Détection de Visages à l'Aide de Réseaux de Neurones

Erwan Le Martelot

17 janvier 2005

Table des matières

Résumé

L'objectif de cet article est de présenter le principe de la détection de visages dans des images à l'aide de réseaux de neurones. Il est basé sur les travaux de Henry A. Rowley, Shumeet Baluja, et Takeo Kanade présentés dans [1]. Plusieurs problématiques sont exposées et amènent à définir une chaîne de traitement. Tout d'abord, les données d'entrée des réseaux doivent être normalisées, d'où la nécessité de définir un filtrage précédent la détection. Ensuite, bien que la technique des réseaux de neurones soit, par construction très polyvalente, l'application étudiée ici va amener à définir une topologie spécifique pour ces réseaux employés pour la détection. Par la suite se pose le problème de la supervision de l'apprentissage. Si en effet un visage se définit selon des critères bien précis, un "non visage" n'a pas de définition. Aussi seront abordés des techniques pour définir les jeux d'exemples et de contre-exemples de l'apprentissage. De plus, comme beaucoup de techniques imparfaites, il y a des détections obsolètes voire fausses. C'est pourquoi des méthodes et heuristiques d'optimisation seront proposées. Enfin, un visage pouvant se présenter sous différents angles, une méthode sera exposée afin d'enrichir les possibilités du système.

Introduction

L'analyse du contenu des images et la reconnaissance de formes sont des domaines d'applications très utilisés de nos jours et rendus de plus en plus efficaces par la puissance croissante des machines.

La détection de visages, traitée ici, illustre bien les difficultés rencontrées dans ce type d'applications. En effet, toute reconnaissance passe par des critères de reconnaissance. Il faut donc pouvoir définir ce qui est recherché dans l'image. Un visage est quant à lui relativement simple à définir. C'est sa définition qui nous amènera à définir une topologie relativement intuitive pour les réseaux de neurones utilisés.

Afin de rechercher des critères, il faut permettre à nos données d'être comparables à nos critères, il faut donc au préalable normaliser ces données. C'est ce qui nous amènera tout d'abord à définir un filtrage pré détection. Ensuite, nous pourrons définir la topologie des réseaux. Puis, afin de permettre un apprentissage, il faudra définir ce qui doit être supervisé par l'utilisateur et ce qui peut être automatiquement supervisé par l'application. Seront à cette fin présentées des méthodes et heuristiques allégeant à la fois l'apprentissage et optimisant les résultats.

La chaîne de traitement de la figure 1 montre l'enchaînement des étapes de la phase de détection du système de base proposé.

FIG. 1 – chaîne de traitement de la détection

Enfin, je porterai l'étude en fin d'article sur le cas des visages frontaux présentés sous différents angles.

Cet article est basé sur les travaux de Henry A. Rowley, Shumeet Baluja, et Takeo Kanade présentés dans [1], lesquels décrivent une méthode précise de détection de visages dans des images. Ce qui sera présenté ici sera moins spécifique à une méthode particulière. L'étude aura pour but, d'une part, de présenter les étapes et méthodes requises pour obtenir un bon système de base de détection, d'autre part de discuter l'utilisation des réseaux de neurones dans le contexte.

1 Collecte des données : Balayage et Filtrages

1.1 Balayage de l'image

L'objectif recherché étant la détection de visages dans une image, il faut d'abord définir une fenêtre qui va parcourir l'image à la recherche d'un visage. Cette fenêtre doit être de taille fixe pour servir de donnée entrante aux réseaux. C'est ce qui nous amènera à balayer l'image à plusieurs échelles. Dans l'article [1], les auteurs proposent une fenêtre de taille 20x20 pixels. Aussi, la détection de visage ne pourra s'opérer que sur des images d'une taille minimale de 20x20 pixels. Le balayage démarrera donc sur une image à sa taille initiale, puis l'image sera successivement réduite à chaque changement d'échelle afin de pouvoir repérer des visages plus ou moins près sur cette image (Figure 2).

Les auteurs proposent pour cette phase de déplacer le fenêtre sur chaque pixel de l'image et de prendre 1.2 pour facteur d'échelle. La vitesse de déplacement ainsi que l'échelle peuvent être modifiées mais si par exemple un déplacement de 2 au lieu de 1 divisera par deux le temps d'analyse total, la détection sera moins précise. Il en va de même pour l'échelle. Aussi, un gain en vitesse se traduit immanquablement par une perte en performances de détection.

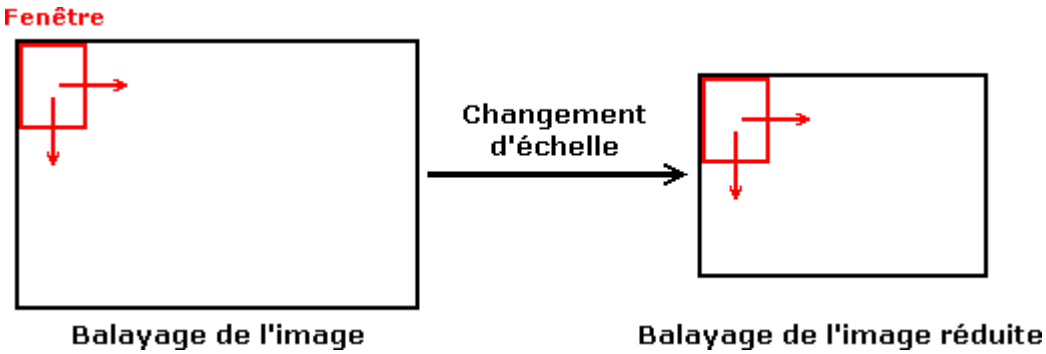


FIG. 2 – Procédé de balayage d'une image : 1) balayage à une échelle donnée, 2) réduction d'échelle, puis 1), etc

1.2 Masquage de données

Afin de cibler l'analyse sur l'image, un visage ayant une forme plutôt ovale verticale, les quatre coins de la fenêtre seront ignorés, ainsi qu'une bande de chaque côté, dans la mesure où ces données représenteront principalement un décor, des habits, ou des cheveux, données très variables et inutile pour de la détection de visages. Cette technique amène donc à définir un masque de taille 20x20 pixels pour cacher des données (Figure 3).

