Reconnaissance de signes de la main à l'aide de réseaux de neurones

Gabrielle DIDELOT et Pauline MARTIN

Abstract

Dans le cadre du projet, nous avons choisi de tester et comparer différentes approches pour détecter automatiquement des signes de la main, en utilisant des outils de vision par ordinateur et d'apprentissage profond par réseau de neurones. Pour ce faire, nous avons travaillé sur une base de 20 000 images labellisées représentant 10 signes de la main différents. Nous avons testé deux méthodes de traitement d'images pour extraire la main de l'arrière-plan : le seuillage et la détection de contours, et nous avons testé deux types de réseaux de neurones pour apprendre sur ces images pré-traitées : un CNN et un MLP. Nous avons ensuite comparé les résultats obtenus en combinant ces différentes méthodes.
L'approche la plus performante que nous retenons à l'issue du projet est d'appliquer une fonction de seuillage aux images puis d'entraîner un réseau MLP sur les données prétraitées.

Introduction

La reconnaissance automatique de gestes, et plus particulièrement celle des gestes de la main, occupe actuellement une place importante dans le domaine de la vision par ordinateur, de par ses nombreuses applications concrètes. Elle permet en effet d'assurer une interaction purement visuelle entre un humain et un ordinateur et se révèle donc particulièrement utile dans le développement d'outils de communication entre humains et robots. Elle est également essentielle à une forme de traduction toute particulière, à savoir la traduction de la langue des signes. L'idée d'une retranscription en live d'une vidéo représentant une personne parlant la langue des signes ne semble alors plus si farfelue. La langue des signes pourrait bientôt être un choix possible sur Google Traduction…
Nous avons donc choisi pour ce projet de travailler sur la reconnaissance de certains signes de la main à partir d'images.

Etat de l'art

La reconnaissance de gestes de la main est un domaine largement traité et de nombreuses études sur le sujet ont déjà été réalisées à ce sujet. Les méthodes et techniques utilisées diffèrent toutefois beaucoup d'une étude à l'autre. Il faut tout d'abord distinguer les travaux se concentrant sur des images, et donc des poses statiques, de ceux traitant des vidéos et donc des gestes dynamiques. Ainsi, Md. Hafizur Rahman et Jinia Afrin [1] se sont focalisés

sur la classification de signes statiques représentant certaine lettres de l'alphabet du langage des signes. Pour cela, ils utilisent un classifieur SVM (machine à vecteurs de support) après un premier filtrage des images. C'est également l'outil qu'utilise Aseema Sultana [2] dans son étude, mais cette fois pour analyser des mouvements et gestes dynamiques. Yiqiang CHEN, Wen GAO et Jiyong MA [3] ont, quant à eux, choisi d'utiliser des arbres de décision afin de classifier plus de 60 gestes différents. D'autres, comme les français Sebastien Marcel, Olivier Bernier, Jean–Emmanuel Viallet et Daniel Collobert [4] se sont appuyés sur des modèles de Markov pour modéliser les gestes de la main. Ainsi, on constate que les approches du problème de classification des signes de la main sont nombreuses. Cependant, on observe depuis quelques années une forte recrudescence de travaux basés sur des réseaux de neurones dans le domaine de la vision par ordinateur, et notamment pour notre problème. Dès 2000, Klimis Symeonidis [5] a cherché à classifier des photos de mains en entraînant un perceptron. Plus récemment, Tasnuva Ahmed [6] a construit un MLP (perceptron multi-couches), tandis que Oyebade K. Oyedotun et Adnan Khashman [7] ont créé un réseau neuronal convolutif pour leur étude.

Approche

Dans le cadre du projet, nous avons décidé de nous concentrer dans un premier temps sur la classification de signes statiques de la main, c'est-à-dire à partir de photos et non de vidéos.

Nous avons choisi de travailler sur des données issues du dataset Kaggle "Hand Gesture Recognition Database" [8]. C'est un ensemble composé de 20 000 photos de mains, représentant chacune un signe clairement identifié parmi dix signes différents. Elles ont été prises par dix personnes différentes, chacun ayant réalisé 200 photos de chaque signe. Ce sont des photos en noir et blanc, prises à l'aide d'une caméra thermique. Elles sont labellisées et peuvent donc être utilisées pour entraîner un algorithme d'apprentissage supervisé.

La taille du dataset est relativement restreinte pour entraîner un réseau de neurones, mais il peut être enrichi avec de nouvelles photos, facilement labellisables.

