M2CAI WORKFLOW CHALLENGE 2016

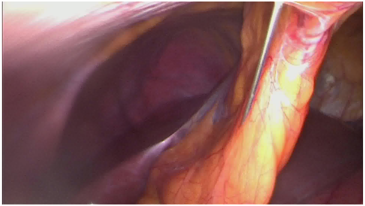
Fine tuning CNN with temporal smoothing and HMM for video frames classification

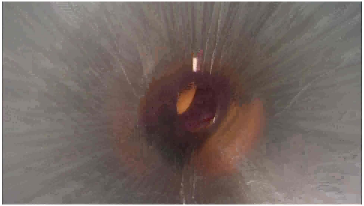
21th October 2016

Rémi Cadène, Thomas Robert, Nicolas Thome, Matthieu Cord

University Pierre and Marie Curie - LIP6 - MLIA

M2CAI Workflow Dataset





Endoscopic videos, resolution of 1920 x 1080, shot at 25 frames per second at the IRCAD research center in Strasbourg, France.

- 27 training videos ranging from 15mn to 1hour
- 15 testing videos

M2CAI Workflow Dataset

1 of 8 classes for each frames :

- TrocarPlacement
- Preparation
- CalotTriangleDissection
- ClippingCutting
- GallbladderDissection
- GallbladderPackaging
- CleaningCoagulation
- GallbladderRetraction

M2CAI Workflow Goal and Measure

Goal : Multi-class classification

- Online prediction : $P(y_i|x_i, x_{i-1}, x_{i-2}, ...)$
  x_i := frame i, and y_i := class of frame i

Algorithm useful to

- Monitor surgeons
- Trigger automatic actions

Evaluation metrics

- **Jaccard index** : $J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|}$
- Accuracy top1 : nb frames well classified / nb total frames

Two fold approach

1. Frames classifier using Deep Learning (Lib : Torch7)

- From Scratch Convolutional Neural Network (CNN) [1]
- Features Extraction CNN [1]
- Fine tuning CNN [1]

2. Smoothing predictions (Lib : Scikit-Learn)

1. Averaging predictions over last 15 frames
2. Hidden Markov Model (HMM) as a "temporal denoizer"

1. Optimized with Adam [Kingma et Ba 2014]

Creating a trainset and valset of images

Creating validation set by random split

- Training set : 22 videos
- Validation set : 5 videos $\{2, 9, 10, 13, 27\}$

Extracting one frame every 25 frames (1 frame per second)

- Training set : 59,493 images
- Validation set : 8,062 images
- Testing set : 28,732 images

Training CNN From Scratch

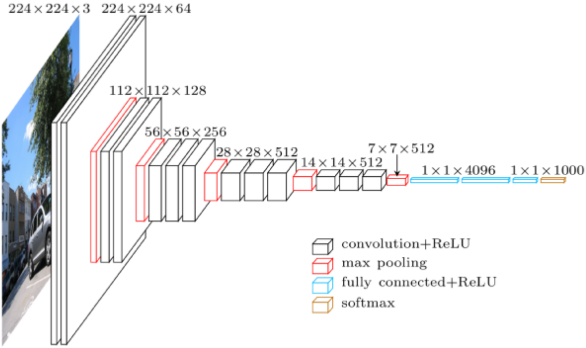
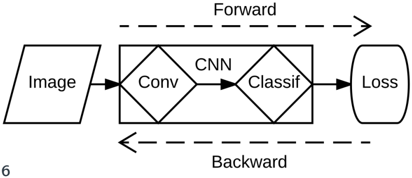
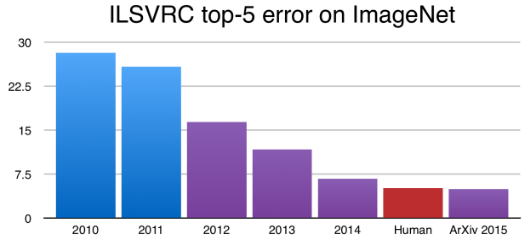