FIG. 3 – Fenêtre et Masque de données

1.3 Passage en niveaux de gris

Dans la recherche de traits de visages, la couleur n'apporte rien et ne peut que gêner la détection, c'est pourquoi l'analyse est effectuée en niveaux de gris. Afin d'obtenir des variations de contraste, l'image, si elle est en couleur, doit donc être passée en niveaux de gris.

1.4 Normalisation

Maintenant que les données en entrée sont définies, il faut les rendre comparables quelles que soient les images originales. Chaque image peut en

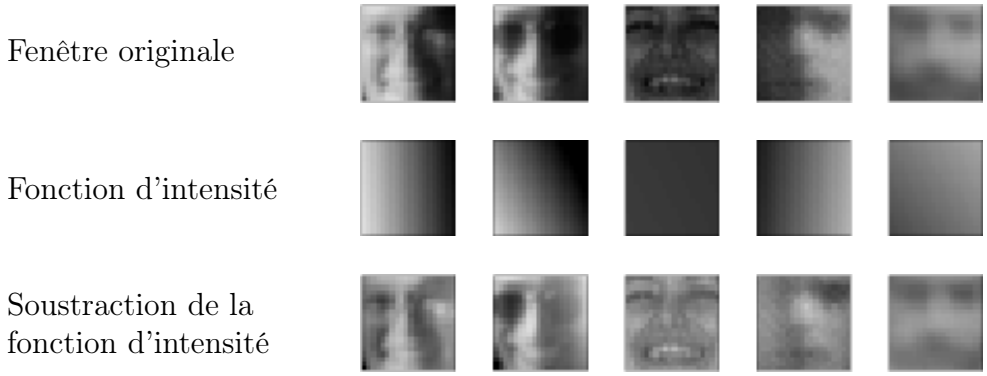


FIG. 4 – Calcul et application du filtre d'égalisation d'intensité

effet provenir d'un milieu plus ou moins lumineux ou avec un éclairage et des ombres différentes. Il est donc important, pour qu'un réseau puisse apprendre, que les données soient normalisées et que les variations de contraste soient représentatives de caractéristiques de visages et non de milieux ou d'expositions.

La normalisation des images va s'effectuer en deux temps :

1. Egalisation de la lumière dans l'image,
2. Egalisation par histogramme.

Egalisation de l'intensité lumineuse de fond

La première étape de ce filtre de normalisation est l'égalisation des intensités rencontrées dans la fenêtre de l'image. Nous nous intéresserons aux pixels non cachés par le masque, l'arrière plan étant hors sujet.

Le principe va être de définir une fonction qui approxime par région de la fenêtre l'intensité globale de cette région et ensuite de la soustraire à la fenêtre. Ceci aura pour effet de conserver les variations locales d'intensité mais de ramener la moyenne globale d'intensité par région à une constante. Ainsi, l'influence de l'exposition initiale de l'image est très atténuée. Cela permet, sur ce premier critère, une normalisation des images issues de milieux hétérogènes et tend donc à rendre possible une comparaison.

L'image 4 montre l'effet du filtre d'égalisation d'intensité. Cette image est tirée de l'article de référence [1].

Egalisation de l'histogramme

Bien que rendues plus homogènes, les images à traiter sont maintenant relativement fades. J'entend par là que les traits du visages ne ressortent que peu. Or, pour entraîner le réseau à reconnaître des traits, il faut être à même

de les mettre en valeur. C'est donc là l'objectif de l'égalisation des contrastes par histogramme.

Ce procédé a pour principe de rendre constante la fréquence des valeurs. Le mot "valeur" est ici à prendre au sens artistique, par opposition à couleur, qui définit en ce terme les nuances de gris.

Pour ce faire, il faut calculer l'histogramme des valeurs de l'image donnant pour chaque valeur sa fréquence, à savoir le nombre de pixels à cette valeur. Cet histogramme doit avoir en mémoire, pour chaque pixel, sa position dans l'image.

Il faut alors calculer la moyenne définie par :

$$m = \frac{Nombre\ total\ de\ pixels}{Nombre\ de\ valeurs}$$

où le nombre de valeurs est 256 lorsque l'on travail en 8 bits par pixels, chaque valeur appartenant à l'intervalle $[0, 255]$.

Cette moyenne calculée, elle nous donne le nombre de pixels par valeur qu'il faut avoir pour obtenir un histogramme plat. Ainsi, l'algorithme va itérativement, de 0 à 255, étaler sur les proches voisins le surplus potentiel de pixels, amenant ainsi à diminuer les fréquences supérieures à la moyenne et augmenter celles qui lui sont inférieures.

A terme l'histogramme obtenu doit être approximativement plat, donc les écarts à la moyenne pour chaque fréquence proches de 0. La figure 5 montre la transformation de l'histogramme.

Une fois l'histogramme égalisé, l'image est reconstruite avec la nouvelle répartition des valeurs. La figure 6 tirée de [1] montre l'image ainsi obtenue.

