Pertinence of Video for Single Image Deep Network

Juliette Chataigner, Stephane Herbin, and Adrien Chan-Hon-Tong

***Abstract*—Using key frames instead of video to train single image deep neural networks make sense as successive images of one video contain almost the same information. However, we show that using all images can significantly increase performances of deep networks on medium size datasets. Considering, that annotating video can be done much more efficiently than annotating disparate images, we argue that using complete videos should be considered where data are naturally collected this way which is often the case in robotic, autonomous driving, or aerial acquisitions.**

***Index Terms*—Deep learning, video, medium size dataset.**

I. Introduction

In robotic, video surveillance, autonomous driving or aerial acquisitions, collected data are naturally in form of video. Due to the large size of these data, it is natural to want to keep key frames only. Keeping key frames seems also interesting especially if the goal of the data is to train single image deep convolutionnal neural networks (CNN) because using all images is widely thought to not increase the performance of single image CNN. Indeed, successive images from a video are visually quite the same. As an example, CITYSCAPE, one of the largest autonomous driving datasets, provides only spaced images.

However, we argue that using all images of video could be interesting. First, using the temporal consistency of the video allows to propagate the expensive human annotation needed for learning. So, for human annotation, using key images or all images of a video costs almost the same. Then, we show, in this paper, that using all images of all videos provides significant performance increases on several medium size datasets and for several deep networks.

More precisely, to provide quantitative evaluation, we fix a use case. We focus on the relevancy of using all images of video datasets to train single image CNN for binary semantic segmentation. Semantic segmentation (see Fig. 1) aims to produce semantic mask of an image. In other words, it aims at deciding a semantic label for each pixel of an image. In our experiment, we will perform such segmentation task but with two classes only. However, binary segmentation should be considered as an example (we are planning to tackle other use cases like object detections).

The contribution of this paper is to present an experimental protocol to evaluate the relevancy of video datasets in this context.

Manuscript received Septembre 25, 2017; revised October 23, 2017.

The authors are with Onera the French Aerospace Lab, France (e-mail: adrien.chan_hon_tong@onera.fr).

We apply state of the art deep learning pipelines which are designed to process single images: each mask is completely estimated using one image only (beside training images will be randomly shuffled even if training set will contain successive images from a common video).

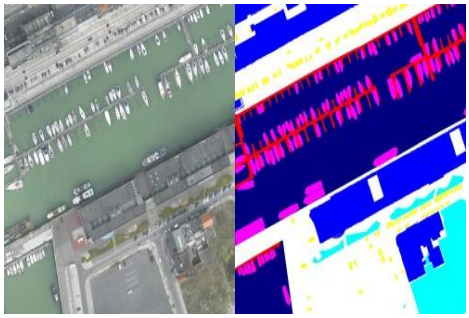

Fig. 1. Example of a semantic mask of an image.

For this purpose, we select (or create) several datasets consisting of very different annotated videos. Then, we perform 4 kinds of experiments to measure the performance of CNN trained with different training datasets extracted from the video. For example, we measure the performance of segmentation pipelines while increasing the number of successive images used from each video. We also perform these experiments with 2 different deep learning pipelines: CNN as feature extractor with support vector margin (SVM) as classifier, and, classical CNN training. Currently, both pipelines require specific hardware. SVM based experiment needs large hard disk to store model file (2To here) but are possible on any modern GPU card. CNN experiments are more scalable but required latest and more expensive GPU (typically a Titan P).

The structure of this paper is the following. In next section, we describe related works. Then, Sections III and IV describe experiments and performances: Section III for SVM based experiments and Section IV for CNN ones. Finally, Section V presents the conclusion and perspectives.

II. Related Works

A. Semantic Segmentation

Semantic segmentation is a growing paradigm [1]. However, comparing to classification, detection or tracking, annotating is even more expensive in semantic segmentation. For applications linked with a large economic market, it is possible and relevant to collect and annotate very large datasets (CITYSCAPE is an example of large datasets designed for autonomous cars – a market friendly application). However, the situation is very different for most

doi: 10.18178/ijmlc.2017.7.6.653

computer vision applications like medical images, aerial images, video surveillance or robotic applications. For example, the 2015 IEEE data fusion contest contains less than 400 car instances which limits the stability of algorithm evaluation.