Figure 2: Vgg16 [Karen Simonyan et Zisserman 2014], top2 ILSVRC2014



ImageNet : 1.2M training images, 1000 classes

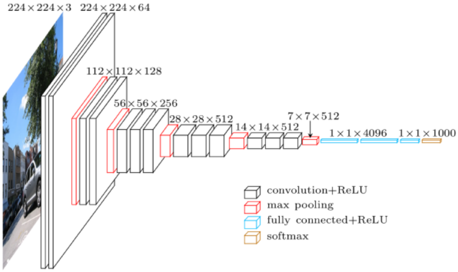


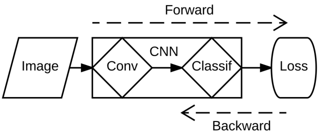
Using representations learned on ImageNet

UPMC SORBONNE UNIVERSITÉS

Pre-trained CNN as Features Extractor

1. Extracting features somewhere
2. Training a Support Vector Machine

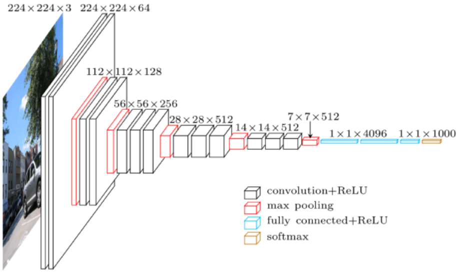


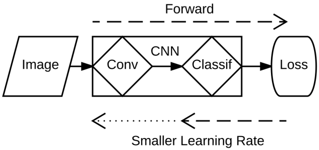


Adapting representations learned on Imagenet

Fine tuning a pre-trained CNN

- Same process than CNN From Scratch
- But smaller learning rate for pre-trained layers





Which CNN to use ? Possible in production ?

Model	Input	Param.	Depth	Implem.	Forward (ms)	Backward (ms)
Vgg16	224	138M	16	GPU	185.29	437.89
InceptionV3 [2]	399	24M	42	GPU	**102.21**	311.94
ResNet-200 [3]	224	65M	200	GPU	273.85	687.48
InceptionV3	399	24M	42	CPU	19918.82	23010.15

Table 1: Forward+Backward with batches of 20 images.

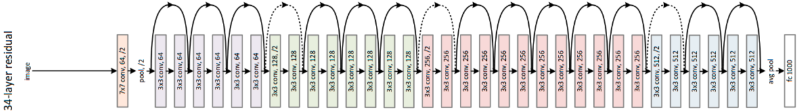
Possible in production thanks to GPUs !

2. [Szegedy et al. 2015]
3. [He et al. 2016]

Comparison of frames classifiers

Model	Type	Accuracy (%)
InceptionV3	Extraction (repres. of ImageNet)	60.53
InceptionV3	From Scratch (repres. of M2CAI)	69.13
InceptionV3	Fine-tuning (both representations)	79.06
ResNet200	**Fine-tuning (both representations)**	**79.24**

Table 2: Accuracy on the validation set.

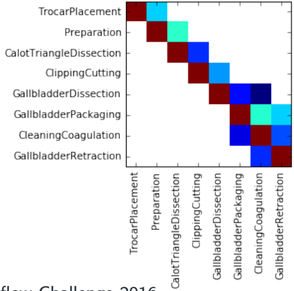


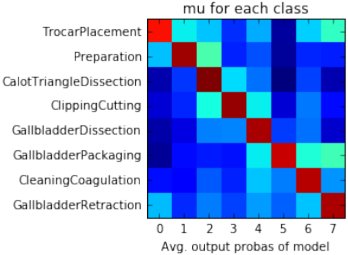
Gaussian Hidden Markov Model

UPMC SORBONNE UNIVERSITÉS

HMM on the smoothed predictions over last 15 frames

- Initial state probabilities :
- Matrix of probabilities of transition between states
- Gaussian parameters for emissions of observations :
  $\rightarrow$ mean and co-variance matrix