2 Fonctionnement du Réseau

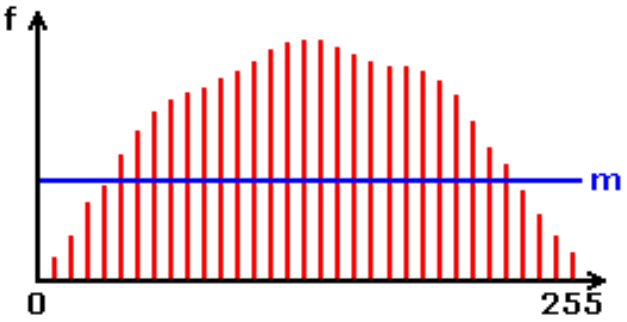
2.1 Topologie

Dans toute utilisation de réseaux de neurones, il faut définir une topologie de réseau. Il n'y a aucune règle pour définir cette topologie et c'est souvent par tests successifs qu'une bonne topologie est définie. Cependant, dans le cas présent, nous sommes à même de définir une topologie de base.

Un visage se distingue en effet surtout par des yeux, un nez et une bouche. Aussi, il semble intuitif de définir une topologie segmentant l'image afin de pouvoir repérer de telles caractéristiques. A cette fin, l'article [1] propose une telle topologie :

- 4 unités de 10x10 pixels,
- 16 unités de 5x5 pixels,

Histogramme original
avec sa moyenne



Histogramme égalisé

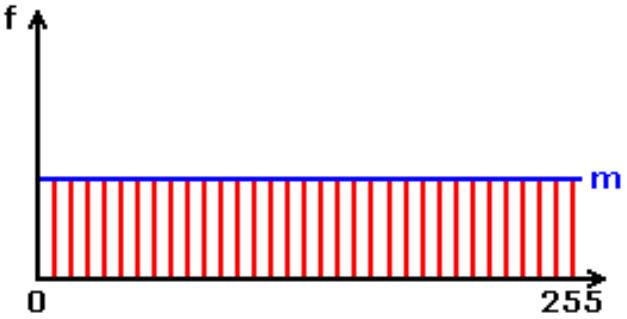


FIG. 5 – Principe de l'égalisation des valeurs dans l'histogramme

Fenêtres après
soustraction de la
fonction d'intensité

Puis égalisation par
histogramme

FIG. 6 – Application de l'égalisation des valeurs par histogramme

– 6 unités de 20x5 pixels.

Une unité sera une donc une couche d'entrée de données. Les unités carrées peuvent notamment repérer un oeil, un nez, ou un coin de bouche tandis que les unités en bande de 20x5 repère la ligne des yeux ou la bouche. Ainsi, les unités seront entraînées chacune à une tache précise et seront donc spécialisées. C'est en réunissant les réponses de ces unités que l'unité finale pourra dire si la fenêtre contient un visage ou non (Figure 7 tirée de [1]).

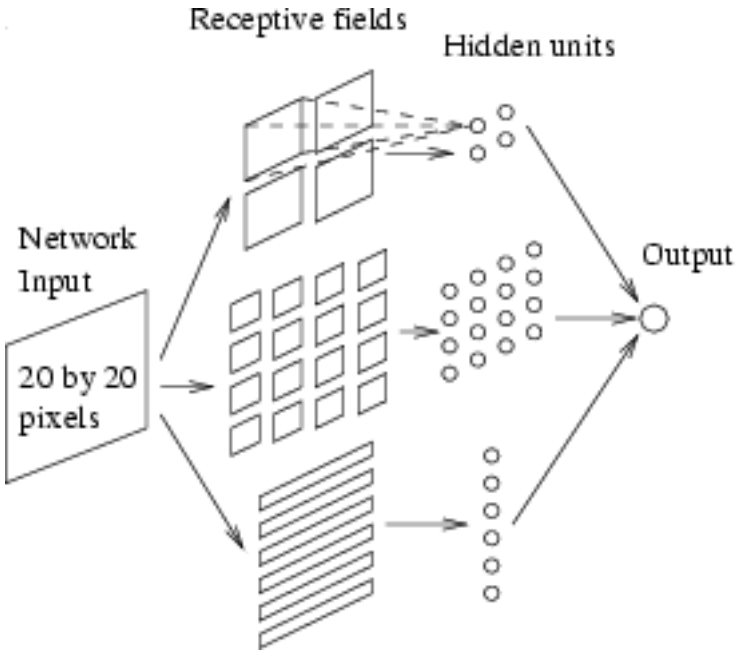


FIG. 7 – Réseau de détection

La topologie de base sera donc d'une unité finale fournissant une réponse binaire (dans l'article [1]) ou probabiliste. On mettra derrière cette unité les couches cachées du réseau, définies plus haut. J'appelle notamment cela une topologie de base car le nombre d'unités, leur taille et leur position restent non empiriques et ne peuvent en conséquence pas à être fermement fixés. L'article [1], adoptant la topologie citée plus haut, précise de même que le nombre de couches de chaque unité peut être augmenté et les auteurs utilisent pour leurs expériences des doubles et triples couches.

2.2 Mise en place

Les réseaux de neurones les plus répandus et les plus simples à la fois restent les perceptrons multi-couches (PMC) qui consistent en une succession

de couches, interconnectées totalement ou partiellement. Il est évident qu'un tel réseau brut ne peut convenir à la topologie citée plus haut. Cependant, une propriété intéressante d'un réseau de neurones est que tout sous réseau est un réseau. Ainsi, tout réseau est un méta réseau. De là, on peut très simplement définir un réseau respectant la topologie citée plus haut à partir de plusieurs PMC. L'algorithme d'apprentissage total reviendra à transmettre à tous les PMC, traitant chacun une unité, le résultat attendu. Si l'exemple à apprendre est un visage, la première étape transmet aux PMC des yeux, de la bouche et du nez, que l'on attend une réponse positive. De là, chaque PMC applique son algorithme d'apprentissage. La figure 8 illustre de manière simplifiée cette mise en place.

