ETRO VUB-DEPARTMENT OF ELECTRONICS AND INFORMATICS

VUB VRIJE UNIVERSITEIT BRUSSEL

Data representation, reduction and analysis

Projects 2018

Nikolaos Deligiannis

Lusine Abrahamyan, Hung Le

Guidelines

The aim of your projects is to study, implement, test and demonstrate machine learning algorithms with applications in multimedia processing. You can use any programming language/framework of your preference. We, however, recommend Python with Tensorflow or SKlearn, as taught during lab sessions.

- Each group consists of up to 3 members.
- Each group selects one topic from the list below. You are not required to strictly follow the referred papers.
- Write **your own code**. Additional libraries can be used for easier pre-processing of the data, but may not constitute the bulk of your project. ***Directly copying existing code from external sources without acknowledgement disqualifies the project, leading to a project score of 0 for the entire group.***
- **Project submission (by email to** dle@etrovub.be **and** alusine@etrovub.be**)**
    - An interim report (not graded) of one page: Describe your understanding of the problem and your chosen algorithms. You will receive our feedback on this report.
    - A final report of 6-10 pages (as pdf). Sections (obligatory): Introduction, Description of your algorithm, Experimental results, Conclusions, References
    - Your code and a README.txt file describing how to run the code, all in a zip file
- **Important dates**
    - The deadline for selecting your topic: 11/11/2018. Send us an email about your group members and topic.
    - The deadline for submitting the interim report: 30/11/2018.
    - The deadline and date for the oral defense will be announced soon.

- **Grading**

  The project grading is based on:
    - Technical quality
    - Final report
    - Oral defense

  The oral presentation and Q&A (20-30 minutes) will be done in groups with a single Power point presentation. Your score will be based on your understanding of the techniques you used in your project

1. Image super resolution

Description

Super resolution from a single low resolution image has always be an ill posed problem. The current state of the art consists in training a convolutional neural network with a large dataset of natural looking images. Corresponding low resolution images are computed and fed as an input to the neural network. The learned feature maps have shown to produce less artifacts than classical methods.

Objectives

The first step in this project is to retrieve the dataset. We suggest using the DIVerse 2K resolution high quality images[1] which provides both high and low resolution images.

The most interesting aspect of this project is the method to upscale the number of pixels in a given layer of your network. A very powerful approach is to interleave pixels of separate feature maps in a so-called sub-pixel convolution. Other works work on a bicubic interpolation of the low-resolution image to avoid this, or use sparse-coding to map low-resolution patches to high-resolution ones.

After training, compute the PSNR of the reconstructed image on a validation dataset.

The deliverables of this project are : (i) a brief description of the literature ; (ii) a working super resolution CNN with a few different architecture parameters; (iii) an experimental study of the reconstructed image PSNR against plain bicubic interpolation.

References

- Shi, Wenzhe, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874-1883. 2016.
- Dong, Chao, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Learning a deep convolutional network for image super-resolution." In *European Conference on Computer Vision*, pp. 184-199. Springer, Cham, 2014.

[1] https://data.vision.ee.ethz.ch/cvl/DIV2K/

2. OCR for handwriting

Description

Handwritten character recognition has been one of the most basic and popular problems in Computer Vision for a long time. The task is to infer the correct label (letter or digit) given an image of handwritten character. This problem, often called *Optical Character Recognition (OCR)*, has been employed in a lot of application. Nowadays, we can see OCR system in our smartphones (Android and iOS), or even in our PDF reader.

Objectives

In this project, you will build an OCR system, employing Convolutional neural network models. You are recommended to use the EMNIST dataset[2] (the full dataset or a subset of it, depending on your computational resources).

Upon completing this project, we expect : (i) a brief description of the literature ; (ii) a working OCR system, which can recognize our handwritten letters, captured by camera (you will need to perform some pre-processing to make it similar to images you have in the EMNIST dataset) ; (iii) an experimental evaluation of the model with different configurations.

References

- G. Cohen, S. Afshar, J. Tapson and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters", *International Joint Conference on Neural Networks (IJCNN), 2017, pp.2921-2926*
- Y. Lecun and L. Bottou and Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, 1998, pp. 2278-2324

[2] https://www.nist.gov/itl/iad/image-group/emnist-dataset

3. Grayscale image colorization

Description

A large amount of historical grayscale images is available online. Adding color information to an existing image can be done by an artist, but this is time consuming. The most effective methods can now assist the artist when propagating colors on the whole image. Colorization with a neural network alone has also been explored, and showed interesting results. The goal of this project is to build your own colorization CNN.

Objectives

In this project, you should start by building a grayscale/color image dataset from a public natural image dataset such as Imagenet[3] or the Berkeley Segmentation dataset[4].

The learned CNN produces a chrominance image (UV space) from a luminance image (Y only). A typical issue that occurs when training the CNN is that most datasets are poorly balanced in the chrominance space: low values are more often present. An ingenious way to tackle this is to discretize the chrominance space, and formulate the problem as per-pixel classification task. Low occurrence of some classes (colors) in the dataset can be compensated by weighting the loss-function for those classes.

Upon completing this project, we expect : (i) a brief description of the literature ; (ii) a working neural-network-based colorization model; (iii) experimental evaluations with different model architectures.

References

- Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." In *European Conference on Computer Vision*, pp. 649-666. Springer, Cham, 2016.
- Cheng, Zezhou, Qingxiong Yang, and Bin Sheng. "Deep colorization." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 415-423. 2015..

[3] http://image-net.org/
[4] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

4. Spam detection

Description

The huge part of machine learning tasks is related to text analyses. Text, as a source of information, has been used in such tasks as recommendation generation, behavior analyses, churn prediction, classification, etc. In this task, you will perform classification task – given real comment from YouTube videos, you should classify if it is a spam or not.

Objective

The first step of this project is to analyze dataset – compute, distribution of unique words, visualize them…, i.e. do exploratory data analysis. Explore text representation techniques - word2vec, tf-idf. In the below dataset, each sample consists of comment_id, author, date, content, tag. Based on the above analysis, design your machine learning algorithm to detect spam comments.

Upon completing this project, we expect: (i) a brief description of the literature; (ii) a working classification model; (iii) experimental evaluations with different approaches.

References

- "YouTube Spam Collection Data Set", https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection
- Alberto, T.C., Lochter J.V., Almeida, T.A. "TubeSpam: Comment Spam Filtering on YouTube. Proceedings of the 14th IEEE International Conference on Machine Learning and Applications"