Using Pre-Trained Models for Fine-Grained Image Classification in Fashion Field

Anna Iliukovich-Strakovskaia
Moscow Institute of Physics and Technology
1 "A" Kerchenskaya st., Moscow, 117303, Russian Federation
+7 495 408 45 54
strakovskaya.am@phystech.edu

Alexey Dral
Moscow Institute of Physics and Technology
1 "A" Kerchenskaya st., Moscow, 117303, Russian Federation
+7 495 408 45 54
dral@phystech.edu

Emeli Dral
Moscow Institute of Physics and Technology & Yandex Data Factory
1 "A" Kerchenskaya st., Moscow, 117303, Russian Federation
+7 495 408 45 54
emeli.dral@phystech.edu

ABSTRACT

Fashion image classification offers online stores a fast and effective way to manage the volume of their products. It could be used in product categorization by their brands, types, styles, etc. It could be also used to make fashion ensembles of clothes and even to create product recommendations. The results of apparel image classification are also useful for such purposes as product filtering and search. A wide range of different models have been successfully applied to coarse level image classification, but only a few of them could be applied to fashion images due to the nature of the data: such images could belong to the one high-level topic such as 't-shirt', 'skirt' or 'sport shoes', but at the lower level be fine-grained (e.g. 'high heels', 'medium heels'). Each apparel shop can have its own fine-grained hierarchy leading to difficulties to acquire a big dataset. This work shows necessity of pre-trained model usage to achieve a good predictive quality. In addition, the article contains a description and an experimental study of an approach to stack features from pre-trained model and features learned directly from an image pixel representation. This approach significantly outperforms baseline methods in fine-grained image classification [1, 2, 3] and a classification based only on a pre-trained model.

CCS Concepts

• Machine learning • Machine learning approaches • Neural networks • Supervised learning • Supervised learning by classification • Ensemble methods • Bagging • Transfer learning

Keywords

Fashion; machine learning; image classification; neural networks; fine-grained image classification

1. INTRODUCTION

Image classification is one of the central tasks in the computer vision area and apart from that image classification is widely used for many different applied cases. Nowadays there are a lot of information systems based on image classification: face recognition systems, objects detection, age recognition, adult content filtering, etc. In the fashion area image classification also became a very popular and important task, because it could be used for a number of applications. As an example, apparel image classification is widely used by online shops for content filtration and categorization (e.g. filtration of clothes by style, season) [19]. Moreover, it is used to make fashion recommendations (e.g. recommendations about stylish ensembles of clothes) [18]. All these tasks are crucial for fashion online stores. Things such as qualitative fashion recommendation, convenient navigation through the amount of available products and good categorization of products significantly influence user experience. User experience directly influences business KPIs (Key Performance Indicators) such as the volume of purchase, the average checkout value. Therefore, it is extremely important for an online apparel store to have good services based on image classification. By improving the quality of such services (e.g. by improving the quality of fashion image classification) businesses can sometimes directly improve their KPIs and gain superior results.

Image classification models usage in the fashion industry can not only bring benefits to end-users, but can also save resources spent on business maintenance. Let us consider the following case: we have a very large online clothes stores aggregator. It means that we have one place (one site) where we can find products from many different shops and buy the products all together in a single order. For unification and quality standard purposes such aggregators might have some rules and limitations regarding products which might be placed on their site. If such limitations exist, aggregators need a mechanism to check if all incoming products match to the limitations before be placed on their pages. Such a mechanism is usually called a "moderation" – a process, that checks all new incoming products and as a result allows or denies it. Basically, the moderation process can be done by humans. We can attract some experts to do it, but such an approach to the moderation process has two big weaknesses. Firstly, we have to pay each expert for his or her work and it can be quite expensive. The second problem is the speed of such work. When businesses are developed then the number of objects to be moderated increases constantly and maintaining a good performance of the moderation process becomes more difficult. In such a situation we can either agree on increased time for the moderation process or attract more experts to support good performance (which basically leads us back to the first problem). All these arguments lead us to a simple decision: the automation of such work.

