

Détection de la coupe L3 par CNN et transfert learning

UCA Deep Learning School 2017 - Deep in France

S. Belharbi, C. Chatelain, R. Hérault, S. Adam, M. Chastan, S.
Thureau, R. Modzelewski

LITIS - Équipe Apprentissage - Centre Henry Becquerel de Rouen

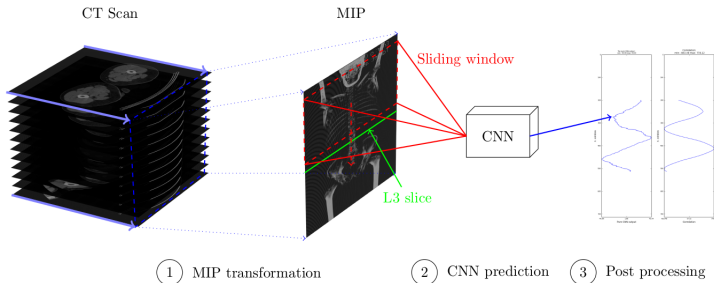
`soufiane.belharbi@insa-rouen.fr`

13 Juin 2017

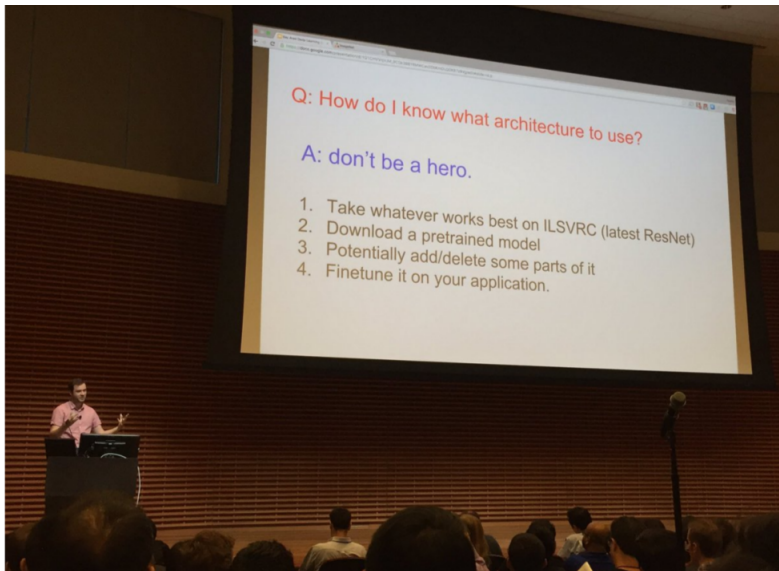
À propos

Publication :

Belharbi Soufiane, Clément Chatelain, Romain Hérault, Sébastien Adam, Sébastien Thureau, Mathieu Chastan and Romain Modzelewski, “***Spotting L3 slice in CT scans using deep convolutional network and transfer learning***”, Computers in Biology and Medicine, 2017.



Message à prendre avec vous



Q: How do I know what architecture to use?

A: don't be a hero.

1. Take whatever works best on ILSVRC (latest ResNet)
2. Download a pretrained model
3. Potentially add/delete some parts of it
4. Finetune it on your application.

Plan

Introduction

Problématique

Approche proposée

Résultats

Conclusion

Sommaire

Introduction

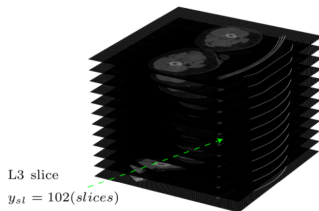
Problématique

Approche proposée

Résultats

Conclusion

Introduction



- Sur une base de **642 CT scans** on a obtenu une **erreur moyenne de localisation < 1.82 coupes (< 5 mm)**.

