

Les réseaux de neurones convolutifs dans le monde des cylindrées

Ghouibi Ghassen

Université Paris 8

ghassen.ghouibi@etud.univ-paris8.fr

23/01/2020

Sommaire

- 1 Introduction
 - Contexte
 - Problématique
- 2 CNN
 - Définition
 - Algorithme
- 3 Solution CNN
 - Détection de signalisation
 - Détection de collision
 - Détection de dommages matériel
 - Voiture autonome
- 4 Mon avis
- 5 Conclusion

Approche

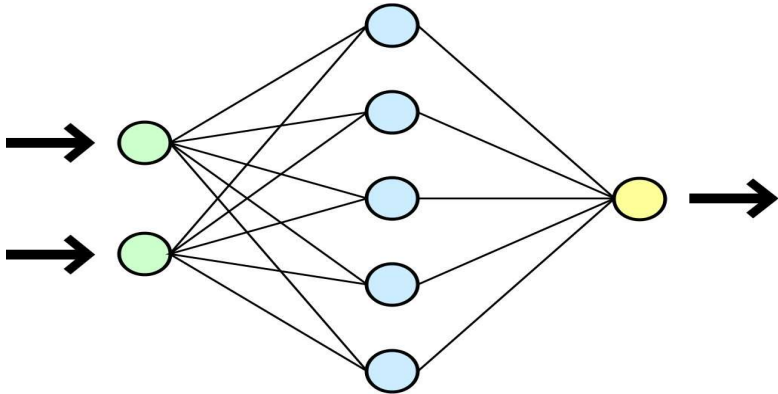


Figure: C'est quoi un réseau de neurones

Sujet

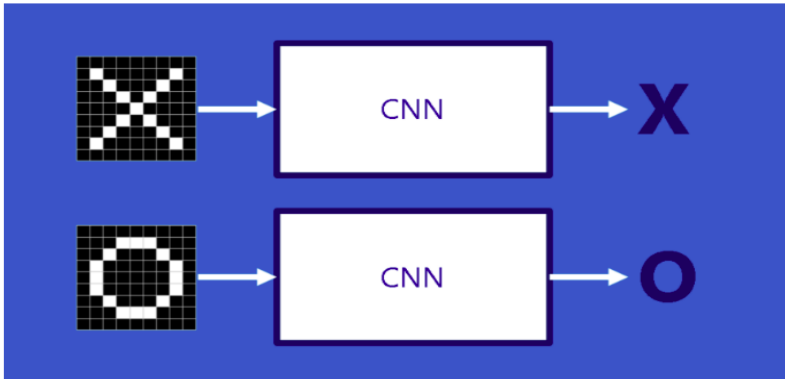


Figure: CNN



Figure: Quelques types d'accident

La distraction est une cause majeure d'accidents de la route:

- Parler au téléphone ou envoyer des messages
- Chercher des objets en arrière de la voiture
- Boire ou consommer des produits stupéfiants



Figure: Quelques types distractions

Problématique

Problématique

Est ce que les nouvelles solutions trouvés dans le domaine de classification d'image, machine learning et prédiction peuvent réduire les accidents de la route voir les éliminer complètement ?

C'est quoi les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs ont une méthodologie similaire à celle des méthodes traditionnelles d'apprentissage supervisé, ils reçoivent des images en entrée, détectent les features de chacune d'entre elles, puis entraînent un classifieur dessus.

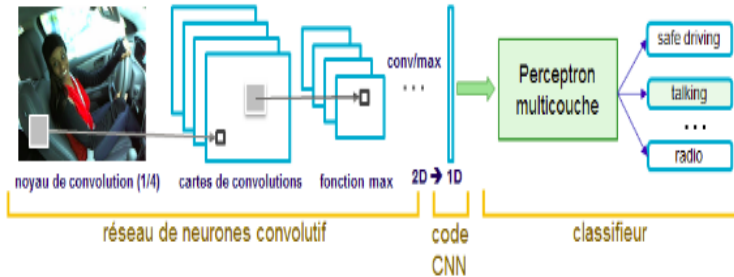


Figure: L'architecture d'un CNN

Pseudo-code entraînement

Algorithm 3. Pseudo code of CNNTTRAIN

Input: cnn.layers{'i', 'c', 's', 'c', 's'};
Output: cnn.layers{'i', 'c', 's', 'c', 's'};

