Image Recognition of Plants and Plant Diseases with Transfer Learning and Feature Compression

Marcin Zięba, Konrad Przewłoka, Michał Grela, Kamil Szkoła, and Marcin Kuta[0000−0002−5496−6287]

Institute of Computer Science,
AGH University of Krakow,
Al. Mickiewicza 30, 30-059 Krakow, Poland,
mkuta@agh.edu.pl

Abstract. This article introduces an easy to implement kick-starting method for transfer learning of image recognition models, meant specifically for training with limited computational resources. The method has two components: (1) Principal Component Analysis transformations of per-filter representations and (2) explicit storage of compressed features. Apart from these two operations, the latent representation of an image is priorly obtained by transforming it via initial layers of the base (donor) model. Taking these measures saves a lot of computations, hence meaningfully speeding up the development. During further work with models, one can directly use the heavily compressed features instead of the original images each time. Despite having a large portion of the donor model frozen, this method yields satisfactory results in terms of prediction accuracy. Such a procedure can be useful for speeding up the early development stages of new models or lowering the potential cost of deployment.

Keywords: transfer learning · feature extraction · feature compression · plant recognition · plant diseases recognition

1 Introduction

The article presents a method for transfer learning of image recognition models using additional dimensionality reduction of latent features as feature compression. This can be used to create mobile applications for automatic recognition of plants and their diseases based on photos. It required acquiring suitable training datasets and development of efficient and memory-frugal machine learning models.

The proposed approach helps to meet the crucial requirement that the deployed models are self-contained and, consequently, are not dependent on the access to the Internet. This requirement is backed by a reasonable assumption that people taking photos (e.g. farmers in the field, tourists in the mountains) may lack a stable Internet connection. Thus, the main focus was on efficiency of the proposed models. Due to limited resources and time constraints, the training process was also optimized.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_19

The visual recognition of plants poses some specific difficulties on its own like similarity of certain species, intra-species variability, different parts of a plant photographed, different growth stages at which the photo of plant is captured and non-obvious image cropping [8].

With transfer learning already being the most time-effective way to start working on computer vision tasks, the method described in this article considers optimizing this process even further, building upon two mechanisms:

- feature extraction using initial layers of a pre-trained model and Principal Component Analysis (PCA) transformation on top of that,
- explicit work on the latent representation of the data (calculating it once and storing in such a form).

Combining these two strategies heavily limits the amount of computation needed to train consecutive models using the compressed features directly as their input. Eventually, one can simply merge feature extractors with the trained model to form a single composite model (or a pipeline).

Six models were prepared, all trained on three datasets: GRASP-125, PlantVillage and PlantDoc. Four proposed models are variations of an approach using two-step feature extraction (exploiting our proposed approach to a full extent) and two models use only single-step feature extraction. We call these steps further (1) primary feature extraction (using initial layers of base model) and (2) secondary feature extraction (using PCA transformation applied in a specific way). Models with only the primary feature extraction achieve the TOP1 test-time accuracy (precise prediction of the true class) of prediction of 81.22% on the GRASP-125 dataset, 97.97% on PlantVillage, 56.24% on PlantDoc (SEResNet-based) and 85.04% on GRASP-125, 98.53% on PlantVillage, 57.55% on PlantDoc (MobileNetV2-based). Meanwhile, models following the approach with two-step extraction result in TOP1 accuracy of 79.58% on GRASP-125, 97.31% on PlantVillage and 51.88% on PlantDoc.

Despite such an aggressive feature extraction (from originally 150528–dimensional problem down to 7840 after primary feature extraction and eventually to 800 after the secondary one), models do not suffer from falling behind too much in terms of the prediction accuracy.

The PCA transformation enabled the creation of a solution that is only slightly worse in terms of performance, but noticeably easier to train compared to more traditional methods (i.e., SEResNet-based and MobileNet-based models). The computational cost of training was further reduced by utilizing the aforementioned latent representation of data. The main advantage of this approach is having a compressed representation of data precomputed ahead of the training itself. This may hurt eventual accuracy of prediction, but enables the use of smaller and simpler models on top of it.

It is important to emphasise that we apply PCA transformation to the data already processed with a deep neural network feature extractor, and not directly to the raw input data itself. Such an approach, to the best of our knowledge, has not been analysed yet. Despite the heavy limitations it may incur in a typical

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_19

model development (which might be a reason for being overlooked), this method may still be successfully utilized in a transfer learning manner.

