Issue 12, 2022

https://doi.org/10.23939/

UDK 004.89

EXPLOIT COMPUTER VISION INPAINTING APPROACH TO BOOST DEEP LEARNING MODELS

Mykola Baranov[1], Yurii Shcherbyna[1], Oles Hodych[2]

[1] Ivan Franko National University of Lviv
[2] Fielden Management Services Pty. Ltd
E-mail: mykola.baranov@lnu.edu.ua, ORCID: 0000-0003-1509-2924
E-mail: yuriy.shcherbyna@lnu.edu.ua, ORCID: 0000-0002-4942-2787
E-mail: oles.hodych@gmail.com, ORCID: 0000-0003-0106-8077

In today's world, the amount of available information grows exponentially every day. Most of this data is visual data. Correspondingly, the demand for the algorithm of image rent is growing. Traditionally, the first approaches to computer vision problems were classical algorithms without the use of machine learning. Such approaches are limited by many factors. First of all, the conditions imposed on the input images are applied – the shooting angle, lighting, position of objects on the scene, etc. Other classical algorithms cannot meet the needs of modern computer vision problems.

Neural network approaches and deep learning models have largely replaced classical programming algorithms. The greatest advantage of deep neural networks in computer vision tasks is not only the possibility of automatically building data processing algorithms that cannot be built in any other way, but also the comprehensiveness of such an approach – actual deep neural networks provide all stages of image processing from start to finish. But. This approach is not always optimal. Training models require a large amount of annotated data to avoid the effect of overfitting such models. In many settings, the conditions have a significant degree of variability, but are limited. In such cases, the combination of both approaches of computer vision is fruitful – pre-processing of the image is performed by classical algorithms, and prediction (classification, object search, etc.) is performed by a neural network.

This article noted an example of the use of damaged images in the classification of tasks (in the extreme cases, the percentage of damage reached 60 % of the image area). We have shown in practice that the use of classic approaches for restoration of damaged areas of the image (inpainting) made it possible to increase the final accuracy of the model by up to 10 % compared to the base model trained under identical conditions on the original data.

Key words: convolution neural networks; image inpainting; image classification.

Introduction

Image processing tasks includes a various subset of computer visions – image classification, object detection, image segmentation, clustering, scene recognition, etc. Despite a difference of such tasks, they exploit almost the same tool set. Examples of such tools may be 2D feature detection [1], contour processing [2], edge detections, threshing, etc. There are limited number of such fundamental processing operations, but all of them are so flexible, so it could satisfy almost every case or task. Having that, combination of the fundamental image processing operations along with its result processing and analyzing gives as a generalized powerful framework for computer vision. All common operations are implemented in such computer vision libraries as OpenCV, sklearn, etc.

The practical issue with such framework that it is so flexible, so in various of complex tasks (like face recognition, arbitrary objects detection or segmentation) it is almost impossible to find the configurations of those operations. That is a fundamental issue of classical computer vision approach.

Modern machine learning approaches gives us an opportunity to automatically built an image processing algorithms (often in end-to-end manner). Actually, it is typically a symbiosis of computer vision framework and optimization processes suggested by machine learning. One of the most powerful AI tools for computer vision is a convolutional neural network. It is implemented based on the idea that sequence of image convolution is able to extract any useful features having the right kernels (filters). Since we cannot directly estimate what kernels are needed for specific task or dataset, CNN propose optimizing the kernels of convolution via gradient descent methods. Thus having an enough amount of annotated data it is possible to discover needed convolution in fully automatically manner. Actually, model are prone to overfitting – finding such kernels that do fit training dataset by performing poorly on validation sets. Usually, it is fixing by adding generalization to the network, hyperparameter tuning. But there is no universal approach to improve performance of network.

In this paper, we analyze a convolution operation from mathematically perspective and found a limitation of convolution layers in specific cases – it suffers from a sharpen edges on the images. We propose usage of a classical computer vision algorithm as a data pre-processing step. We showed that using such pre-processing gives us an 8 % accuracy boost of exactly the same convolution neural network.

Methodology

A convolution network consists of a sequence of convolution operations followed by activation layers (those layers are needed since convolution operation form group over input matrices, i.e., a sequence of convolution may be composed into one single convolution operation).