Notre étude s'organise en deux parties : une première partie orientée vision par ordinateur, doit permettre de détecter la forme de la main dans l'image, puis une deuxième partie axée deep learning, qui doit permettre la prédiction du signe correspondant à la forme de la main extraite précédemment.

Computer Vision

Le premier objectif est de détecter la main, et par conséquent de pouvoir la faire ressortir par rapport à l'arrière-plan. Pour ce faire, les images sont pré-traitées à l'aide d'outils de Computer Vision afin d'en faire ressortir les caractéristiques qui seront utiles à

l'apprentissage du réseau de neurones a posteriori. Nous avons testé deux approches pour extraire les caractéristiques utiles, expliquées ci-dessous : le seuillage et la détection de contours selon la méthode de Canny. Les images sont ainsi soumises à l'une des chaînes de traitements suivantes :

A. Méthode par seuillage

1. Réduction du bruit par filtre médian
2. Extraction de la main par seuillage
3. Recadrage de l'image au format 240*240, centré sur la main

B. Méthode par détecteur de Canny

1. Réduction du bruit par filtre médian
2. Extraction de la main par un détecteur de contours de Canny
3. Recadrage de l'image au format 240*240, centré sur la main

1. Réduction du bruit par filtre médian

Pour détecter au mieux la main dans l'image, il est nécessaire de réduire le bruit, car certains éléments de l'arrière-plan peuvent perturber le seuillage ou la détection de contours. Cependant, les filtres linéaires classiques (moyenneur ou gaussien) lissent le bruit et induisent un effet de flou, notamment au niveau des contours. Nous avons donc choisi d'appliquer un filtre médian : chaque pixel est remplacé par la valeur médiane de son voisinage. Ceci permet de réduire l'influence des valeurs extrêmes dans un voisinage. Ainsi le bruit impulsionnel est réduit, sans effet de lissage au niveau des contours.

2.A Extraction de la main par seuillage

Comme les images ont été capturées par une caméra thermique et qu'elles sont en noir et blanc, la main se distingue relativement bien de l'arrière plan car elle est plus claire. Une première approche pour détecter la main dans l'image est donc d'appliquer une fonction de seuillage : tous les pixels d'intensité inférieure au seuil s prennent la valeur 0 (pixel noir), et tous les autres prennent la valeur 255 (pixel blanc). En choisissant un seuil s pertinent, on peut ainsi séparer les pixels décrivant la main des pixels de l'arrière plan. Cette représentation simplifiée de l'image condense l'information utile pour l'apprentissage du réseau de neurones pas la suite.

2.B Extraction de la main par un détecteur de contours de Canny

Une deuxième approche, très classique pour l'extraction de caractéristiques dans une image, est la détection de contours. En traitement d'image, un contour est défini comme une discontinuité de la fonction d'intensité. Pour localiser ces contours, on étudie donc le gradient de l'intensité. Le détecteur de Canny, implémenté dans OpenCV, fonctionne de la façon suivante:

- il applique son propre traitement de réduction du bruit: un filtre gaussien de dimension 5*5
- il approxime, par une approche discrète, le gradient de l'intensité dans les 2 directions verticale et horizontale. En combinant l'analyse dans les 2 directions, il obtient ainsi le module et la direction du gradient de l'intensité en tout point de l'image
- en comparant chaque pixel aux pixels voisins dans la direction locale du gradient, il supprime tous ceux qui ne correspondent pas à des maxima locaux du gradient

- pour déterminer les véritables contours à partir de tous les maxima locaux du gradient, il applique un seuillage par hysteresis : tous les maxima locaux dont la valeur est supérieure à Vmax sont des contours, et tous ceux dont la valeur est inférieure à Vmin ne sont pas des contours. Les points entre les deux seuils sont des contours seulement s'ils sont reliés à des contours.

En sortie, on obtient une image où les pixels de contours ont la valeur 255 (pixel blanc), et tous les autres ont la valeur 0 (pixel noir). De même que précédemment, cette représentation simplifiée de l'image condense l'information utile pour l'apprentissage du réseau de neurones pas la suite.