Currently, most computer vision applications still deal with medium size datasets (at least for the annotated data). If we discard unsupervised algorithms which have not yet reached a sufficient efficiency, and algorithms which are able to learn from few data (which are at the opposite of current trends), there are only one possibility to increase performances for these applications: increase annotation productivity.

B. Semi Automatic Annotation

As, annotating large annotated datasets is a very expensive human process, there are research to increase annotation productivity. It includes active learning where computer vision proposes best next thing to annotate [2], crow engineering [3] and, optimization of human time [4]. Crow engineering does not really increase productivity, it allows to share the work. However, by sharing the work, it introduces annotation errors, and thus, needs to use validation procedures to check annotation consistency between several user.

Thus, crow engineering has made possible the creation of very large datasets but does not solve the human cost problem (it even worsen the problem by needing multi stage validation). So crow engineering does not seem to be the solution for medium budget computer vision applications.

Optimization of the human time and active learning are relevant when sufficiently good algorithms are already available. [4] just moves from a sparse annotation to a dense annotation by using deep network trained on the same kind of datasets. Also, as learning algorithm are more and more complex, it is not trivial to perform an efficient online interactive semi automatic annotation system. To our knowledge, there is no widely used semi automatic annotation system whose efficiency has been successfully evaluated.

C. Video Annotation

But, there exist a case where computer vision algorithm may provide a productivity gap for annotation: video. The interest of video is that information can be propagated (by tracking or optical flow [5]) from one frame to the next. This allows to save costly human annotation when the propagation behaves correctly. This propagation uses low level clue, which would not be sufficient if not helped by the temporal consistency.

However, video datasets are not large in the same sense than image datasets: they are large but correlated. This is exactly why we focus on this paper on the question of the relevancy of training single image CNN from correlated images of one video i.e. even if two successive frames of a video are quite the same.

For the purpose of experiments, we develop tracking based annotation tools. These tools are close to [3] but uses off the shelf trackers (e.g. opencv implementation of dsst tracker) to propagate annotations from one frame to the other. As we take binary semantic segmentation, this tracking tool is sufficient for propagating a coarse semantic map. With this tool, human correction is only needed when tracker fails to be sufficiently accurate.

So to summarize, to apply semantic segmentation state of the art deep networks on medium budget applications (like most robotic applications), we need to find a way to increase the annotation productivity to produce larger datasets. One possible way is the use videos. But it raises the question of the relevancy of correlated data to train single image deep networks. In the next two sections, we show that deep networks indeed take advantage of these kind of correlated data: using more images than key frame images from videos provides significant performance increases.

III. CNN+SVM on Video Datasets

A. Datasets

There is a lot of semantic segmentation datasets. But most of them (MSCOCO, IEEE data fusion contest, ...) are images datasets. They are thus irrelevant to focus on video datasets. Again, there exists very large datasets like CITYSCAPE (which it is composed of 25000 HD frames pixelwise annotated into 30 semantic classes).

However, this dataset has already been purged to keep only spaced image. So it can not be used to perform a successive vs spaced comparison.

Thus, to tackle this paper issues, we rely on MOT 2016 (MOT16) [6], VIRAT aerial dataset [7] (not video surveillance data - we use only the aerial videos). As no annotation has been released for VIRAT aerial, we annotate it ourselves using our annotation tools.

MOT16 is a multi objects tracking dataset: detection are provided, the goal is to keep temporal id on the detection. Here, we use this dataset only to learn to produce a semantic mask corresponding to the detections (we only keep person detection). We convert the detection ground truth into a semantic segmentation ground truth by considering that all pixels in a detection are from class 1 and all pixels outside detections are from class -1. We do some experiment to check if applying graphcut to refine mask was relevant. Currently, we find it is not critical for deep learning. Thus, these experiment uses not refined masks. The training part of MOT16 is composed of 6 HD videos taken from a pedestrian or car or surveillance camera. For MOT16, we will use accuracy, gscore and iou score to evaluate all algorithms. Accuracy is the stablest measure as both gscore and iou have non linear behavior especially if the threshold between precision and recall is poorly adjusted.