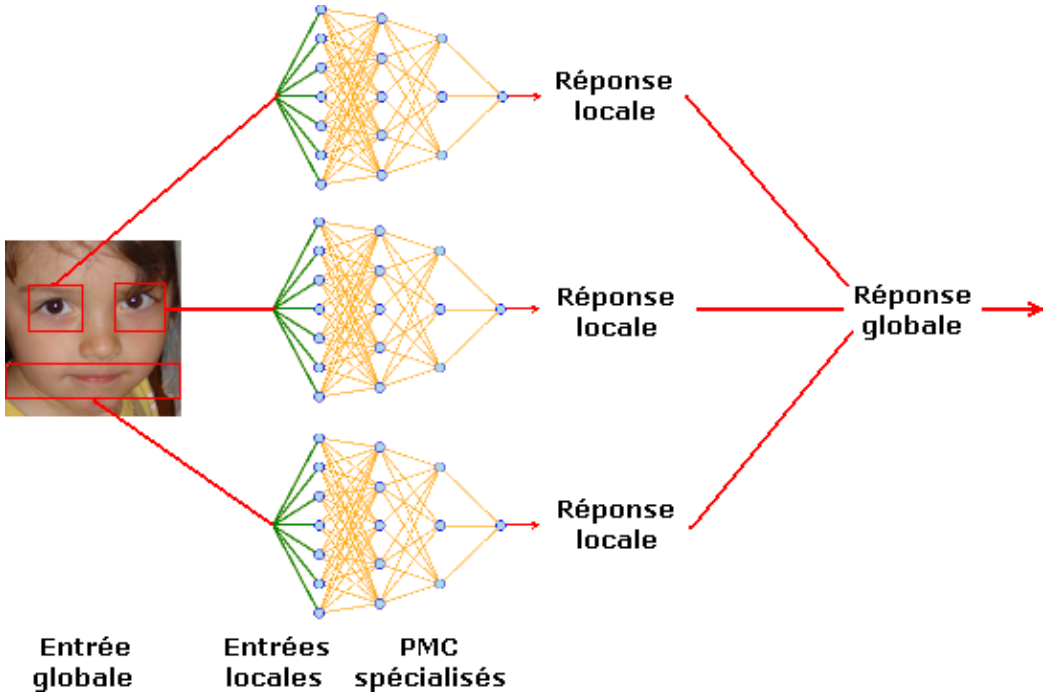


FIG. 8 – Mise en place d'un réseau de PMC

3 Apprentissage

L'apprentissage d'un réseau de neurones étant supervisé, il faut pouvoir définir des exemples d'apprentissage, ainsi que des contre-exemples.

3.1 Exemples

En premier lieu, il faut posséder une certaine base d'images de visages. A partir de ces images, il va être possible d e générer d'autres exemples, proches des originaux mais qui apporteront de la robustesse aux réseaux. En effet, si nous nous intéressons à la détection de visages verticaux, la verticalité et le centrage ne sont jamais parfaits. On peut donc créer à partir d'un visage exemple des images de ces visages tournées de quelques degrés de chaque côté, translatées de quelques pixels, ou zoomés quelque peu (Figure 9 tirée de [1]). On peut également faire une symétrie planaire, à savoir l'image du visage dans un miroir. Ainsi, on extrait de chaque exemple plusieurs exemples ayant de nouvelles caractéristiques malgré leur proximité au visage original.

FIG. 9 – Visages des auteurs [1] retournés, pivotés, zoomés légèrement

Cependant, il faut rester prudent sur ce type d'opération. En effet, autoriser une trop grande rotation ou translation aura pour effet de rendre la détection moins précise. Par exemple, en ne translatant que d'un pixel, le filtre tend à être insensible à un décalage de la fenêtre de un pixel dans l'image originale. Cette insensibilité est raisonnable mais on se retrouve une fois encore face à la dualité rapidité/qualité. Cette technique apportant plus d'exemples amène aussi à plus l'imprécision si on la surexploite.

3.2 Contre-exemples

S'il est relativement simple de réunir des exemples de visages, il n'en va pas de même pour les "non visages". En effet, un visage est quelque chose de définit. Or ce qui n'est pas un visage ne se définit pas. Tout ce qui n'en est

pas un, à savoir une infinité de choses, est un contre exemple potentiel. Dans l'idéal, un réseau doit pouvoir apprendre ce qui est et ce qui n'est pas l'objet à détecter. Pour approcher la diversité des contre-exemples de visages, il est cependant possible de proposer des solutions efficaces apportant des images suffisamment hétérogènes.

Tout d'abord, toute image, naturelle ou générée, ne contenant pas de visage, contient des contre-exemples. Afin que le réseau apprenne, on peut lui faire analyser ce type d'image et toute détection de visage sera une erreur, donc un contre-exemple sur lequel il devra apprendre. Ainsi, on peut définir de manière relativement automatisée une procédure d'extraction de contre-exemples correcteurs. Cette méthode est d'autant plus intéressante qu'il peut exister dans des images des configurations possédant des points communs avec des traits de visages. Des taches sombres environs aux emplacements des yeux, du nez et de la bouche peuvent induire un réseau en erreur. Ainsi, avec ce type de contre-exemple, il apprendra à être plus exigeant sur la précision dans ces zones. La figure 10, tirée de [1], illustre ce procédé.