2 Related Work

Modern systems for automatic plant species and plant diseases recognition are based on deep convolutional neural networks and transfer learning. Such networks achieve amazing accuracies on a wide range tasks, but their excessive need for memory and computational resources may be prohibitive from deployment on devices with limited resources. One of the main frontiers of advancement in image classification is a strive to minimize the number of parameters a model uses while retaining a reasonable accuracy. Examples of research regarding the high-efficiency image recognition are the mobile networks from MobileNets architectures family: MobileNet [3], MobileNetV2 [9] and MobileNetV3 [2]. The MobileNetV2 model [9] is used in our work as a base model for transfer learning. To reduce computations, MobileNets introduce depthwise separable convolutions, which consist from depthwise convolutional filters and pointwise convolutions.

Speedup in test-time prediction can be achieved by exploiting redundancy between different filters and feature channels. This was obtained with filter factorization, implemented with SVD transformation, and was applied to 15 layer CNN [1]. A related approach was presented in [5]. Our work is related to [1], [5] through the usage of PCA.

3 Datasets

The experiments were conducted on three datasets: the GRASP-125 dataset [6], PlantDoc [10] and PlantVillage [7]. The GRASP-125 and PlantDoc datasets were already split into training and testing sets. To further validate the performance of our models, we created additional validation datasets for these sets. Conversely, the PlantVillage dataset was not originally divided and thus we split it into training, validation, and testing sets for the purpose of evaluating our models.

The collected images were subsequently augmented for the training set to gain a more stable and comprehensive measure of models' predictive power. Image augmentation involved typical operations like random rotation, random cropping, horizontal flip, and random adjustment of brightness, saturation, hue and contrast. Finally, each image was scaled to 224×224 pixels, the format accepted by the utilized models.

GRASP-125 contains 16 327 images of vascular plants belonging to 125 classes (plant species). The test set contains 1704 images. The validation set contains 12 randomly chosen images per plant from the initial training set, in total 1500 images. The training set contains the remaining 13 119 images.

As a result of the augmentation process, each plant species was represented by 400 training examples and 12 validation examples. The test examples were not changed.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_19

The PlantVillage dataset [7] is dedicated to plant disease detection based on plant leaves. It contains 55 448 images of healthy and unhealthy leaf images, divided into 39 classes by species and disease. As no predefined split was provided, we divided the dataset into test, validation and training sets. The division was stratified, so the original proportions of classes were maintained.

The test set comprised 11 090 samples, constituting 20% of the dataset. The validation set was the same size as the test set (11 090 samples, representing 20% of the dataset). Prior to the data augmentation process, the training set contained 33 268 images, comprising 60% of the original dataset. Following the augmentation process, the number of images in the training set increased to 46 541, from 1000 to 3304 images per class, depending on the class.

PlantDoc [10] is a newer dataset containing 2576 images of healthy or diseased plant leaves belonging to 27 classes (17 diseased, 10 healthy) from 13 plant species. It comes with a predefined train-test split into 2340 images in the training set and 236 images in the test set. From 15% of the original training set, we created the validation set, containing 351 samples. The class distribution of the original dataset was maintained in the validation set. The training set was aggressively augmented to contain between 1037 and 1152 images per class, depending on the class, which gives a total of 28 983 images. This is in contrast to the original dataset, which contained between 44 and 179 images per class before the augmentation process.

4 Proposed Approach

All the further experiments were conducted using the MobileNetV2 model as the base model [9]. This architecture is known to be a good off-the-shelf image classifier, achieving both satisfactory prediction accuracy and computational efficiency. MobileNetV2 has been chosen out of the MobileNet architectures due to the availability of the pre-trained model with weights on a task similar to plant species or plant diseases recognition. The MobileNetV2 model was used mostly as the primary feature extractor – specifically, initial layers making up for 1 364 864 out of all 3 538 984 parameters of the model were utilized.

For the purpose of transfer learning, a model pre-trained on the ImageNet dataset was chosen. This base model most likely did not see an overwhelming majority (if any) of the images from the three datasets used here during its training.

The overall transfer learning procedure for preparing the compressed representation of the dataset, as well as feature extractors (both primary and secondary), consists of the following steps:

1. Specifying of the primary feature extractor (initial layers of some pre-trained model).
2. Encoding the whole dataset using the (primary) feature extractor from the previous step.
3. Saving the compressed examples to a persistent storage.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_19

4. Initialization of separate PCA units, one per each channel (filter output), with an arbitrary number of principal components specified.
5. Fitting the PCA units, passing to each of them only the corresponding channel contents, using only the training dataset (data per channel may require its flattening first).
6. Encoding the whole compressed dataset (training, validation, test sets) from step 3 by applying the learnt PCA transformations (secondary feature extractor).
7. Saving the compressed examples to a persistent storage.

