Deep Learning: Solving the detection problem

M2R ISI in Paris-Dauphine University - Master thesis defense
Master internship at Image & Pervasive Access Lab (IPAL)

Anne MORVAN

14 septembre 2015

Plan

1. Introduction
2. Propositions
   - Building a large dataset
   - Sampling algorithm
   - Training a CNN-based classifier
3. Results
   - Sampling algorithm
   - Training a CNN-based classifier
4. Conclusion

Outline

What is the detection problem ?

Pipeline

- to know how classifiy an image into the classes pedestrian presence or not : **CLASSIFICATION TASK**
- to know the localization within the image and even the number of instances

Issues

- DEEP learning : lack of data, overfitting, computation time

Outline

Collecting the data (1/2)

N	dataset	source	format	colour	size	nb frames	nb annots
1	Daimler	video	PNG	no	640x480	21790	56484
2	ETH	video	PNG	RGB	640x480	1804	14166
3	INRIA	holidays photos	PNG	RGB	varies	2120	1826
4	TudBrussels	video	PNG	RGB	640x480	508	1498
5	USA	video (ev. 30 frames)	PNG	RGB	640x480	8274	11504
6	MSCOCO	photos	JPG	RGB	≈ 578x484	123287	269886
7	PETA	crops	varies	RGB	≈ 72x170	19000	19000
8	CBCL	closed photos	JPG	RGB	1280x960	3547	1449
9	CVC	closed photos	PNG	RGB	640x480	593	2008
					Total	**57636**	**377821**

N	dataset	av. per frame	Person	Ignore	People	Person?	Person-fa	**Total**
1	Daimler	2.59	14131	40925	1428	0	0	**56484**
2	ETH	7.85	14166	0	0	0	0	**14166**
3	INRIA	0.86	1826	0	0	0	0	**1826**
4	TudBrussels	2.95	1498	0	0	0	0	**1498**
5	USA	1.39	9479	0	1636	258	131	**11504**
	USA train	-	*5080*	*0*	*1152*	*92*	*41*	*6365*
	USA test	-	*4399*	*0*	*484*	*166*	*90*	*5139*
6	MSCOCO	2.18	269886	0	0	0	0	**269886**
7	PETA	1	19000	0	0	0	0	**0**
8	CBCL	0.40	1449	0	0	0	0	**1449**
9	CVC	3.39	2008	0	0	0	0	**2008**
		Total	**333443**	**40925**	**3064**	**258**	**131**	**358821**

N	dataset	mean w	mean h	std w	std h	mean w/h
1	Daimler	27	54	20	40	0.51
2	ETH	51	101	32	63	0.50
3	INRIA	119	289	61	148	0.41
4	TudBrussels	28	74	14	37	0.39
5	USA	24	59	19	46	0.43
6	MSCOCO	82	133	98	131	0.64
7	PETA	72	170	22	55	0.43
8	CBCL	89	195	67	117	0.46
9	CVC	54	142	29	70	0.38

Collecting the data (2/2)

Daimler, ETH, INRIA, TudBrussels, USA, MSCOCO, PETA, CBCL, CVC

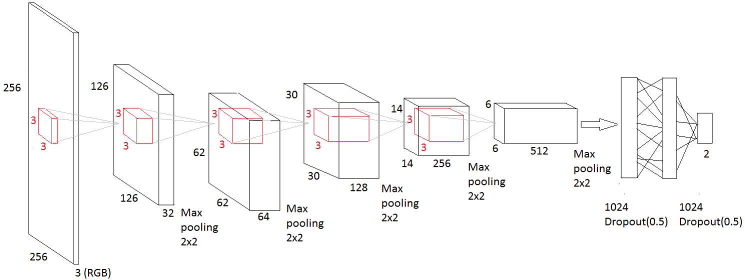
Sampling algorithm

Algorithm 1 Sampling algorithm

```
isPositive ← 0 or 1
dataset ← rand_dataset(isPositive)
if isPositive == 1 then
    class ← rand_class(dataset)
    annot ← rand_annot(class)
    while the constraints in parameters are not respected do
        annot ← rand_annot(class)
    end while
    bb ← get_bb(annot)
else
    while we can't build a neg bb respecting the constraints in parameters do
        frame ← rand_frame()
        if frame contains no positive annotations then
            bb ← create_bb(frame)
        else
            bb_list ← getBB_in_frame(frame)
            bb ← create_bb(frame, bb_list)
        end if
    end while
end if
return bb
```

Training a CNN-based classifier (1/2)

Model architecture



- 5 convolutional layers
- 2 fully-connected layers

Cost function : **Cross entropy loss**

$$E = \frac{-1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \left(p_{n_k} \; . \; log \; \hat{p_{n_k}} + (1 - p_{n_k}) \; . \; log \; (1 - \hat{p_{n_k}})\right)$$

Training a CNN-based classifier (2/2)

Pre-processing and data augmentation

- normalization with mean and std pixel values for each channel, for each dataset
- mirror (proba = 0.5)
- shift (max. 10% of height or width)
- rotation

