Machine Learning and Big Data Processing

Projects 2023

Nikolaos Deligiannis, Adrian Munteanu (course titulars)

Remco Royen, Yuqing Yang (TAs)

Guidelines

The aim of the project is to study, implement, test and demonstrate machine learning algorithms with applications in multimedia processing. For your implementations, you should use Python, with some libraries for machine learning (e.g., sklearn, tensorflow, pytorch).

- You have to **write your own Python code**. Additional libraries/open-source source codes can be used 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**.
- You have to submit a written project report of 6-10 pages.
- Each group submits ONE report, with the contributions of group members clearly listed.
- Each group consists of up to 4 members.
- Sections (obligatory): Introduction, State of the Art, Description of your algorithm/approach, Experimental results, Conclusions, References.

- **Project submission**

  Submit a **pdf** report, along with the code and a README.txt file describing how to run the code (all in a zip file) to: Remco.Royen@vub.be and yuqing.yang@vub.be

- **Grading**

  The project accounts for **30%** of the final score for the exam.

  The project grading is based on:
  - Technical quality (score per group)- 40%
  - Written report (score per group)- 25%
  - Oral defense (individual score)- 35%

  The oral presentation and Q&A (~30 minutes of which 15-20min presentation) will be done in groups with a single Power point presentation. Your score will be based on your understanding of the machine learning and big data processing techniques you used in your project.

  The selection of a topic and group will be available starting from **March 28th (20:00)** (exact procedure is described in the introduction slides) and will be closed **Friday, March 31st (23:59)**.

  The project submission deadline is set on **Sunday, May 28th (23:59)**. Please note that a late submission within 24 hours after the deadline (thus before Monday, May 29th (23:59)) will automatically lead to a -2 on the final project score. Any submission after this 'grace period' will NOT be accepted.

  The date of the oral defense will be announced soon.

1. Image super resolution

Description

Super resolution from a single low-resolution image has always been 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.
- Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee. “Enhanced Deep Residual Networks for Single Image Super-Resolution”. The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 136-144, 2017
- Other papers found on Google Scholar/IEEE Xplore

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

2. Image denoising

Description

When measuring real-world color values, any digital imaging sensor will be disturbed by noise. When correcting the value of a given pixel, methods involving the weighted average of neighboring pixels are often used. However, it has recently been shown that the mapping from a noisy signal to a noise-free version can be learned very efficiently with a convolutional neural network. The non-linearity of this mapping outperforms classical signal processing methods. The goal of this project is to train a CNN to the task of removing additive Gaussian noise on grayscale images.

Objectives

For the image dataset, consider using the Berkeley Segmentation dataset[2]. You can add generated Gaussian noise of a fixed variance to produce the noisy images yourself.
An effective method is called residual learning: the output from the CNN is not the denoised image, but the noise pattern alone. It can then easily be subtracted from the noisy image to produce the result. You should compare your results with and without residual learning.
Upon completing this project, we expect: (i) a brief description of the literature; (ii) a working neural-network-based denoising method; (iii) experimental evaluations with different model architectures.

References

Zhang, Kai, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising." IEEE Transactions on Image Processing 26, no. 7 (2017): 3142-3155.

Burger, Harold C., Christian J. Schuler, and Stefan Harmeling. "Image denoising: Can plain neural networks compete with BM3D?" In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2392-2399. IEEE, 2012.

Y. Zhang, L. Sun, C. Yan, X. Ji and Q. Dai, "Adaptive Residual Networks for High-Quality Image Restoration," in IEEE Transactions on Image Processing, vol. 27, no. 7, pp. 3150-3163, July 2018.

Other papers found on Google Scholar/IEEE Xplore

[2] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

3. Voice isolation

Description

In human perception, a sound source is easily isolated, even in a crowded environment. A showcase task in this context is to isolate the singer's voice in a song. Neural Networks have been shown very effective for this task, thanks to their flexibility. Typically, we work on a spectrogram version of the sound, and require the network to produce a mask for each frequency bin. This mask indicates whether the bin belongs to the singer or the instruments.

Objectives

The first step in this project is to build a music dataset, along with the singer audio track isolated from the instruments. For this purpose, we suggest using the MedleyDB dataset[3] which contains individual tracks for 122 songs, or the more general MUSDB[4] database.

Short time Fourier transform must be used to produce a spectrogram of the sound, that can then be processed by a neural network. You can use an external library to process audio with Python[5].

