# Projet de mémoire

## Détection des replays dans les vidéos de sport

Billal Boudjoghra

July 18, 2019

# Outline

1. Introduction
2. Détection par analyse d'image
3. Réseau à convolution
4. Détection par apprentissage profond
5. Suite de la recherche

# Sportagraph : Présentation

- son produit : Digital Asset Manager
- mon poste : développeur Scala
- ma tâche : détections des replays dans les vidéos de sport

# Sujet de recherche : Détection des replays dans les vidéos de sport

- en lien avec mon travail en entreprise
- thème vaste : deep learning / computer vision

# Détections des replays

- Les replays sont les moments forts de la vidéo
- Hypothèse : les replays sont compris entre deux logos
- Objectif : détection/reconnaissance des logos


Figure 1: Un logo

# ORB (1/3)

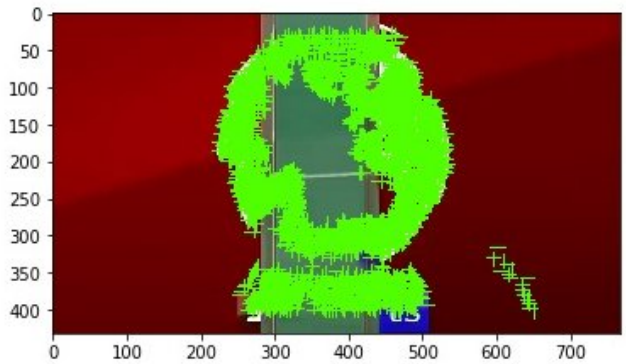
- outil : OpenCV

**Idée**

- obtenir des features pour chaque frame de la vidéo (SIFT, ORB)
- appliquer des algorithmes de machine learning sur ces features (K-NN)

# ORB (2/2)


16564.jpg
16565.jpg
16566.jpg
16567.jpg
16568.jpg
16569.jpg
16570.jpg
16571.jpg
16572.jpg
16573.jpg
16574.jpg
16575.jpg


0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700

# ORB : Résultats (3/3)

| **Feature extracted by ORB on :** | Football:PL<br>150k frames<br>48 logos | Football:Ligue 1<br>80k frames<br>16 logos | Football:Ligue 1<br>150k frames<br>34 logos | Football:Liga<br>300k frames<br>36 logos | Tennis:AusOpen<br>400k frames<br>32 logos |
|---|---|---|---|---|---|
| 1 frame / shot | 29.73% 68.75% | 14.94% 40.63% | 30.86% 73.53% | 19.19% 52.78% | 11.36% 46.88% |
| 1 frame window / shot | 55.10% 84.38% | 31.51% 71.88% | 34.67% 76.47% | 34.62% 75.00% | 14.29% 53.13% |
| Process time | 2700 s | 1600 s | 2700 s | 4000 s | 5300 s |