The final representation is a highly compressed version of the original features. Due to the fact that now we have all three datasets in such a form, we can continue working at this abstraction layer as if these were the actual input data. Once some model is finally obtained, it is sufficient to attach it on top of the feature extractors, thus producing a full model capable of working on raw (non-compressed) data. To achieve this, it might be necessary to define a custom model layer for carrying the PCA dimensionality reduction, given the previously learnt parameters of this transformation.

Let n be batch size, c – number of channels, s – size of a channel, and p – number of principal components to be extracted. In a batch, multichannel and multi-PCA-unit case let us assume further the following notation:
$X \in \mathbb{R}^{n \times s \times c}$ – data tensor, $M \in \mathbb{R}^{1 \times s \times c}$ – all empirical means tensor, $W \in \mathbb{R}^{s \times p \times c}$ – all principal components tensor, $Y \in \mathbb{R}^{n \times p \times c}$ – data tensor of reduced dimensionality. Then, for each channel i we have $Y_{:,:,i} = \big(X_{:,:,i} - M_{0,:,:}\big)W_{:,:,i}$.

For practical use, a custom layer or component should be implemented, so that it is initialized with empirical means and basis vectors of principal components (obtained from PCA units used during training), and during runtime it applies above transformation to its inputs.

Initial layers of convolutional deep learning models for computer vision can be effectively pictured as the extractors of some abstract features. They are successful at learning edges, color gradients and simple shapes, and then using them for further, more high-level, reasoning. Representation of latent features at a specific level of the network gives at our disposal a dozen of abstract feature maps, one per channel. In particular, each channel here is an output of a corresponding filter from the previous layer. Each of these filters is specialized in identifying some specific phenomena. Thus, it may be presumed that intra-channel features are likely to be noticeably correlated with each other. This can result in the suboptimality of the representations. To exploit this fact, a PCA dimensionality reduction can be applied – importantly, a distinct one per each channel. As a result, we obtain a heavily compressed representation of the original input.

Applying the PCA transformation channel-wise is a computationally preferred option compared to the straight-forward use of a single PCA unit. Most notable distinction is the size of the principal axes matrix. Following the earlier notation, we can describe the number of parameters as:

– $c * s * p$ for calculating transformations for each channel separately (in our case $160 * 49 * 5 = 39\,200$),

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_19

– $(c * s) * (c * p)$ for calculating transformations for whole input at once (in our case $160 * 49 * 160 * 5 = 6\,272\,000$).

Both approaches reduce the dimensionality to the same extent. The observed discrepancy in the number of parameters originates from disregarding the inter-channel interactions.

It is viable to use Linear Discriminant Analysis (LDA) instead of PCA for dimensionality reduction. The transformation to be executed is analogous to the one depicted for PCA – using empirical means and linear projection matrix in a similar way. The prediction accuracies for both approaches are comparable.

5 Experiments and Results

Three distinctively different model architectures have been eventually determined. One of them is fully built upon the approach presented in Sect. 4 (steps 1–7 of the above procedure – both feature extractors were used) and is presented in four variants. The other two models apply only the primary feature extractor (steps 1–3), providing a good point of reference.

It is essential to keep in mind that the original dimensionality of the problem is 150 528 as the input images are of size 224×224 pixels (3 channels). Primary feature extraction reduces it to 7840 (7×7×160) and the secondary one further decreases the dimensionality to 800 (5×160) due to PCA transformation from 49 features per channel down to 5 principal components.

The implemented architectures are:

– PCA+Dense – The architecture embraces both feature extraction stages applying the full procedure introduced in Sect. 4 – as such, this model is our main object of interest here. It operates on the input dimensionality of size 800. Its input is flattened and fed into a multi-layer perceptron.
– PCA+SepConv – The architecture is structurally similar to PCA+Dense, except for not flattening the extracted features immediately, instead using separable convolutions first.
– LDA+Dense – LDA versions of PCA+Dense architecture.
– LDA+SepConv – LDA versions of PCA+SepConv architecture.
– SEResNet – The architecture is designed as a single block of the Squeeze & Excitation Residual Network (SEResNet) architecture [4]. The choice of the SEResNet was rather arbitrary; nonetheless, it offers a decent predictive power while remaining relatively simple.
– MNv2 – The architecture, consisting of the leftover layers of the original MobileNetV2 [9], is adapted for distinguishing the relevant number of classes (125 for GRASP-125, 39 for PlantVillage, 27 for PlantDoc) instead of 1000 by reducing the number of units at the output layer and adding regularization to mitigate the risk of overfitting.