Gaussian Hidden Markov Model

Training process

- Counting using the training set annotations
- Counting using the predictions

Testing process

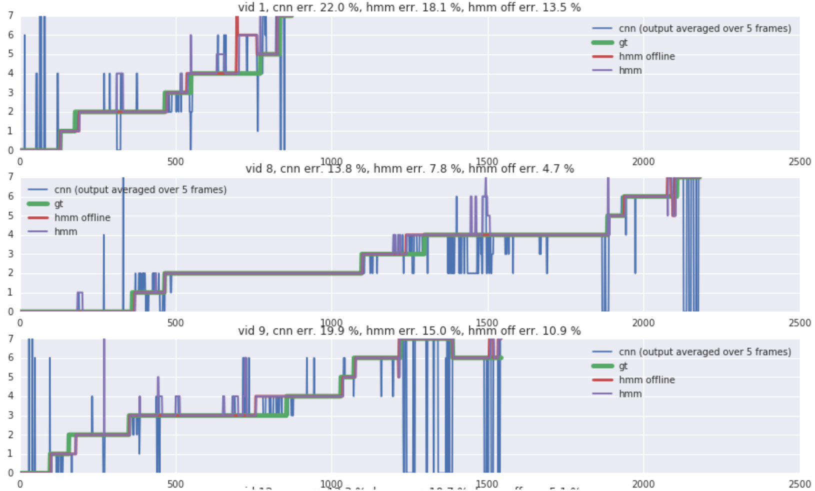
- Offline testing : Viterbi algorithm to obtain the most likely sequence of states
- **Online testing** : to predict x_t we apply Viterbi on the sequence $y_1, \ldots, y_t$

Comparison of temporal smoothing methods

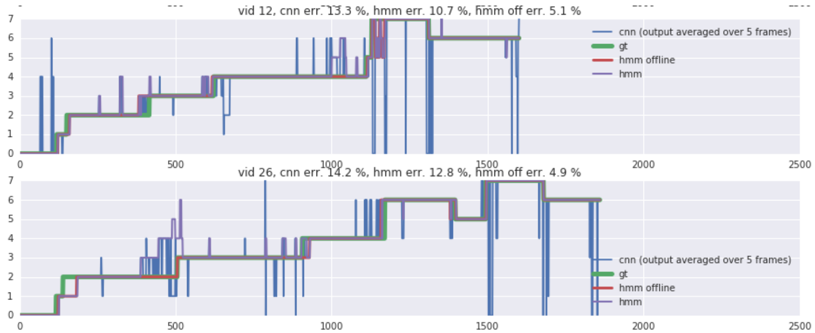
Temporal Method	Accuracy Val (%)	Jaccard Val	Jaccard Test
No Smoothing	79.24	–	–
Avg Smoothing	85.97	74.67	–
Avg + HMM Online	**88.90**	**81.60**	**71.9**
Avg + HMM Offline	93.47	87.59	–

Table 3: With the predictions of our fine tuned ResNet-200

Visualization



Visualization



Conclusion

Conclusion

- Deep Learning efficient even without medical knowledge
- Fine Tuning most accurate approach
- HMM is usefull to smooth the predictions

Future work

- Fine tuning CNN on full trainset (not only 80%)
- Ensembling several fine tuned CNNs

Code available : `github.com/Cadene/torchnet-m2caiworkflow`

References I

He, Kaiming et al. (2016). "Identity Mappings in Deep Residual Networks". In : *ECCV*.

Karen Simonyan et Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In : *ICLR*.

Kingma, Diederik P. et Jimmy Ba (2014). "Adam : A Method for Stochastic Optimization". In : *ICLR*.

Mishkin, D. et J. Matas (2015). "All you need is a good init". In : *arXiv preprint arXiv :1511.06422*.

Remi Cadene, Nicolas Thome et Matthieu Cord (2016). "Master's Thesis : Deep Learning for Visual Recognition". In : *arXiv preprint arXiv :1610.05567*.

Szegedy, Christian et al. (2015). "Rethinking the inception architecture for computer vision". In : *CVPR*.