**Score : Precision (left), Time (middle, in italic), Recall (right)**

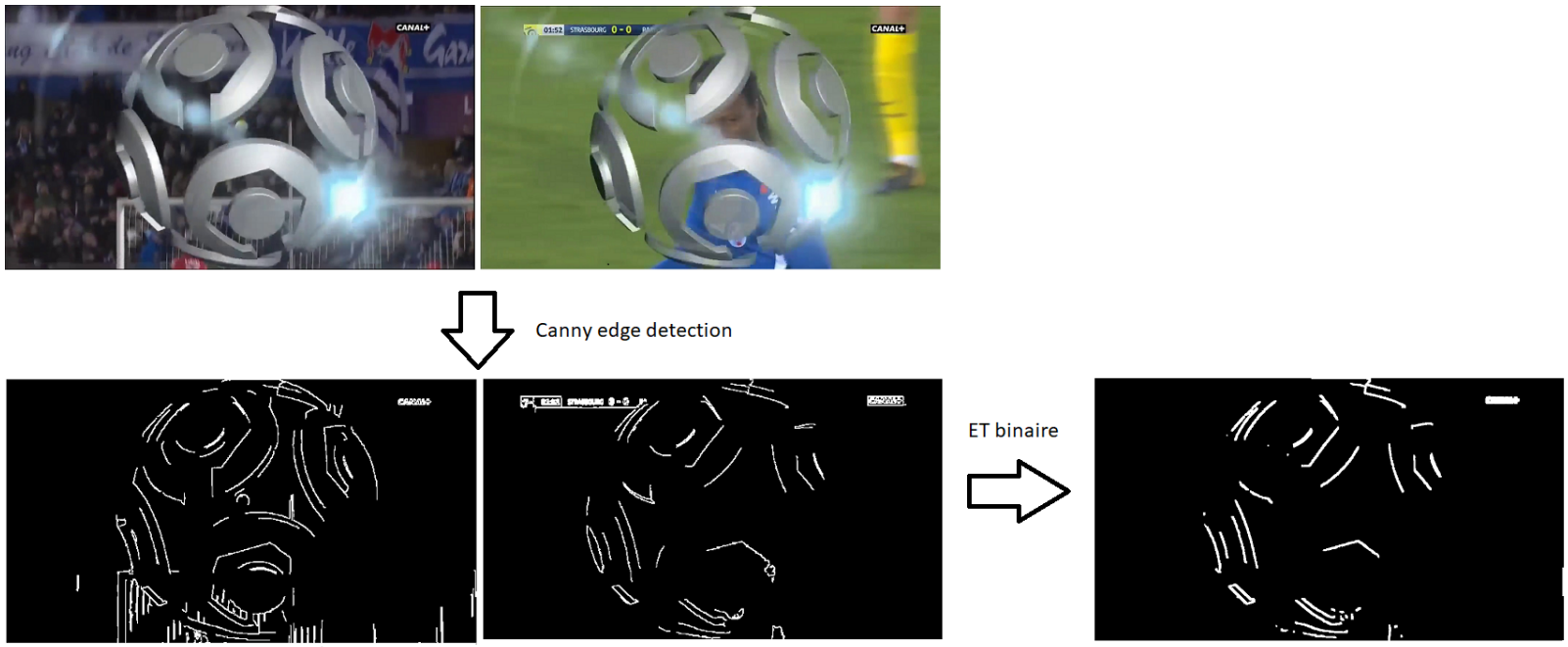
# Détection de contours (1/3)

- outil : OpenCV

## Idée

- détecter les contours pour chaque frame de la vidéo (canny edge detection)
- logo image fixe => les contours des frames logo sont les mêmes

# Détection de contours (2/3)



# Détection de contours : Résultats (3/3)

| **Video width / Video height / Luminance threshold** | Football : Prem League 150k frames 48 logos | | Football: Ligue 1 80k frames 16 logos | | Football: Ligue 1 150k frames 34 logos | | Football: Liga 300k frames 36 logos | | Tennis: Australia Open 400k frames 32 logos | |
|---|---|---|---|---|---|---|---|---|---|---|
| **With delete BG + dilate contour** | *200 seconds* | | *100 seconds* | | *150 s* | | *300 s* | | *200 s* | |
| 5000,100,100,5,11,2,2, 0.64 | 100.00% | 100.00% | 93.75% | 93.75% | 100.00% | 100.00% | 65.63% | 65.63% | 85.71% | 88.24% |
| 5000,100,100,5,11,0,4, 0.64 | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 71.88% | 71.88% | 94.12% | 94.12% |
| **Without delete BG + Gauss contour / mosaic** | *340 s* | | *160 s* | | *150 s* | | *290 s* | | *350s* | |
| 10000,100,100,10,85,0,0, 0.81 (gaussian blur mosaic + contourDiff ) | 97.96% | 100.00% | 50.00% | 56.25% | 100.00% | 100.00% | 62.50% | 62.50% | 94.12% | 94.12% |
| **With delete BG + gauss contour Gauss contour / mosaic + group shot by seconds (rounded) + deleteBG = 0.8 sec** | *300 s* | | *180 s* | | *150 s* | | *290 s* | | *300 s* | |
| 10000,100,100,10,10,1,1, 0.81 | 95.92% | 97.92% | 55.56% | 62.50% | 97.14% | 100.00% | 62.86% | 68.75% | 97.06% | 97.06% |
| **With delete BG + gauss mosaic +dilate (2) contourDiff + deleteBG = 0.8 sec** | *300 s* | | *280 s* | | *330 s* | | *320 s* | | *600 s* | |
| 10000,100,100,10,20,1,1, 0.81 | 97.96% | 100.00% | 100.00% | 100.00% | 91.89% | 100.00% | 75.00% | 75.00% | 97.06% | 97.06% |
| **With delete BG + gauss mosaic +dilate (2) contourDiff + deleteBG = 0.8 sec + group shot by half-second + saveWindowSize relative to fps** | *430 s* | | *260 s* | | *640 s* | | *260 s* | | *1620 s* | |
| X*1000,100,100,X,20, 1,1, 0.81 | 97.96% | 100.00% | 100.00% | 100.00% | 91.89% | 100.00% | 75.00% | 75.00% | 91.89% | 100.00% |
| X*1500,100,100,X,10, 1,1, 0.81 | 93.75% | 93.75% | 100.00% | 100.00% | 85.29% | 86.11% | 68.75% | 68.75% | 100.00% | 100.00% |

