Object recognition and computer vision 2015

Jean Ponce, Ivan Laptev, Cordelia Schmid and Josef Sivic

Final projects

Description:

The final project amounts to 50% of the final grade. You will have the opportunity to choose your own research topic and to work on a method recently published at a top-quality computer vision conference (ECCV, ICCV, CVPR) or journal (IJCV, TPAMI). We also provide a list of interesting topics / papers below. If you would like to work on another topic (not from the list below), which you may have seen during the class or elsewhere, please consult the topic with the class instructors (I. Laptev and J. Sivic). You may work alone or in a group of 2-3 people. If working in a group, we expect a more substantial project, and an equal contribution from each student in the group.

Your task will be to:
(i) read and understand a research paper,
(ii) implement (a part of) the paper, and
(iii) perform qualitative/quantitative experimental evaluation.

Evaluation and due dates:

- **Project proposal (due on Nov 24).** You will submit a 1-page project proposal indicating (i) your chosen topic, (ii) the plan of work, i.e. what are you going to implement, what data you are going to use, what experiments you are going to do, (iii) if working in a group, who are the members of the group and how you plan to share the work. The project proposal will represent 10% of the final project grade.
- **Project report (due on Jan 11).** You will write a short report (<=3 pages) summarizing your work. The report will represent 70% of the final project grade.
- **Project presentation (on Jan 7 and Jan 8)**. You will present your work in the class. The project presentation will represent 20% of the final project grade.

Collaboration policy for final projects

You can discuss the final projects with other students in the class. Discussions are encouraged and are an essential component of the academic environment. You may work **alone or in a group of maximum of 2 people**. If working in a group, we expect a more substantial project, and an equal contribution from each student in the group. The final project report needs to explicitly specify the contribution of each student. Both students are expected to present the project at the oral presentation and contribute equally to writing the

report. *The final projects will be checked to contain original material. Any uncredited reuse of material (text, code, results) will be considered as plagiarism and will result in zero points for the assignment / final project. If a plagiarism is detected, the student will be reported to MVA. See below the policy on re-using other's people code.*

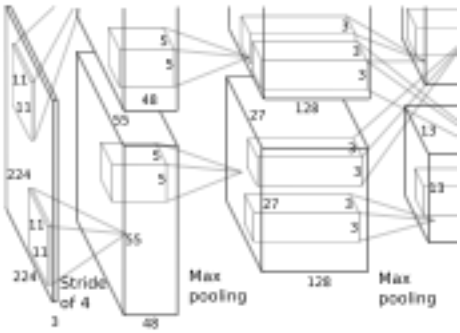
Re-using other's people code:

You can re-use other people's code. However, you should clearly indicate in your report/presentation, what is your own code and what was provided by others (don't forget to indicate the source). We expect projects balanced between implementation / experimental evaluation. For example, if you implement a difficult algorithm from scratch, only few qualitative experimental results may suffice. On the other hand, if you completely use someone else's implementation, we expect a strong quantitative experimental evaluation with analysis of the obtained results and comparison with baseline methods.

Suggested papers / topics:

Below are some suggested papers and topics for the final projects. If you would like to work on a different topic, please consult your choice with the course instructors (I. Laptev and J. Sivic).

Special 2015 theme: convolutional neural networks and deep learning



Description: Convolutional neural networks have emerged as a powerful image representation that can be learnt from large amount of labelled image data and transferred to different recognition tasks. Please see the recent computer vision (CVPR'15, ICCV'15) and machine learning conferences (ICLR'14, ICLR'15, NIPS'14, Deep learning workshop at NIPS'14) for papers related to this theme. A good introduction to deep learning for visual recognition is in this CVPR'14 tutorial and the two CVPR'15 tutorials (Torch and Caffe).

If you choose a paper from the theme, you will be expected to read and understand the paper (before the project proposals are due) and in your project proposal describe in detail the plan of work, i.e. what are you going to implement, what data you are going to use, what experiments you are going to do.

Topic A - A Neural Algorithm of Artistic Style

Papers:

[1] A Neural Algorithm of Artistic Style, L. Gatys, A. Ecker, M. Bethge, 2015

[1a] Texture Synthesis Using Convolutional Neural Networks, L. Gatys, A. Ecker, M. Bethge, to appear in NIPS 2015

Additional details: https://bethgelab.org/deepneuralart/

[2] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014
[3] Image Analogies, A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, D. Salesin. SIGGRAPH 2001.