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

References

- Simpson, Andrew JR, Gerard Roma, and Mark D. Plumbley. "Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network." In International Conference on Latent Variable Analysis and Signal Separation, pp. 429-436. Springer, Cham, 2015.
- Daniel Stoller, Sebastian Ewert, Simon Dixon. "Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation". E-Print Archive: https://arxiv.org/abs/1806.03185
- Jansson, A., Humphrey, E., Montecchio, N., Bittner, R., Kumar, A. and Weyde, T ."Singing voice separation with deep U-Net convolutional networks". In 18th International Society for Music Information Retrieval Conference (ISMIR), pp. 745-751, 2017.
- Other papers found on Google Scholar/IEEE Xplore

---

[3] http://medleydb.weebly.com/

[4] https://sigsep.github.io/datasets/musdb.html

[5] https://github.com/librosa/librosa

4. 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[6] or the Berkeley Segmentation dataset[7]. 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.
- Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. "Image Colorization using Generative Adversarial Networks". E-Print Archive: https://arxiv.org/pdf/1803.05400.pdf
- Other papers found on Google Scholar/IEEE Xplore

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

5. Image inpainting

Description

Image inpainting is also an interesting topic in the field of image restoration besides image denoising (project topic 3). This project aims at inpainting missing region(s) of an image using remaining pixels. Missing pixels can be a square in the middle, randomly localized blocks or random regions. Recently, with the development of Convolutional Neural Networks (CNNs), especially Generative Adversarial Nets (GANs), AI systems show an outstanding power of imagination to infer out missing information.

Objectives

We suggest using ImageNet[1] (or other similar datasets) and cropping out ¼ central or bottom-right square region of each image. Then, a simple method to recover this region is to train a convolutional autoencoder. However, you can also use the GAN model, a very hot model recently, in which the autoencoder works as the generator. Another classifier network is set as the adversarial discriminator to classify the output of the generator as an original (real) missing patch or a generated (fake) patch. The discriminator aims at minimizing the classification error while the generator seeks to maximize the classification error of the discriminator. Through alternative training of these two networks, the generator is equipped with a strong generating ability.
Upon completing this project, we expect: (i) a brief description of the literature; (ii) a working neural-network-based image-inpainting model; (iii) experimental evaluations with different model architectures.

References

- Chao Yang, Xin Lux, Zhe Liny, Eli Shechtmanz, Oliver Wang, and Hao Li. “High-Resolution Image Inpainting Using Multi-scale Neural Patch Synthesis”. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4076-4084. 2017
- Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. “Context Encoders: Feature Learning by Inpainting”. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2536-2544. 2016
- Other papers found on Google Scholar/IEEE Xplore

6. 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 applications. Nowadays, we can see OCR systems in our smartphones (Android and iOS), or even in our PDF reader applications.

Objectives

In this project, you will build an OCR system, employing Convolutional neural network models. You are recommended to use the EMNIST dataset[8] (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); An interesting addition would be to recognize letter per letter in words. (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

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

7. Music classification

Description

Recent advances in machine learning and especially deep learning have enabled several applications in multimedia. A very interesting application is automatic analysis of musical contents. Some examples are Shazam and Spotify Discover – online content-based music discovery services. These services belong to the category of music retrieval problem. A related problem, which has also received a lot of attention is content-based Music Classification. This problem involves identifying the characteristics of a song from the content (i.e. rhythm rather than metadata). Interesting characteristics include genre, artists, era and so on. These characteristics are useful for applications such as automatic organization of music collections, music recommendation and retrieval.

Objectives

In this project, you can choose to infer one specific characteristic from musical signals. As a suggestion, you can work on the genre classification problem, as its literature is relatively rich. Nevertheless, we encourage you to have your own proposal and creativity.

A basic pipeline to approach this problem contains two steps: feature extraction and classification models. As of the dataset, a suggestion is to use a subset of the Million Song dataset[9]. You can use an external library to process audio with Python[10].

Upon completing this project, we expect: (i) a brief description of the literature ; (ii) a working demo with different feature extraction methods and at least one classification algorithm ; (iii) an experimental study with different components you have built.

References

- v. d. Oord and S. Dieleman and B. Schrauwen, Deep content-based music recommendation", Internal Conference on Neural Information Processing Systems, 2013
- P. Ahrendt, "Music Genre Classification Systems - A Computational Approach", 2006
- T. Li and M. Ogihara and Q. Li, "A Comparative Study on Content-based Music Genre Classification", Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, 2003, pp. 282-289.
- Other papers found on Google Scholar/IEEE Xplore

---

9 https://labrosa.ee.columbia.edu/millionsong/

10 https://github.com/librosa/librosa

8. Matrix completion

Description

Matrix completion is one of the key problems in machine learning. It concerns the problem of recovering a matrix from partial observations, where a matrix cannot be fully sampled or directly observed. Examples of machine learning tasks that employ matrix completion algorithms include clustering, classification, and especially recommender systems. The Matrix Factorization (Collaborative Filtering) exercise that you saw during the lab sessions is also a Matrix completion method. Recently, deep neural network-based solutions have been proposed to tackle the matrix completion problems, and they are achieving state-of-the-art results.

