Machine Learning Methods for Computer Vision

Eros Innocenti[1], Alessandro Vizzarri[2]

[1]Deptartment of Engineering Science, Guglielmo Marconi University, Italy
[2]Deptartment of Enterprise Engineering, University of Rome Tor Vergata, Italy

Abstract

Over the last years, deep learning methods proved to outperform previous machine learning techniques, especially in high computational task such as computer vision. This review paper aims to provide a preliminary overview of the machine learning tasks where computer vision in involved. Furthermore, a brief review of their history and state-of-the-art techniques is presented in the fields of image classification and object detection.

Keywords

Machine Learning, Computer Vision, Artificial Intelligence, Deep Learning

1. Introduction

Nowadays, computer vision is one of the most studied artificial intelligence and machine learning subfields. Its applications are many and various, ranging from industry applications to manufacturing [1], healthcare and autonomous vehicles. The CV main goal is to replicate the capabilities of humans' vision. Although for our brain this kind of task appears fairly simple, there is a lot of information processing under the hood. Over the years, the field of computer vision is shifting from a statistical approach, based on hand-crafted methods, to deep learning neural networks ones. This change of perspective is driven not only by an increasing performance demand [2]. In fact, deep learning models proved that they can learn semantic representations of images, thus adapting better to different scenarios without requiring human interventions [3]. In this paper we want to take a brief review on the problems, which CV could solve and the state-of-the-art technologies developed in the last few years of research. In Section 2 we illustrate how the machine learning problems are categorized in different tasks, each one with different goals. Section 3 presents the subtasks specifically related to computer vision, subsequently in Section 4 some mainly used object detection techniques are described. Eventually, in Section 5 an overview of future directions is presented, presenting some of next years open challenges.

Machine learning includes an extensive set of tasks, which can be classified in three broad categories: Supervised Learning, Unsupervised Learning and Reinforcement Learning. In the next subsections we will briefly describe these three categories.

ICYRIME 2021 @ International Conference of Yearly Reports on Informatics Mathematics and Engineering, online, July 9, 2021
eros@newtechweb.it (E. Innocenti);
alessandro.vizzarri@uniroma2.it (A. Vizzarri)
0000-0002-7793-4974 (E. Innocenti); 0000-0002-6274-991X (A. Vizzarri)

2. Machine Learning Tasks

2.1. Supervised learning

In supervised learning the goal is to infer a function starting from a collection of labeled training data. The training data, typically consists in a set of image examples annotated with extra information such as the image class, or the position of the depicted object(s). The training in most cases is hand-made, but semi-supervised approaches are available too. This possibility is useful if the training set size is small, and it is difficult or even impossible to obtain more samples. Moreover, image augmentations techniques (e.g., horizontal and vertical flip, shear, brightness and contrast variations) can be used to artificially increase the training set size, thus achieving better training performances.

The steps required to train a computer vision model using supervised learning can be summarized in the following:

1. Decide the kind of training examples which represent accurately the problem.
2. Collect a sufficient number of examples. In the case of many classes, make sure to balance the number of examples across all of them.
3. Decide an input feature vector which is descriptive for the selected task. The number of features should not be too large, in order to avoid overfitting.
4. Decide the learning function structure and pick a loss function which has to be minimized during the training phase.
5. Run the model on the training set, iteratively optimizing its parameters until the target metric (e.g., loss, accuracy, average precision) reaches the target value.

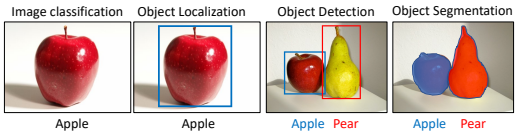

Figure 1: Computer vision tasks

6. Evaluate the trained model on a test set. In order to obtain an unbiased evaluation of the model, it's important that the test set is composed only by unseen examples.

2.2. Unsupervised learning

Unsupervised learning, unlike the supervised one, does not need a labeled training set. Instead, the goal is to infer a function which describes the underlying structure from unlabeled data. It is worth noting that since the examples are not annotated, it is not possible to evaluate the performance of the model using the methods applied in supervised learning. Unsupervised learning is used in many situations, some of them are *dimensionality reduction*, *search of clusters*, *data compression*. One popular example of unsupervised learning is the k-means clustering algorithm [4].