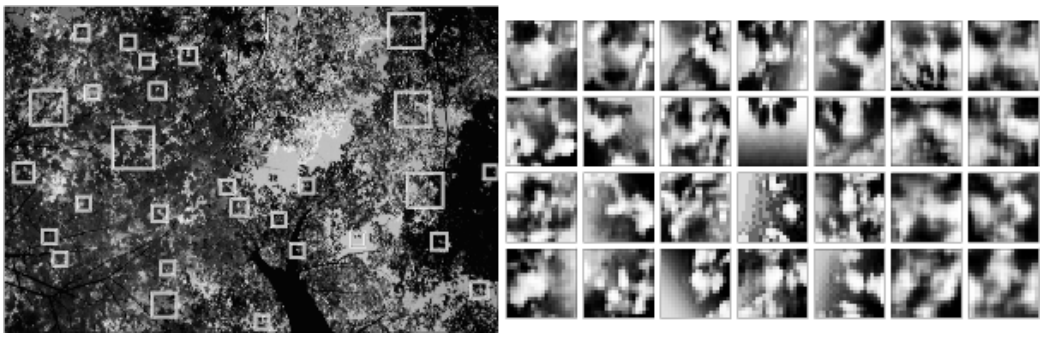


FIG. 10 – Détection dans une image ne contenant pas de visages et visualisation des fenêtres trouvées. Ces fenêtres ont pour certaines en effet des caractéristiques assez flagrantes de visage au niveau des variations de valeurs.

4 Sélection pertinente de solutions

L'imprécision et l'insensibilité à quelques pixels près amène très souvent de multiples détections d'un même visage dans un voisinage proche. Il faudrait pouvoir, parmi ces multiples propositions, en sélectionner ou en agréger une.

De même, considérant cette propriété, une détection isolée peut être considérée comme une erreur.

FIG. 11 – Détections multiples de visages et erreurs isolées

Cette observation amène les auteurs de [1] à proposer deux méthodes de sélection :

– l'agrégation de multiples détections,
– la sélection selon les avis de plusieurs réseaux de neurones.

4.1 Agrégation de multiples détections

La figure 11 extraite de [1] montre la détection multiples de visages ainsi que des détections erronées isolées.

Les auteurs proposent donc une heuristique éliminant une bonne partie des détections erronées. Il s'agit de compter, dans un voisinage restreint, le nombre de détections d'échelles proches. Si ce nombre est supérieur à un certain seuil définit au préalable, la zone est considérée comme un visage. Si par contre ce nombre est inférieur à ce seuil, alors la détection est considérée comme erronée. Afin de définir la détection conservée, le centre de chaque fenêtre est calculé. Tous ces centres vont prendre, dans une image appelée pyramide, la valeur du nombre de centre de détections qu'ils approchent dans un voisinage définit par un seuil. Cela va amener au calcul de centroïdes, centres unissant les points du voisinage approchant un minimum, définit par un seuil, de détections. Ces centres, jusqu'ici, peuvent définir des premières détections.

Maintenant, une deuxième heuristique est ensuite proposée, plus arbitraire à mon sens, dans laquelle il s'agit d'éliminer les détections qui débordent sur une autre détection validée, à savoir considérée comme correcte.

Appliquée au calcul des centroïdes, cette heuristique va donc traiter sé-

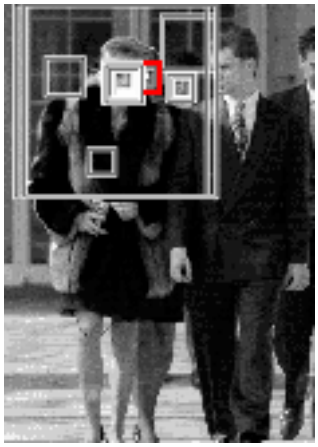


FIG. 12 – Elimination à tort (détection marquée en rouge) par l'heuristique éliminant les détections débordant sur des visages détectés

quentiellement les centroïdes en commençant par celui ayant le plus de détections à son actif. Ensuite, tout centroïde, représentant donc moins de détections, qui amène à une détection débordant sur une précédente, est supprimé. Ainsi, à la fin, il ne reste que des centroïdes ayant un certain poids de détection et ne concurrençant aucun autre centroïde.

Cette heuristique élimine dans la plupart des cas des détections fausses. Cependant, dans une recherche de visage par exemple dans un lieu où beaucoup de gens se croisent, les personnes situées aux premiers plans seront repérées là où ceux plus en retraits seront élagués. Ce type d'heuristique doit donc être utilisée avec prudence selon l'image analysée et l'objectif recherché pour cette détection. Si en effet, il s'agit de rechercher un visage dans une foule en combinant détection et reconnaissance de visages, cette heuristique présente des risques certains (Figure 12 créée depuis [1]).

4.2 Sélection selon plusieurs avis de réseaux

Une autre approche pour réduire le nombre d'erreurs consiste à entraîner plusieurs réseaux et ensuite à choisir en fonction de leurs réponses, celles qui doivent être conservées ou éliminées.

Afin de rendre efficace ce type de méthode, on considère que les poids initiaux des réseaux sont distribués aléatoirement. De même, la sélection et l'ordre des exemples pour l'apprentissage sont propres à chaque réseau. Ainsi on élimine la possibilité de duplication parallèle, même partielle, des réseaux.