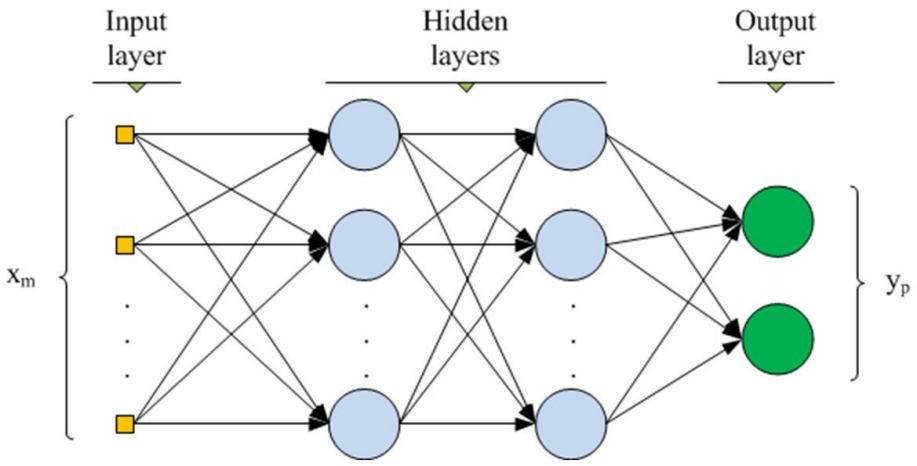
3. Recadrage de l'image au format 240*240, centré sur la main

Pour servir ensuite de base d'apprentissage à un réseau de neurones, les images doivent toutes être au même format. De plus, afin de réduire le nombre de calculs et ainsi améliorer la rapidité d'apprentissage du réseau, on cherche à réduire au maximum la taille des images tout en gardant l'information utile. On choisit donc de couper les images, de taille initiale 240*640, en un format carré 240*240, centré sur la main. L'image est ainsi gardée intacte dans sa hauteur, et est tronquée sur ses bords gauche et droit. Pour tronquer en centrant sur la main, on détecte les points de contour (ou de la partie seuillée) le plus à gauche et le plus à droite de l'image.

Deep Learning

Une fois la forme de la main extraite, il nous faut construire un algorithme d'apprentissage qui soit capable de classifier au mieux les différentes images. Pour cela, nous nous sommes appuyés sur des techniques de deep learning et avons donc utilisé des réseaux de neurones. Afin de pouvoir se faire une première idée de l'efficacité des différents types de réseaux de neurones dans le cadre de notre problème, nous avons décidé de tester deux structures très différentes et de comparer leurs performances : un perceptron multi-couches (MLP) et un réseau convolutif (CNN).

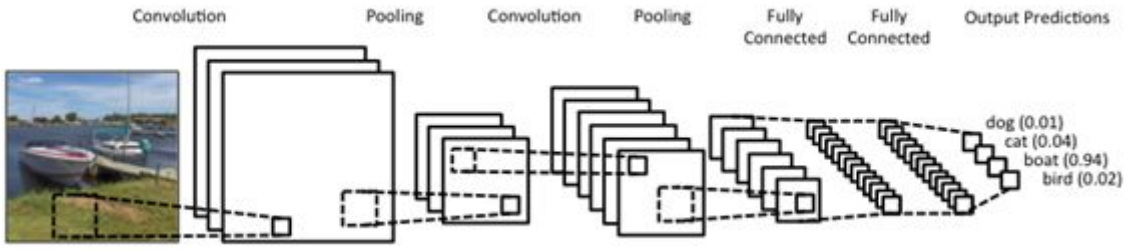
Le perceptron multi-couches (MLP) est une structure classique de réseau de neurones. Les données d'entrée sont fournies sous la forme d'un vecteur et sont ensuite retraitées dans différentes couches cachées.



Source : HackerEarth

Le MLP est un type de réseau très flexible et donc pertinent dans de nombreux cas, y compris certaines analyses d'images. Cependant, la forme imposée de l'entrée (un vecteur) empêche de conserver toute information sur la structure spatiale d'une image. Les pixels sont considérés indépendamment les uns des autres.

A l'inverse, une des caractéristiques principales d'un réseau convolutif (CNN) est qu'il permet de conserver ces informations d'espace et de "pixels voisins". Il n'est pas affecté par des modifications de type translation ou distorsion. C'est donc le type de réseau tout désigné pour la classification d'images.



Source : NCIA Lab

Un CNN est composé de couches de convolution, à travers lesquelles les pixels sont retraités, en fonction notamment de leurs voisins, suivies de couches de pooling qui permettent de consolider les caractéristiques apprises grâce aux couches de convolution précédentes. Ces couches, de par leur caractère généralisant, permettent de réduire le surapprentissage.