Sommaire

Introduction

Problématique

Approche proposée

Résultats

Conclusion

Détection de la coupe L3

Étant donné un scan d'une partie du corps, trouver la coupe correspondante à la coupe L3 parmi quelques centaines de coupes

difficultés

- variabilité inter-patients
- ressemblances coupe L3 avec d'autres coupes
- nécessité d'utiliser le contexte pour décider de la L3

⇒ Machine Learning

Approches envisageables

Classification de coupes (prédire une valeur discrète)

Pour chaque coupe décider "L3"/"pas L3"

- ▶ Approche simple, 😊
- ▶ Pas de contexte 😞

Étiquetage de séquences

Chercher à trouver toutes les vertèbres L1, L2, L3, ...

- ▶ Analyse globale : contexte, 😊
- ▶ Travaux existants semblent prometteurs, 😊
- ▶ Nécessite étiquetage de chaque coupe 😞

Regression (prédire une valeur continue)

Étant donnée toute la séquence, estimer la hauteur de la coupe L3

- ▶ Analyse globale : contexte, 😊
- ▶ Pas de travaux existants,
- ▶ Nécessite seulement la position de la L3 en étiquetage 😊

Sommaire

Introduction

Problématique

Approche proposée

Résultats

Conclusion

Regression pour la détection de la coupe L3

Choix du modèle statistique pour la regression

- ▶ Deep learning, Convolutional Neural Network (CNN)
- ▶ Pas de caractéristiques à extraire
- ▶ State-of-the-art sur l'image
- ▶ Modèle statique : requiert un espace d'entrée de taille fixe

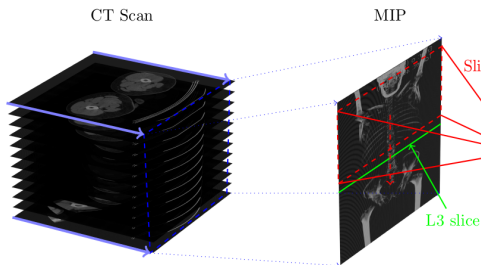
Parlons chiffres

- ▶ Espace d'entrée : 1 scan = $N \times 512 \times 512$, avec $400 < N < 1200$
- ▶ Scans avec L3 annotée et vérifiée : 642 patients
- ▶ La hauteur des scans N est variable d'un scan à un autre

Solution au problème N°1

Problème N°1 : dimension espace d'entrée

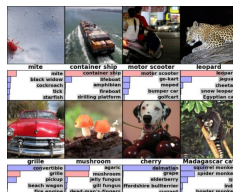
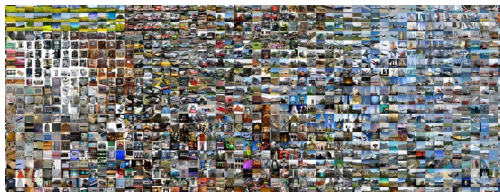
- ▶ 131 M entrées par exemple : ça fait beaucoup même pour un CNN !
- ▶ → Maximum Intensity Projection (MIP) facial ou latéral
- ▶ On passe de $512 \times 512 \times N$ à $512 \times N$
- ▶ Information pertinente *a priori* conservée



Solution au problème N°2

Problème N°2 : Peu de données annotées (642 patients)

- ▶ On a essayé quand même ! → résultats mitigés
- ▶ Solution : utilisation de modèles pré-appris sur des grandes quantités de données
- ▶ AlexNet, GoogleNet, VGG16, VGG19, etc.
- ▶ Modèles appris sur ImageNet : 14 millions d'images naturelles annotées [Fei-Fei and Russakovsky 2013]



Solution au problème N°2 (suite)

Idée générale : Transfer Learning

Bénéficier des filtres convolutionnels préappris sur des images naturelles (!!!), qu'on affine ensuite sur nos données.