- 1: **Calculate** the number of training samples
 $\text{numbatches} = m / \text{opts.batchsize};$
- 2: **IF** $\text{rem}(\text{numbatches}, 1) \neq 0$
 $\text{error}(\text{'numbatches not integer'})$;
End $\quad \text{//rem: Remainder after division. rem}(x, y) \text{ is } x - n * y$

$$\text{Loss} \leftarrow \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

- 3: Initialize net.rL $\quad \text{net.rL} = []$;
- 4: **FOR EACH** $i = 1 : 1 : \text{opts.numepochs}$
- 5: **DO** Model Selection $\quad \text{kk} = \text{randperm}(m); \quad \text{// } x_i = (x_{i-1} * A + C) \bmod M$
- 6: **FOR** $l = 1 : \text{numbatches}$ **//** Sign samples and corresponding tags
 $\text{batch_x} = x(:, :, \text{kk}((l-1) * \text{opts.batchsize} + 1 : l * \text{opts.batchsize}))$;
 $\text{batch_y} = y(:, \quad \text{kk}((l-1) * \text{opts.batchsize} + 1 : l * \text{opts.batchsize}))$;
 $\quad \quad \quad \text{// } \frac{\partial \delta}{\partial a_i} = \sum_i \sum_{y_i} \frac{\partial \delta}{\partial f(y_i)} \frac{\partial f(y_i)}{\partial a_i}$
 $\text{net} = \text{CNNNET}(\text{net}, \text{batch_x})$; **//** Calculate net output in CNNNET.
 $\text{net} = \text{CNNTRAIN}(\text{net}, \text{batch_y})$; **//** Train algorithms.
 $\text{net} = \text{CNNAPPLYGRADS}(\text{net}, \text{opts})$; **//** Update weights of networks.
- 7: **END** $\quad \text{net.rL}(\text{end} + 1) = 0.99 * \text{net.rL}(\text{end}) + 0.01 * \text{net.L}$; **//** Record the error.
- 8: **END** **Return** net.layers

Figure: Pseudo code de CNN

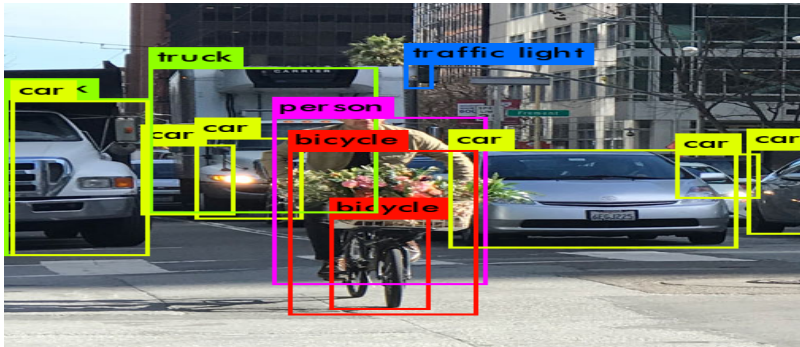


Figure: Détection de traffic

Additionneur pondéré

$$a_i^j = \sigma\left(\sum_k a_k^{i-1} w_{jk}^i\right) \quad (1)$$

La croix entropie

$$H(p, q) = -\sum_i Y(i) \log y(i) \quad (2)$$

Fonction d'activation

$$ELU(x) = \begin{cases} \exp(x) - 1, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (3)$$

Résultat



Figure: Résultat obtenu dans l'article

Détection de collision

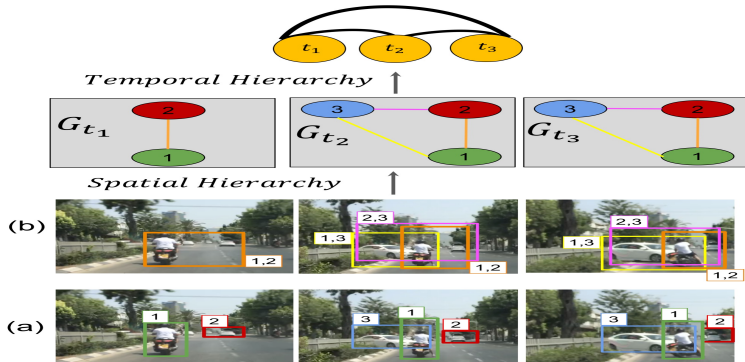


Figure: détection de colission

Algorithme

ALGORITHM 1: Collision Detection

Input: current frame $t_{current}$, collision state list *Collision*

Output: collision state list *Collision*

```

for  $t_L \leftarrow t_{previous}$  to  $t_{current}$  in steps of  $L$  frames do
    for each pair of object trajectory  $(p_{:t_L}^1, p_{:t_L}^2)$  do
        if  $(p_{:t_L}^1 \text{ intersects } p_{:t_L}^2 \text{ as of } t_L)$  then
            | add  $o_1, o_2$  to Collision
        end
    end
    if (Collisions) then
        |  $t_{previous} \leftarrow t_L$ ; return TRUE
    end
end
 $t_{previous} \leftarrow t_d$ ; return FALSE