Si l'on utilise un algorithme d'apprentissage de type rétro propagation

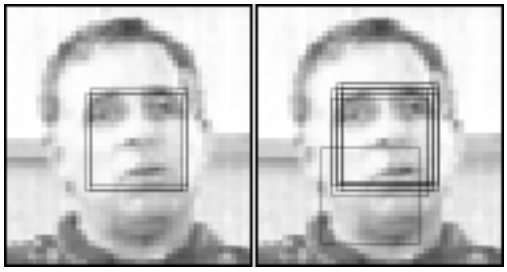


FIG. 13 – Détections de deux différents réseaux

FIG. 14 – Sélection effectuée par l'opérateur "ET"

du gradient, il n'y a pas de facteur aléatoire, il faut donc impérativement un maximum de différences initiales et d'apprentissage. Dans le cas d'un algorithme génétique, une base commune aura moins de conséquences car les croisements et mutations seront basés sur des facteurs aléatoires.

Quelle que soit la méthode employée, on peut donc arriver à des réseaux faisant des erreurs différentes, et si l'apprentissage est suffisant, leur convergence d'avis donnera en général les détections justes (Figure 13 et 14 créées depuis [1]). Les détection sélectionnées seront celles qui coïncident précisément en terme de position et d'échelle, l'équivalent d'un "ET" logique. Ainsi, il est peu probable que deux réseaux s'accordent sur une erreur, compte tenu de leurs différences initiales et d'apprentissage. Cependant, si un réseau ne détecte pas bien un visage, même si l'autre l'a bien fait, cette détection ne sera pas sélectionnée. Toutefois, ces cas sont très rares.

Cette technique peut être appliquée avec n réseaux et la sélection des visages peut aussi se faire par ou "OU" logique. Bien sur, davantage de réseaux peut entraîner plus de désaccords. De même un "OU" n'élimine pas les fautes mais les réuni. Cependant, couplé avec la technique d'agrégation, les multiples détections étant de fait plus nombreuses encore (somme de plusieurs réseaux), cela permet d'être plus exigeant sur le seuil minimal des détections multiples

et donc d'éliminer avec moins d'erreurs les mauvaises détections. Il y a donc plusieurs approches à utiliser et à combiner, chacune possédant des propriétés intéressantes.

5 Extension à des visages non verticaux

Jusqu'ici, les visages détectés sont supposés être droits. La technique employée ne permet pas de détecter un visage écarté de plus de quelques degrés de l'axe vertical. Même si cette configuration verticale est la plus courante, il n'est pas envisageable dans le cas réel d'en faire une hypothèse aussi forte.

Une première idée pourrait être de conserver le système en place, et, de même que le processus est répété sur l'image réduite, il pourrait être répété sur l'image pivotée. Cependant, en admettant par exemple que notre détection soit insensible à 10ř degrés près, ce type de technique demanderait au moins 18 répétitions du processus (donc une rotation de 20ř à chaque fois), déjà lourd en soi. Cette technique, trop lourde en calcul, n'est donc pas envisageable.

L'article [2] apporte à ce problème une solution bien plus rapide. Dans la même idée que la chaîne de traitement vu précédemment, il va s'agir d'introduire une étape pré détection supplémentaire.

5.1 Principe

L'idée proposée va être d'utiliser un nouveau réseau de neurones, déterminant l'angle d'un visage par rapport à l'axe vertical. Ce réseau suppose que l'image présentée est un visage, il n'y a pas à ce niveau d'attente sur la détection. En effet, il sera entraîné à reconnaître l'angle d'un visage par rapport à l'axe vertical. Si la fenêtre n'en contient pas, il renverra une valeur angulaire sans importance puisque la détection ne retiendra pas l'image.

La puissance de cette méthode va être que pour toute fenêtre, l'étape de rotation n'est appliquée qu'une seule fois.

5.2 Fonctionnement

Les étapes de mise en oeuvre de la rotation d'image ont une structure assez similaire à ce qui a été vu auparavant. Chaque fenêtre passe à travers l'égalisation par histogramme. Ensuite l'image résultante est présentée à un réseau de neurones qui renvoie le degré d'écart à l'axe vertical. Ensuite, la fenêtre initiale est pivotée, puis la chaîne précédente s'applique, à savoir égalisations pré détection puis détection.

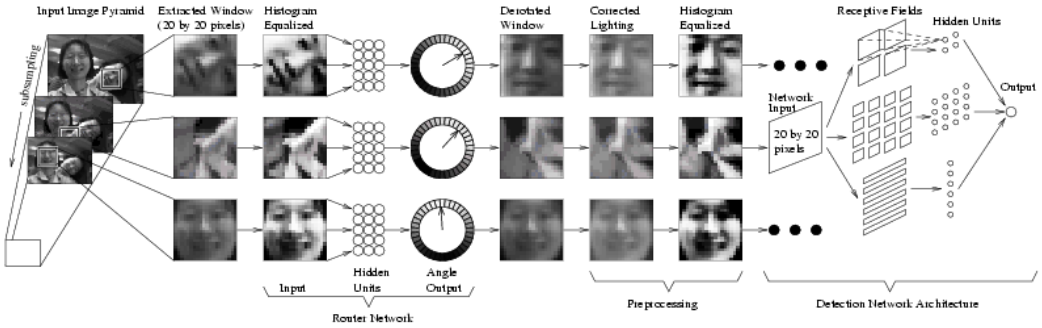


FIG. 15 – chaîne de traitement avec rotation de l'image

La figure 15 extraite de [2] illuste la chaîne de traitement ainsi obtenue.

