)$$

On obtient alors ce qu'on appelle un gradient de facteur d'échelle σ. Ce filtre a pour effet de lisser l'image et d'atténuer les contours. Pour construire la pyramide :

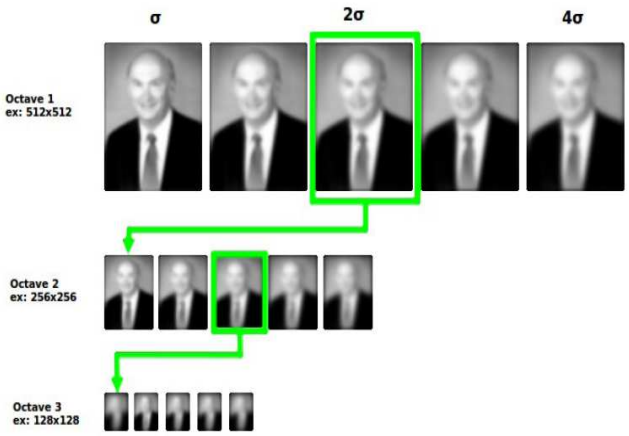

Figure 01 : *Pyramide Gaussien*

2.2 Détection des extrema dans les DoG

L'idée consiste à faire une différence de gaussiennes (DoG), entre deux images consécutives d'une même octave dans la pyramide de gaussiennes pour obtenir une pyramide de DoG.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (08)$$

Où k est un nombre constant afin d'obtenir un nombre fixe d'images lissées par octave, et de garantir que nous aurons le même nombre de DoG par octave.

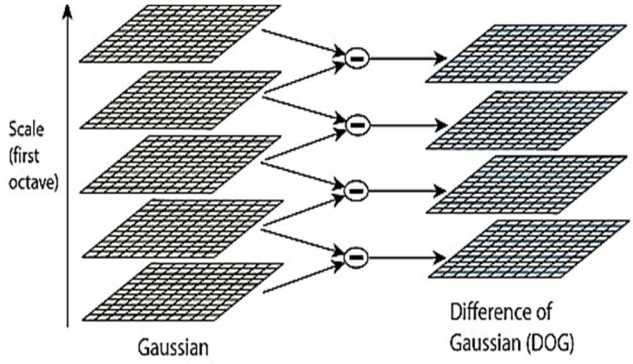

Figure 02 : *Différence de Gaussien*

Ainsi on peut récupérer les extremas, en comparant chaque pixel par ses 8 voisins et également à ses 9 voisins au-dessus et en dessous, Si la valeur du pixel est inférieure ou supérieure à celles des 26 voisins, alors celui-ci est considéré comme un *extremum* local,

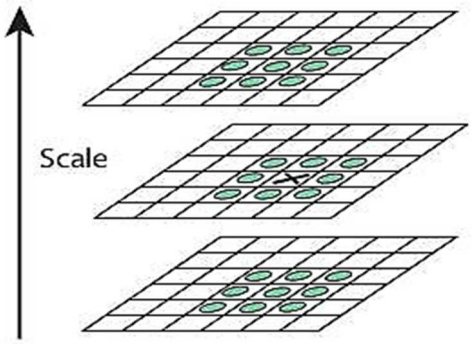

Figure 03 : *Détections des extremas par DoG*

2.3 Localisation précise des points d'intérêt

2.3.1 L'interpolation des coordonnées

En effet, il est possible d'effectuer une interpolation des coordonnées des points où se trouvent les extremums.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (09)$$

La localisation de l'extremum $\hat{x}$ réel est déterminée en prenant la dérivée de cette fonction par rapport à x en 0, ce qui donne :

$$\frac{\partial D}{\partial x} + \frac{\partial^2 D}{\partial x^2}\hat{x} = 0 \quad (10)$$

$$\frac{\partial^2 D}{\partial x^2}\hat{x} = -\frac{\partial D}{\partial x} \quad (11)$$

Où :

$$\hat{x} = \begin{bmatrix} x \\ y \\ \sigma \end{bmatrix} = -\begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial x} \\ \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y\sigma} \\ \frac{\partial^2 D}{\partial x\sigma} & \frac{\partial^2 D}{\partial y\sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix}$$