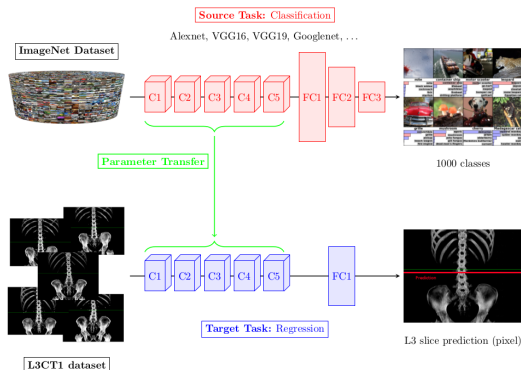
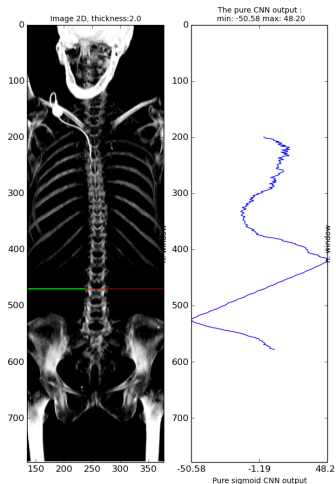


FIGURE 1 – Transfert des filtres

Solution au problème N°3

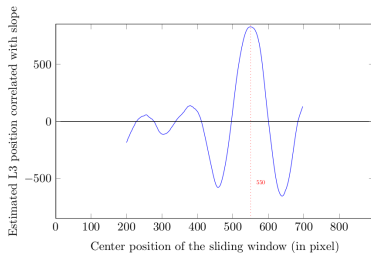
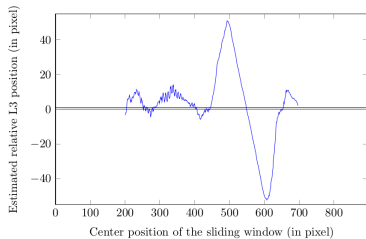
Problème N°3 : Espace d'entrée de taille variable

- ▶ Problème assez classique
- ▶ Utilisation d'une fenêtre glissante
- ▶ Post-traitement pour analyser la sortie du CNN aux différentes positions



Post-traitement

- Corrélation.



Système complet : Prédiction

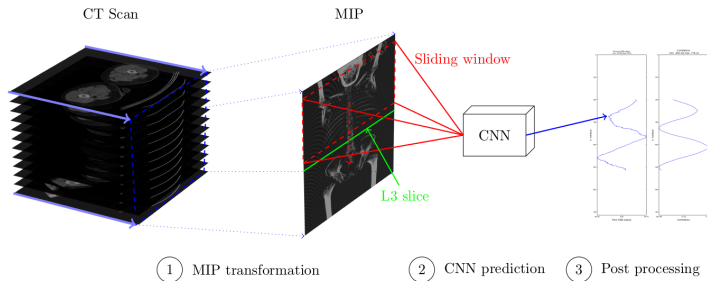


FIGURE 2 – Système complet : Prédiction

Sommaire

Introduction

Problématique

Approche proposée

Résultats

Conclusion

Résultats quantitatifs

Erreur en cross validation : comparaison d'un CNN "maison" et des CNN pré-appris + transfer learning

	RF500	CNN4	Alexnet	VGG16	VGG19	Googlenet
fold 0	7.31 ± 6.52	2.85 ± 2.37	2.21 ± 2.11	2.06 ± 4.39	1.89 ± 1.77	1.81 ± 1.74
fold 1	11.07 ± 11.42	3.12 ± 2.90	2.44 ± 2.41	1.78 ± 2.09	1.96 ± 2.10	3.84 ± 12.86
fold 2	13.10 ± 13.90	3.12 ± 3.20	2.47 ± 2.38	1.54 ± 1.54	1.65 ± 1.73	2.62 ± 2.52
fold 3	12.03 ± 14.34	2.98 ± 2.38	2.42 ± 2.23	1.96 ± 1.62	1.76 ± 1.75	2.22 ± 1.79
fold 4	8.99 ± 7.83	1.87 ± 1.58	2.69 ± 2.41	1.74 ± 1.96	1.90 ± 1.83	2.20 ± 2.20
Average	10.50 ± 10.80	2.78 ± 2.48	2.45 ± 2.42	1.82 ± 2.32	1.83 ± 1.83	2.54 ± 4.22

TABLE 1 – L'erreur de test en cross validation (avec post-traitement) (coupes).