Rad	0.26	0.13	0.07	0.03	0
Prob	0.1	0.1	0.2	0.3	0.3

- aspect ratio

Ratio	(1,1)	(2,2)	(1,2)	(2,1)
Prob	0.25	0.25	0.25	0.25

- hard negatives or bootstrapping (proba = 0.5 with 2001 samples)
- learning rate policy

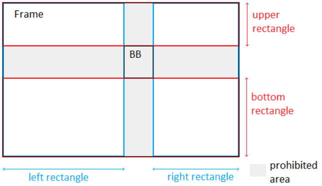
Outline

Sampling algorithm

Parameters for choosing the data

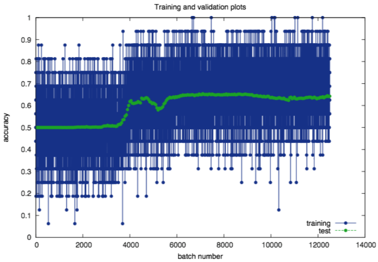
- training data : USA with proba. 1
- test data : USA with proba. 1
- positive crop proba. : 0.5
- positive classes : only person
- min. dimensions : $w = 5,\ h = 20$
- no constraints on the distance from the bounds or proportions of the w or h
- h and w : two dependent normal distributions
- bounding box : 4 rectangles method with $jaccard\_index = 0.1$

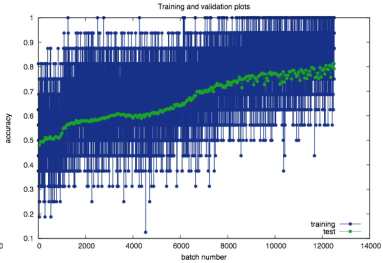
$$h \rightsquigarrow \mathcal{N}(\mu_h,\ \sigma_h^2)$$
$$w \rightsquigarrow \mathcal{N}(\mu_w, \sigma_w^2)$$
$$w|h \rightsquigarrow$$
$$\mathcal{N}(\mu_w + \frac{\sigma_{hw}}{\sigma_w}(h - \mu_h), \sigma_w - \frac{\sigma_{hw}^2}{\sigma_h})$$



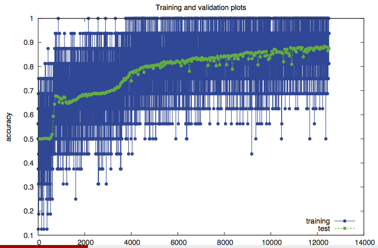
Influence of learning rate

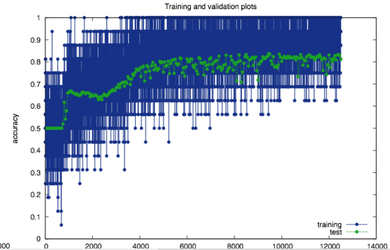
$lr = 0.01$ & $lr = 0.02$





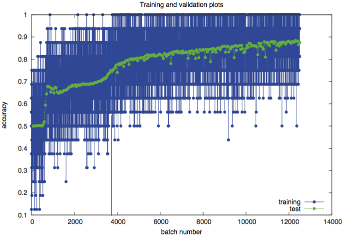
$lr = 0.03$ & $lr = 0.05$

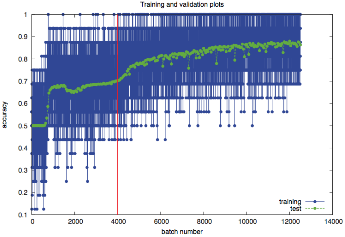




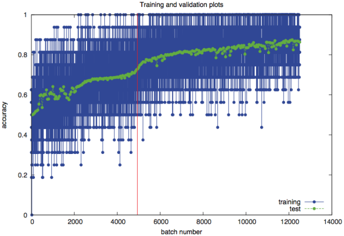
Role of data augmentation methods (1/2)

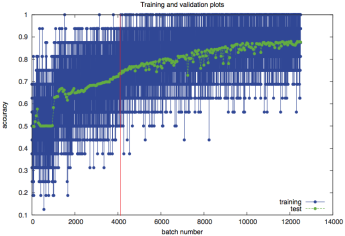
nothing mirror





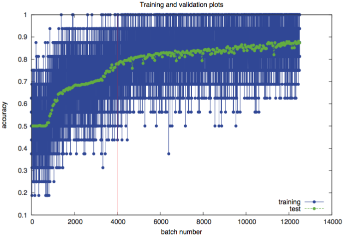
shift aspect ratio



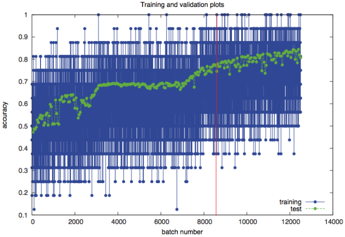


Role of data augmentation methods (2/2)

