Reconnaissance d'un langage de signes grâce au réseau Alexnet

Quentin Vieillard and *Jerry* Mba Wendji

[1] *École CentraleSupélec - Option ISIA*

Abstract. L'avancée du deep learning permet aujourd'hui la réalisation de nombreuses applications allant de la prédiction à la classification. Il peut s'agir par exemple de prévision météorologique, ou encore de classification d'images. Cette dernière application est l'objet de cet article qui se propose d'apprendre la reconnaissance de signes de main sur la base d'images associées chacune à une lettre de l'alphabet. Pour y parvenir, nous avions à notre disposition un dataset d'images recueilli sur internet. Nous avons alors procédé à un prétraitement de ces images pour en extraire les informations pertinentes avant de les fournir au réseau Alexnet pour un entraînement par "finetuning".

Introduction

Cet article s'intéresse aux travaux[1] réalisés par Nicolas Pugeaut et Richard Bowsen à l'université de Surrey[1]. Ces derniers ont proposé une interface utilisateur capable de reconnaître de manière interactive le langage des signes américain (ASL - *American Sign Language*). Cette interface est un outil précurseur dans le domaine des interactions homme/machine sans clavier ainsi que dans l'apprentissage du langage des signes.

La reconnaissance du langage des signes reste difficile à cause de la multiplicité des signes à reconnaître : mouvements et/ou forme de la main, position du corps et expression du visage. Afin de se soustraire à cette complexité, Pugeaut et Bowsen se sont intéressés uniquement aux images présentant les différentes formes de la main représentant l'alphabet ASL. Toutefois, 2 niveaux de complexité demeurent :

- similarité de certains signes (ex : les lettres *a, e, m, n, s et t*)
- mutliplicité des representations de signes en fonction des individus (position, orientation...)

Pugeaut et Bowsen ont constitué un dataset d'images qui prend en compte ces différents critères. Ils ont produit deux catégories d'images : l'une portant l'information sur l'intensité lumineuse (une photo classique) et l'autre, portant l'information sur la profondeur (cette dernière a été recueillie grâce à la Kinect de Microsoft). Ils se sont ensuite servis de cette double information comme attribut de classification pour entraîner leur outil de reconnaissance de signes.

Dans cet article, nous avons souhaité étendre la reconnaissance de signes aux appareils ne possédant pas l'information de profondeur (car ce sont les plus répendus sur le marché), qui est utile pour la détection de la main. Nous nous sommes donc intéressés, uniquement, à la catégorie d'images du dataset pourtant l'information d'intensité. Cette restriction soulève ainsi deux interrogations auxquelles nous répondons dans la suite de cet article :

1. comment détecter la main sur une image ?
2. Quel outil de classification utiliser pour préserver une forme de généralisation ?

Dans un premier temps, afin de se substituer à l'information de profondeur pour extraire la forme de la main, nous avons appliqué un ensemble de traitements sur nos images. Ces traitements ont pour but d'extraire l'information pertinente de l'image, à savoir la main. Ainsi, nous avons appliqué :

- un filtre préliminaire pour nettoyer le bruit
- une égalisation d'histogramme pour augmenter le contraste
- un nouveau filtre de bruit pour éliminer les singularités apparues
- un filtre de peau pour éliminer l'environnement inutile

Dans un second temps, nous avons choisi d'entraîner un réseau de neurones convolutif pour réaliser la classification. Soumis à une contrainte temporelle, nous avons choisi de travailler avec un réseau pré-entraîné : ALEXNET. Ce réseau a été entraîné sur IMAGENET qui est un très grand dataset d'images variées ; c'est pourquoi nous avons choisi ce réseau pour notre projet. Nous avons réalisé du *finetuning* sur ce réseau et cette étape est décrite dans la sous-section 2.2.

Nous présenterons dans une première section un état de l'art dans la reconnaissance de signes avec les travaux de Pugeaut et Bowsen. Puis dans une deuxième section, nous détaillerons les choix que nous avons opérés dans notre approche, avant de présenter dans une troisième section, les résultats des expériences que nous avons menées.