Comparaison avec Random Forests (RF) avec 500, 100, 10 arbres (Local Binary Patterns (LBP)).

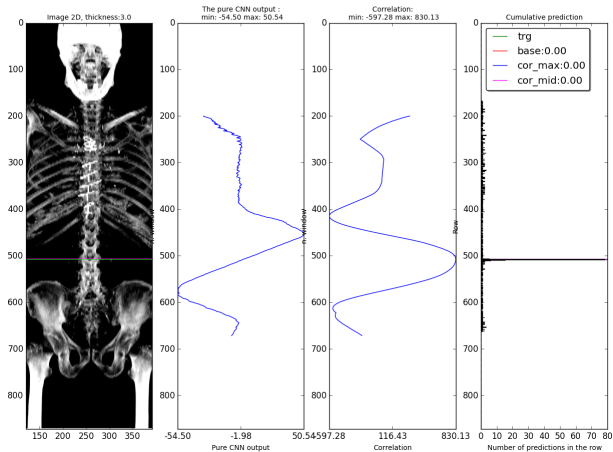
Temps d'évaluation

	Nombre de paramètres	Temps moyen de traitement (seconds/CT scan)
CNN4	55 K	04.46
Alexnet	2 M	06.37
VGG16	14 M	13.28
VGG19	20 M	16.02
Googlenet	6 M	17.75

TABLE 2 – Nombres de paramètres et temps d'évaluation d'un scan sur un GPU (K40).

Résultats Qualitatifs

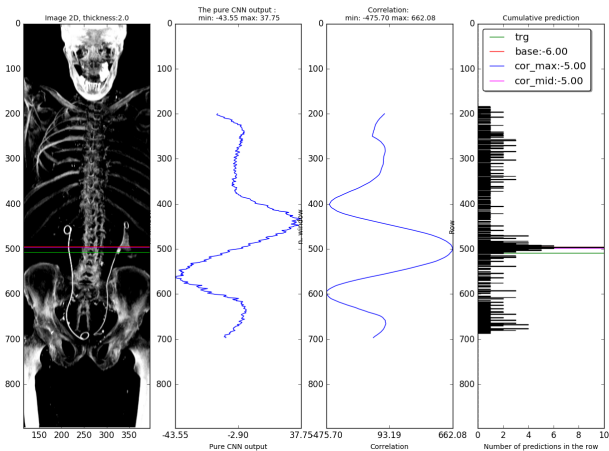
Prediction for pt_id: 165_5112614581.



Erreur : 0 coupes.

Résultats Qualitatifs

Prediction for pt_id: 1_9352086790.



Erreur : 6 coupes.

CNN vs. radiologistes

- Nouvelle base : 43 CT scans annotés par le radiologiste de référence.
- Faire prononcer 3 autres radiologistes sur la base.

Errors (slices) / operator	CNN4	VGG16	Radiologist #1	Radiologist #2	Radiologist #3
Review1	2.37 ± 2.30	1.70 ± 1.65	0.81 ± 0.97	0.72 ± 1.51	0.51 ± 0.62
Review2	2.53 ± 2.27	1.58 ± 1.83	0.77 ± 0.68	0.95 ± 1.61	0.86 ± 1.30

TABLE 3 – Comparaison : CNN et radiologistes.

- 1.70/1.58 coupes \sim 5 mm.
- Nous avons perdu le combat avec les radiologistes mais avec les honneurs 😊.

Accélération de calcul (prédiction)

Réduire le taux d'échantillonnage \Rightarrow réduire le nombre des fenêtres à traiter.

Exemple pour VGG16 :

1 fenêtres à chaque **1 pixel** : **13.28 seconds/CT scan**, erreur : **1.82 ± 2.32** .

\Rightarrow

1 fenêtres à chaque **6 pixel** : **2.36 seconds/CT scan**, erreur : **1.91 ± 2.69** .

Perspectives : Comment accélérer plus ?