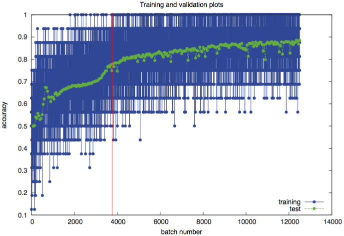
rotation



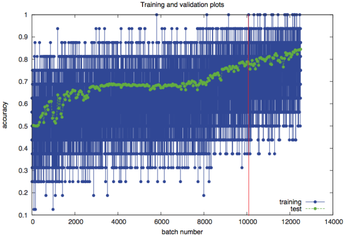
hard negatives



learning rate policy

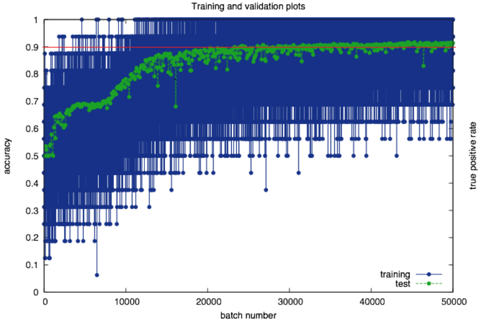


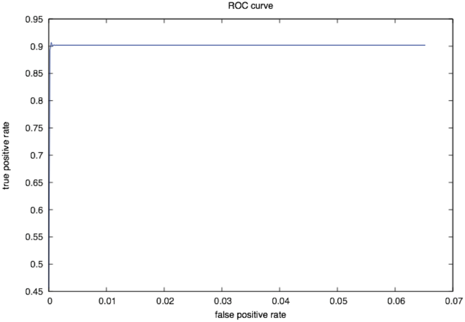
everything



Accuracy and ROC curve (1/2)

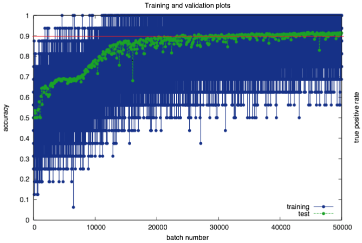
With all data augmentation methods + learning rate policy at the 2500-th batch

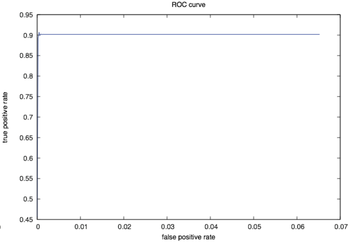




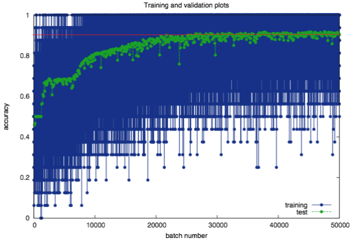
Fresh results

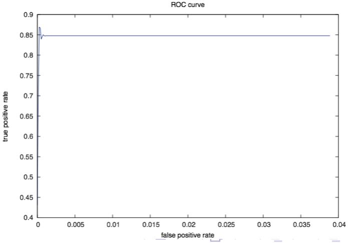
With all data augmentation methods + lr policy at the 2500-th batch





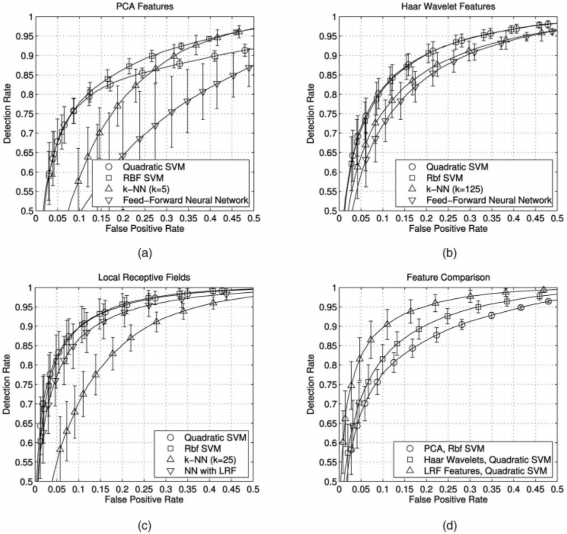
With all data augmentation methods + deformation ($\approx$ blurring)





Accuracy and ROC curve (2/2)

State of the art



"A comparison of different feature extraction and classification methods. Performance of different classifiers on
(a) PCA coefficients,
(b) Haar wavelets, and
(c) Local Receptive Field (LRF) features.
(d) A performance comparison of the best classifiers for each feature type."

Outline

Conclusion & further perspectives

Our work

- classification task (sampling algorithm + data augmentation methods + classifier)
- goal : 98% of well-classified images rate
- obtained : $\approx$ 93% for the validation by training with the USA training set and validating on the USA test set

Perspectives

- merge more and more datasets (KITTY ...)
- perform other data augmentation methods from the elastic transformations family (perspective distorsion transformations...)
- use synthetic images
- use temporal information + motion
- define part-based models