We can also use an image classification model to develop a system for content auto moderation. For instance, a model can be trained to distinguish between two classes of images: "approved" and "denied". Such a system can dramatically reduce the cost for the moderation process, which is basically desired by any business model.

All these observations lead us to a conclusion that image classification is an important task for the fashion industry, especially for its online presence. In this paper, we propose an approach for fine-grained fashion image classification applicable for small datasets based on usage of pre-trained neural network and model stacking.

2. RELATED WORK

Some image classification methods are based on low-level image representation. Such methods usually consider image as a set of low-level features such as size, shape, color, texture, etc. For example, bag-of-word is a very popular way to obtain image numeric representation, where each image representation vector component is related to a visual word. Thus each image gets a numeric vector representation by counting the frequency of visual words in the image. Then a predictive model can be learned based on such numeric representation of an image.

One another collection of classification methods are based on a mid-level features representation. Aforementioned methods consider image as a set of pre-trained generic object detectors. There is a number of works, where attributes are used as mid-level representations for images and videos [1, 2, 3]. In [3] authors proposed an approach for mid-level visual features construction for image classification. An image is represented as a vector consisting of outputs from a collection of binary classifiers. These binary classifiers are trained to differentiate pairs of object classes in an object hierarchy. Similar approach is used in the development of the Classme descriptor [4], where images are represented by the output of a large number of weakly trained object classifiers.

Nowadays, usage of neural-networks and deep neural-networks to obtain image representation is trending. Such approach allows us to extract features from a specified layer of trained neural network and then use extracted features as a numeric image representation. There is a number of works related to the image processing with neural networks [8 - 13].

Our work is related to the line of research, where deep neural-networks are used for image classification.

3. PROBLEM STATEMENT

When average online clothes store starts building image classification model, it faces a number of serious problems. Firstly, you have to obtain train and test datasets before you can train any classification or regression model. Such datasets should be large enough to train a model, especially when we speak about multiclass classification with a wide range of classes. Not any clothes store can collect such dataset itself. The reason of it is a lack of available data and difficulties to acquire it. It could take months to collect dataset with an appropriate size from store's own data.

The next challenge to overcome is to build a prediction model with a good enough quality. It is a challenging task when we speak about image classification. We have to use advanced machine learning algorithms to process such a complicated type of content. To name a few, there are deep neural-networks, model ensembles such as bagging or boosting techniques. Such models have a lot of different parameters and hyper-parameters to optimize. Even if someone has an appropriate dataset, he or she still needs to have an expertise to obtain an accurate and reliable model. For example, if we train deep neural-network classification model, we have to set a structure of the network in advance, define a number of layers, order of convolution layers, and so on. Worth mentioning that all these parameters influence the result dramatically.

Finally, when we speak about fashion image classification, we should keep in mind that in most cases we work with fine-grained objects' classification. It means that such objects could all belongs to the one high-level topic such as 't-shirt', 'dog' or 'cat', but at the lower level be fine-grained. For example, if we work with a dataset about cats, at the high level all objects are related to the same category 'cat', but if we need to distinguish between different breeds of cats, we move to the lower level of topics such as 'American Bobtail', 'Bengal' or 'Maine Coon'. Working with fine-grained classification often requires a special approach, which takes into account hierarchical structure of the explored data.

In this work we focus on fashion image fine-grained classification for small datasets. We show drawbacks and shortcomings of simple and complex methods. Therefore, we show necessity of transfer learning usage in this task. Usage of an external dataset is the well-known and promising approach for image classification. In this work we use this approach and show how we can utilize available knowledge from initial dataset to achieve significantly better performance. We show what complexities we faced and how to overcome them.

In this paper we used several pre-trained deep neural-network models: Inception_BN [11] and Inception_21k. The first one is trained on 1000 topics from ImageNet [6]. The last one is trained on all 21000 topics from ImageNet. Pre-trained model usage allows us to achieve good enough image feature representation even for small collection of objects, where we simply do not have enough data to train a complex and strong model. Apart from that we extract low-level features from an initial dataset. Final classification model is built on joined image feature representations that allows us to benefit both from external and initial data.