2.3. Reinforcement learning

Lastly, reinforcement learning substantially differs from the previous ones because it lacks the initial training data completely [5]. In this kind of machine learning, the running program (i.e., the agent) interacts with the environment making use of sensors and actuators with a certain goal to achieve. The agent is provided by feedbacks that could be rewards or penalties based on the actions taken in the previous one or more time spans.

In the next sections of this paper we will focus mainly on supervised learning.Specifically we will analyze the most frequent computer vision related subtasks and the techniques commonly used to solve this kind of problems.

3. Computer Vision Tasks

As stated before, in computer vision, we can further split these tasks, mainly into 4 categories:

- Image classification
- Object localization
- Object detection
- Object segmentation

In figure 1 an example of these categories is depicted.

3.1. Image classification

Image classification is probably the most well-known computer vision task. The main goal is to assign an input image to one of a set of predefined categories. The simplest case is represented by binary classification, it means that the output of the model consists in only two possible values: true or false. An example could be a classifier which given a picture returns if that picture contains a person or not. A more complex version of the same classifier could have more than two categories (e.g., person, cat, dog, car).

3.2. Object localization

Starting from the previous image classification task, we could improve the output of the neural network adding the information about the location of the object. The common way to describe the location of an object is to define a bounding box which encloses the object in the picture.

3.3. Object detection

Object localization is limited to one object per image. The computer vision task whose goal is to localize multiple object of different classes in the same picture is called Object Detection. This task introduces major complexities if compared with the previous one, and the required effort to scale from Object Localization to Object Detection can be significant. Some problems encountered can be difficult even for humans. Some objects could be partially visible, because they overlap each other or may be partially outside the frame. Moreover, the sizes of the objects belonging to the same class could vary noticeably.

3.4. Object segmentation

In the previous localization and detection tasks, the main goal is to place a bounding box (and a class label) over all the objects present in the input image. Segmentation differs from localization and detection because the output is no more a set bounding box. Instead, in segmentation, the computer vision model tries to annotate every pixel of the image whether part of a specific class from a set of predefined ones.

Object segmentation can be further divided in two types: *semantic segmentation* [6, 7] and *instance segmentation* [8, 9].

The main difference between these two kinds is that semantic segmentation treats multiple objects belonging to the same class as a single entity. On the other hand, instance segmentation treats multiple objects of the same class as individual instances.

3.5. Object tracking

Object tracking applies to a sequence of images instead of a single input, because of this reason it has not been listed at the beginning of this section. The purpose of object tracking is to track a moving object over subsequent frames. This kind of functionality is essentials for robots or autonomous cars. A straightforward approach to perform object tracking is to apply the object detections techniques to a video instance and then compare every object instance in order to determine the direction and the speed of the movement. However, it is worth noting that, in many cases, the object tracking does not need to recognize objects of different classes, but could simply rely on motion criteria without being aware of the objects classes.

4. Techniques

4.1. Object classification

The emergence of large scale annotated training sets such as ImageNet [10] or COCO [11], required significant computational power and deeper network architectures. In the last few years, high performance parallel computational systems, such as GPUs, enabled new challenges in computer vision that can be solved by the means of deep learning. The most representative models of deep learning applied to computer vision are Convolutional Neural Networks (i.e., CNNs). The first convolutional neural network appeared in 1998 with LeNet-5 [12], a 7 layers convolutional neural network developed by Yann LeCun. LeNet was used to recognize hand-written numbers from the famous MNIST dataset [13], a collection of 32x32 pixels greyscale input images. The architecture was pretty simple, mainly because for the time there were computational power constraints.

In 2012, AlexNet [14] won the ILSVRC [15] (ImageNet Large Scale Visual Recognition Challenge) 2021 competition, with a similar architecture but with more filters and layers, thus becoming one of the first deep neural networks.

The next year, ZFNet [16] won the ILSVRC mostly tweaking the hyper-parameters of AlexNet, maintaining the same base structure.

In 2014 VGGNet [17] entered the scene becoming one of the reference architecture for object classification. The first version (i.e., VGG16) had a very uniform architecture, composed by sixteen 3x3 convolutional layers followed by max pooling operations. The main drawback of VGG is the number of parameters (i.e., 138 million), which can be challenging to handle. Anyhow, VGG is still one of the preferred architecture used for feature extraction from images.

In 2015, ResNet by Kaiming He et al [18] introduced a novel CNN architectured called Residual Neural Network. The main difference from the previous is the introduction of skip connections between layers. Such skip connections permitted to obtain better training results with fewer parameters. ResNet obtained a top-5 error rate of 3.5% on ImageNet, which beats human-level performances (approximately 5%) on the same dataset.