Bare convolution operation is described by the following formula:

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\, d\tau$$

where *f* stands for input function and *g* denotes a convolution kernel. In case of computer vision, the image is represented as a 2-dimension *f* function from coordinates *x* and *y*. Thus, we have to deal with 2D convolution operation:

$$f(x,y) * g(x,y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1,\tau_2) \cdot g(x-\tau_1, y-\tau_2)\, d\tau_1\, d\tau_2$$

Such operations allow us to control convolution strength by the horizontal and vertical axis separately.

Having that convolution notations expressed though the integration, we can easily draw a limitation of such convolution – both functions *f* and *g* should be continuous since it should be integrated. The simplest example of a discontinuous function is a step function. We will put this example on practice in the next section.

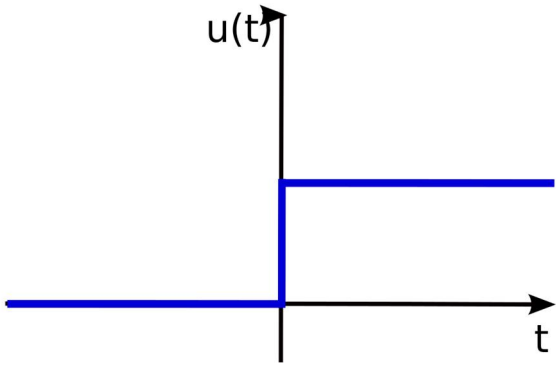

Fig. 1. Diagram of the model pipeline

In the practice of computer vision, we are usually dealing with discrete images. It means, that there is no technical possibility to face a discontinuous image, since we apply a discrete convolution:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

But general idea that there is a problem of applying a convolution discontinuous function gives as an intuition to be careful even on images - at those areas where a gradient is too high (we may expect there a some kind of step function).

Dataset

In this paper, we set up our experiments with an open-source Intel Image Classification dataset. This is image data of Natural Scenes around the world. It contains around 25 000 images of size 150×150, distributed under 6 categories:

1. Buildings.
2. Forest.
3. Glacier.
4. Mountain.
5. Sea.
6. Street.

Note, that classes are not semantically distributed, e.g., mountains and glacier is much similar than mountain and sea, so it gives as extra bias of data classes.

Fig. 2. Intel Image Classification dataset samples

There are a lot of image classification approaches: VGG [3], ResNet [4], Inception [5], EfficientNet [6], etc. Those models can easily achieve up to 95 % class accuracy.

But in this paper we are focused on corrupted data. ML Olympiad context hosted by Google Inc. on the kaggle platform provide us a corrupted image dataset based on Intel Image Classification dataset. The data contains masked images with random patches. These random patches can cover anything between 20 % to 60 % of the image. All images were resized up to 256×256 pixels. Here on Fig. 3 are example of corrupted images:

Fig. 2. Example of corrupted images

Image pre-processing approach

Obviously, we have much fewer data in comparison to original animation. It makes the classification task much harder since we have fewer features, and it is more probably to confuse glacier and mountain. But we discover one more problem with such a setup. Moreover, it is more fundamental than previous once. Considering the image as a function, indeed a discrete function from coordinates x and y, edges of corrupted regions forms some sort of discontinuous. Technically, it is not so, since there are discrete values, but intuition of such corruptions suggest us to pay attention to the edges of missed areas. We suppose, that quite big gradients produced by masks may be harmful for convolution (one more time going out of convolution operation expressed by integral). So, we propose to pre-process such missed regions with a classic free open source inpainting algorithm provided by OpenCV.

We processed both training and validation parts of the dataset, so the results is shown on the Fig. 3.

Fig. 3. Example of inpainting result

Note, that there is no new information, corrupted regions are just deducted from its surrounding thus it helps model just to uniform data and avoid discontinuous of image.

On Fig. 4 we demonstrate how does inptaining poisonous an image:

We trained a MobileNet image classifier on the pre-processed image data with a cross-entropy loss over soft-max activation. In order to perform stable evaluation of our model, we trained the model using K-Fold validation with $K = 5$.

We also evaluate a baseline model of exactly the same architecture on the original data without fixing discontinuous. The result of comparison is provided in the Table 1.

Fig. 4. Example of inapinting poisoning

Table 1

Experiments details overview

Dataset	Accuracy
Original	0.823
Preprocessed ***(our)***	**0.923**

Here we got up to 10 % accuracy boost by an inpainting of corrupted data.

