Issue 11, 2022

https://doi.org/10.23939/sisn2022.11.103

UDK 004.89

COMPREHENSIVE ANALYSIS OF FEW-SHOT IMAGE CLASSIFICATION METHOD USING TRIPLET LOSS

Mykola Baranov[1], Yurii Shcherbyna[2]

Ivan Franko National University of Lviv
[1] e-mail: mykola.baranov@lnu.edu.ua, ORCID: 0000-0003-1509-2924,
[2] e-mail: yuriy.shcherbyna@lnu.edu.ua, ORCID: 0000-0002-4942-2787

Image classification task is a very important problem of a computer vision area. The first approaches to image classification tasks belong to a classic straightforward algorithm. Despite the successful applications of such algorithms a lot of image classification tasks had not been solved until machine learning approaches were involved in a computer vision area. An early successful result of machine learning applications helps researchers with extracted features classification which was not available without machine learning models. But handcrafter features were required which left the most complicated classification task impossible to solve. Recent success in deep learning allows researchers to implement automatic trainable feature extraction. This gave significant progress in the computer vision area last but not least. Processing large-scale datasets bring researchers great progress in automatic feature extraction thus combining such features with precious approaches led to groundbreaking in computer vision. But a new limitation has come - dependency on large amounts of data. Deep learning approaches to image classification task usually requires large-scale datasets. Moreover, modern models lead to unexpected behavior in distribution datasets. A few-shot learning approach of deep learning models allows us to dramatically reduce the amount of required data while keeping the same promising results. Despite reduced datasets, there is still a tradeoff between the amount of available data and trained model performance. In this paper, we implemented a siamese network based on triplet loss. Then, we investigate a relationship between the amount of available data and few-shot model performances. We compare the models obtained by metric-learning with baselines models trained using large-scale datasets.

Key words: few-shot learning; zero-shot learning; metric learning; computer vision; deep learning; image classification.

Introduction

Image classification tasks have a lot of real-life applications. Real-life images have uncountable factors of distortions, as well as an uncountable number of classes to be predicted. One more important note is that most natural-scene images cannot be directly classified with a handy-crafted algorithm (like edge detection [1], thresholding, morphology operations [2], etc). A basic machine learning approach allows researchers to engineer features and apply machine learning models to those features. Moreover, deep learning models automate a feature extraction process by incorporating it. Despite the great access achieved by using deep learning models, this approach cannot be directly applied to some business cases (especially when classes to be predicted changes time by time).

Few-shot learning approaches [3] dramatically decrease the amount of required training data to perform image classification. In metric learning [4] (a subfield of few-shot learning) it is done by rethinking the task of image classification: it is better to find the nearest image (in some metric space) from the training dataset rather than performing image classification from scratch. The key idea of such a

concept is that images of the same classes are samples from the same data distribution. The task of metric learning – is to build such a metric space that leads to tight similar images into one cluster. Moreover, following that approach, we can perform even zero-shot classification which means classifying previously unseen classes.

Methodology

Metric learning is a well-known approach to few-shot learning, especially for face recognition tasks. Suppose, you are working with a dataset A. So, your dataset A consists of pairs:

$$A = \{ (X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n) \},$$

where $X_i \in \mathbb{R}^{W \ H \times 3}$ stands from the input image (in RGB colorspace) and $Y_i \in \mathbb{N}$ represent class index of image X_i.

The task of metric learning is to construct such an operator $M(\cdot,\cdot)$ which satisfies the following condition:

$$M(A, B) = \infty \text{ if } A, \ B \text{ belong to different class}$$

and

$$M(A, B) = 0 \text{ if } A, \ B \text{ belong to the same class}$$

In case of metric learning we are able to build such an operator with a deep neural network. It is a good choice to use a convolutional neural network for computer vision to build distance functions. The main challenge is to design a loss function. The Triplet loss [4] function gives us a good differentiable formulation of operator M.

It is worth mentioning that triplet loss provides us a way to train an embedding extraction model, not a distance operator itself. But such embeddings are presented by a vector that is separable by a distance used in a triplet loss function.

The whole pipeline of image classification is presented in Fig. 1.