Code / demos:

1. Torch implementation with links to several other implementations : http://gitxiv.com/posts/jG46ukGod8R7Rdtud/a-neural-algorithm-of-artistic-style
2. Another Torch implementaiton: https://github.com/jcjohnson/neural-style
   Variant of above applied to video: https://github.com/mbartoli/neural-animation
3. Torch: https://github.com/kaishengtai/neuralart
4. Torch: https://www.terminal.com/snapshot/054f5a4d8d576779ee4b8cd77718e250027955205e7db80dc2cd3f1b7add13cd
5. Python (using DeepPy): https://github.com/andersbll/neural_artistic_style
6. iPython (using Theano): https://github.com/Lasagne/Recipes/blob/master/examples/styletransfer/Art%20Style%20Transfer.ipynb
7. Sklearn/Theano based implementation: https://github.com/memisevic/artify
8. pyCaffe based implementation: https://github.com/fzliu/style-transfer

Online demo: https://www.instapainting.com/ai-painter
Press coverage: http://www.boredpanda.com/computer-deep-learning-algorithm-painting-masters/

Data: For the artistic style experiments, you can use images from: https://github.com/jcjohnson/neural-style
Please see the individual sub-projects below for data related to each project.

A

Description:
Convolutional neural networks have emerged as a powerful image representation for various recognition tasks. However, their use for image editing applications remains still one of the key open problems. Imagine for example turning a photograph into a painting (see left), changing the illumination of an image from day to night, season from summer to winter, or even changing a car to look more “retro” or your “salami pizza” into a “vegetarian” one. This project will explore the use of the recent very deep convolutional neural networks [2] for image editing tasks.

C

The project will be based on the recent paper on changing artistic style [1] that uses the very deep convolutional neural network of [2]. The project will proceed along these steps:

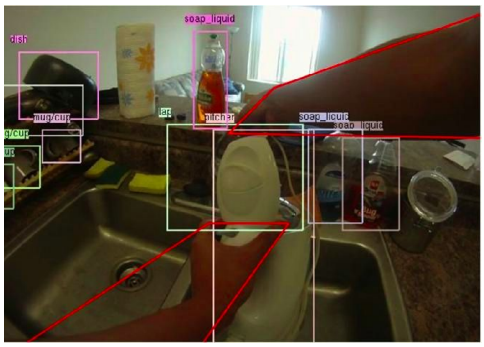
1. Read and make sure you understand very well the method in [1] and the structure of the network described in [2].
2. Choose one the implementations above (or make your own) and reproduce results from figure 2 and figure 3 in the paper to make sure you can run the code correctly and understand the different parameter choices. In addition to reproducing figures 2 and 3 from the paper, run the code on one or more pictures of your choice (e.g. your self-portrait). Show how the algorithm works on an image of your choice from [3].

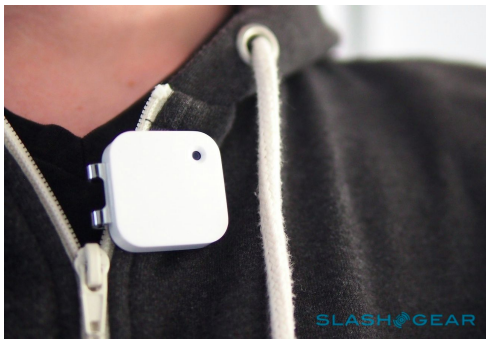
3. Choose one or more of the steps below.
   a. **Changing illumination/season.** Can the algorithm be applied to changing the illumination or the season of an image, e.g. from day to night, or from summer to winter? Try applying “style” of another similar looking photograph captured in the night or in a different season. Show results on example images from the dataset of Laffont et al., SIGGRAPH’14 available at http://transattr.cs.brown.edu/. Report results for different parameters of the model in [1]. Suggest, implement and test possible changes of the model/algorithm in [1] to improve the results.
   b. **Style/content mixing from multiple images.** Adapt the algorithm so that the style of two (or more) different paintings are applied to the same image. Can the style of the different paintings be applied to different spatial locations of the input photographs? For example, apply different styles to the foreground objects and to the background. Suggest, implement and test possible changes of the model/algorithm in [1] to improve the results.
   c. **Learning artistic styles.** Can you learn artistic styles, such as drawing or watercolor that will be independent of the source photo? In particular see the examples here under “Examples where the source image is available”.

