논문 2021-1-8 http://dx.doi.org/10.29056/jsav.2021.06.08

A Study on Deep Learning Model-based Object Classification for Big Data Environment

Jeong-Sig Kim*, Jinhong Kim**†

Abstract

Recently, conceptual information model is changing fast, and these changes are coming about as a result of individual tendency, social cultural, new circumstances and societal shifts within big data environment. Despite the data is growing more and more, now is the time to commit ourselves to the development of renewable, invaluable information of social/live commerce. Because we have problems with various insoluble data, we propose about deep learning prediction model-based object classification in social commerce of big data environment. Accordingly, it is an increased need of social commerce platform capable of handling high volumes of multiple items by users. Consequently, responding to rapid changes in users is a very significant by deep learning. Namely, promptly meet the needs of the times, and a widespread growth in big data environment with the goal of realizing in this paper.

keywords : Big Data, Social Commerce, Deep Learning, Predictive Model, Object Classification

1. Introduction

Ten years ago, it has marked a significant growth of the social commerce automation, driven by the growth of the social market. The significant change from offline shopping to online shopping anywhere and anytime has caused a change in appearance order fulfillment process. This paradigm shift has sparked a widespread growth in automation systems, with the goal of realizing fully autonomous order fulfillment systems [1-5]. The biggest challenge in realizing fully autonomous order fulfillment systems lies in the identification and handling of foreign objects. Current image segmentation techniques provide sufficient accuracy to handle foreign objects for pick and place operations. These image segmentation techniques could be identify and distinguish geometrical shapes and edges that are sufficient for robust handling of foreign objects. However, these features are insufficient for the identification of foreign objects [6-8], and this limitation in technology necessitates the presence of a manual worker that provides the identification of foreign objects in an order fulfillment process [9]. In order to full automation to be realized, their system should be able to autonomously recognize the object, so that the object can be transported to its corresponding destination. Accordingly, we

* 경기과학기술대학교 컴퓨터모바일융합공학과
** 배재대학교 지능SW공학부 컴퓨터공학과
† 교신저자: 김진홍(email: jinhkm@pcu.ac.kr)
접수일자: 2021.06.02. 심사완료: 2021.06.14.
게재확정: 2021.06.20.

propose object classification using deep learning prediction model for social commerce for big data environment.

2. Machine Learning

ML(Machine learning) is a field that is focused on the construction of algorithms that make predictions based on data [10]. A machine learning task aims to identify a function : X to Y that maps the input domain X onto output domain Y. Functions f are chosen from different function classes, dependent on the type of learning algorithm that is being used [11].

2.1 MLA(Machine Learning Algorithms)

MLA(Machine learning algorithms) can be classified into three categories by data sets type largely that are used as experience. SL(Supervised Learning) systems make use of labeled data sets $(x, y) \in X \times Y$, where x represents a data point and y the corresponding true prediction for x. This training set with both input and output pairs is used to find a deterministic function that maps any input to an output, predicting future input and output observations while minimizing errors as much as possible. UL(Unsupervised Learning) systems use unlabeled data sets to train the system. The objective of UL is to derive structure from unlabeled data by investigating the similarity between pairs of objects. It is usually associated with density estimation or data clustering [12].

2.2 OSLA: Optimal Supervised Learning Algorithm Selection

To determine which machine learning techniques are suitable for a particular application [13], it can analyze the necessary aspects of an optimal supervised machine learning pipeline. It describes a pipeline that can be used to create successful classifiers that generalize well for new data instances. This pipeline is shown in Figure 1.

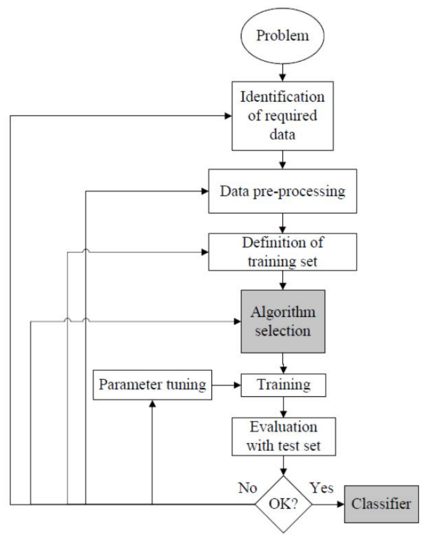

Fig. 1. SML(Supervised Machine Learning pipeline) Flow

The first two steps of this pipeline are the most important and largely define the performance of the classifier. Identifying required data consists of determining and choosing the most relevant features [14]. By excluding irrelevant or redundant features, data dimensionality can be reduced. Too much irrelevant or redundant information can prevent a learning algorithm from finding patterns in data or even result in false results. The pre-processing step is used to deal with this information redundancy, and is often used to deal with noise or missing values. The end result of the pre-processing step is the input of the training dataset.