[1] Centre for Vision, Speech and Signal Processing

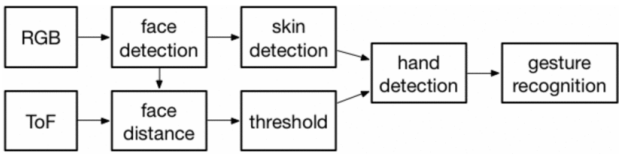


Figure 1: Aperçu du système avec les caméras RGB et ToF.

1 État de l'art

Différents personnes se sont déjà intéressées à ce sujet de reconnaissance de signes par le passé. Nous présentons par la suite quelques uns de ces travaux ainsi que les résultats obtenus par ces derniers.

1.1 Travaux de J. Isaacs et S. Foo

J. Isaacs et S. Foo ont amorcé les travaux dans la reconnaissance gestuelle qui, selon eux, sera l'avenir dans les interfaces hommes-machines[2]. Ils ont également travaillé à la reconnaissance de l'alphabet ASL.

Dans leur approche, ils ont utilisé une décomposition de chaque image représentant une lettre, sous la forme d'un vecteur de features. Ces vecteurs, ils les ont ensuite donnés en entrée d'un réseau de neurones artificiel.

Ils ont obtenu de résultats qui semblent intéressants : 99.9% de taux de reconnaissance. Toutefois, ces résultats ne précisent pas la taille du dataset utilisé, ni la qualité des images qui le composent.

1.2 Travaux de Michael Van den Bergh et Luc Van Gool

Van den Bergh a introduit un système d'interaction temps-réel par les gestes de la main. Ce système est basé sur de la reconnaissance de peau et sur la capture de l'information de profondeur pour une meilleure reconnaissnace 3D[3].

La grande innovation de leur travaux est l'apport de l'information de profondeur captée à l'aide de caméras ToF - Time-of-Flight. En combinant les images portant l'information d'intensité de lumière et celles portant l'information de profondeur, ils arrivent à faire une distinction entre les gestes de la main et le reste de l'environnement comme le montre la figure1. La main ainsi détectée est alors fournie à un algorithme de reconnaissance de gestes basé sur *Haarlet*.

1.3 Travaux de N. Pugeaut et R. Bowsen

Dans la même veine d'idée que les précédents travaux, Pugeaut et Bowsen ont combiné les informations d'intensité de couleur et celles sur la profondeur pour parvenir à une meilleure détection de la main dans les images. Leur processus se décline 3 étapes qui sont les suivantes :

- **Extraction et suivi de la main :** qui est réalisé grâce à l'interface de la Microsoft Kinect
- **Sélection des features :** chaque image est représentée par un vecteur concaténant les intensités lumineuses et les informations de profondeur
- **Classification par forêts aléatoires :** les vecteurs ainsi formés sont enfin classés par un algorithme de classification

2 Approche

La plupart des méthodes existantes sont basées sur une approche reposant entièrement sur l'extraction d'attribut et la classification à l'aide de méthodes spécifiques à la vision par ordinateur. Dans notre cas, nous nous sommes concentrés sur une classification des signes en deux parties:

- l'utilisation de filtres d'images pour nettoyer et améliorer l'information disponible dans les images
- L'utilisation d'un réseau de neurones pour classifier les images résultant de ce prétraitement

2.1 Prétraitement des images

Le prétraitement des images se fait dans notre cas en 5 étapes:

- Un traitement de bruit préliminaire
- Une égalisation d'histogramme pour augmenter le contraste sur les données et 'centrer' les caractéristiques des images
- Un autre traitement de bruit pour atténuer les extrêmes générés par le filtre précèdent
- Un filtre de peau pour réduire au maximum les données inutiles dans les images

2.1.1 Traitement de bruit

La première étape dans tout traitement des images consiste à réduire le bruit générer par la prise de vue de l'image (principalement de type *poivre et sel*).

Dans notre cas, la conservation des bordures internes à l'image est un point important de notre analyse, nous avons donc eu à choisir un filtre de bruit adapté. Nous avons choisi un filtre *médian 3x3* qui est très efficace pour ce type de réduction de bruit et permet de conserver les contours. Le choix de la taille du filtre est obtenu de manière expérimentale afin de conserver une définition de l'image maximale tout en retirant les pixels aberrants.