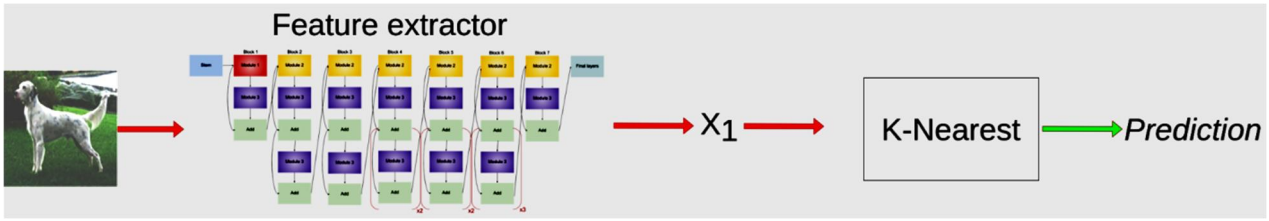

Fig. 1. Diagram of the model pipeline

Dataset

In this paper, we set up our experiments with an open-source FSS-1000 [5] dataset. This dataset consists of 1000 various classes and is specially designed for few-shot learning purposes. Presented classes in this dataset in unbiased which means that there are no bunches of similar classes (like more than 200 dog breeds in the ImageNet [6] dataset). The original dataset was collected and prepared for an instance segmentation task, but exactly one class is presented per image, so this dataset is suitable for the image classification task as well.

Fig. 2. FSS-1000 dataset samples

All images have exactly the same resolution – 224×224. Our exploration data analysis shows that a significant part of objects does not cover the whole image, so a lot of background data is presented which makes the classification task harder in terms of overfitting prevention.

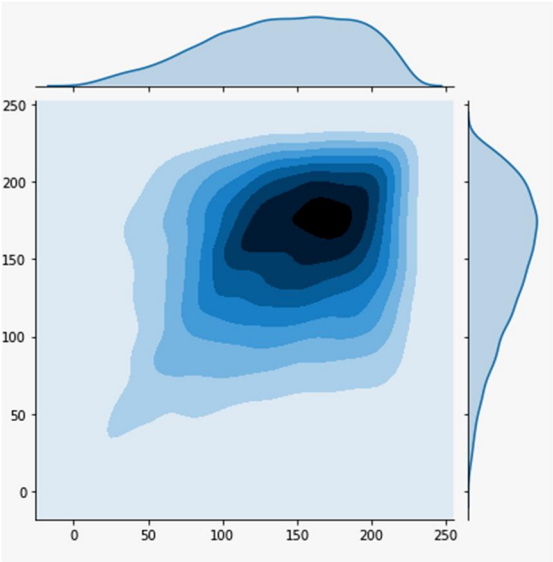

Fig. 3. Object sizes on images

In the FSS-1000 dataset each class are represented by exactly 10 images. It allows us to prepare perfectly class-balanced dataset splits.

Triplet loss-driven model

1. Model

We build a tiny deep learning model which consists of three blocks:

- Feature extractor (EfficientNet B2 feature vector, pretrained on ImageNet);
- Emdeding learner (fully-connected layer, 256 neurons);
- Scale normalization layer (L2 normalization).

The overall model structure is presented in Fig. 4.

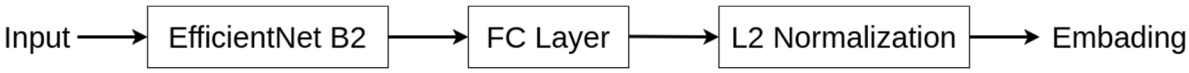

Fig. 4. Embedding model structure

Detailed model structure is shown in the table:

Table 1

Model's layers description

Layer	Input shape	Output shape	#Params
Feature extractor	[224, 224, 3]	[1408]	8769374
Dense	[1408]	[256]	360704
L2 Norm	[256]	[256]	0

Note, that the last layer contains no trainable parameters.

2. Training process

Since we use a pretrained feature extractor we freeze in order to train the embedding learner first. The training schedule is provided in the following table.

Table 2

Training stages description

Stage	Trainable layers	Epochs
#1	Dense	20
#2	The whole model	10

3. Experiments setup

Given an FSS-1000 dataset we prepare various training splits while keeping the testing split the same for all experiments. All available data was splitted onto three parts – training, validation, test.

We take 200 classes from the dataset as a test set. The rest data was used for training and validation purposes. The test part remains the same for all experiments. By doing this, we are able to compare the influence of the training dataset to the final performance of the model. Note, that classes that were incorporated for test purposes were never presented in a training/validation part. In other words, we set up a zero-shot experiment.

The rest 800 classes were distributed between training and validation sets as shown in the following table:

Table 3

Experiments details overview

#Experiment	#Training images	#Validation images
1	7	3
2	6	4
3	5	5
4	4	6
5	3	7

Training images were used in backpropagation steps during embedding model training. Validation samples were only used in validation loss calculation and never were utilized for backpropagation. Test images were not utilized before the final score evaluation.