2.3 Processing Datasets of OSLA

A method to describe the data sets that are used to train and test machine learning algorithms is with a design matrix. The design matrix is a matrix that consists of all data points, where each column can correspond to a certain feature. For example, if one uses a set of 10 photos taken of an object with a resolution of 1600 x 1200 with three features for each photo, the dataset can be represented with a design matrix $X \in R^{10*1600*1200*3}$. Data sets that are used to train and test machine learning algorithms vary from relatively simple, to large and complex data sets. An example of a relatively simple dataset is the Iris dataset. This dataset contains 150 samples, each with four data instances and can be described by a design matrix $X \in R^{150*4}$. On the other hand, when dealing with photographs data sets can become very large due to high image resolutions. An image with a resolution of 1900 × 1080 pixels, with each pixel representing a data point with an x, y and z value can easily produce an amount of 6156000 data points per image. Due to the large amount of data points per example, a lot of processing power is necessary to train and test the machine learning algorithm. As the amount of data produced keeps growing each day, more and more complex data can be used to train machine learning algorithms. In the field of computer vision, data sets can consist of up to thousands of images that can each again contain dozen of features.

3. Object Identification Framework

A new object identification framework is discussed. Our system requirements are used to define the desired objective, input and outputs of the framework real world. In order to build the barcode localization framework, the input, output and class definition are first defined. It makes automation solutions for different markets: parcel, warehouse and airports, which means that objects that these solutions handle generally exhibit a large variety in shape, color, weight and other physical properties. It is important to realize that for this project, the focus lies on the warehousing segment. Furthermore, products are chosen from standardized categories, because this avoids ambiguities regarding the class definition and interpretation.

The dataset that is created consists of images of box category item, taken from a top side view, and the images will be imported in the framework as RGB data, with a resolution of 400x300. Thus input data will consist of a design matrix X 2 $R^{N*400*300*3}$, where 'N' represents the number of samples in the data set. By the convolutional neural networks, these are a type of supervised learning, labeled data is needed for the training phase of the framework. In order to label all data efficiently, the following class definition is chosen:

$$C_n = 1,2,3,4,5,6$$

In this class definition each of the numbers represent one of the six faces of a box sized object. This class definition is always applied from the perspective of the camera. For all cases, the face that is closest to the camera is defined as the top face (c=1), from there on the other faces are defined. The face opposite of the top face is the bottom face (c=3). The faces adjacent to the top and bottom faces are the front (c=2), back (c=4), right (c=5) and left face (c=6). With the camera perspective in mind, the definition is interpreted and illustrated as follows in Figure 2:

C_n = {top, front, bottom, back, right, left}

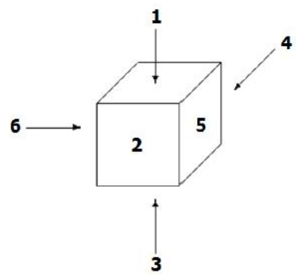

Fig. 2. Proposed object Recognition Framework

4. Deep Learning Prediction Model

Convolutional neural networks is currently one of the most prominent algorithms for deep learning with image data. Whereas for traditional machine learning relevant features have to be extracted manually, deep learning uses raw images as input to learn certain features. CNNs consist of an input- and output layer, and several hidden layers between the input and output. Examples of in between layers are convolutional layers, max-pooling layers and fully connected layers. CNN architectures vary in the number and type of layers implemented for its specific application. For continuous responses, the network should include a regression layer at the end of a network, whereas for categorical responses the system must include a classification function and layer. Neurons in each CNN layer are arranged in a 3D arrangement, and transform a three dimensional output from a three dimensional input. For our particular application, the input layer holds the images as 3D inputs, with the height, width and RGB values as dimensions. Hereafter, in the convolutional layer neurons are attached to the regions of the image and transformed into a three dimensional output, see Figure 3.