2.1.2 Égalisation d'histogramme

Le choix de l'application d'une égalisation d'histogramme est basée sur deux besoins:

- L'augmentation du contraste sur l'image existante pour rendre les caractéristiques de l'image plus proéminentes et ainsi accélérer l'apprentissage dans les étapes suivantes
- L'harmonisation de la luminosité dans l'image. Cela permet de réduire l'impact de l'intensité de l'éclairage sur les caractéristiques de l'image.

Afin de remplir ces deux conditions, on a donc effectuer la conversion dans l'espace *YCrCb* [4] pour faciliter le traitement selon les besoins précédents. La composante qui nous intéresse est la *luminance relative* Y qui est déterminer par la formule suivante:

$$Y = 0,3R + 0,6G + 0,1B \quad (1)$$

Il s'agit alors d'égaliser l'histogramme de cette composante avec la transformation suivante:

$$T(x_k) = \frac{224}{n} * \sum_{j=0}^{k} n_j \quad (2)$$

Avec n est le nombre de pixel total, x_k le niveau d'un pixel, et n_j le nombre d'occurrence au niveau x_k.

Expérimentalement, on peut voir une amélioration du contraste apparaître tout en restant proche de la répartition initial des couleurs. Le plus gros apport est l'harmonisation de la luminosité qui permet de rendre les images partiellement indépendantes de l'intensité de l'image initiale (qui dépend beaucoup de la prise de vue).

On voit toutefois l'apparition de nouveaux extrêmes non significatifs dans les pixels correspondant à des composantes de bruit, précédemment de faible intensité, qui ont été ignorer par le premier filtre médian. L'application d'un second filtre de bruit médian est alors nécessaire pour réduire l'impact des pixels de valeur extrême.

2.1.3 Filtre de peau

Notre reconnaissance étant basée sur les caractéristiques de la main, on va appliquer un filtre de peau afin de réduire les données inutiles présentes dans l'image. Notre méthodologie de filtrage de peau est inspirer des travaux existant [5] que nous avons adapter à notre problème.

L'approche des travaux précédents consiste en 9 différentes conditions dans les espaces, RGB, YCrCb, et HSV combiner mais dans notre cas, nous nous sommes limités aux conditions sur l'espace RGB. L'expérimentation nous a montré que les règles sur les espaces YCrCb et HSV, qui ont pour but de restreindre les pixel détectés en tant que peau, sont trop restrictives et génèrent une trop grande perte de données sur les images. La présence de 'bruit' additionnel correspondant à des pixels n'appartenant pas à la main est moins gênante que l'absence de caractéristiques de la main qui survient par moment.

Les conditions appliquées sont donc, dans l'espace RGB:

$$(B > 20)AND(G > 40)AND (R > 95)AND(R > G + 15)AND(R > B) \quad (3)$$

$$(R > 220)AND(G > 210)AND(B > 170)AND (|int(R) - int(G)| <= 15)AND(R > B)AND(G > B) \quad (4)$$

Les deux conditions correspondent respectivement à le couleur de peau sous éclairage naturel uniforme, et sous éclairage naturel latéral ou lumière artificielle. On va donc utiliser comme règle de détection:

$$(3)OR(4) \quad (5)$$

Cette détection de peau va nous permettre de construire un masque que l'on va ensuite dilater pour faire disparaître les 'trous' qui peuvent apparaître à cause de problèmes mineurs de détection. On va donc appliquer le masque résultant à l'image afin de mettre en noir les pixels en dehors du masque et ainsi garder seulement les pixels utiles à la détermination du signe.

2.1.4 Redimensionnement des images

La dernière étape avant de fournir nos images au réseau de neurones consiste à redimensionner les images pour standardiser leurs dimensions. Les proportions d'images étant variés, ce redimensionnement va se faire en deux étapes:

- Un redimensionnement à une taille de 227*227 pixels. Cela se fait très simplement en réduisant la dimension la plus grande à une taille de 227 pixels et la dimension la plus petite à une taille $227/DimMax * DimMin$
- L'ajout d'une bordure pour augmenter la taille de la dimension la plus petite à 227 pixels. Cette bordure, ajoutée de façon symétrique, est noire pour suivre la convention définit lors de l'application du filtre de peau qui consiste à mettre les pixels inutiles en noir.