4. DATA

For presented case we used Yahoo! Shopping Shoes dataset [5]: this dataset consists of 5250 images of 107 shoes classes. Each shoes class belongs to one of the 10 superclasses. For instance, in this dataset we have "high_heels" and "slipons" super-classes, and each of them contains 14 classes inside. For example, at 'high_heels" superclass we have "betsey_johnson", "jimmy_choo", "michael_kors" classes, etc. At "slipons" superclass we have "crocs", "ecco", "timberland" classes, etc. This dataset was collected as a small subset of products from Yahoo! Shopping Shoes to reflect the interesting real-world problem of fine-grained object recognition. That is why we have chosen it for our study.

We used cross-validation technique for all estimations we made. We randomly chose 90% of the examples for training and the remaining ones were used for testing. The same technique was used in [3]. Thus our results became comparable.

5. OUR APPROACH

An overview of our approach is illustrated in Fig. 1. Input image is consumed by several processing flows. In the first flow (we call it "left branch"), image considered as a feature vector of raw pixels and reduced to a low-dimensional feature space with help of standard dimensionality approaches. In the second flow (we call it "right branch"), image goes through pre-trained deep neural-network. Features from one of the layers (which are known to work best) are used in the following processing stages. Similar to the left branch, such layer serves a role of a dimensionality reduction algorithm. The main difference is that the right flow leverages a model trained on crafted external datasets. Finally, features from different flows are combined and used to fit a nonlinear classifier. In the article we will reference this model as a "Two-Flow Model".

When algorithm [3] works on top of low-level features such as color histogram, SIFT histogram, GIST, LBP; our approach works with even more low-level representation of an image – raw pixels. Nevertheless, our approach uses raw pixel image representation, "left branch" flow in isolation gives better quality than classifier trained on low-level histogram features.

Let us discuss in more details ideas behind model architectural choices. Quite standard image resolution is 640x480x3, which is 0.92 million features in total. In such a case we just cannot choose arbitrary dimensionality reduction algorithm. Far from all dimensionality reduction algorithms would be feasible to be used in a such installation. Even if we do not have a big collection of images we cannot easily afford usage of algorithms linear on one dimension of a dataset (e.g. number of features, P) and quadratic on another (e.g. number of images, N). As an example, not optimized version of PCA has a complexity $O(PN^2)$. As suggested in the article [16] Random Projection is a perfect fit for image preprocessing in such scenarios. In short, it has two major benefits: efficiency and effectiveness.

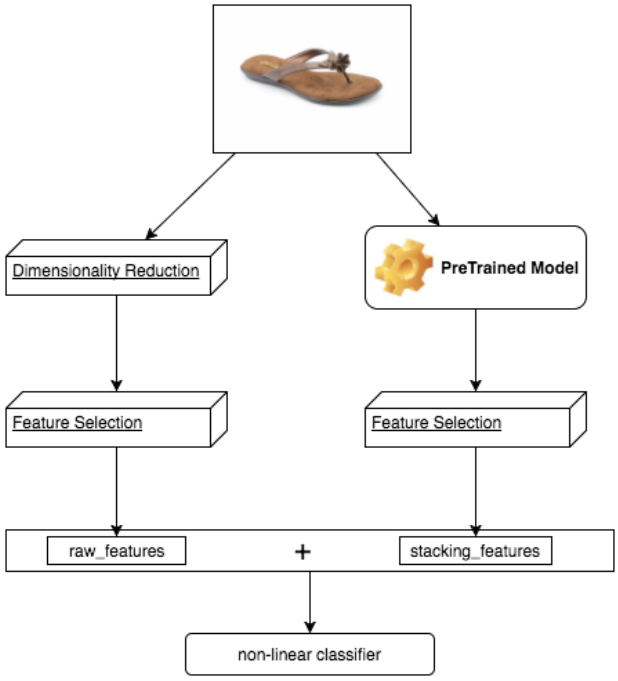

Figure 1. Architecture of the Two-Flow Model.