```

Figure: Pseudo code détection de colission

Résultat

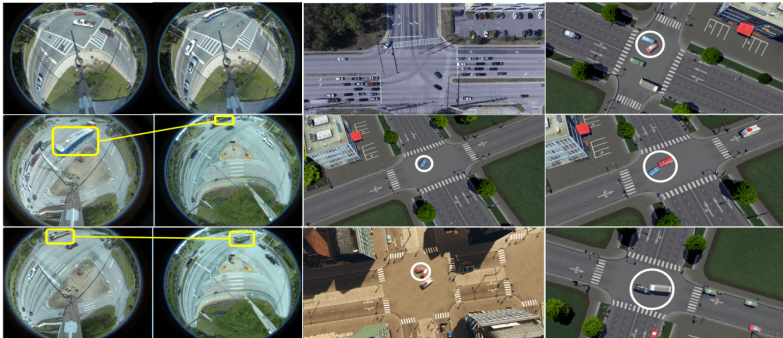


Figure: Résultat obtenu dans l'article

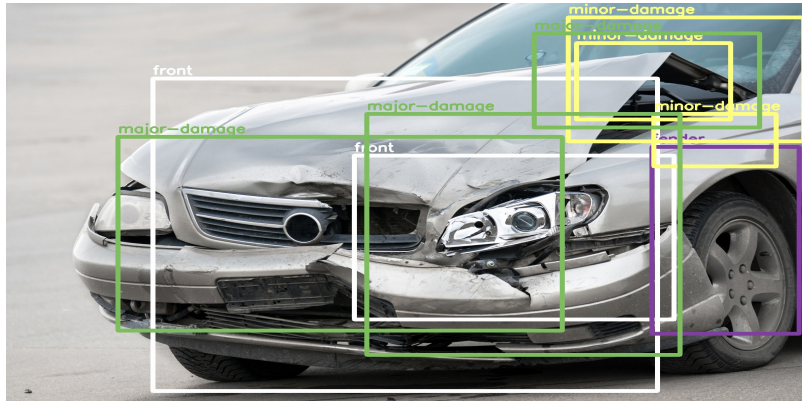


Figure: Détection des dommages causé après un accident

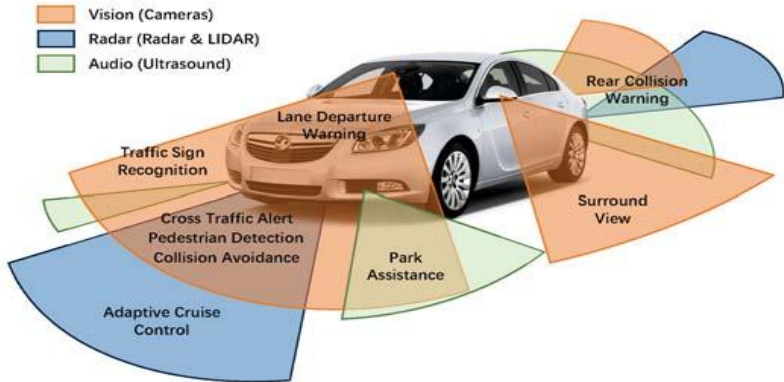


Figure: Capteur dans une voiture

Conduite autonome

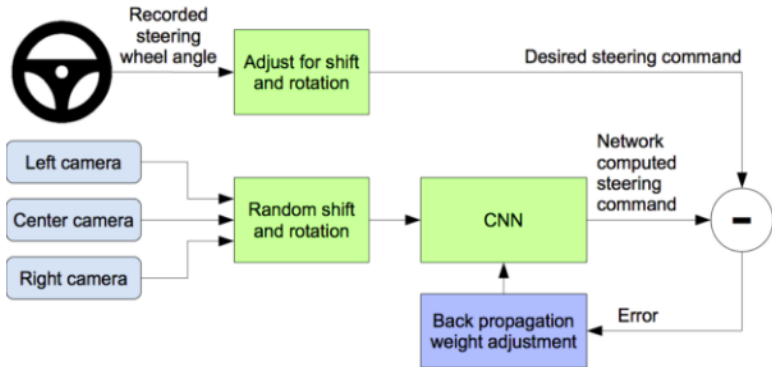


Figure: Vue de haut niveau du système de collecte de données.

Mon avis

A mon avis la solution c'est avoir plus des voitures autonome, à traves la fiabilité des CNN on peut construire des voitures qui pourrons sauver beaucoup de personnes.

- une réduction des accidents
- une réduction des embouteillages
- être seul en voiture sans avoir à conduire

Conclusion

Pour les travaux futurs, il serait important d'étudier d'autres algorithmes d'apprentissage automatique pour la gravité des accidents de la circulation prédiction et appliquer ces algorithmes sur différents ensembles de données sur le trafic pour examiner leur efficacité.

Sources



<https://hal.archives-ouvertes.fr/hal-00853274/document>



<https://arxiv.org/pdf/1901.01138.pdf>



<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn>



<http://vision.ia.ac.cn/zh/senimar/reports/ConvNet-slide-luohl.pdf>



<https://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicite/>



<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8661485>



<https://towardsdatascience.com/cnn-application-detecting-car-exterior-damage-full-implementable-code-1b205e3cb48c>



<https://www.sciencedirect.com/science/article/pii/S1474667017526464>



<https://hal.inria.fr/hal-00737443/document>

Merci pour votre attention !