**Score : Precision (left), Time (middle, in italic), Recall (right)**

# CNN

- efficace pour la reconnaisance d'image
- utilise la convolution au lieu de la multiplication matricielle
- deux caractéristiques importantes : l'intéraction parcimonieuse et le partage de paramètres

# CNN : opération de convolution

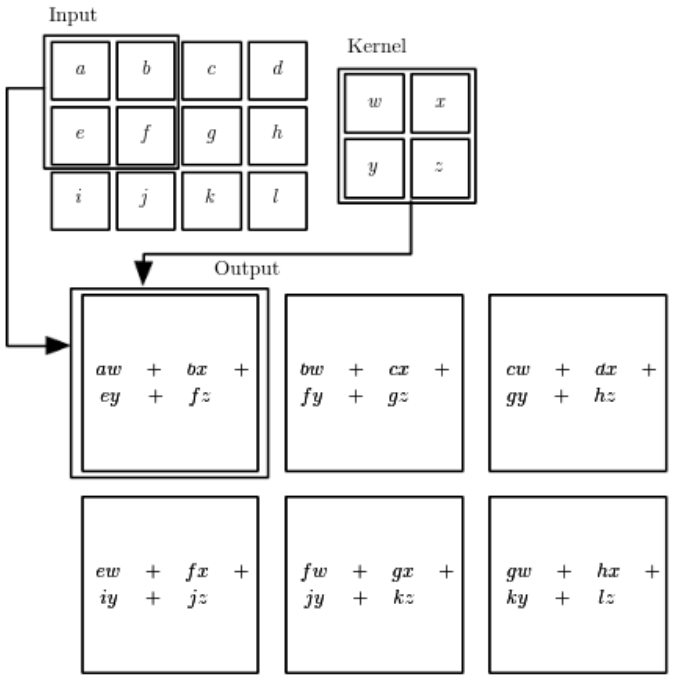



Figure 2: Opération de convolution label:convolution

- entrée : une matrice
- applique le kernel sur l'entrée
- sortie : une carte des caractéristiques (feature map)

# CNN : intéraction parcimonieuse

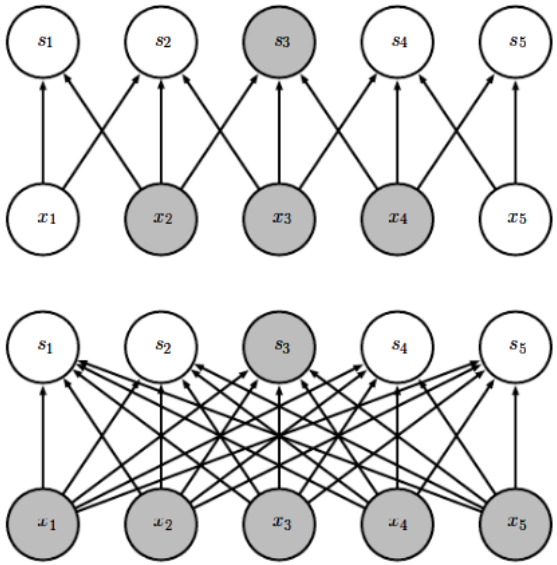



Figure 3: Intéraction parcimonieuse (en haut), intéraction non parcimonieuse (en bas) label:sparse-vs-dense

- taille kernel $<$ taille entrée
- le kernel ne parcourt qu'une petite partie de l'entrée à la fois
- moins de calculs à effectuer

# CNN : partage de paramètres

- un seul kernel itére sur l'entrée de la couche
- le réseau n'apprend que les poids du kernel
- taille kernel « taille entrée

=> beaucoup moins de paramètres à apprendre

# CNN : pooling

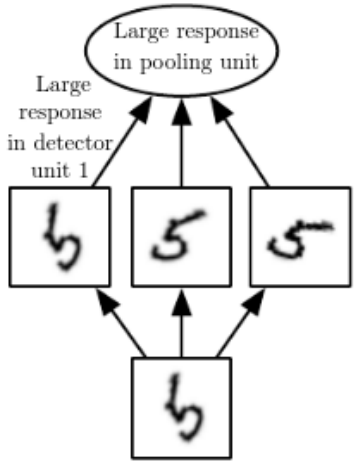



Figure 4: Pooling & invariance label:pooling

- modifie la sortie de la couche de convolution
- fait une approximation de la sortie
- rend la représentation invariante à de petits changements sur l'entrée
- améliore la capacité de généralisation des CNN