In 2017, MobileNet [19] was presented as a solution for mobile and embedded visual applications. This lightweight network is particularly suited for low power system [20]. The network is very flexible and can be easily adapted to the specific application, tweaking its hyper-parameters.

Lastly, in 2019 Mingxing T. and Quoc V. [21] studied a novel neural network (i.e., EfficientNet) which can be scaled up as needed in a very efficient way. The main novelty about this method is that the scaling process involves not only the depth of the network, but also the width and the resolution of the input, thus proving that this compound method obtains better results with less parameters.

4.2. Object detection

Deep Neural Networks for Object Detection can be categorized in two different types:

- Region proposal networks
- Single shot detectors

Historically, the first detectors were based on the previous described image classification networks. The basic idea to obtain object detection is based on a sliding window approach. Substantially, a fixed size rectangular window crops the image at different positions and a subsequent image classification network is in charge of predicting the object class. At each iteration, the window is moved by a stride value until the whole image is analyzed. The main drawback of this method is the low speed because it is computational expensive. An improvement over the sliding window approach, is called selective search [22], which consists in a hierarchical grouping segmentation algorithm that combines multiple grouping strategies. This algorithm starts with an initial set of regions and at each iteration merges the most similar regions together, until the whole image is represented as a single region. Finally, a set of regions of interests (ROI) are selected and fed into an image classification network. The resulting object detection network is called Region-based ConvNet (R-CNN) [23, 24]. Although selective search improved quite noticeably the overall speed of the process, it is still not enough when speed is a key factor. In 2015 other two improvements of region proposal based networks were proposed, Fast R-CNN [25] and soon after Faster R-CNN [26]. The main

novelty about these new architectures was the integration of ROIs generation into the neural network itself. In fact, the previous version of R-CNN used selective search for ROI extraction as a separated process.

In the same year, YOLO (You Only Look Once) [27, 28] revolutionized the object detection scene presenting an algorithm substantially different from the classical region proposal networks. A new kind of architecture started to emerge, called Single Shot Detectors. Instead of using a ROIs extraction phase, single shot detectors divides the image in a grid, giving at each cell the task to detect objects in that region. For each grid cell, multiple predefined boxes (i.e., anchors or priors) are considered. These boxes have multiple sizes, aspect ratio in order to be able to detect objects of different shapes. Immediately after, Single Shot MultiBox Detectors [29] followed the same approach obtaining similar results to YOLO in terms of speed and accuracy.

Over the years many variations of these architectures were presented, each one with its particularities and strengths. Although there are exceptions, nowadays region proposal based networks are preferred when accuracy is of main importance and speed is secondary. Moreover, R-CNNs are considered better in detecting small objects.

On the other hand, single shot detectors overtake R-CNNs in real-time tasks, edge or mobile computing [30]. The inference time of these networks is less, at the cost of lower accuracy [31].

5. Conclusions

In this paper, a brief review of commonly used deep learning methods has been made, emphasizing its application in the field of computer vision. In the last years, especially using GPUs clusters, we obtained the computational power to enable the design of deeper neural networks [32]. Moreover, the availability of large datasets such as COCO or ImageNet allowed training accurate models, which can be adapted to a variety of scenarios. With the increasing importance of mobile devices and edge computing, the high power requirements of the reviewed techniques will inevitably conflict with the low power resources offered by edge devices. Although cloud computing can help, many situations such as rural areas, make internet access problematic, thus invalidating the remote processing possibility. Moreover, supervised learning, which is the commonly used method for computer vision tasks, allows obtaining noticeably results at the cost of long training times. In the future, self-learning methods should be considered, in order to skip the whole dataset creation and focus in the learning phase, as it happens for the humankind.

References

[1] A. Jaber, R. Bicker, Fault diagnosis of industrial robot bearings based on discrete wavelet transform and artificial neural network , International Journal of Prognostics and Health Management 7 (2016) art. no. 017.

[2] G. Capizzi, G. Lo Sciuto, C. Napoli, E. Tramontana, A multithread nested neural network architecture to model surface plasmon polaritons propagation, Micromachines 7 (2016) 110.

[3] F. Fallucchi, M. Petito, E. De Luca, Analysing and Visualising Open Data Within the Data and Analytics Framework, Communications in Computer and Information Science 846 (2019) p.135–146.

[4] Y. Li, H. Wu, A clustering method based on k-means algorithm, Physics Procedia 25 (2012) 1104–1109.