SEResNet and MNv2 rely only on the primary feature extractor and are kept here for reference. In particular, MNv2 is initialized (where it is possible)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_19

with the weights from the original model per-trained on ImageNet – it is likely that this model approximately determines the upper bound of the achievable accuracy in this setting.

The models were trained utilizing a single GPU instance. Each model was trained for 10 epochs. The optimizer used in each case was Nadam (with default Keras parameters: learning_rate=0.001, beta_1=0.9, beta_2=0.999) and the loss function was categorical cross entropy. Primary feature extractor consisted of initial layers of the base model with 1 364 864 parameters in total. Moreover, a learning rate scheduler halving the learning rate when arriving at plateau (with patience of 2 epochs) was used.

The prepared models achieved the accuracies shown in Table 1. Discrepancies between validation set and test set accuracy are due to the reasons mentioned in Sect. 3. Despite utilizing a significant number of data augmentation techniques, the models were unable to fully generalize their classification of images from the PlantDoc dataset due to its diversity.

Table 1: Accuracy of TOP1 and TOP5 prediction for each tested model. TOP1 accuracy means exact prediction of the true class. For TOP5 accuracy, the real class should be among the five most probable outcomes.

	Validation set		Test set	
	TOP1 [%]	TOP5 [%]	TOP1 [%]	TOP5 [%]
		GRASP-125		
PCA+Dense	78.45	93.27	78.64	93.34
PCA+SepConv	80.33	92.60	79.99	94.54
LDA+Dense	77.53	92.07	78.64	92.96
LDA+SepConv	78.80	92.47	81.04	93.84
SEResNet	79.33	93.80	81.22	94.84
MNv2	84.27	94.33	85.04	94.89
		PlantVillage		
PCA+Dense	97.67	99.92	97.41	99.94
PCA+SepConv	97.65	99.95	97.57	99.94
LDA+Dense	97.24	99.97	97.10	99.90
LDA+SepConv	96.82	99.95	97.13	99.90
SEResNet	98.00	99.96	97.97	99.94
MNv2	98.46	99.95	98.53	99.95
		PlantDoc		
PCA+Dense	56.67	88.02	52.43	87.17
PCA+SepConv	52.69	88.31	53.67	88.44
LDA+Dense	52.40	87.44	51.16	85.83
LDA+SepConv	52.97	88.58	50.27	85.00
SEResNet	61.52	90.30	56.24	87.65
MNv2	61.52	88.87	57.55	88.01

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_19

6 Conclusions

In the setup of the conducted experiments the dimensionality of the problem was reduced from 150 528 (original photo scaled to 224×224 pixels, 3 channels), to 7840 (7×7×160) after primary feature extraction, and finally to 800 (5×160) after secondary feature extraction (PCA transformation). During these experiments the deterioration of the PCA-reliant model due to extensive feature compression was not as severe as it could be anticipated and satisfactory results could still be provided. Thus, PCA can be successfully used for the extraction of heavily compressed features from the latent representation of the data already processed by a convolutional feature extractor. The proposed procedure is useful for speeding up the early development stages of new models or lowering the potential cost of their deployment.

Acknowledgements. The research presented in this paper was supported by the funds assigned to AGH University of Krakow by the Polish Ministry of Education and Science.

References

1. Denton, E.L., Zaremba, W., Bruna, J., LeCun, Y., Fergus, R.: Exploiting linear structure within convolutional networks for efficient evaluation. In: Annual Conference on Neural Information Processing Systems, NIPS 2014. pp. 1269–1277 (2014)
2. Howard, A., Pang, R., Adam, H., Le, Q.V., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., Chu, G., Vasudevan, V., Zhu, Y.: Searching for mobilenetv3. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019. pp. 1314–1324 (2019)
3. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017)
4. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018. pp. 7132–7141 (2018)
5. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions. In: British Machine Vision Conference, BMVC 2014 (2014)
6. Kritsis, K., Kiourt, C., Stamouli, S., Sevetlidis, V., Solomou, A., Karetsos, G., Katsouros, V., Pavlidis, G.: Grasp-125: A dataset for greek vascular plant recognition in natural environment. Sustainability **13**(21) (2021)
7. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Frontiers in Plant Science **7** (2016)
8. Nilsback, M., Zisserman, A.: Automated flower classification over a large number of classes. In: Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008. pp. 722–729. IEEE Computer Society (2008)
9. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018. pp. 4510–4520 (2018)
10. Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., Batra, N.: Plantdoc: A dataset for visual plant disease detection. In: CoDS-COMAD 2020: 7th ACM IKDD CoDS and 25th COMAD. pp. 249–253. ACM (2020)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_19