# Two-Stream Convolutional Networks (1/4)

Séparation de la tâche de reconnaissance dans les vidéos en 2 parties :

- composante spatialle
- composante temporelle

Un CNN est associé à chaque composante

# Two-Stream Convolutional Networks : Composante spatialle (2/4)

- Classifieur d'image classique (imageNet, GoogLeNet)
- Donne un indice fort sur l'action
- Bénéficie des avancées dans le domaine de l'image

# Two-Stream Convolutional Networks : Composante temporelle (3/4)

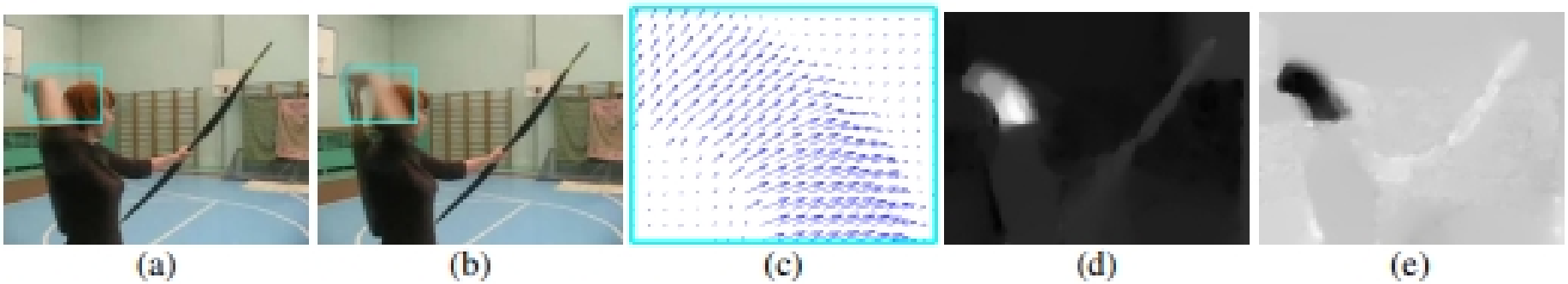



Figure 5: Flux optique label:optical-flow label:opt-flow

- utilise l'algorithme de flux optique
  - détecte le mouvement entre les images de la vidéo
- entrée du CNN temporel : image flux optique

# Two-Stream Convolutional Networks : (4/4)

Table 4: **Mean accuracy (over three splits) on UCF-101 and HMDB-51.**

| Method | UCF-101 | HMDB-51 |
|---|---|---|
| Improved dense trajectories (IDT) [26, 27] | 85.9% | 57.2% |
| IDT with higher-dimensional encodings [20] | **87.9%** | 61.1% |
| IDT with stacked Fisher encoding [21] (based on Deep Fisher Net [23]) | - | **66.8%** |
| Spatio-temporal HMAX network [11, 16] | - | 22.8% |
| "Slow fusion" spatio-temporal ConvNet [14] | 65.4% | - |
| Spatial stream ConvNet | 73.0% | 40.5% |
| Temporal stream ConvNet | 83.7% | 54.6% |
| Two-stream model (fusion by averaging) | 86.9% | 58.0% |
| Two-stream model (fusion by SVM) | **88.0%** | **59.4%** |

Figure 6: Résultats obtenus par l'approche Two-stream model label:two-stream-res

Apport de la composante temporelle : +15%

# Réseau à convolution 3D (1)

- article Learning Spatiotemporal Features with 3D Convolutional Networks cite:Tran$_{2015}$
- idée :
  - 2D : image
  - 3D : video = image + temps
- apprendre la temporalité grâce à la convolution 3D