"Left branch" processing flow is focused to process raw data (image pixels). "Right branch" processing flow is focused on leveraging knowledge from models trained on external crafted image datasets. Finally features are combined and consumed by nonlinear classifier.

The choice of pre-trained model is not straightforward. Due to popularity of ImageNet competitions [17], a lot of fine-tuned pre-trained deep learning networks are available in the Internet. Each of them has its own benefits and shortcomings. In ideal scenario, you can find a model trained specifically for your domain of usage (e.g. animals, clothes, cars, …). In our experiments we were using pre-trained general purpose deep neural-network called Inception, which showed state-of-the-art prediction quality in ImageNet competition, 2014.

You can see "feature selection" steps in the classification pipeline in Fig. 1. Feature selection found to be beneficial in improving model generalization or testing accuracy. Random Forest was used at both final stages: "feature selection" and "nonlinear classifier". The reason is a two-fold. The first major benefit of trees as they can easily handle numerous classes in classification problems and provide feature importance for free (by ranking features by variable importance and choosing top K). The second major benefit is the following. Training ensemble of tress is easy parallelizable from multi-core machine to a multi-node cluster. Thus, you will be able to scale the model based on your business requirements.

Lastly, we do not combine and train deep neural-networks in our architecture, though it has its own merits. Deep neural-networks give state-of-the-art prediction quality, but require a lot of data (which is barely holds true in our experiments), need normalization and regularization to achieve a good performance. In the next section, we show for comparison purposes how much quality you can achieve with deep neural-networks without tuning on a small image dataset. Thereby, we show importance and ease of usage of pre-trained models.

6. EXPERIMENTS

We evaluated our approach on publicly available dataset Yahoo! Shopping Shoes [5]. Approach presented in [3] is used as a baseline as it provides the best quality on this dataset with mid-level features learning (64.7% accuracy). All accuracy scores presented in this section are reported on 107 classes classification task. In our experimental study we show the fallacy of a dumb feature stacking. And we show the benefits of presented dimensionality reduction algorithm usage. It is a fine-tuned tradeoff between low-level histograms and raw pixel image representations. Reported accuracy and comparison of different methods presented in Table 1.

Table 1. Comparison of overall accuracies of the Two-Flow Models with other approaches.

	mean (± std)	**gain**
Convolution NN on raw pixels	0.40	-38%
low-level histograms	0.44	-32%
random projections	0.48	-26%
Inception_BN (pool: global)	0.55	-15%
raw pixels	0.58	-10%
raw pixels + Inception_21k	0.60	-7.26%
Cao et al. [3]	0.624	-3.55%
Inception_BN (pool: 3c)	0.636	-1.70%
Baseline [3]	**0.647**	**--**
Inception_BN (pool: 5b)	0.658	+1.70%
Inception_BN (pool: 4e)	0.680	+5.10%
low-level + Inception_21k	0.688	+6.34%
Inception_21k	0.693 (± 0.011)	+7.11%
Two-Flow Model	0.705 (± 0.015)	+8.96%

We achieved 69.3% accuracy (+7.11% over baseline) with pre-trained deep neural-network and we show in this section how to boost classification quality to 70.5% (+8.96% over baseline) with stacking procedure described in the previous section. It is important to use Cross Validation to address deviation in measurements. For instance, test accuracy of Two-Flow Model varies between 67.6% and 73.5% on different splits.

Each name in a row describes features used to train Random Forest classifier (see section 5 for reasoning behind usage of this classifier) except the first row. Test accuracy of a convolution neural-network trained on raw images is 40% and train accuracy is 54%. In comparison, all other considered approaches can easily achieve 100% train accuracy. It is a well-known fact that tuning parameters such as initialization, momentum and regularization [13] are crucial for neural-networks to achieve good generalization accuracy. We trained neural-network without hypermeters tuning to show the accuracy you can achieve without much investment. The latest argument is an important factor in business domain.