Cette étape sert à approximer un peu plus finement les valeurs du point $X = (x, y, \sigma)$. Pour déterminer si la position du point candidat est exacte, on regarde $\hat{x} - x$ dans les 3 dimensions i.e, x, y et σ. On regarde si la position obtenue est stable et si la localisation obtenue par interpolation est suffisamment proche de celle trouvée au départ :

- si $\hat{x} - x > +0.5$ dans n'importe quelle dimension alors on réévalue l'interpolation au point $x + 1$
- si $\hat{x} - x < -0.5$ dans n'importe quelle dimension alors on réévalue l'interpolation au point $x - 1$

2.3.2 Rejet des points à faible contraste

Afin d'améliorer encore la méthode et pour justifier l'invariance des PI à l'illumination, Lowe propose de rejeter des points à faible contraste. On regarde alors la valeur de D en $\hat{x}$:

$$D(\hat{x}) = D + \frac{1}{2}\frac{\partial D^T}{\partial x}\hat{x} \quad (12)$$

Si $|D(\hat{x})| < 0.03$, alors $\hat{x}$ est considéré comme un point à faible contraste et il est rejeté de l'ensemble des points d'intérêts. Il est également mieux d'éliminer les points de contours, sur lesquels la DOG a de fortes réponses ce qui peut donner naissance à des extremums locaux instables.

2.3.3 Matrice des dérivées partielles secondes Hessienne

Afin de ne pas prendre en compte les points non pertinents on se base sur le critère de la courbure de ces points le long du contour. La courbure principale peut-être calculée grâce à la Hessienne H à la position et à l'échelle du PI.

$$H(x, y, \sigma) = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2}(x, y, \sigma) & \frac{\partial^2 D}{\partial x \partial y}(x, y, \sigma) \\ \frac{\partial^2 D}{\partial x \partial y}(x, y, \sigma) & \frac{\partial^2 D}{\partial y^2}(x, y, \sigma) \end{bmatrix} \quad (13)$$

H est symétrique donc ses valeurs propres sont réelles. Les valeurs propres de cette matrice nous donnent des informations sur la courbure principale de D car elles sont proportionnelles avec celle-ci. On utilise les propriétés de cette matrice. Ces propriétés nous permettent de ne pas calculer explicitement les valeurs propres et de gagner en temps de calcul.

$$tr(H) = Dxx + Dyy = \lambda_1 + \lambda_2 \quad (14)$$

$$Det(H) = Dxx * Dyy - (Dxy)^2 \quad (15)$$
$$= \lambda_1 * \lambda_2$$

Avec

- $Dxx = \frac{\partial^2}{\partial x^2} D$
- $Dyy = \frac{\partial^2}{\partial y^2} D$
- $Dxy = \frac{\partial^2}{\partial x \partial y} D$

On suppose que λ_1 est la plus grande valeur propre et λ_2 la plus petite. On peut calculer le rapport $r = \lambda_1/\lambda_2$ et $\lambda_1 = r * \lambda_2$. On étudie le nombre :

$$R = \frac{tr(H)^2}{Det(H)} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} \quad (16)$$
$$= \frac{(r\lambda_2 + \lambda_2)^2}{{\lambda_2}^2} = \frac{(r+1)^2}{r}$$

SIFT choisit un seuil empirique r (r = 10 dans l'article) et si $R < (r+1)^2/r$ alors on retient le point considéré. Ce calcul permet d'éliminer les PI dont le rapport entre les deux courbures principales est plus grand que 10.

2.4 Affectation d'orientation aux points d'intérêt

Un voisinage est pris autour de l'emplacement du point clé en fonction de l'échelle. La magnitude et la direction du gradient sont calculées dans cette région. Un histogramme d'orientation avec 36 cases couvrant 360 degrés est créé.