[5] L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re, S. Spanò, Multi-agent reinforcement learning: A review of challenges and applications, Applied Sciences 11 (2021) 4948.

[6] C. Napoli, G. Pappalardo, E. Tramontana, An agent-driven semantical identifier using radial basis neural networks and reinforcement learning, volume 1260, 2014.

[7] A. Venckauskas, A. Karpavicius, R. Damasevicius, R. Marcinkevicius, J. Kapociute-Dzikiene, C. Napoli, Open class authorship attribution of lithuanian internet comments using one-class classifier, 2017, p. 373 – 382. doi:10.15439/2017F461.

[8] G. De Magistris, S. Russo, P. Roma, J. Starczewski, C. Napoli, An explainable fake news detector based on named entity recognition and stance classification applied to covid-19, Information (Switzerland) 13 (2022). doi:10.3390/info13030137.

[9] C. Napoli, E. Tramontana, G. Lo Sciuto, M. Woźniak, R. Damaševičius, G. Borowik, Authorship semantical identification using holomorphic chebyshev projectors, 2015, p. 232 – 237. doi:10.1109/APCASE.2015.48.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[11] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: common objects in context, CoRR abs/1405.0312 (2014). URL: http://arxiv.org/abs/1405.0312. arXiv:1405.0312.

[12] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 1998, pp. 2278–2324.

[13] Y. LeCun, C. Cortes, MNIST handwritten digit database, prova (2010). URL: http://yann.lecun.com/

exdb/mnist/.

[14] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90. URL: https://doi.org/10.1145/3065386. doi:10.1145/3065386.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[16] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, CoRR abs/1311.2901 (2013). URL: http://arxiv.org/abs/1311.2901. arXiv:1311.2901.

[17] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015. arXiv:1409.1556.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015. arXiv:1512.03385.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. arXiv:1704.04861.

[20] G. M. Bianco, R. Giuliano, G. Marrocco, F. Mazzenga, A. Mejia-Aguilar, LoRa System for Search and Rescue: Path-Loss Models and Procedures in Mountain Scenarios, IEEE Internet of Things Journal 8 (2021) p.1985–1999.

[21] M. Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, CoRR abs/1905.11946 (2019). URL: http://arxiv.org/abs/1905.11946. arXiv:1905.11946.

[22] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders, Selective search for object recognition, International Journal of Computer Vision (2013). URL: http://www.huppelen.nl/publications/selectiveSearchDraft.pdf. doi:10.1007/s11263-013-0620-5.

[23] R. B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CoRR abs/1311.2524 (2013). URL: http://arxiv.org/abs/1311.2524. arXiv:1311.2524.

[24] N. Brandizzi, V. Bianco, G. Castro, S. Russo, A. Wajda, Automatic rgb inference based on facial emotion recognition, volume 3092, 2021, p. 66 – 74.

[25] R. B. Girshick, Fast R-CNN, CoRR abs/1504.08083 (2015). URL: http://arxiv.org/abs/1504.08083. arXiv:1504.08083.

[26] S. Ren, K. He, R. B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, CoRR abs/1506.01497 (2015). URL: http://arxiv.org/abs/1506.01497. arXiv:1506.01497.

[27] J. Redmon, S. K. Divvala, R. B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, CoRR abs/1506.02640 (2015). URL: http://arxiv.org/abs/1506.02640. arXiv:1506.02640.

[28] R. Avanzato, F. Beritelli, M. Russo, S. Russo, M. Vaccaro, Yolov3-based mask and face recognition algorithm for individual protection applications, volume 2768, 2020, p. 41 – 45.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, A. C. Berg, SSD: single shot multibox detector, CoRR abs/1512.02325 (2015). URL: http://arxiv.org/abs/1512.02325. arXiv:1512.02325.

[30] F. Mazzenga, R. Giuliano, F. Vatalaro, FttC-based fronthaul for 5G dense/ultra-dense access network: Performance and costs in realistic scenarios, Future Internet 9 (2017).

[31] A. Simonetta, M. Paoletti, Designing digital circuits in multi-valued logic, International Journal on Advanced Science, Engineering and Information Technology 8 (2018) pp. 1166–1172.

[32] G. Capizzi, F. Bonanno, C. Napoli, Hybrid neural networks architectures for soc and voltage prediction of new generation batteries storage, in: 2011 International Conference on Clean Electrical Power (ICCEP), IEEE, 2011, pp. 341–344.