The VIRAT aerial dataset is a set of videos which are low resolution and contains camera motions, highly textured background and very small object. As no public annotation has been released for this dataset, we annotated a small subset of the data in a person detection setting (see https://github.com/achanhon/VIRAT-AERIAL-ANNOTATION). We convert the ground truth in the same way than for MOT16. In order to provide a diversity of situations, we chose to annotate about thirty sub videos of 400 frames containing at least one person distributed over the dataset (but discarded infrared images). For VIRAT, we will use only

gscore and IoU score to evaluate all algorithms. Accuracy is not relevant as 99,2\% of the pixel are background pixel.

B. Global Pipeline

In this section, we use a imagenet CNN to extract feature map for each training image. Then, we train SVM for semantic segmentation (see Fig. 2).

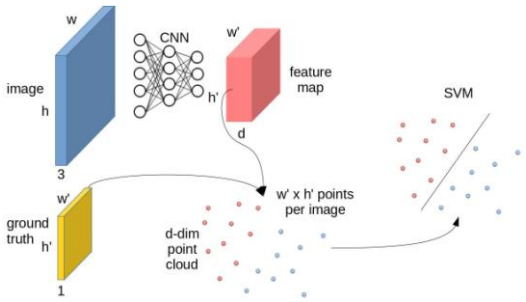

Fig. 2. global pipeline for CNN+SVM experiments.

We rely on VGG16 [8]: we forward each image into VGG16 (pretrained on imagenet), extract several layers (conv12, conv22, conv33, conv43 and pool5). We resize all extracted layers to the ground truth size, obtaining a feature map in which each pixel is described per a vector.

We learn a SVM with each pixel of the feature map being a training point. We allow the ground truth to be eventually smaller than the original image when spatial accuracy of the human annotation is not relevant enough.

This leads to very very large SVM model: if we have **n** training images with ground truth size **w'**x**h'**, then the SVM see **n**x**w'**x**h'** points. With **n**=100, **w'**=320 and **h'**=240, there are yet 7680000 points for the SVM. Each point will have a dimension corresponding to the number of neurons in the extracted layers (typically if we use only pool5 it will be 512 but if we merge several layers in a UNET [1] fashion this lead to much more large dimension).

We learn a SVM with liblinear or liblinear block when we reach the RAM limit (8Go). We also learn a SVM with a simple stochastic gradient descent (SGD) for comparison.

C. Different Types of Experiments

We perform 4 kinds of experiments to evaluate the impact of video datasets:

- **1 vs 20**: we compare the performance of a CNN trained with 1 image per video with the same CNN trained with 20 successive images per video
- **1by20 vs 400**: we also compare the performance of a CNN trained with the images {0,20,40,...,380} of each videos with the same CNN trained with 400 successive images per video (notice that one can concatenate experiment 1 and 2 to get the evolution of the performance from 1 image per video to 400)
- **1x20 vs 20**: we compare the performance of a CNN trained with 20 successive images per video with the same CNN trained with 20 randomly noised images computed from the image 1 - so the number of image is the same but in one hand it is real successive images and on the other hand randomly generated one
- **20x20 vs 400**: Finally, we do the same with images 1 to 400 versus 20 images generated from each image {0,20,40,...,380}

Experiments of types 1vs20 and 1by20vs400 aim to measure the performance improvement while increasing the number of correlated images.

The global idea of the synthetic experiments (1x20vs20 and 20x20vs400) is to remove any possible size bias: both training data contain the same number of images. But, in one side, there are images generated from a small set by adding different Gaussian noise and on the other side this is just 20 successive images.

Subsection III.D will deal about the two first set of experiments, III.E will deal about the last two. In all experiment, the name of the setting is formed by the frame setting (e.g. 1 frame or 400 or 1by20) and the pipeline (e.g. liblinear or sgd).

D. Performances VS Frames Per Video

Performances of CNN+SVM experiments of type 1 vs 20 or 1by20 vs 400 are reported in Tables I, II, III and IV.