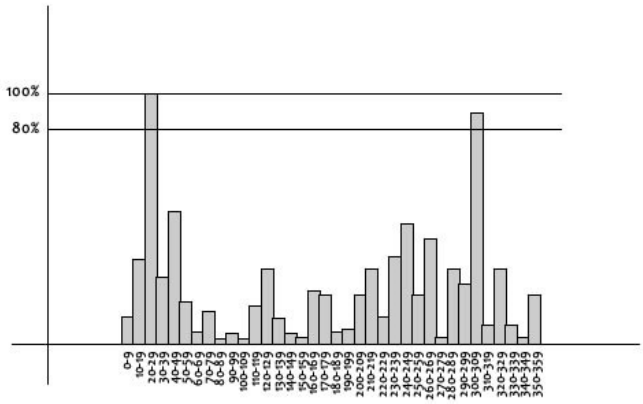

Figure 04 : *Histogrammes d'orientations des voisinages du point clé de 0 à 360 degrés*

Le pic le plus élevé de l'histogramme est pris et tout pic supérieur à 80% de celui-ci est également considéré pour calculer l'orientation.

2.5 Descripteur de point clé

À ce stade, chaque point clé a un emplacement, une échelle et une orientation. Ensuite, il faut calculer un descripteur pour la région d'image locale autour de chaque point clé qui est hautement distinctif et invariant autant que possible aux variations telles que les changements de point de vue et d'éclairage.

Pour ce faire, une fenêtre 16x16 divisé en 16 sous-blocs de 4x4 autour du point-clé est prise. Il est divisé en 16 sous-blocs, Pour lesquels, un histogramme des orientations du gradient est construit comme montre la **Figure 05** ci-dessous. Ensuite, comme dans la **Figure 06**, pour chaque sous-bloc, un histogramme d'orientation de 8 bacs est créé.

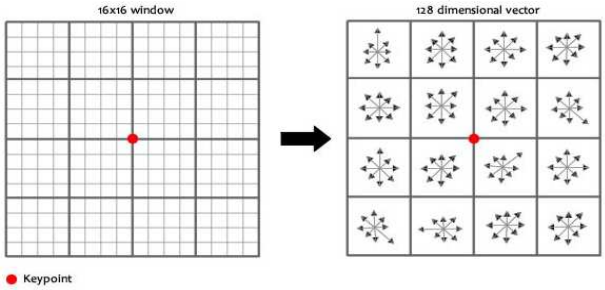

Figure 05 : *Histogramme d'orientation des 16 sous blocs*

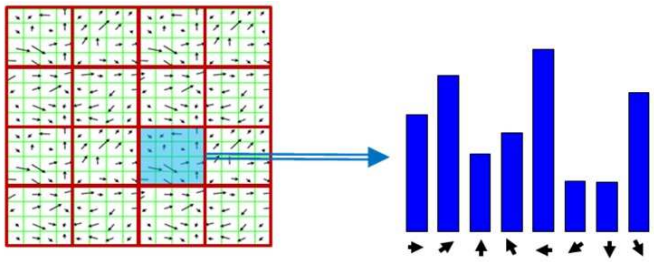

Figure 06 : *Histogramme d'orientation de 8 bacs*

Les directions 4 x 4 x 8 donnent 128 valeurs de casier. Ces valeurs sont représentées comme un vecteur de caractéristiques pour former un descripteur de point-clé.

3. Algorithme de Dense SIFT

L'algorithme de Dense SIFT est une bonne alternative de SIFT. Avec SIFT, nous obtenons un descripteur à chaque emplacement d'un point d'intérêt, tandis qu'avec Dense SIFT, nous obtenons un descripteur à chaque emplacement dans l'image.

3.1 Descripteur de Dense SIFT

Une opération SIFT dense commence par segmenter une image en échelle de gris en segments plus petits, ou patchs, de taille 8 × 8 pixels. Chacun de ces patchs est en outre divisé en 2 x 2 segments plus petits. Pour chacun de ces seize (4 × 4) segments, qui représentent les voisinages autour du point caractéristique (centre du patch), les gradients d'images ont été calculés. Comme il existe huit directions (45° chacun), un histogramme pondéré lissé de huit cases est créé en fonction de la valeur du gradient.

Donc la procédure de descripteur est le même qu'avec SIFT mais il n'y a pas de détections de point d'intérêt en Dense SIFT.