d. **Semantic image editing (advanced).** Can "style" of an object in one image be applied to another similar looking object in another image? Try this on the car dataset of Lee et al., ICCV 2013, available here. Choose one car from the dataset as the target style and apply it to another car image. Make sure the two cars are captured from roughly the same viewpoint to make the problem easier. Suggest, implement and test possible changes of the model/algorithm in [1] to improve the results.
e. **Find the difference (advanced).** Modify the algorithm in [1] to emphasize the image features that are different between two images. Apply this to two different images of the car dataset from Lee et al., ICCV 2013, available here. Can you localize the car parts that are different between the two images?
f. **Video editing (very advanced).** Modify the algorithm of [1] to use the convolutional neural network of Tran et al., ICCV'15 for simple video editing task. For example, can you apply the motion of one dynamic texture to another?

Topic B - Activities in first-person camera view

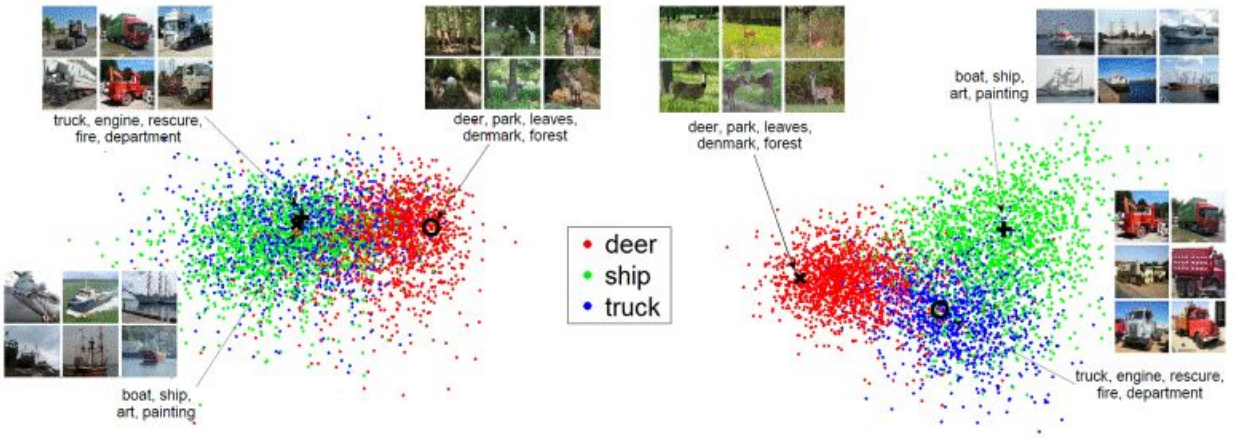
Paper: [1] Detecting Activities of Daily Living in First-Person Camera Views
H. Pirsiavash and D. Ramanan, CVPR 2012
Project page: http://people.csail.mit.edu/hpirsiav/codes/ADLdataset/adl.html
[2] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014
[3] M. Cimpoi, S. Maji, A. Vedaldi, Deep Filter Banks for Texture Recognition and Segmentation, CVPR 2015.
[4] Ren et al., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015.





Description: Wearable cameras such as the Narrative.com clip or GoPro are easily available. Imagine you have a personal assistant that watches what you do and can answer questions like, "Where did I leave my car keys?" or "Did I close the balcony door before I left home?". This project will investigate recent work [1] on detecting activities of daily living (such as washing dishes, making tea, or watching TV) in video footage from a wearable sensor. The goals of the project are: (i) reproduce results from the paper on their dataset, available here. While the code and intermediate results are available, you may also choose to re-implement portions of this work. (ii) Improve on the reported clip classification and temporal sliding window detection results (table 3 in the paper) by trying out the recent CNN descriptors [2], [3] and object detection methods [4] (see also assignments 2 and 3). (iii) Finally, you can capture your own data and apply the algorithm on your own videos. Wearable camera will be available from the class instructors.

Topic C - Joint representations for images and text
Multimodal retrieval: image-to-image search, tag-to-image search, and image-to-tag search.