5.3 Apprentissage et Topologie en sortie

L'apprentissage du réseau de rotation va être différent de l'apprentissage utilisé pour la détection. En effet, le réseau ne doit pas renvoyer une valeur booléenne mais permettre d'obtenir un angle de rotation. Il n'y aura donc dans cet apprentissage aucun contre-exemple. La sortie du réseau ne va pas être une valeur mais un vecteur de 36 valeurs. Lors de l'apprentissage, ces valeurs, supervisées, seront :

$$cos(\theta - i \times 10)$$

avec

- θ l'angle connu du visage sur l'exemple d'apprentissage,
- i une valeur comprise entre 0 et 35.

De même que pour la phase de détection, les images d'apprentissage sont légèrement tournés, translatés et réduites et produisent d'une part une batterie d'exemple et d'autre part par voie de conséquence une insensibilité à de légères rotations, translations ou réductions.

5.4 Résolution de l'angle de rotation

Maintenant que le réseau est entraîné, il faut pouvoir déduire des 36 valeurs de sortie un angle de rotation à appliquer à la fenêtre.

Simplement, on pourrait vouloir chercher le cosinus le plus proche de 1, à savoir celui pour lequel $\theta - i \times 10$ est le plus proche de 0. On aurait alors la valeur de i fournissant à 5ř près la meilleure approximation de l'angle.

Cependant, la fonction cosinus ne marcherait que pour des angles compris entre 0ř et 180ř. Aussi, une autre approche est proposée, dans laquelle la valeur de sortie pour tout i va être vue comme un poids attribué à i.

L'idée est de sommer, sur tous les i le produit de leur poids multiplié par le cosinus ou le sinus de l'angle $i \times 10$ selon l'axe de coordonnées. On obtient le vecteur définit par les valeurs :

$$\left(\sum_{i=0}^{35} output_i \times cos(i \times 10), \sum_{i=0}^{35} output_i \times sin(i \times 10)\right)$$

où $output_i$ est la i^e sortie du réseau.

Les coordonnées horizontale et verticale de ce vecteur vont ainsi définir l'angle du visage.

On peut sentir intuitivement pourquoi ce vecteur approche l'angle voulu. La bonne valeur de i donne théoriquement un poids proche de 1. Donc le produit de ce poids par $cos(i \times 10)$ et $sin(i \times 10)$ donne une première bonne approximation. Cependant, les autres valeurs contiennent elles aussi de l'information. En effet, plus $i \times 10$ est loin de θ, plus $cos(\theta - i \times 10)$ diminue. Donc en sommant également ces valeurs, on utilise l'information comme quoi ces valeurs possèdent une information, mais étant moins précise, leur poids est moindre. On peut voir ce calcul comme une synthèse spectrale avec pondération. Cela évite de mettre toute la confiance en une seule valeur.

5.5 Topologie générale

Dans cette partie, il n'est plus possible de définir une topologie aussi précise que lors de la détection. Le réseau n'a à priori aucune information sur la disposition des formes à rechercher comme les yeux, le nez ou la bouche pour la détection. Ainsi, dans ce genre de cas, le perceptron multi-couches à connection totale entre les couches est souvent employé. Les auteurs de [2] proposent une première couche en entrée de 400 unités, soient autant de pixels que la fenêtre donne en entrée, puis une couche cachée de 15 unités, et enfin la couche de sortie de 36 unités (Figure 16). Ce type de topologie relève plus de l'expérience de tests réalisés que d'une étude formelle. Plus le réseau est gros, plus il est coûteux en calculs, mais souvent plus il est précis. Il faut donc trouver, et souvent par le test, un juste milieu. Il en va de même pour les fonctions d'activations. Ici, ils ont choisi une tangente hyperbolique. L'algorithme d'apprentissage est quant à lui très souvent le célèbre algorithme de rétro propagation du gradient comme c'est le cas dans leur méthode. Cependant, les algorithmes de type génétique, d'une approche totalement différente, non formelle, donnent souvent des résultats comparables voire meilleurs selon les cas.