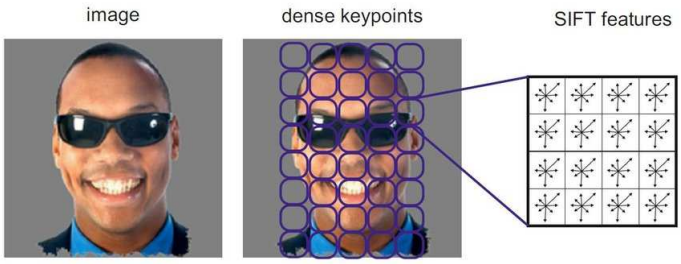

Figure 07 : *Descripteur de Dense SIFT*

4. Description des données FER-2013

Les données FER-2013 consistent en des images en niveaux de gris de 48 x 48 pixels de visages. La tâche consiste à classer chaque visage en fonction de l'émotion montrée dans l'expression du visage dans l'une des sept catégories (0 = colère, 1 = dégoût, 2 = peur, 3 = heureux, 4 = triste, 5 = surprise, 6 = neutre).

Le fichier de FER-2013 contient deux colonnes, "émotion" et "pixels". La colonne « émotion » contient un code numérique allant de 0 à 6 inclus, pour l'émotion présentée dans l'image. La colonne "pixels" contient les valeurs des pixels.

L'ensemble de formation comprend 28 709 exemples. L'ensemble de tests publics utilisés pour le classement se compose de 3 589 exemples. L'ensemble de tests privés, qui est utilisé pour tester le modèle après son entrainement afin de

l'évaluer, se compose de 3 589 exemplaires supplémentaires.

5. Implémentations des algorithmes hybrides CNN_SIFT et CNN_D SIFT

Les fonctionnalités SIFT et Dense SIFT sont fusionnées avec les fonctionnalités CNN dans la première couche entièrement connecté de CNN. L'existence des fonctionnalités SIFT ou D_SIFT au cours de la formation CNN l'aide à apprendre la représentation des différentes fonctionnalités de SIFT et à faire en sorte que CNN et SIFT se complètent.

Les **Figures 08** et **09** montrent l'architectures des algorithmes hybrides CNN_SIFT et CNN_Dense SIFT.

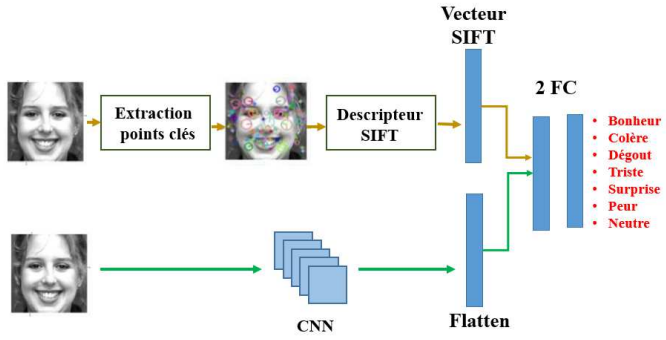

Figure 08 : *CNN avec SIFT*

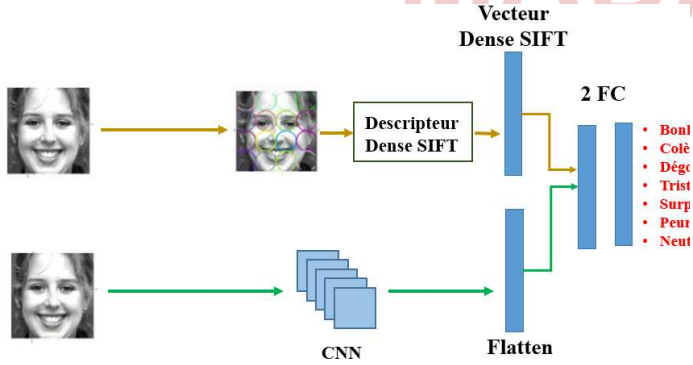

Figure 09 : *CNN avec Dense SIFT*

6. Progressions de l'entrainement

Les deux courbes des **figures 10 et 11** suivantes montrent les valeurs des pertes calculées durant la phase d'apprentissage.