(a) Visualization of CCA (V+T) (b) Visualization of CCA (V+T+C)

Papers:
[1] Normalized CCA: http://www.unc.edu/~yunchao/crossmodal.htm
[2] Word2Vec: http://code.google.com/p/word2vec/
[3] Overfeat: http://cilvr.nyu.edu/doku.php?id=software:overfeat:start
[4] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014
[5] M. Cimpoi, S. Maji, A. Vedaldi, Deep Filter Banks for Texture Recognition and Segmentation, CVPR 2015.
[6] Devlin et al. Exploring Nearest Neighbor Approaches for Image Captioning, 2015.
[7] Karpathy and Fei Fei, Deep Visual-Semantic Alignments for Generating Image Descriptions, CVPR 2015.

Data: Microsoft COCO dataset.

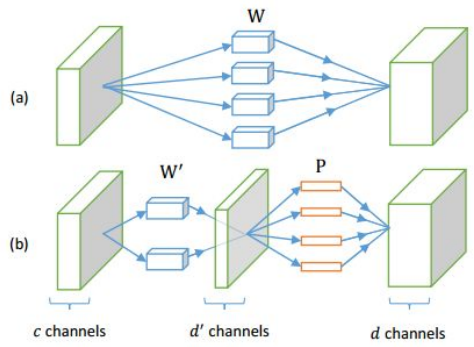
Description: Automatically producing natural text describing the content of an image is a very hard problem. In this project you will investigate joint representations for images and text suitable for this task. In particular, you will investigate the canonical correlation analysis (CCA) [1], a popular and successful approach for mapping visual and textual features to the same latent space. You will experiment with canonical correlation analysis on several sources of data to find correlations between image features and sentence features.
Detailed step-by-step instructions:

1. Implement CCA following [1]. Show it works on toy synthetic data.
   You can focus only on the standard two-view CCA. No need to work on the multi-view version.
2. Extract text word representations from the sentences associated with MS COCO data using [2].
3. Extract CNN image representations using [3].
4. Apply CCA on the features extracted in 2. and 3.
5. Implement a retrieval pipeline using the computed correlations as in [1].
   a. Tag-to-image search (T2I)
   b. Image-to-tag search (I2T)

6. Show qualitative example results on Microsoft COCO dataset
7. Pick 5-10 objects and quantitatively evaluate the results for tag-to-image search using the ground truth object labels provided with MS COCO. Plot a precision recall curve for each object and report average precision (AP).
8. Use the object ground truth for MS COCO to quantitatively evaluate the image-to-tag search as described in section 6.7 of [1], i.e. compute the average precision for all tags ranked number 1, number 2, etc. You can do this on a randomly sampled test set.
9. You can pick one or more of steps below:
    a. Quantitatively compare plain CCA with normalized CCA (on T2I and I2T) tasks.
    b. Quantitatively compare different CNN features, e.g. [3] with [4] or even [5].
    c. Show results for generating full sentence captions (rather than only single tag prediction) using the nearest neighbour method of [6].
    d. Choose few example images from this website. On those images compare sentences generated using the method in c. with outputs of recurrent neural network (RNN) model of [7]. You do not need to run the model of [7], just use their outputs available on the website above.

Topic D - Accelerating Convolutional Neural Networks

Paper: Accelerating Very Deep Convolutional Networks for Classification and Detection
Xiangyu Zhang, Jianhua Zou, Kaiming He and Jian Sun, Technical report, arXiv, May 2015



Description: In the last three years CNNs have taken over the majority of image recognition tasks including object, scene and face recognition. Most of the current CNN architectures build on the work of Krizhevsky et al. NIPS12 and differ mainly in the number of layers, the number of filters and the size of convolutional kernels. Hence, they all share the same problem: they are slow. The speed of CNNs becomes essential in situations with limited resources such as real-time and/or mobile applications. The speed is also critical for further research in CNNs since the long training times prevent wide exploration of potentially interesting architectures, loss-functions, training methods, etc.
Several recent methods focus on speeding-up CNNs. The main time-consuming step of CNNs is the multiplication of network responses at a layer N with the weights of filters at the layer N+1 which can be expressed as a matrix multiplication. One idea to speed up this operation is by means of low-rank matrix approximations reducing matrix multiplications to less operations. Your task in this project is to implement a variant of such an approach described in the paper Zhang et al. 2015 and to evaluate the proposed low-rank approximations of CNN layers, applying them e.g. to the popular VGG16 network. The time and the accuracy

	should be evaluated for different low-rank approximations. Results should be compared to results in the paper using corresponding experimental setups.

