# M2CAI WORKFLOW CHALLENGE 2016

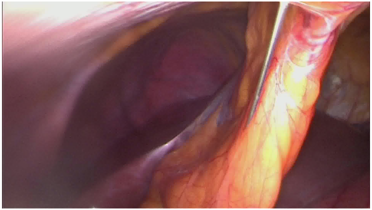
## Fine tuning CNN with HMM smoothing

21th October 2016

Rémi Cadène, Thomas Robert, Nicolas Thome, Matthieu Cord

University Pierre and Marie Curie - LIP6 - MLIA

# M2CAI Workflow Dataset



Videos resolution is 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 test videos

# M2CAI Workflow Dataset

1 of 8 classes for each frames :

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

# M2CAI Workflow Goal and Measure

Online prediction : $P(y|x_i, x_{i-1}, x_{i-2}, ...)$

$x_i$ := frame $i$, and $y$ := classes

Useful to :

- monitor surgeons
- trigger automatic actions

Measures : - Jaccard similarity coefficient :
$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$

- Accuracy top1 : nb frames well classified / nb total frames

# Two fold approach

## 1. Model to classify frames as images

- Extract features from pre-trained CNN
- CNN From Scratch
- Fine tuning pre-trained CNN

## 2. Smoothing predictions of our frames classifier

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

# 1. Creating validation set and extracting images

UPMC SORBONNE UNIVERSITÉS

Spliting randomly the full training set of 27 videos

- 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

# 2. Training a frame classifier

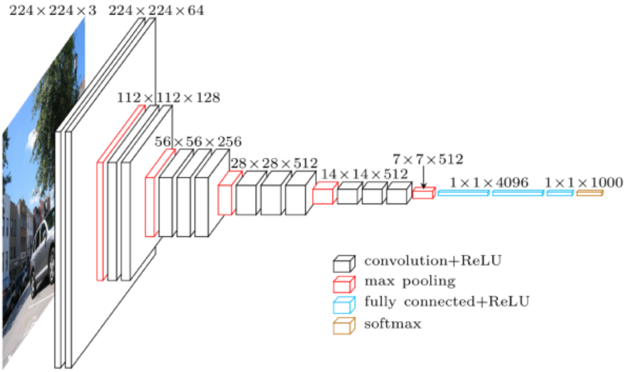
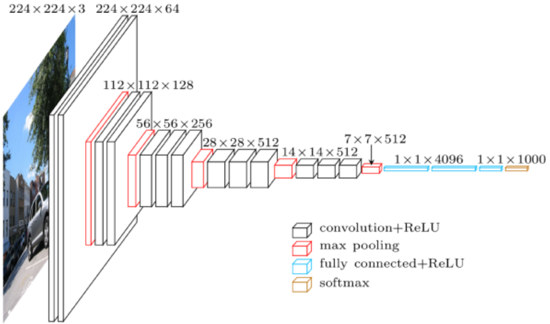



Figure 1 – Vgg16 [**simonyan2014very**], top2 ILSVRC2014

# 2. Training a frame classifier

## Pre-trained CNN as Features Extractor

- Remove last layer, Add new layer output size 8, Train with SGD fixing the pre-trained layers
- Extract features somewhere, Train a SVM

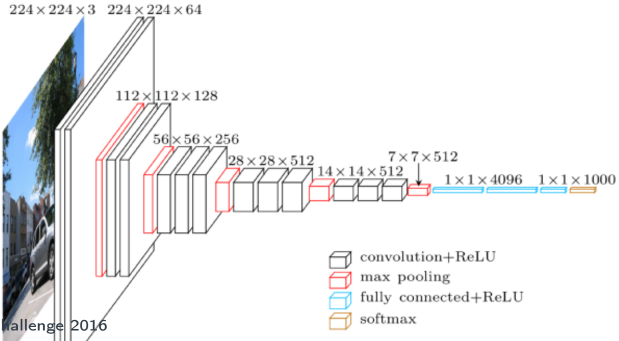


# 2. Training a frame classifier

UPMC SORBONNE UNIVERSITÉS

## Training a CNN From Scratch

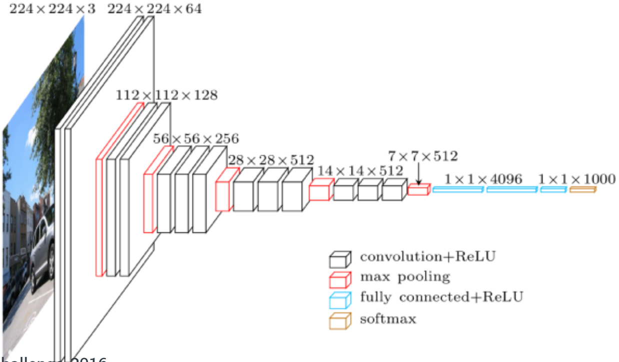
- Design specific CNN architecture
- Reinitialized architecture designed for large datasets with strong regularization