TABLE I: 1 VS 20 FOR CNN+SVM ON MOT16

setting	*accuracy*	*iou*	*gscore*
1 liblinear	90	33	29
20 liblinear	91	36	31
1 sgd	88	12	10
20 sgd	89	38	30

TABLE II: 1BY20 VS 400 FOR CNN+SVM ON MOT16

setting	*accuracy*	*iou*	*gscore*
1by20 liblinear	87	40	34
400 liblinear	89	45	39
1by20 sgd	87	35	27
400 sgd	89	48	33

TABLE III: 1 VS 20 FOR CNN+SVM ON VIRAT

setting	*iou*	*gscore*
1 liblinear	17	8
20 liblinear	12	6
1 sgd	13	9
20 sgd	11	6

TABLE IV: 1BY20 VS 400 FOR CNN+SVM ON VIRAT

setting	*iou*	*gscore*
1by20 liblinear	12	4
400 liblinear	20	10
1by20 sgd	11	6
400 sgd	21	9

These results are interesting especially on MOT because performances exhibit clear trends. Thus, we can state that using video dataset significantly (even if not largely) increases performances from 1 to 20 and from 1 by 20 to 400.

On VIRAT, scores largely increases between 1 to 400. However, performances decreases in 1vs20. However, scores on VIRAT are low especially the gscore. When checking this decrease, we find that it is due to a worse balance between positive and negative in 20 than in 1. However, we compute a biased accuracy (accuracy when removing easy negatives) and we find that this biased accuracy increases by more than 3% from 1 to 20. So results on VIRAT are less clear but score still increase from 1 to 400.

These experiments shows that, contrary to what could have been thought, using all the videos instead of key frames can increase performance of single image deep learning pipeline (here for binary segmentation).

E. Successive Images Versus Data Augmentation

To implement experiments 1x20vs20 and 20x20vs400, we generate images by adding Gaussian noise to a real images. The variance of the noise is the same that the variance measured in the corresponding set of successive real images. Images generated are converted into 8 bits images to enter the CNN. As we want to make multiple runs of the training to average random noise effect, we perform this type of experiment with sgd only (and not with liblinear), because, the great advantage of sgd is that the noised images can be generated on the fly. However, we have see that sgd globally behaves like liblinear in Tables I-IV.

The results of these experiments are clear: real images outperforms synthetic ones (we even observe no average increases of performance between using only the raw images vs using 20 noised version of the images).

We acknowledge that other types of noise (or a noise not converted into 8 bits) could have been better, but still the result of this experiment show that the increase of performance observed in Tables I-IV is due to the use of successive images and is not trivial to obtain by another way.

IV. CNN on Video Dataset

A. Global Pipeline

In this section, we did a second time some of the experiments described in section III but in a straightforward deep learning fashion.

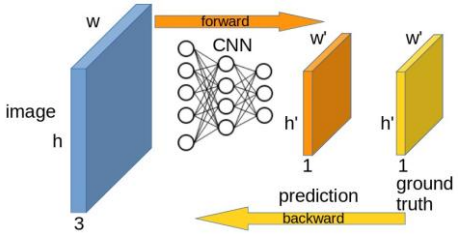

Fig. 3. Global pipeline for CNN experiments.

The networks are designed to directly produce an output with same shape as the ground truth. Back propagation is directly performed by computing the gradient corresponding to the loss between the ground truth and the network output (the loss is the average on all the pixels of the cross entropy - learning is done by stochastic gradient descent).

Training is done with NVIDIA DIGITS tools (we work on random crop of size 256x256 at training time). We evaluated different state of the art networks: VGG [8] (the same as the svm experiment) and UNET [1], and a deeper UNET (we add a level to the UNET structure). All tested networks are based on VGG and are finetuned from VGG weight on imagenet.

In order to strengthen the interest for video, we constraint the training to have close durations. Typically, even if we use 400 images instead of 1 from each video, we do not let the training to run 400 more times. More precisely, we divide the number of epochs by 4 when scaling by 20 the number of frames. No training takes more than a week.