Conclusions

In this paper, we take a close look at the limitation of convolution in terms of applying it to discontinuous function. We discovered that even discrete function may suffer from such issue. Fixing that on practice gives as a significant accuracy boosting in comparison to baseline model. We also discover the same issue with ha binary images - images where edges are so sharp without any smooth transition.

We also show that despite a powerful technology of deep learning, it is possible to obtain a better model by exploiting a classic technics of computer vision as a preprocessing step,

References

1. Merino, Ibon & Azpiazu, Jon & Remazeilles, Anthony & Sierra, Basilio (2020). 2D Image Features Detector And Descriptor Selection Expert System. DOI: 10.5121/csit.2019.91206.

2. Gong, Xin-Yi & Su, Hu & Xu, De & Zhang, Zhengtao & Shen, Fei & Yang, Hua-Bin (2018). An Overview of Contour Detection Approaches. *International Journal of Automation and Computing*, 15, 1–17. 10.1007/s11633-018-1117-z. DOI: 10.1007/s11633-018-1117-z.

3. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. DOI: 10.1109/TPAMI.2015.2502579.

4. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778. DOI: 10.1109/CVPR.2016.90.

5. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9. DOI: 10.1109/CVPR.2015.7298594.

6. Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. *In International conference on machine learning*, 6105–6114. PMLR. DOI: 10.1109/ECTI-CON54298.2022.9795496.

ВИКОРИСТАННЯ КЛАСИЧНИХ ТЕХНІК ВІДНОВЛЕННЯ ВТРАЧЕНИХ РЕГІОНІВ ЗОБРАЖЕННЯ ДЛЯ ПОКРАЩЕННЯ РОБОТИ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ

Микола Баранов[1], Юрій Щербина[1], Олесь Годич[2]

[1] Львівський національний університет імені Івана Франка
[2] Fielden Management Services Pty. Ltd
E-mail: mykola.baranov@lnu.edu.ua, ORCID: 0000-0003-1509-2924
E-mail: yuriy.shcherbyna@lnu.edu.ua, ORCID: 0000-0002-4942-2787
E-mail: oles.hodych@gmail.com, ORCID: 0000-0003-0106-8077

У сучасному світі щодня кількість наявної інформації зростає експоненційно. Велика частина цих даних належить до візуальних даних. Відповідно зростає попит на алгоритми опрацювання зображень. Традиційно першими підходами до задач комп'ютерного зору були класичні алгоритми без використання машинного навчання. Такі підходи зазвичай обмежені багатьма чинниками. Це стосується насамперед умов, накладених на вхідні зображення, – ракурс знімання, освітлення, положення об'єктів на сцені тощо. З іншого боку, класичні алгоритми не можуть задовольнити потреби сучасних задач комп'ютерного зору.

Нейромережеві підходи та моделі глибинного навчання багато в чому замінили класичне програмування алгоритмів. Найбільшою перевагою глибоких нейронних мереж у задачі комп'ютерного зору і не тільки є можливість автоматичної побудови алгоритмів оброблення даних, які неможливо побудувати іншим способом, а й всеосяжність такого підходу – зазвичай глибинні нейромережі виконують усі етапи оброблення зображень від початку до кінця. Проте такий підхід не завжди оптимальний. Для тренування моделей необхідна наявність великої кількості проанотованих даних, щоб уникнути ефекту перенавчання таких моделей. У багатьох задачах для умов середовища характерний значний ступінь варіативності, проте вони є обмеженими. У таких випадках плідною є співпраця обох підходів комп'ютерного зору – попереднє оброблення зображення виконують класичні алгоритми, а безпосередньо передбачення (класифікація, пошук об'єктів тощо) – нейромережа.

У статті розглянуто приклад використання пошкоджених зображень у задачі класифікації (у найгірших випадках відсоток пошкодження досягав 60 % площі зображення). Ми показали на практиці, що використання класичних підходів реставрації пошкоджених ділянок зображення (inpainting) дало змогу покращити фінальну точність моделі до 10 % порівняно з базовою моделлю, тренованою у ідентичних умовах на оригінальних даних.

Ключочі слова: згорткові нейронні мережі; реставрація зображень; класифікація зображень.