# 2. Training a frame classifier

## Fine tuning a pre-trained CNN

- SGD : lr, lrd, ftfactor
- Adam : lr, lrd



# 2. Training a frame classifier

| Model | Input | Param. | Depth | Implem. | Time (ms) |
|---|---|---|---|---|---|
| Vgg16 | 224 | 138M | 16 | GPU | |
| InceptionV3 | 399 | 24M | 42 | GPU | **0** |
| ResNet-200 | 224 | | 200 | GPU | |
| InceptionV3 | 399 | 24M | 42 | CPU | 0 |

Table 1 – Forward+Backward with batches of 30 images.

# 3. Smoothing the predictions

Averaging the predictions across the last 15 frames (15 seconds)

Hidden Markov Model on the predictions 3 kind of parameters the initial state probabilities the matrix of probabilities of transition between states the emissions of observations

# 3. Smoothing the predictions

HMM has

Training : counting

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 frames classifiers

| Classification Model | Accuracy (%) |
| --- | --- |
| InceptionV3 Extraction | 60.53 |
| InceptionV3 From Scratch | 69.13 |
| InceptionV3 Weldon | 78.18 |
| InceptionV3 Fine-tuned | 79.06 |
| ResNet200 Fine-tuned | 79.24 |

Table 2 – Accuracy on the validation set.

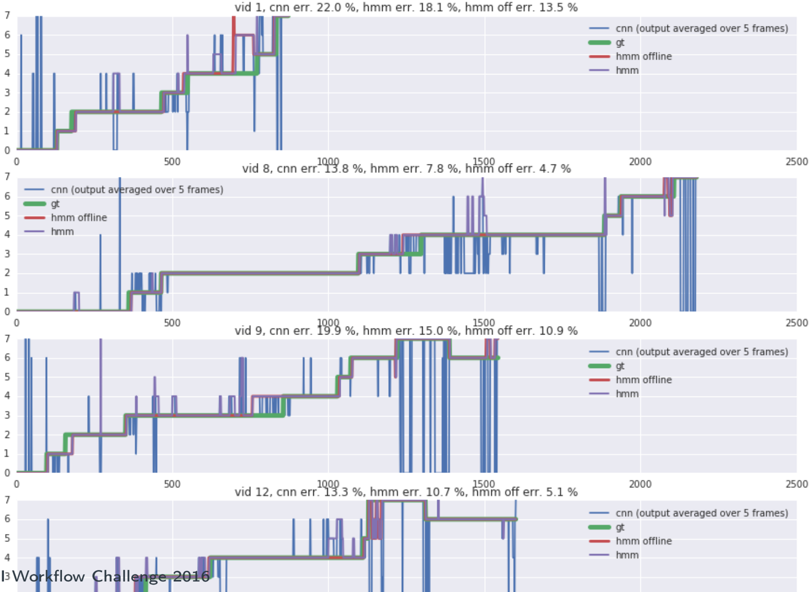
# Comparison of temporal smoothing methods

UPMC SORBONNE UNIVERSITÉS

| Temporal Method | Accuracy (%) | Jaccard |
|---|---|---|
| Avg Smoothing | $85.97 \pm 3.75$ | $74.67 \pm 7.87$ |
| HMM Online | $88.90 \pm 3.55$ | $81.60 \pm 10.49$ |
| HMM Offline | $93.47 \pm 3.59$ | $87.59 \pm 6.97$ |

Table 3 – Accuracy Top1 and Jaccard score on the validation set. The variance is computed over all classes.

# Visualization



# Conclusion

## Conclusion

- Deep Learning efficient
- Fine Tuning most accurate approach
- HMM is usefull to smooth the predictions

## Future work

- train on 100%
- ensembling

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

# References I