B. Results

Finetuning experiments scale linearly with the data (compare to SVM which is super linear) but these experiments lock expensive (and thus shared) GPU hardware. So, it was not easy for us to replicate all SVM experiments. Instead, we evaluate the increase of performances while increasing the number of successive images between 1, 20 and 400 which correspond to a partial mix of experiments 1 vs 20 and 1by20 vs 400. Available results are presented in Tables V and VI.

TABLE V: CNN on MOT16

setting	*accuracy*	*iou*	*gscore*
1 vgg	84	8	3
20 vgg	85	13	6
1 unet	88	9	6
20 unet	88	12	8
400 unet	83	17	11
1 deep unet	87	0	0
20 deep unet	89	20	16
400 deep unet	91	31	35

TABLE VI: CNN on VIRAT

setting	*iou*	*gscore*
20 vgg	0	0
1 deep unet	11	9
400 deep unet	22	21

These results are globally similar to the ones of Tables I, II, III and IV.

Finetuning reaches comparable performance than CNN+SVM in the 400 images setting but not on all setting despite that CNN+SVM corresponds to a finetuning of the last layer only.

However, the most important point, in our opinion, is that these experiments confirm the trend observed in SVM ones: performance tends to largely increase when increasing the number of successive images used from the videos. This is especially clear for 1 DEEP UNET to 400 DEEP UNET on VIRAT with a gscore who jumps from 9\% to 21\%. This is also especially clear for deep UNET on MOT who jumps from 0\% to 31\% of gscore when using 400 images instead of 1 per video.Again, training times have not been allowed to increase by 400 even when data have

V. Conclusion

This paper focuses on successive images of video datasets for single image semantic segmentation. The question is about the relevancy of using video on which temporal information can be used to help human annotation to train single image semantic segmentation pipeline (especially deep learning ones). In our experiments, using these successive images of video increases performance of the pipelines whereas using data augmentation (to form a dataset with the exact same size) does not.

Of course, using uncorrelated images would be more efficient. But as using video is still useful, we argue that this should be considered when data are naturally collected as video. Because annotating video can be done efficiently, this could provide a nearly free increase of performances (free for human annotation time).

Seeing our contribution, we hope that people who plan to form datasets from video compatible devices will, at least, consider the possibility to keep all available data - both for

video pipelines but also for single image ones.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Springer International Publishing, 2015.

[2] A. Yao, J. Gall, C. Leistner, and L. V. Gool, "Interactive object detection," *Computer Vision and Pattern Recognition*, 2012.

[3] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, 2008.

[4] O. Russakovsky, L. J. Li *et al*., "Best of both worlds: Human-machine collaboration for object annotation," *Computer Vision and Pattern Recognition*, 2015.

[5] L. S. Lara, D. Q. Sun, V. Jampani, and M. JBlack, "Optical flow with semantic segmentation and localized layers," in *Porc. Conference on Computer Vision and Pattern Recognition*,2016.

[6] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, *MOT16: A Benchmark for Multi-Object Tracking*, 2016.

[7] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis *et al*., "A large-scale benchmark dataset for event recognition in surveillance video," in *Porc.Conference on Computer Vision and Pattern Recognition*, 2011.

[8] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, In CoRR, 2014.

Juliette Chataigner is a student of the Institut d'Optique Graduate School.

She has done a reseach intership at Onera in 2017 on deep learning pipeline for semantic segmentation of images.

Stéphane Herbin is a senior researcher of Onera.

He has received an engineering degree from the Ecole Supérieure d'Electricité (Supélec), the M.Sc. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign, and the Ph.D. degree in applied mathematics from the Ecole Normale Supérieure de Cachan. He was employed by ONERA since 2000, he works on computer vision. His current main research interests are explanable artificial intelligence (XAI) and few exemples learning (e.g. 0 shot learning).

Adrien Chan-Hon-Tong is a junior researcher of Onera.

He has received an engineering degree from the Ecole Polytechnique and a Ph.D. from Université Pierre et Marie Curie (UPMC).

He mainly works on object detections in aerial images or drone videos.