CNN configurations comprise of a multitude of hidden layers. In each layer, activation volumes are altered with the use of differentiable functions. Four principle layer types exist that are used to build CNN configurations.

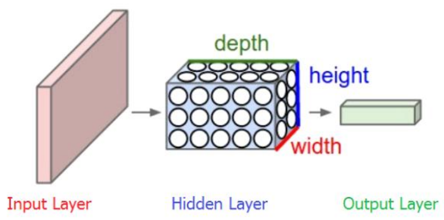

Fig. 3. CNN-based image transformation into a 3D arrangement

1. Convolutional Layer (CONV): Convolu-tional filters are used to derive an activation map from the input data.
2. Rectified Linear Unit Layer (ReLU): Filters negative values to provide only positive values for a much faster training time.
3. Pooling Layer (POOL): Performs nonlinear down-sampling and cuts down the amount of parameters for a simpler output.
4. Fully Connected Layer (FC): Computes the class probability scores by outputting a vector of C dimensions, with C being the number of classes. All neurons are connected to this layer.

When importing the labeled dataset, a division of 75%/25% is made between the training data and validation data. This means that 75% of the data is used to train the network, and 25% of the data is used to validate the network.

After a sufficient validation accuracy is achieved, an additional test set can be used to see how well the network performs. The first configuration is ran on a CPU based computation model. The CPU used is an Intel Core i7 720QM processor, with four physical and four virtual cores. The training progress and result are shown in Figure 4.

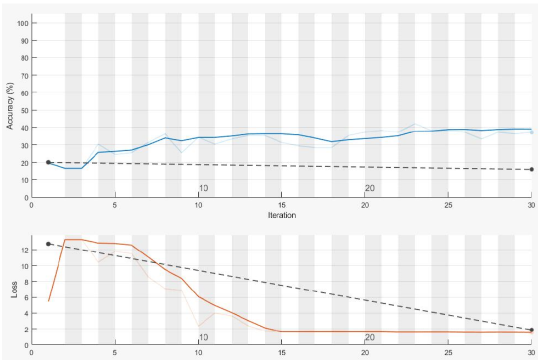

Fig. 4. CNN training with the use of a CPU

Even though the CPU used has four physical cores, it can be immediately noted by the elapsed time, that training with a CPU still takes an extreme amount of time and is not really feasible for fine-tuning in the scope of this project. Even when a CPU consists of multiple cores and is top of the line, high processing times still remain an issue. Looking at the test results, the validation accuracy of 16% is very insufficient and it looks like the CNN has not learned any valuable features that can be used for correct classification. By altering the amount of features that the convolutional layers calculate, and the filter size that the convolutional layer uses, the system is fine-tuned further to increase validation accuracy. However, from the second run on a GPU based computation is taken. For the following runs, a NVidia GTX1060 GPU is used and an Intel i7-6700HQ CPU is used. The result are shown in Figure 5.

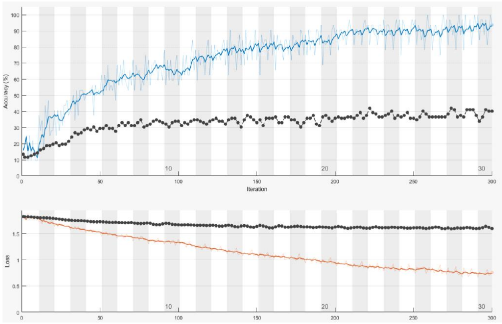

Fig. 5. CNN training with the use of a CPU ; System parameters are already tweaked a bit, by decreasing filter size and number of filters a validation accuracy of 40.18% is reached with a dazzling training time.

What can be noted immediately is that the computation time decreased from 431 minutes to about 1 minute. This means that the computation time decreased with a factor of around 400. This at the very least allows for efficient fine-tuning of the CNN. Looking at the validation results, an accuracy of 40.16% is achieved. While the training set accuracy does increase to around 100%, the validation accuracy keeps oscillating around 40%. Comparing the training loss with the validation loss it can be seen that although the training set does converge to zero, the validation set stays around 1.75. Further finetuning is done to the CNN to investigate whether altering the learning rate, minibatch or epoch size, or layer configurations and settings can improve the CNN even more. This is done numerous times consecutively, showing some of the interesting results.