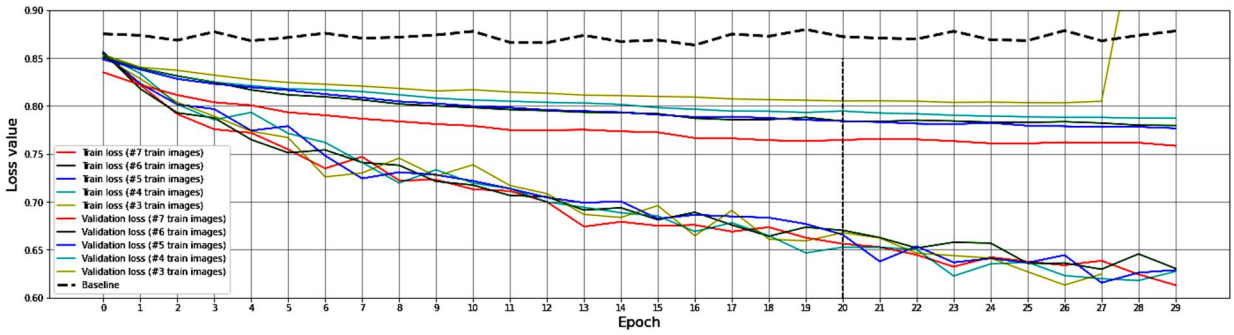

Fig. 5. Comparison of experiment losses

In Fig. 5 we show a training and validation loss per experiment. We may observe that there are almost no differences between training losses. Such noisy losses are caused by randomness in triplets

mining [7]. But for experiments where more training examples were utilized loss is much lower. It means that using more data makes our embedding model better in terms of embedding separability.

We also evaluate a baseline model (bare EfficentNet B2 feature vector). In order to compare the non-trained model with our experiments, we evaluate training and validation losses 30 times, so we can estimate a validation loss as well as training loss along with its deviation. Baseline losses are visualized in bold gray color in Fig. 5.

4. Estimate model accuracy

In order to be able to classify images we have to compare embeddings extracted from our model. In our work, we find KNN the best candidate to work with extracted embeddings.

For each experiment, we extract training, validation, and test set embeddings. Train embeddings were used to find KNN hyperparameters (numbers of neighbors, distance weights, and others) using grid search and cross-validation (5 folds were used). Once we find the best hyperparameters final model is built according to those parameters. Now, we can use our model to predict novel classes from the test set.

We fit KNN to the test embeddings and evaluate the final score using a stratified cross-validation strategy. The following metrics were calculated – top 1, top 3, top 5, and top 10 accuracies. We observe a logarithmic dependency between Top N accuracy and score. There is no reason to use more than Top 10 predictions since it brings almost no improvements. So, we don't take into account KPIs more than the top 10 accuracies.

We expose achieved scores in the following table 4.

Table 4

Accuracy scores per experiments

#Experiment	Top 1 accuracy	Top 3 accuracy	Top 5 accuracy	Top 10 accuracy
#1	0.932	0.980	0.983	0.983
#2	0.928	0.976	0.980	0.980
#3	0.928	0.980	0.984	0.985
#4	0.924	0.979	0.980	0.985
#5	**0.933**	**0.981**	**0.985**	**0.988**

The visualization of top-K scores is provided on Fig. 6.

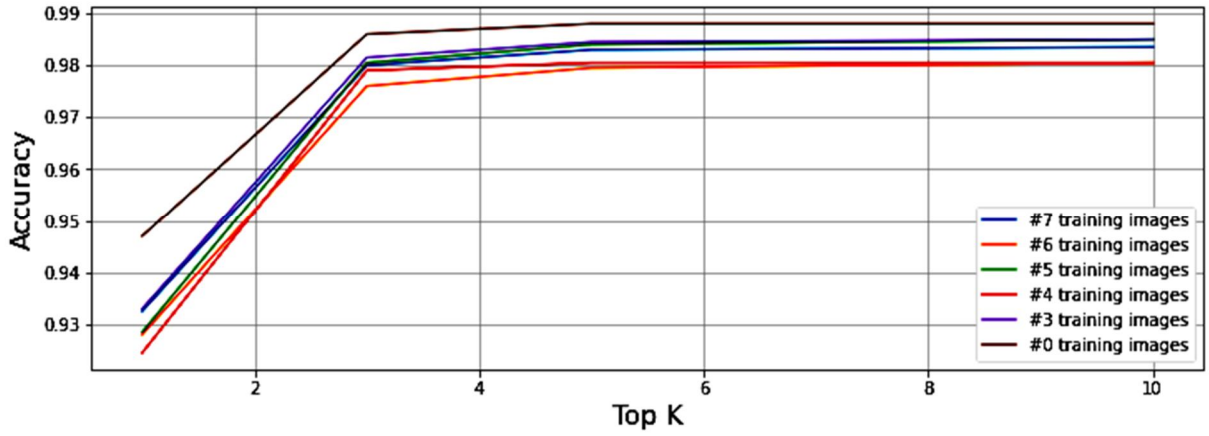

Fig. 6. Top K accuracies

Conclusions

In this paper we take a close look at the few-shot image classification problem using triplet loss. We utilize an FSS-1000 dataset for a classification purpose in order to investigate the influence of the amount of training data on the obtained model. In this work, we run a bunch of experiments with the same test data but different training data in order to be able to compare the results. Namely, we train the EfficientNet B2 based model using 7, 6, 5, 4, and 3 images per class (1 image was not possible as triplet loss requires two images of the same class; at the same time 2 images is very unliked to be sampled so often during training).