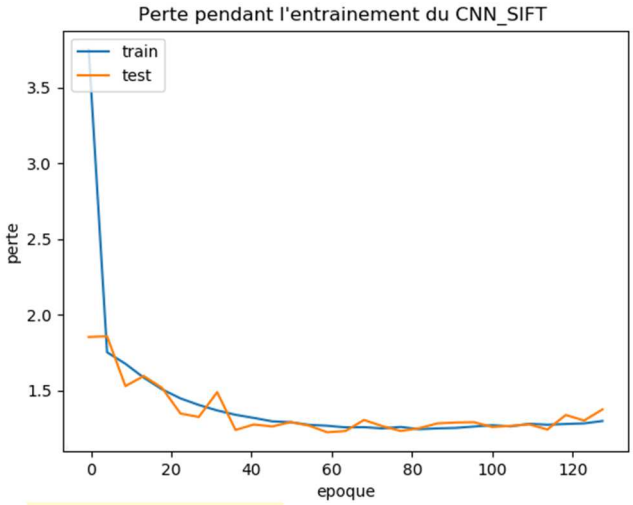

Figure 10 : *Perte pendant l'entrainement du CNN_SIFT*

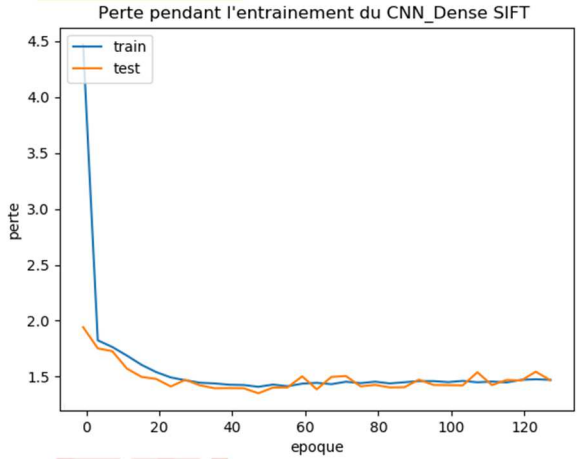

Figure 11 : *Perte pendant l'entrainement du CNN_D SIFT*

7. Apprentissage d'ensemble des sous modèles

Maintenant, nous avons quatre modèles entrainés séparément avec Fer2013. Pour augmenter la précision du modèle final, les sorties des CNN_1, CNN2, CNN_SIFT et CNN_D SIFT sont agrégées en utilisant la somme moyenne. Donc la probabilité qu'une image d'entrée x contienne une expression e est :

$$P(e|x) = \frac{Pr_1(e|x) + Pr_2(e|x) + Pr_3(e|x) + Pr_4(e|x)}{4} \quad (17)$$

Avec Pr_1, Pr_2, Pr_3, Pr_4 les prédictions des modèles CNN_1, CNN2, CNN_SIFT et CNN_D SIFT respectivement.

8. Accuracy du modèle

Nous avons testé notre modèle avec un ensemble de donné priver du fer2013 et calculé la précision des modèles par MAPE.

$$MAPE = \frac{100}{n}\sum\left|\frac{y - \hat{y}}{y}\right| \quad (18)$$

Avec y la valeur de l'étiquette et $\hat{y}$ la valeur prédite. Ainsi nous pouvons évaluer la performance du modèle avec l'ensemble de test privé.