5. Conclusions

In this paper, we proposed Convolutional neural networks that it is the most prominent algorithms for deep learning with image data. Reflecting back on the proposed our research in light of the defined requirements and constraints, it is concluded that a sufficient accuracy is achieved of 44.64%, higher than the requirement of 40%. In order to reach a framework that robustly automates these phase of the order fulfillment process, the classification accuracy of the CNN has to be improved. The logical conclusion that could be made from the training results, is that the dataset has to be drastically increased to prevent overfitting of the data. Furthermore datasets of all other product categories have to be added to the CNN for the framework to be implemented. Although generating and collecting this data might take a lot of time, from the results that have been achieved up until now it can be seen that even with small datasets, promising results are shown. Furthermore, we will try to do improvement this various deep learning method for restrict environment.

References

[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Communications of the ACM, Volume 60, Issue 6, pp.84 - 90, June 2017. https://doi.org/10.1145/3065386.

[2] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks" in Advances in Neural Information Processing Systems, pp.91-99, 2015. https://ieeexplore.ieee.org/abstract/document/7485869.

[3] Y. Li, H. Qi, J. Dai, X. Ji and Y. Wei, "Fully convolutional instance-aware semantic segmentation", CVPR, 2017. https://openaccess.thecvf.com/content_cvpr_2017/html/Li_Fully_Convolutional_Instance-Aware_CVPR_2017_paper.html.

[4] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge", International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015. https://link.springer.com/article/10.1007/s11263-015-0816-y.

[5] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: Common objects in context", European Conference on Computer Vision, Jan. 1, 2014. https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48.

[6] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo", CVPR, pp.3485-3492, 2010. https://ieeexplore.ieee.org/abstract/document/5539970.

[7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The pascal visual object classes challenge: A retrospective", International Journal of Computer Vision, vol. 111, no. 1, pp.98-136, 2015. https://link.springer.com/article/10.1007%252Fs11263-014-0733-5.

[8] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite", CVPR, 2012. https://ieeexplore.ieee.org/abstract/document/6248074.

[9] S. R. Richter, V. Vineet, S. Roth and V. Koltun, "Playing for data: Ground truth from computer games", European Conference on Computer Vision, 2016. https://link.springer.com/chapter/10.1007/978-3-319-46475-6_7.

[10] P. Dollar, C. Wojek, B. Schiele and P. Perona, "Pedes-trian detection: An evaluation of the state of the art", IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, pp.743-761, 2012. https://ieeexplore.ieee.org/abstract/document/5975165.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection" in CVPR, IEEE Computer Society, vol. 01, pp.886-893, 2005. https://ieeexplore.ieee.org/abstract/document/1467360.

[12] M. Everingham, A. Zisserman, C. K. I. Williams et al., "The 2005 PASCAL visual object classes challenge", First PASCAL Machine Learning Challenges Workshop, pp.117-176, 2006. https://link.springer.com/chapter/10.1007/11736790_8.

[13] K. Gauen, R. Rangan, A. Mohan, Y. H. Lu, W. Liu and A. C. Berg, "Low-power image recognition challenge", Asia and South Pacific Design Automation Conf., pp.99-104, 2017. https://ieeexplore.ieee.org/abstract/document/7858303.

[14] M. Taiana, J. C. Nascimento and A. Bernardino, "An improved labelling for the inria person data set for pedestrian detection", Pattern Recognition and Image Analysis: Iberian Conference, pp. 286-295, 2013. https://link.springer.com/chapter/10.1007/978-3-642-38628-2_34.

Authors

김정식(Jeong-Sig Kim)

2007.2 Ph.D. in Electrical, Electrical and Computer Engineering from Sungkyunkwan University
2012.3-Present : Professor in Gyeonggi University of Science and Technology

<Research Interests> Wireless Sensor Networks, Mobile Computing, Network Security Protocols, Proxy Caching Systems

김진홍(Jin-Hong Kim)

2006.2 Ph.D. in Electrical, Electrical and Computer Engineering from Sungkyunkwan University
2017.3-2020.2 Assistant Professor in Dept. of Information Communication Eng., Seoil University
2020.3-Present Assistant Professor in Department of Computer Engineering, Paichai University

<Research Interests> Big Data, Artificial Intelligent, Intelligent Software, Smart Vehicular Network, Smart Platform