2.2 Réseau de neurones

Les prétraitements des images ont pour but principal d'améliorer la discrimination des différents caractéristiques définissant les différents signes. Une fois ce prétraitement effectué, nous passons l'image résultante à une réseau de neurones basé sur AlexNet, modifié pour notre usage.

2.2.1 Alexnet

AlexNet[6] est un réseau de neurones profond de convolution qui a été créé et entraîné pour la classification de 1.2 million d'images haute résolution en 1000 classes. Il consiste en cinq couches de convolution, dont certaines d'entre elles sont suivies d'un *Max Pooling* et de trois couches complètement connectées (MLP - Multi-Layer Perceptron).

Ce réseau a été classé parmi les meilleurs du concours qui lui a donné naissance, et a l'avantage d'être publiquement disponible, en implémentation et surtout en version pré-entraîné. Le choix de ce modèle est justifié par les très bonnes performances qu'il affiche sur la classification d'images et par sa disponibilité en version pré-entraînée.

2.2.2 Dataset

Le dataSet[7] utilisé consiste en une série de captures des 24 signes de l'alphabet américain (ASL) à l'aide de la *Kinect*. Les lettres j et z sont exclues du dataset à cause du

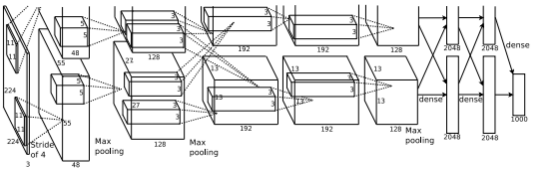


Figure 2: illustration de l'architecture du réseau AlexNet

Figure 3: Exemple du dataset utilisé

mouvement nécessaire pour les exprimer. Le dataset contient les images en RGB des signes fait par 5 différentes personnes ainsi que les données de profondeur associées.

Dans notre cas, nous avons souhaité pouvoir utiliser notre détecteur de signe sans accès à la *kinect* ; c'est pourquoi nous avons ignoré les données de profondeur. Les résultats de ces captures consiste ainsi en un set d'environ 65000 images, soit environ 2700 par classe.

Pour notre utilisation avec le réseau de neurones, nous avons extrait 100 images par classe pour effectuer un test final des résultats, le reste des images a été séparé en 3/5 pour l'apprentissage et 2/5 pour les tests associés.

Afin d'éviter un biais d'apprentissage dû à la similarité d'images successives et d'avoir des résultats semi-indépendants lors des différents processus de Learning, les données sont également mélangées de manière aléatoire à chaque exécution.

2.2.3 Finetuning

Dans le but d'accélérer la phase d'entraînement et d'améliorer la capacité de généralisation de notre réseau, nous sommes partis du réseau pré-entraîné AlexNet. Nous en avons modifié la dernière couche afin qu'elle renvoie les 24 classes qui nous concernent. Nous avons ensuite "oublié" les poids des deux dernières couches du réseau et relancé l'apprentissage avec nos données.

L'apprentissage s'effectue alors par une descente de gradient stochastique, paramétré par :

- Un **taux de dropout de 0.5** pour éviter l'overfitting
- Une prédiction générer par la **classe de probabilité maximale d'appartenance**
- Un estimateur de perte consistant d'un ***soft max* de l'entropie croisé entre la prédiction et le vrai label**
- Une précision valant la moyenne des prédictions correctes
- un système de mini-batchs de taille 48 (2 images/classe) pour assurer une bonne représentativité de chaque classe à chaque calcul de gradient
- un **learning rate** (taux d'apprentissage) de **0.01**

Nous effectuons également une évaluation sur les données de test toutes les 100 itérations afin d'examiner la progression de l'apprentissage.