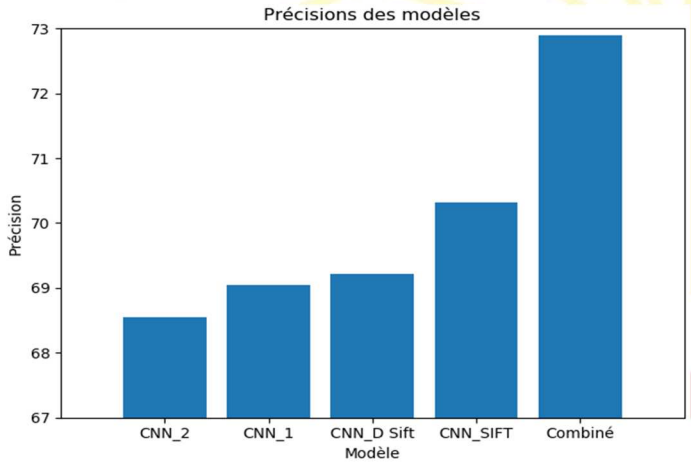

Figure 12 : Valeurs de précisions de chaque modèle

8.1 Interprétation

Donc nous pouvons observer maintenant CNN_1 a un accuracy plus élevé que CNN_2 puisqu'elle présente beaucoup plus de couche. Mais les deux modèles CNN ne peuvent rivaliser avec les précisions des deux hybrides CNN_SIFT et CNN_D SIFT. La précision de CNN_SIFT est plus élevée que celui de CNN_D SIFT puisque sa méthode d'extraction est plus efficace donc plus précise. Enfin nous pouvons affirmer que la précision du modèle combiné s'élève à 72.89 %. Ce qui augmente de 2.57% de la précision de CNN_SIFT.

9. Matrice de confusion du modèle

Comme avec l'accuracy, nous allons comparer la matrice de confusion du modèle combiné avec les deux modèles hybrides. Les **figures 13, 14, et 15** sont les trois matrices de confusion que nous allons comparer dans l'interprétation.