Topic E - Visual Question Answering

Paper: VQA: Visual Question Answering
S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, D. Parikh, ICCV 2015.
Project page: http://visualqa.org/

Description:
Computer vision now can detect and classify objects in images rather well. Is the problem of artificial vision nearly solved? Compared to human capabilities to analyze visual scenes, computer vision is still far behind on many tasks such as predicting future events or answering queries about images. Visual Question Answering (VQA) is a new task where computers are expected to understand questions about images and to provide correct answers. With VQA one can easily increase the level of complexity by changing questions (e.g., Is there a dog in the image? vs. How old is the person that plays with a dog?). Compared to other related tasks, such as automatic image caption generation, the evaluation of VQA can be done in a more objective way. VQA can also be easily re-focused on particular domains, enforcing automatic systems to understand e.g. the fashion, weather, human emotions, etc.
The goal of this project is to experiment with the VQA task on the dataset provided at http://visualqa.org/ The project should implement some of the baseline methods reported in the ICCV15 paper, in particular methods that analyze query text only and methods that analyze text and images. Results should be reported using the standard train/val/test split of the dataset, the evaluation setup should enable direct comparison to results reported in the paper. The project is open-ended and students are encouraged to try their own ideas.

Your own chosen topic.

You can also choose your own topic, e.g. based on a paper which has been discussed in the class. Please validate the topic with the course instructors (I. Laptev or J. Sivic) first. You can discuss the topic with the course instructors after the class or email to Ivan.Laptev@ens.fr or Josef.Sivic@ens.fr.

Joint topics with the "Introduction to graphical model" class (F. Bach and G. Obozinski).

The joint project between two classes is expected to be more substantial and will have a strong machine learning as well as computer vision component. Please contact the instructors of both courses if you are interested in the joint project. We will discuss and adjust the requirements from each course depending on the size of the group. An example of a topic for a joint project is **"Topic C - Joint representations for images and text"**.

You can also define your own topic for a joint project between the two classes. You need to validate the topic with the instructors for both courses.

Instructions for writing and submitting the project proposal.

- You will submit a 1-page project proposal indicating (i) your chosen topic, (ii) the plan of work, i.e. what are you going to implement, what data you are going to use, what experiments you are going to do, (iii) if working in a group, who are the members of the group and how you plan to share the work. The due date for the proposal is given at the beginning of this page. The project proposal should be a single 1-page pdf file.
- The proposal pdf should be named using the following format: FP_lastname1_lastname2.pdf, where you replace "lastname*" with last names of all members of your group in alphabetical order, e.g. for a group consisting of 2 people: I. Laptev, J. Sivic, the file name should be **FP_Laptev_Sivic.pdf**.
- Send the pdf file of your proposal to **Ivan Laptev <Ivan.Laptev@ens.fr>**.

Instructions for writing and submitting the final project report

- You will hand-in a 3 page report in the format of the submission to the IEEE Computer Vision and Pattern Recognition conference (CVPR) . Use the latex or word templates provided at the CVPR Author Guidelines webpage. Note, that your are asked to produce only a 3-page double-column report (in contrast, a standard CVPR submission is up-to 8 pages).

- At the top of the first page of your report include (i) names of all members of your group (up to 3 people), (ii) date, and (iii) the title of your final project.

- The report should be a single pdf file and should be named using the following format: FP_lastname1_lastname2.pdf, where you replace "lastname*" with last names of all members of your group in alphabetical order, e.g. for a group consisting of 2 people: I. Laptev and J. Sivic, the file name should be **FP_Laptev_Sivic.pdf**.

Send the pdf file of your report to **Ivan Laptev <Ivan.Laptev@ens.fr>**.

Instructions on preparing the project presentation.

- Each group will present their final project work in the class.
- **Timing.** Depending on the size of the group you will have 10-20 min slot to present your work. The exact timing and schedule of the presentations will be determined during the course.
- **Who should speak?** If you are working in a group, you can have one person presenting for the whole group, but it is preferable that all members of the group get to present a part of the project.
- **Content.** You should introduce the topic, clearly state what the goal of the project is. Show the work you have done. When describing results, please show both qualitative and quantitative results you have obtained and any interesting observations / findings you have made. Your audience are the other students in the class and the class instructors. You want to show us that you have done interesting work. Remember, it is good to illustrate your findings with images.
- **Re-using material / figures / slides from other people.** You can take figures from papers or other people's slides to illustrate an algorithm or explain a method. However, allways properly acknowledge the source if you do so.