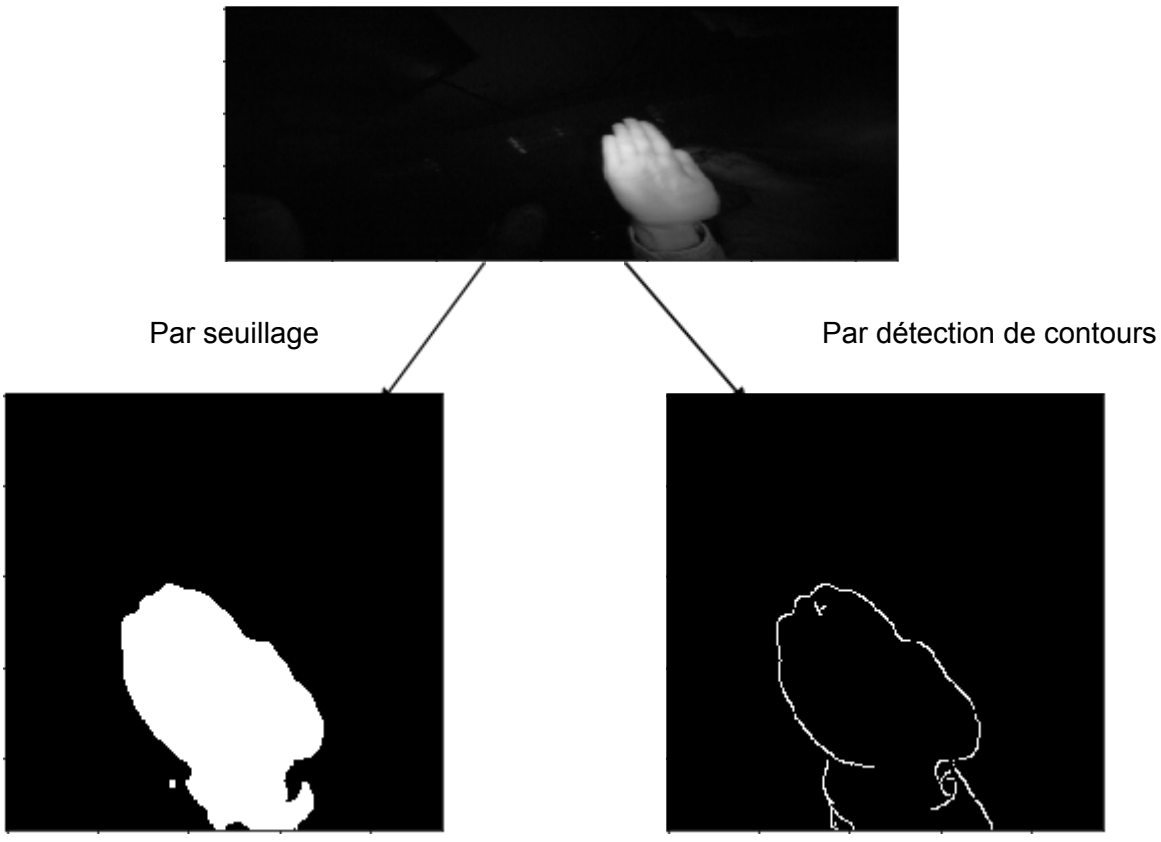
Ainsi, nous avons appliqué ces deux types de réseaux de neurones aux données précédemment pré-processées, afin de pouvoir évaluer leur performance et par conséquent leur pertinence pour le problème à traiter.

Expériences et résultats

Prétraitements de Computer Vision :

En testant sur une sélection d'images, nous avons déterminé empiriquement les paramètres qui permettent de bien détecter la forme de la main. Ainsi, nous appliquons en premier lieu un filtre médian de taille 7*7.
Lors de l'extraction de la main par seuillage, nous appliquons un seuil s=50 pour discriminer les pixels de la main et ceux de l'arrière plan.
Lors de l'extraction de la main par détection de contours, nous appliquons un détecteur de Canny de paramètre Vmin = 20 et Vmax = 150.

Les traitements appliqués aux images sont visualisés ci-dessous:



Après mélange (à l'aide d'une permutation) et normalisation des images, elles peuvent être fournies en entrée au modèle (entraînement puis test).

Perceptron multi-couches (MLP):

Nous avons créé un réseau composé de deux couches cachées, chacune d'une largeur de 64 neurones, dont la fonction d'activation est la fonction non-linéaire ReLU (Rectified Linear Unit). La couche de sortie est une couche 'softmax' de largeur 10, correspondant aux probabilités que l'image d'entrée représente chacun des 10 signes possibles. Pour des raisons de temps de calcul, le nombre d'epochs a été limité à 3.