Abbreviations used in the table:

- low-level histograms – collection of standard image representations such as color histogram, SIFT histogram, GIST, LBP;
- raw pixels – low-level pixel image representation. All images have size 640x480x3 (overall 0.92 million features per image);
- random projections – part of Two-Flow Model which uses Random Projections method to reduce dimensionality (we use from 32- to 1024- dimensional sub-space in different experiments);
- Inception_BN - pre-trained deep neural-network model, trained on 1000 topics form ImageNet with Batch Normalization [11] to speed-up convergence and increase model performance. There are different layers that can be used as a dimensionality reduction stage before classification. They are marked as (pool: *pool_name*) where *pool_name* corresponds to the pooling layer described in the article [12];
- Inception_21k - pre-trained deep neural-network model, trained on 21000 topics from ImageNet. Results presented in the table are based on the features from *global_pool* layer (1024 features in total);
- features_A + features_B (e.g. "raw pixels + Inception_21k) – Random Forest model learned on a collection of features.

Reading Table 1 from top to bottom we can make a conclusion about significance of usage different approaches for dimensionality reduction and stacking.

The first observation is that there is an expected and significant margin between usage of different pre-trained models. For instance, Inception_BN test accuracy varies between 55% and 68% (depends on feature layer used). While Inception_21k gives 69.3% prediction accuracy out-of-the-box on *global_pool* layer.

The second observation is that dumb feature stacking negatively influences prediction quality. It can be described by the fact that bigger number of features makes it easier to over-fit on the training set. So, the main challenge is to correctly combine different features extracted from existing dataset and features extracted from external pre-trained models. Test accuracy score based on "low-level histograms" and "raw pixels" are 44% and 58% correspondingly. The quality of Inception_21k is 69.3%. Stacking variants are named as "low-level + Inception_21k" and "raw pixels + Inception_21k". Instead of increasing quality we move back from 69.3% to 68.8% for stacking with low-level features and to 60% for stacking with raw pixels. It is quite difficult for a classifier to focus on good (informative) features, especially when we have 0.92 million features which contain a lot of noise and only 1024 informative ones.

Two-Flow Model provides an approach to overcome this limitation. We reduce raw pixels' image representation in a cost effective way to the size that we don't lose much information from the one hand side (48% accuracy) and can combine them with highly informative features in such a way that classifier will be able to utilize them without quality sacrifice (70.5% accuracy). The difference in accuracy between Two-Flow Model and the second best model (Inception_21k) is statistically significant. We used paired two-sample t-test. The null hypothesis is that the average quality of Two-Flow Model is equal to the average quality of Inception_21k versus two-sided alternative (that average model qualities are different). We got the following results: statistic = 3.027 and p-value = 0.013. It means that we can reject zero hypothesis on a pretty high significance level (for instance - 0.95). Confidence interval for the difference between model qualities is [0.3%, 1.7%] (calculated on each fold). It also shows that quality difference between Two-Flow Model and Inception_21k is higher than zero, and again shows that we have statistically significant quality gain.

Our final model has the following characteristics:

- "left branch" flow uses Random Projections to reduce 0.92 million features to 128. Random Forest classifier is used in a feature selection stage to choose most informative 55 features. Feature selection was performed by ranking features by variable importance and then by choosing top K whose value is above mean;
- "right branch" flow uses pre-trained Incpetion21k and use features from global pooling layer (1024 features overall). Similarly, Random Forest classifier is used to choose most informative 400 features;
- Random Forest is used in the final classification stage (nonlinear classifier in Figure 1).

From technical perspective, dimensionality reduction stage takes around 2-3 minutes for the whole Yahoo! Shopping Shoes collection on a standard laptop. Time to train a final classifier is even less than a minute (there are 200 trees in ensemble). To summarize, Two-Flow Model provides a cost-effective way to stack pre-trained models and a significant boost in prediction quality without much hassle.