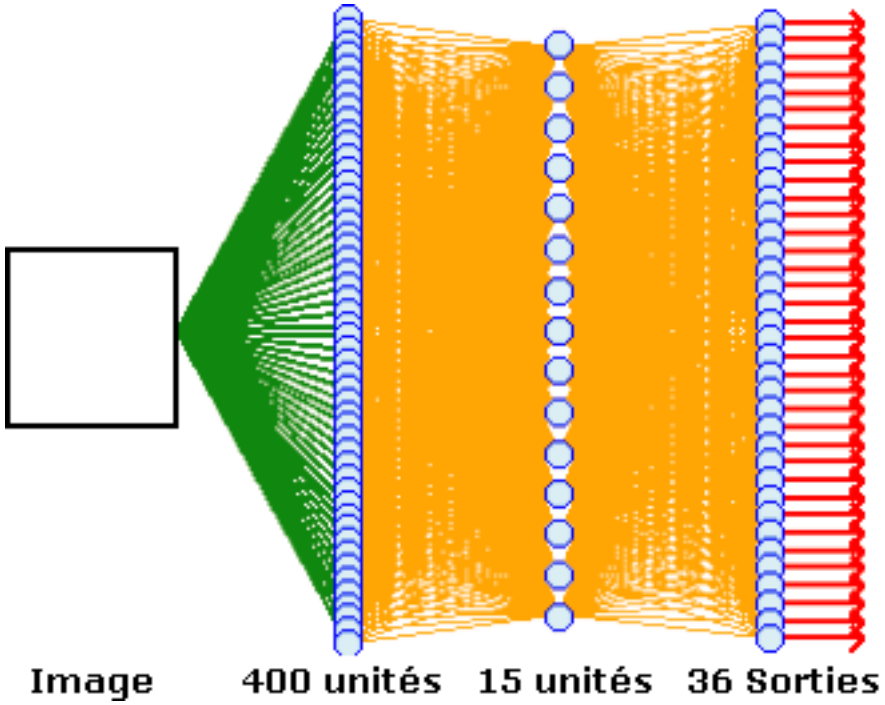


FIG. 16 – Réseau de rotation

Conclusion

Cette application des réseaux de neurones montre la simplicité avec laquelle, exemples en main, cet outil est utilisable pour obtenir rapidement un premier résultat. Cependant, ce sont la précision de la détection et le taux d'erreurs qui sont sujets à la plupart des travaux. L'efficacité et la robustesse d'un système se jugeront surtout sur les heuristiques et les méthodes employées pour l'optimiser. Ces méthodes doivent d'une part minimiser le taux de mauvaises détections, et d'autre part maximiser le score de bonnes détections. Ces deux scores ne sont pas totalement complémentaires, nous avons en effet discuté des dangers de l'heuristique éliminant toute détection en chevauchant une autre plus probable. Ainsi, l'erreur se définit à la fois en fonction du nombre de mauvaises détections, mais aussi du nombre de détections omises. C'est ce balancement entre ces deux facteurs qui rend difficile l'optimisation. Les méthodes sont en général efficaces pour l'un mais présentent des effets de bords pour l'autre. C'est ainsi que traiter l'ensemble des cas peut complexifier exponentiellement le traitement pour une amélioration parfois logarithmique.

De plus, si l'on considère tous les angles, en trois dimensions, sous lesquels

peuvent être présenté un visage, la reconnaissance devient beaucoup moins triviale. Pour traiter la reconnaissance de visages tournés en dehors du plan de l'image, comme des profils, plusieurs méthodes peuvent être envisagées. L'utilisation de la symétrie et de la forme du visage peut amener à déduire depuis un visage vu de côté sa forme frontale. Une autre méthode, analogue à celles vues ici, serait d'entraîner séparément des réseaux à reconnaître l'angle de présentation, comme profil gauche, semi profil gauche, face, semi profil droit, profil droit. Cependant, ces angles, bien que les plus courants, ne sont pas les seuls.

Références

[1] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Neural Network-Based Face Detection", Janvier 1998 School of Computer Science, Carnegi Mellon University, Pittsburg Justsystem Pittsburg Research Center, Pittsburg

[2] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Rotation Invariant Neural Network-Based Face Detection", School of Computer Science, Carnegi Mellon University, Pittsburg Justsystem Pittsburg Research Center, Pittsburg

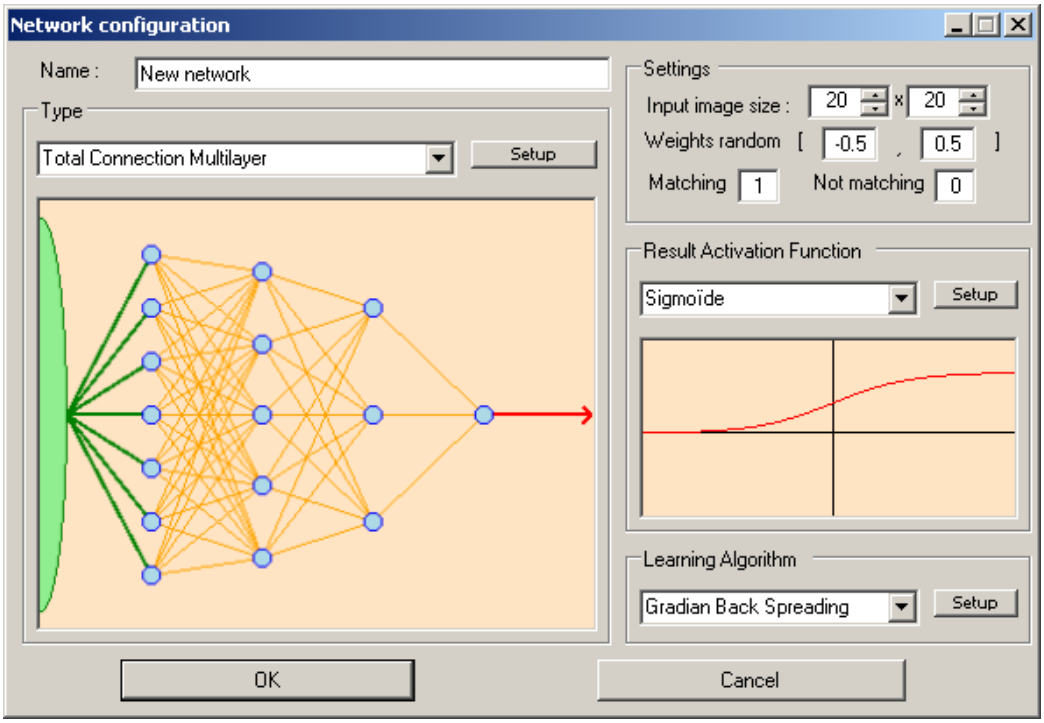


FIG. 17 – Interface graphique de la création d'un réseau de neurones

Implémentation

Une librairie personnelle de réseaux de neurones implémentant le Perceptron multi-couches avec :

- Apprentissage par rétro propagation du gradient,
- Apprentissage génétique,

et avec les fonctions d'activations

- Seuil (Heaviside),
- Sigmoïde exponentielle,
- Gaussienne,
- Tangente hyperbolique,

est disponible sous licence GPL à l'adresse :

http ://rone56.free.fr/Prog/NeuralNetworks.html

Elle peut être employée pour la mise en place d'un réseau de détection de visage selon la méthode exposée plus haut.