# Réseau à convolution 3D (2)

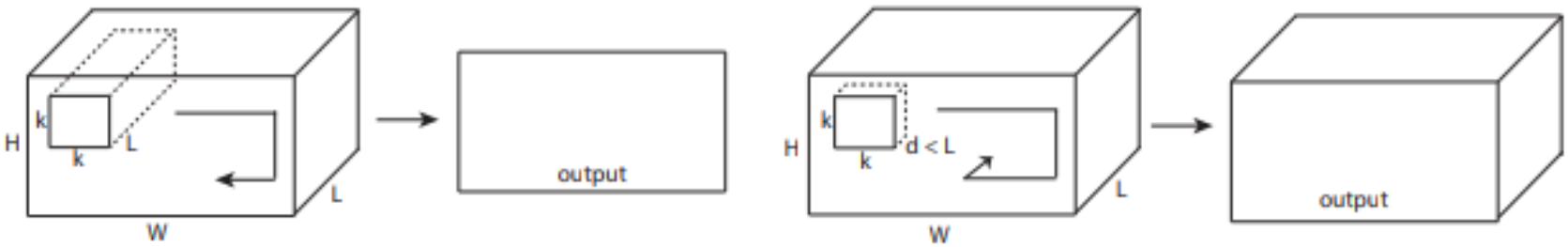



Figure 7: Convolution 2D sur une séquence d'image (gauche), convolution 3D sur une séquence d'image (droite) label:c3d-idea

- convolution 2D : produit une image (2D) => perte de l'info temporelle
- convolution 3D : produit une réprésentation 3D => garde l'info temporelle

# Réseau à convolution 3D (3)

# Objectif

- implémenter l'approche par convolution 3D
- comparer avec l'approche par détection de contours

# Référence

# Table des figures

ref:arch-lstm Ng, J. Y., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G., Beyond short snippets: deep networks for video classification, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (), (2015). http://dx.doi.org/10.1109/cvpr.2015.7299101. Figure 3

ref:convolution Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning (2016), : MIT Press. Chapitre 9. Figure 9.1

ref:pooling Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning (2016), : MIT Press. Chapitre 9. Figure 9.9

ref:opt-flow Simonyan, K., & Zisserman, A., Two-stream convolutional networks for action recognition in videos, CoRR, abs/1406.2199(), (2014). Figure 2

ref:two-stream-res Simonyan, K., & Zisserman, A., Two-stream convolutional networks for action recognition in videos, CoRR, abs/1406.2199(), (2014). Table 4

# Articles (1/4)


- Weiss, Y., Torralba, A., & Fergus, R., Spectral Hashing, In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), Advances in Neural Information Processing Systems 21 (pp. 1753–1760) (2009). : Curran Associates, Inc.
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G., Orb: an efficient alternative to sift or surf, 2011 International Conference on Computer Vision, (), (2011). http://dx.doi.org/10.1109/iccv.2011.6126544
- Abd-Almageed, W., Online, simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing, 2008 15th IEEE International Conference on Image Processing, (), (2008). http://dx.doi.org/10.1109/icip.2008.4712476
- Raventós, A., Quijada, R., Torres, L., & Tarrés, F., Automatic summarization of soccer highlights using audio-visual descriptors, SpringerPlus, 4(1), (2015).

# Articles (2/4)


- Duan, L., Xu, M., Tian, Q., & Xu, C., Mean shift based video segment representation and applications to replay detection, 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, (), (2004). http://dx.doi.org/10.1109/icassp.2004.1327209
- Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning (2016), : MIT Press.
- Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M., Learning spatiotemporal features with 3d convolutional networks, 2015 IEEE International Conference on Computer Vision (ICCV), (), (2015). http://dx.doi.org/10.1109/iccv.2015.510
- Simonyan, K., & Zisserman, A., Two-stream convolutional networks for action recognition in videos, CoRR, abs/1406.2199(), (2014).

# Articles (3/4)


- Farabet, C., Couprie, C., Najman, L., & LeCun, Y., Learning hierarchical features for scene labeling, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1915–1929 (2013). http://dx.doi.org/10.1109/tpami.2012.231
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P., Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86(11), 2278–2324 (1998). http://dx.doi.org/10.1109/5.726791
- Ng, J. Y., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G., Beyond short snippets: deep networks for video classification, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (), (2015). http://dx.doi.org/10.1109/cvpr.2015.7299101
- Pan, H., Li, B., & Sezan, , Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions, IIEEE International Conference on Acoustics Speech and Signal

# Articles (4/4)


- Chu, W., Song, Y., & Jaimes, A., Video co-summarization: video summarization by visual co-occurrence, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (), (2015). http://dx.doi.org/10.1109/cvpr.2015.7298981
- Javed, A., Irtaza, A., Khaliq, Y., Malik, H., & Mahmood, M. T., Replay and key-events detection for sports video summarization using confined elliptical local ternary patterns and extreme learning machine, Applied Intelligence, 49(8), 2899–2917 (2019). http://dx.doi.org/10.1007/s10489-019-01410-x
- Xu, W., & Yi, Y., A robust replay detection algorithm for soccer video, IEEE Signal Processing Letters, 18(9), 509–512 (2011). http://dx.doi.org/10.1109/lsp.2011.2161287


bibliography:summary.bib