Réseau convolutif (CNN) :

Nous avons construit un CNN composé de 3 couches de convolution, chacune suivie d'une couche de max pooling.

La première couche de convolution est moins large que les deux autres (32 filtres, puis 64) car il est d'usage d'utiliser relativement peu de filtres pour la couche d'entrée et d'en augmenter le nombre par la suite. La taille du kernel utilisé diffère également pour la première couche de convolution : 5x5 d'abord puis 3x3. En effet, l'image passée en entrée est relativement grande (240x240 pixels) et il est donc cohérent d'analyser ensemble des blocs relativement grands. Ce kernel doit ensuite être réduit à mesure que la taille de l'image décroît. Le stride utilisé est fixé à 1 et le padding à 'valid' ; ce sont les valeurs par défaut. Les couches de max pooling sont quant à elles toutes identiques : elles ne conservent que la valeur maximale parmi chaque groupe de 4 pixels (2x2). Ainsi, la taille de l'information traversant ces couches est à chaque fois réduite d'un facteur 4 (2 en hauteur, 2 en largeur).

En sortie de réseau, on retrouve des couches *'fully connected*', et notamment la même couche 'softmax' de largeur 10 qu'en sortie du MLP.

Le même optimiseur a été choisi pour les deux types de réseaux, à savoir la descente de gradient stochastique (SGD). C'est un optimiseur très répandu qui permet de trouver simplement les paramètres optimaux d'un réseau. De même, nous avons entraîné tous les modèles avec pour métrique l'accuracy. C'est en effet la métrique qui nous semble la plus pertinente pour notre problème de classification multi-classes.

Résultats:

Les scores sont calculés sur des données de test, grâce aux performances du réseau préalablement entraîné sur les données d'entraînement. L'ensemble de test représente 20% de l'ensemble total.

Les résultats sont répertoriés ci-dessous :

	MLP	**CNN**
Seuillage	**0.989**	0.974
Détection de contours	0.891	0.961

Conclusion

A l'issue de cette étude, nous déduisons que la méthode la plus performante pour qu'une machine détecte des signes de la main est d'appliquer une fonction de seuillage aux images, puis d'entraîner un réseau de neurones MLP à partir de ces images prétraitées. Cette méthode nous a permis d'atteindre une accuracy de 0.989 sur nos données de test, ce qui est particulièrement performant.
Cependant, pour ce travail nous disposions d'une base de 20 000 images, ce qui est relativement peu pour entraîner un réseau de neurones profond. Pour affiner le modèle et le rendre d'autant plus robuste aux traitements de données inconnues, nous pourrions donc l'enrichir de nouvelles données.

Références

[1] Md. Hafizur Rahman and Jinia Afrin, "Hand Gesture Recognition using Multiclass Support Vector Machine". International Journal of Computer Applications (0975 – 8887) Volume 74– No.1, July 2013.

[2] Aseema Sultana and T Rajapuspha 2, "Vision Based Gesture Recognition for Alphabetical Hand Gestures Using the SVM Classifier". International Journal of Computer Science & Engineering Technology (IJCSET). Vol. 3 No. 7 July 2012.

[3] Yiqiang CHEN, Wen GAO, Jiyong MA, "Hand Gesture Recognition Based on Decision Tree," Institute of Computing Technology, Chinese Academy of Sciences, Beijing.

[4] Sebastien Marcel, Olivier Bernier, Jean–Emmanuel Viallet and Daniel Collobert, "Hand Gesture Recognition using Input–Output Hidden Markov Models," France Telecom CNET.

[5] Klimis Symeonidis, "Hand Gesture Recognition Using Neural Networks," Degree of Master of Science in Multimedia Signal Processing communications, School of Electronic and Electrical Engineering, On August 23, 2000.

[6] Tasnuva Ahmed, "A Neural Network based Real Time Hand Gesture Recognition System".International Journal of Computer Applications (0975 – 8887) Volume 59– No.4, December 2012.

[7] Oyedotun, O.K. & Khashman, A. Neural Comput & Applic (2017) 28: 3941. https://doi.org/10.1007/s00521-016-2294-8

[8] Kaggle, Hand Gesture Recognition Database. Updated in September 2018. https://www.kaggle.com/gti-upm/leapgestrecog