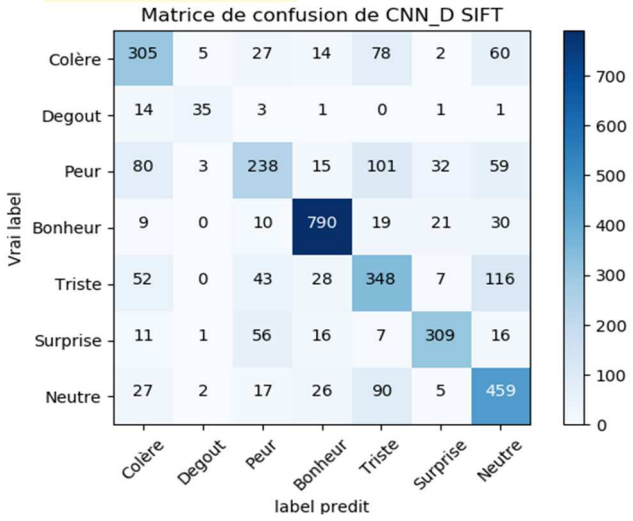

Figure 13 : *Matrice de confusion de CNN_Dense du modèle SIFT*

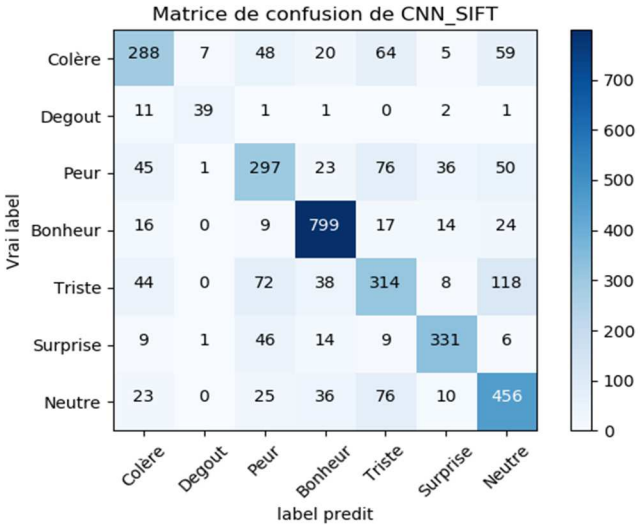

Figure 14 : *Matrice de confusion du modèle CNN_SIFT*

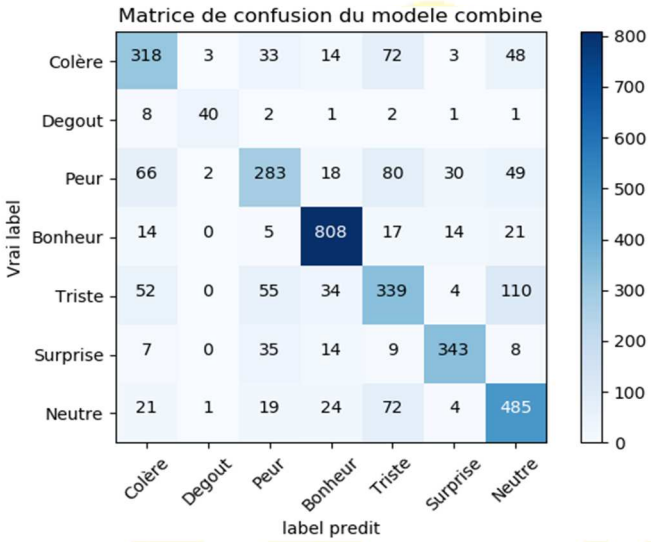

Figure 15 : *Matrice de confusion du modèle combiné*

9.1 Interprétation

Nous avons comparé la matrice de confusion du modèle combiné avec les modèles hybrides CNN_SIFT et CNN_D SIFT. Nous pouvons affirmer d'abord que chaque modèle présente des difficultés à distinguer les expressions peur, triste, colère et neutre.

Mais les nombres de vrais positifs, et de vrais négatifs de la plupart des sept classes du modèle combiné sont supérieurs par rapport aux deux autres modèles hybrides.

10. Test avec la webcam

Les **figures 16 et 17** suivantes sont des captures d'écran de mon visage sur la webcam détecté par notre modèle.

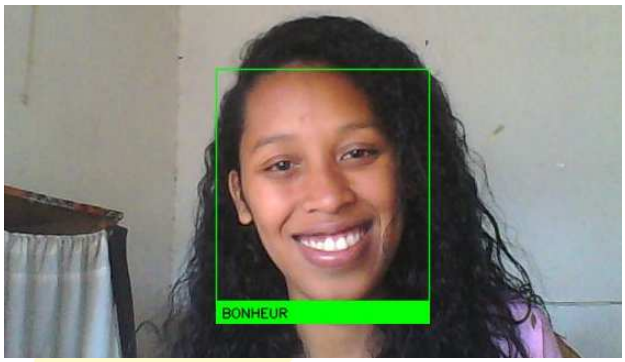

Figure 16 : *Capture d'écran d'une détection d'expression faciale via la webcam*

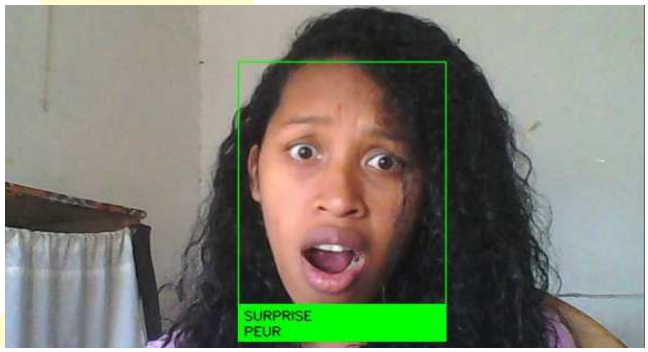

Figure 17 : *Capture d'écran d'une détection d'expression faciale via la webcam*

11. Conclusion

Nous avons montré comment les fonctionnalités SIFT et Dense SIFT et le réseau neuronal à convolution pouvaient se compléter pour améliorer la précision du résultat. Et la méthode d'apprentissage d'ensemble qui combine quatre modèles a obtenu une précision de 72.89 %.

12. Références

[1] T. Liu, S. Fang, « *Implementation of Training Convolutional Neural Networks* », University of Chinese, Juin 2015

[2] M. Zakaria « *classification des images avec les réseaux de neurone convolutif* », Université Abou Bakr, 03 Juillet 2017

[3] T. Jefkine, « *Backpropagation In Convolutional Neural Networks* », http://www.jefkine.com, 5 Septembre 2016

[4] D. Tyagi, « *Introduction to SIFT (Scale Invariant Feature Transform)* », Data Breach, 16 Mars 2019

[5] P. Poublang « *L'algorithme des SIFT* », Université de la Méditerranée., A.U. : 15 mai 2012

[6] V. Tan, M. Wandi, « *Tensor Decomposition of Dense Sift Descriptor in Object Recognition* », University of Canberro, Australia, 23 avril 2014

[7] J. Ramzai, « *Simple guide for ensemble learning methods* », Towards Data Science, 26 fierier 2019.