Tendance actuelle : (déploiement de deep learning)

Comment déployer VGG16 sur une machine à faible puissance de calcul (téléphone, cpu) et

faire des prédictions sur plusieurs images ? 1 images ~ 1 second sur cpu.

⇒ **Élagage des couches de convolution (filtre/cartes)**. (inclut le transfert learning)

Perspectives : Comment accélérer plus ?

Tendance actuelle : (déploiement de deep learning)

Comment déployer VGG16 sur une machine à faible puissance de calcul (téléphone, cpu) et

faire des prédictions sur plusieurs images ? 1 images ~ 1 second sur cpu.

⇒ **Élagage des couches de convolution (filtre/cartes)**. (inclut le transfert learning)

Références :

- ▶ *Pruning Convolutional Neural Networks for Resource Efficient Transfer Learning.* (<https://arxiv.org/abs/1611.06440>)
- ▶ *Pruning Filters for Efficient ConvNets.* (<https://arxiv.org/abs/1608.08710>)
- ▶ *Faster CNNs with Direct Sparse Convolutions and Guided Pruning.* (<https://openreview.net/pdf?id=rJPcZ3txx>)

Perspectives : Comment accélérer plus ?

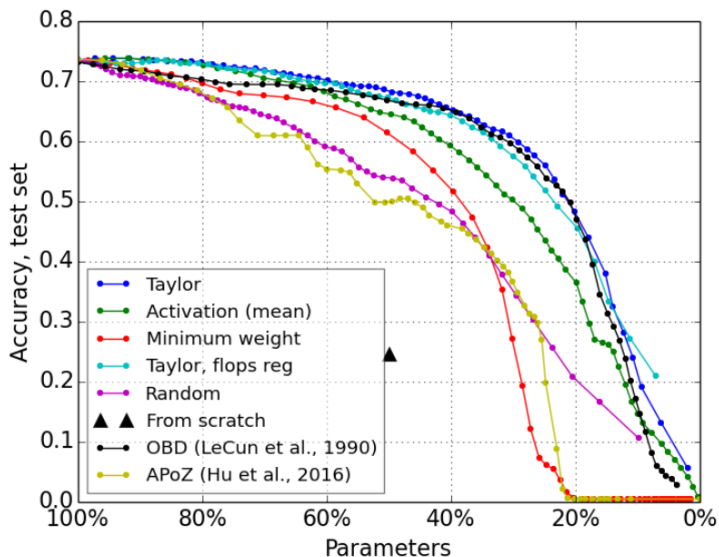


FIGURE 3 – Élagage des carte caractéristiques dans VGG16 fine-tuné sur la base Birds-200. (<https://arxiv.org/abs/1611.06440>)

Perspectives : Comment accélérer plus ?

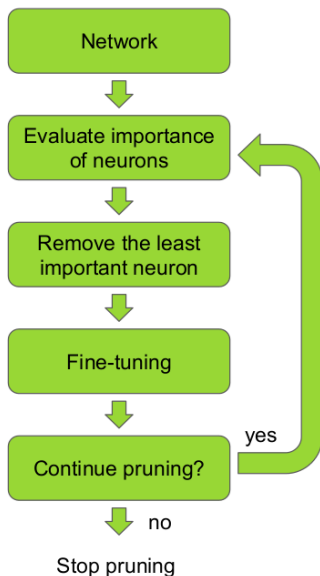


FIGURE 4 – Comment élaguer ?. (<https://arxiv.org/abs/1611.06440>)

Sommaire

Introduction

Problématique

Approche proposée

Résultats

Conclusion

Conclusion

Conclusion

- ▶ Résultats intéressants
- ▶ Pipeline adapté : prétraitement / CNN / post traitement
- ▶ Généricité : utilisation du transfer learning requiert assez peu de données et de temps d'apprentissage. → test sur d'autres problèmes ?
- ▶ Le déploiement est important (à réfléchir).

Questions ?

soufiane.belharbi@insa-rouen.fr
<https://sbelharbi.github.io>