7. CONCLUSIONS

The purpose of this work was to find out fast, effective and robust approach for fashion image fine-grained classification for small datasets. We conducted series of experiment on a small Yahoo! Shopping Shoes dataset consisting of 107 fine-grained classes to support presented approach. We used existing in the literature experimental studies based on mid-level feature representation as a baseline (62-65% accuracy). We showed fallacy of complex methods usage to achieve good performance on the small dataset (40-44% accuracy). Thus, we came to a conclusion of necessity of transfer learning usage (55-69% accuracy, it highly depends on pre-trained model).

Our approach uses pre-trained deep neural-network model to obtain image feature representation. This makes our approach applicable for small collections of images, because it allows us to obtain qualitative feature representation enriched with knowledge from external crafted datasets. We also use features extracted directly from the considered dataset. Feature stacking from several models based on external and internal data does not give better performance out-of-the-box. Moreover, it sometimes

can drastically reduce predictive quality. Presented Two-Flow Model provides a way to significantly increase classification accuracy (up to 71% accuracy).

In the future we are going to conduct a series of experiments to validate presented Two-Flow Model on different image collections and to scale this approach to address a diverse range of publicly available pre-trained models. Our research shows that presented approach allows us to achieve gain in classification accuracy which is statistically significant. We are going to explore the ways to significantly increase classification quality to be suitable for industrial usage from practical point of view.

8. ACKNOWLEDGMENTS

This work was supported by chair of Algorithms and Theory of Programming, Department of Innovation and High Technology in Moscow Institute of Physics and Technology (ATP DIHT MIPT).

9. REFERENCES

[1] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2009.

[2] J. Liu, Q. Yu, O. Javed, S. Ali, A. Tamrakar, A. Divakaran, H. Cheng, and H. S. Sawhney. Video event recognition using concept attributes. In IEEE Workshop on Applications of Computer Vision, 2013.

[3] Somayah Albaradei, Yang Wang, Liangliang Cao and Li-Jia Li 2014. Learning Mid-Level Features from Object Hierarchy for Image Classification.

[4] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In European Conference on Computer Vision, 2010.

[5] Yahoo research labs. Yahoo! Shopping Shoes Image Content, 2013.

[6] Full ImageNet Network. https://github.com/dmlc/mxnet-model-gallery/blob/master/imagenet-21k-inception.md

[7] Model http://data.dmlc.ml/mxnet/models/imagenet/

[8] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, Chu-Song Chen 2015. Deep Learning of Binary Hash Codes for Fast Image Retrieval.

[9] Peter Kokol, Mateja Verlič, Miljenko Križmarić 2006. Modelling Teens Clothing Fashion Preferences Using Machine Learning. In Proceedings of the 10th WSEAS International Conference on COMPUTERS (Vouliagmeni, Athens, Greece, July 13-15, 2006).

[10] Neal Khosla, Vignesh Venkataraman 2015. Building Image-Based Shoe Search Using Convolutional Neural Networks. Stanford University. CVPR2015 workshop.

[11] Sergey Ioffe, Christian Szegedy 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv preprint arXiv:1502.03167v3.

[12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich 2014. Going Deeper with Convolutions. arXiv preprint arXiv:1409.4842v1.

[13] Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning (Atlanta, Georgia, USA, 2013).

[14] Vinod Nair, Geoffrey E. Hinton 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. Department of Computer Science, University of Toronto, Toronto, ON M5S 2G4, Canada.

[15] Tianqi Chen, Mu Li, Tianjun Xiao, Yutian Li, Bing Xu, Min Lin, Naiyan Wang, Minjie Wang, Chiyuan Zhang, Zheng Zhang 2015. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems.

[16] Ella Bingham and Heikki Mannila 2001. Random projection in dimensionality reduction: Applications to image and text data; KKD.

[17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei 2014. ImageNet Large Scale Visual Recognition Challenge. arXiv prepring arXiv:1409.0575

[18] Hanbit Lee, Sang-goo Lee 2015. Style Recommendation for Fashion Items using Heterogeneous Information Network. RecSys 2015

[19] Wei Di, Catherine Wah, Anurag Bhardwaj, Robinson Piramuthu, and Neel Sundaresan 2013. Style Finder: Fine-Grained Clothing Style Recognition and Retrieval. CVPR 2013 workshop