2.3 Expériences

2.3.1 Pré-traitement

Le prétraitement des données est assez long à effectuer et l'expérimentation a porté principalement sur :

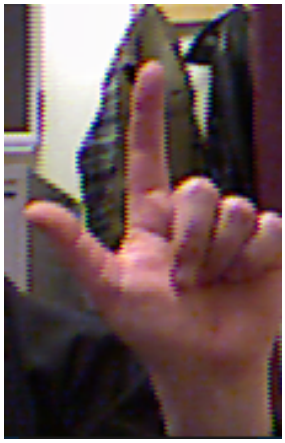
- *la taille des filtres médian pour la réduction de bruit* : les valeurs de 3, 5, et 7 ont été testées et de façon subjective nous avons convenu que la valeur 3 était la plus adaptée. Le critère principal fut au niveau des jointures des doigts qui perdent leur définition quand le filtre devient trop grand
- *l'égalisation d'histogramme appliqué avant ou après le filtre de peau* : son application préalable permet d'égaliser la luminosité des images, ce qui rend le filtre de peau plus performant. Son application a posteriori, pour bénéficier d'une égalisation plus large, donne des résultats très variables en fonction de la peau détectée précédemment
- *la détection de peau* :les différentes conditions issues du papier étudié[5] donnent des résultats assez variés. Une application stricte de la procédure suggérée est beaucoup trop restrictive sur les pixels détectés. Après avoir testé les différentes conditions, nous avons constaté que les deux conditions sur l'espace RGB sont celles qui permettent la meilleure discrimination tout en conservant toute la main.

2.3.2 Apprentissage du réseau de neurones

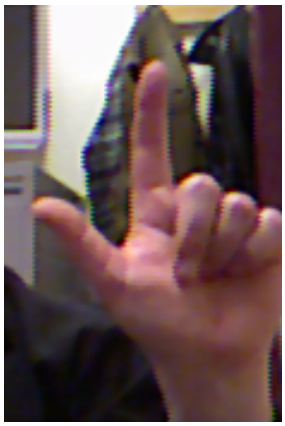
L'apprentissage du réseau de neurones fut un processus long ; c'est pourquoi nous avons conduit moins d'expérimentations dans cette phase par rapport à la précédente.

Centrage du dataset

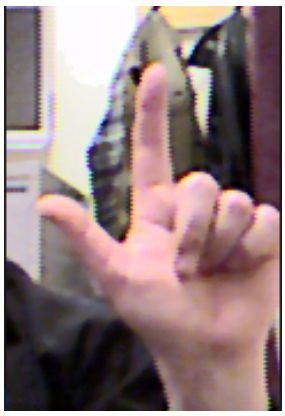
Nous avons, tout d'abord, voulu évaluer l'apprentissage sur le dataset en retirant ou non l'image moyenne (moyenne de l'ensemble des images du dataset). Cette opération a pour but d'améliorer la vitesse d'apprentissage du réseau en harmonisant les échelles de valeur entre les différentes features des images. Toutefois nous n'avons pas pu observer de différences notables sur l'évolution du taux précision, lors des premières itérations. Étant donné la contrainte du temps de calcul long lors de l'apprentissage, nous nous sommes restreints aux images directement issues du pré-traitement.



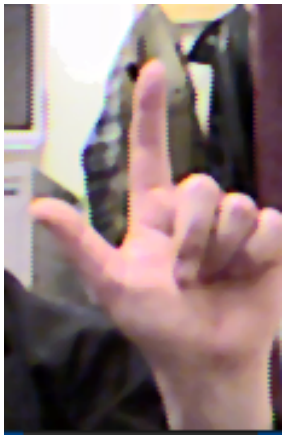
(a) Image initiale



(b) Image après un filtre median 3x3

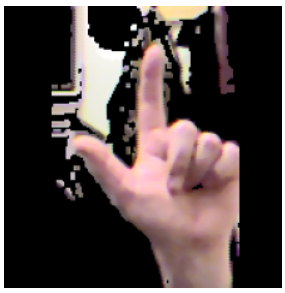


(c) Image après l'égalisation d'histogramme



(d) Image après le second filtre médian

(e) Image après le filtre de peau



(f) Image après le redimensionnement

Figure 4: Résultats des prétraitements

Similarité entre les images du dataset

Le dataset utilisé a été construit à partir de séquences d'images recueillies à l'aide d'une caméra ; c'est pourquoi il peut y avoir une forte similarité entre images successives. Pour palier à ce problème, nous avons ajouté un mélange aléatoire de la liste des images. Ce changement a permis d'avoir un apprentissage plus rapide du modèle