We found that lower images in the training outperform all other experiments. This result may be counterintuitive, but we found it consistent due to an unbalanced dataset (in terms of 1 class versus the rest). At the same time, we carefully examined validation losses obtained from the experiments. The most valuable result is that lower loss corresponds to a bigger training set. We found that gap between validation losses and scores is very interesting in set it up for further work and investigation.

References

1. Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851.

2. Said, K. A. M., Jambek, A. B., & Sulaiman, N. (2016). A study of image processing using morphological opening and closing processes. International Journal of Control Theory and Applications, 9(31), 15–21. https://doi.org/10.1109/ICED.2016.7804697.

3. Ye, H. J., Ming, L., Zhan, D. C., & Chao, W. L. (2021). Few-shot learning with a strong teacher. arXiv preprint arXiv:2107.00197. https://doi.org/10.1109/TPAMI.2022.3160362.

4. Hoffer, E., & Ailon, N. (2015, October). Deep metric learning using triplet network. In International workshop on similarity-based pattern recognition, 84–92. Springer, Cham. https://doi.org/10.1007/978-3-319-24261-3_7.

5. Li, X., Wei, T., Chen, Y. P., Tai, Y. W., & Tang, C. K. (2020). Fss-1000: A 1000-class dataset for few-shot segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2869–2878. https://doi.org/10.1109/CVPR42600.2020.00294.

6. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, 248–255. Ieee. https://doi.org/10.1109/CVPR.2009.5206848.

7. Xuan, H., Stylianou, A., & Pless, R. (2020). Improved embeddings with easy positive triplet mining. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2474–2482. https://doi.org/10.1109/WACV45572.2020.9093432.

КОМПЛЕКСНИЙ АНАЛІЗ ТЕХНІКИ НАВЧАННЯ НА МАЛОМУ НАБОРІ ДАНИХ ДЛЯ ЗАДАЧІ КЛАСИФІКАЦІЇ МЕТОДОМ ОПТИМІЗАЦІЇ ТРІЙОК

Микола Баранов[1], Юрій Щербина[2]

Львівський національний університет імені Івана Франка
[1] e-mail: mykola.baranov@lnu.edu.ua, ORCID: 0000-0003-1509-2924,
[2] e-mail: yuriy.shcherbyna@lnu.edu.ua, ORCID: 0000-0002-4942-2787

Задача класифікації зображень є дуже важливою сучасною проблемою в області комп'ютерного зору. Перші підходи до розв'язання цієї задачі полягали у використанні класичних алгоритмів. Незважаючи на певний прогрес, отриманий класичними підходами, більшість

складніших задач класифікації зображень залишались нерозв'язаними до початку використання алгоритмів машинного навчання. Перші спроби застосування машинного навчання до задачі розпізнавання зображень допомогли класифікувати набори ознак, які опрацювати прямими алгоритмами не вдавалось. Проте видобування множини ознак залишалося за прямими алгоритмами тривалий час. Нещодавний прогрес у сфері глибокого навчання відкрив можливість побудови систем автоматичного видобування множини ознак. Це зумовило значний прогрес у області комп'ютерного бачення і не тільки. Обробка великомасштабних наборів даних призвела до прориву у задачах розпізнавання зображень. Проте з'явилося нове обмеження– залежність від кількості наявних проанотованих даних. Методи глибинного навчання для задачі класифікації зображення зазвичай потребують великої кількості проанотованих зображень. І більше, сучасні моделі схильні до неочікуваної поведінки на наборах даних з іншого домена (нових класів у випадку розпізнавання зображень). Методи навчання на малому наборі даних дозволяють під час тренування глибоких нейронних мереж використовувати значно менше даних, зберігаючи таку саму точність розпізнавання. Незважаючи на це, залишається компроміс між кількістю наявних даних та точністю моделі. В цій роботі ми побудували сіамську нейронну мережу на основі функції втрат трійки і дослідили, як наявна кількість даних впливає на точність розпізнавання сіамської нейронної мережі. Ми порівняли моделі, отримані навчанням на основі метрик, та базову модель, натреновану на великомасштабних наборах даних.

Ключові слова: методи навчання на малому наборі даних; навчання на одному прикладі; навчання на основі метрик; комп'ютерний зір; класифікація зображень.