Objectives

In this project, you will study the literature of neural-network-based matrix completion and implement and experiment with one model. Concerning the dataset, you can select and use (at least two) from: different versions of the MovieLens dataset[11], the Book-Crossing dataset[12], the Jester dataset[13] and the Netflix dataset[14].
Upon completing this project, we expect: (i) a brief description of the literature; (ii) a working neural-network-based matrix completion model; (iii) experimental evaluations with different model architectures.

References

- J. Fan and T. Chow, "Deep learning-based matrix completion", Neurocomputing (266), pp. 540-549.
- S. Suvash & A. K. Menon and S. Scott and X. Lexing, "AutoRec: Autoencoders Meet Collaborative Filtering", WWW 2015, pp.111-112
- F. Strub and R. Gaudel and J. Mary, "Hybrid Recommender System Based on Autoencoders", Workshop on Deep Learning for Recommender Systems, 2016, pp.11-16.

[11] https://grouplens.org/datasets/movielens/
[12] https://grouplens.org/datasets/book-crossing/
[13] https://grouplens.org/datasets/jester/
[14] https://archive.org/download/nf_prize_dataset.tar

9. Content-based Image Retrieval

Description

Content-based image retrieval (CBIR) concerns the problem of searching for relevant images given a query image. Traditional image retrieval methods rely on metadata associated to the images (e.g., users' provided tags on Flickr). Nevertheless, this approach is not suitable nowadays since the metadata normally does not describe well the sheer amount of the available content. CBIR methods, on the other hand, perform the matching between the query and images in the database using their contents and hidden (learned) features.

Objectives

In this project, you will study the literature of content-based image retrieval, implement the method(s) and carry out experiments. There are two main components in an image retrieval system: (i) feature extractor, and (ii) a retrieval component, which matches query to images in the database. A lot of features have been studied in the literature, e.g., BoW, VLAD and Convolutional-neural-network-based features (most recent). You are encouraged to use more than one feature extraction methods so that you can make comparison. For the retrieval component, some distance measures, e.g., cosine distance, Euclidean distance, etc., can be used on the features of the query image and the images in the database. For the experiments, you can use the popular benchmark datasets, such as the Holidays dataset, Oxford5k buildings, Paris6k buildings.

Upon completing this project, we expect: (i) a brief description of the literature; (ii) a working implementation; (iii) experimental evaluation of the method and comparison between different feature extraction methods and/or distance measures (with suitable evaluation metric).

References

- A. S. Razavian and H. Azizpour and J. Sullivan and S. Carlsson, "CNN Features off-the-shelf: an Astounding Baseline for Recognition", IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2014, pp. 512-519.
- J. Y-H. Ng, F. Yang, L. S. Davis, "Exploiting Local Features from Deep Networks for Image Retrieval", IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2015, pp. 53-61.

10. Rumour Detection on Social Media

Description

The aim of this topic is to automatically detect rumours that are spread on social networks, such as Facebook, Twitter, Sina Weibo. In this context, a rumour assumed to be related to an event. We will consider the setting in which each event is associated with a set of posts, e.g., tweets, re-tweets on Twitter from different users, based on which it gets spread out. By concatenating the texts from all these posts, we can create a "document" for the event. From this document, the textual features can be extracted to represent the event. In this project, we will classify if an event is a rumour or not based on their textual features (It should be noted that beside textual features, different types of features, e.g., users' profiles, can also be extracted to represent the events).

Objectives

In this project, you will first study the literature of rumour detection and text classification. After that, you will select a method to implement and carry out experiments. There are commonly two main components in text classification, namely feature extractor and classifier. For the feature extractor, you can use one or multiple features, e.g., BoW, TF-IDF, or deep-learning-based features (word2vec, doc2vec). For the classifier, you have seen different models during the lectures, namely logistic regression and neural networks. You can build different systems by combining different feature types and classifiers. For the experiments, you can use the Twitter and Weibo datasets (see the first reference).

Upon completing this project, we expect: (i) a brief description of the literature; (ii) a working implementation; (iii) experimental evaluations with different feature types and/or classifiers.

References

- J. Ma and W. Gao and P. Mitra and S. Kwon and B. J. Jansen and K-F. Wong and M. Cha, "Detecting Rumors from Microblogs with Recurrent Neural Networks", Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI) 2016, pp. 3818-3824
- N. Ruchansky and S. Seo and Y. Liu, "CSI: A Hybrid Deep Model for Fake News Detection" Proceedings of the ACM on Conference on Information and Knowledge Management (CIKM), 2017, pp. 797-806