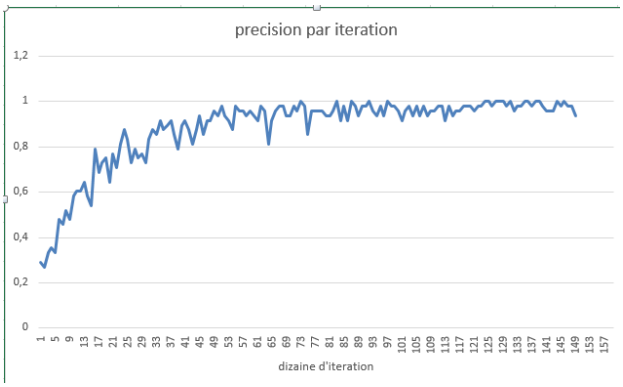


Figure 5: Évolution du taux de précision en fonction du nombre d'itérations lors de l'apprentissage

Figure 6: Évolution du taux de précision en fonction du nombre d'itérations sur l'ensemble de test

(environ 1/5 d'itérations en moins pour une performance équivalente).

Taux d'apprentissage

Nous avons également évaluer l'impact de différents taux d'apprentissage : 0.0005, 0.001 et 0.002. Un taux d'apprentissage trop élevé est mauvais pour la valeur final de précision mais augmente la vitesse de convergence. La valeur 0.002 fut un bon compromis.

Nombre d'itérations

Nous avons constater que pour un nombre d'itérations au delà 1000 (tout autre paramètre fixé par ailleurs - taille des mini-batchs, taux d'apprentissage), le niveau de précision n'excède pas 0.96. En effet, cette précision ne croit plus que très faiblement, voire elle diminue par moment. Il est donc inutile d'aller plus loin et de risquer l'*overfitting*.

Comme le montre la figure 5, notre modèle atteint de bonnes performances à partir de 500 itérations. En effet, nous avons obtenons plus de 90% de précision. Nous avons donc choisi de préserver une valeur finale enregistrée au bout de 1500 itérations : **0.975**.

2.4 Conclusion

Nous pouvons constater qu'à l'aide d'un réseau pré-entraîné, il est assez rapide et assez efficace d'entraîner un

nouveau classificateur. Dans notre cas, les tests de notre détecteur de signes sur des photos prises à l'aide de notre webcam (figure 4) montre que la détection reste toutefois soumise à des limites dues au dataset utilisé. En effet, ce dernier est basé sur des éclairages assez standards.

Bien que les signes soient effectivement détectés dans la plupart des cas (17 sur 20 dans nos photos personnelles), il existe des exemples où la particularité de l'éclairage pose problèmes. D'une part, sur la détection de peau qui devient hasardeuse et d'autre part, sur la classification par le réseau de neurones qui n'a pas été entraîné avec assez de diversité. L'augmentation de données à l'aide de modifications programmées sur les images existantes n'a pas pu résoudre ce problème. En effet, de simples modifications de contraste ou de luminosité (sous atténuée par l'égalisation d'histogramme) ne modélise pas suffisamment bien la variation d'éclairage qui survient naturellement.

References

[1] N. Pugeault, R. Bowden, Proceedings of the 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision (2011)

[2] J. Isaacs, S. Foo, System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on (2014)

[3] M.V. den Bergh, L.V. Gool, Applications of Computer Vision (WACV), 2011 IEEE Workshop on (2011)

[4] A. Taguchi, tomoaki kimura, Proceedings of 2009 APSIPA Annual Summit and Conference (2009)

[5] K.C.W. Nusirwan Anwar bin Abdul Rahman, J. See (2006)

[6] G.E.H. Alex Krizhevsky, Ilya Sutskever (2012)

[7] *Asl finger spelling dataset* (2011), `http://empslocal.ex.ac.uk/people/staff/np331/index.p`