https://doi.org/10.2352/ISSN.2470-1173.2018.2.VIPC-175

Transfer Learning for Data Triage Applications

Felix Mayer, Marcel Schäfer, Martin Steinebach

Abstract

Convolutional neural networks (CNNs) have improved the field of computer vision in the past years and allow groundbreaking new and fast automatic results in various scenarios. However, the training effect of CNNs when only scarce data are available is not yet examined in detail. Transfer learning is a technique that helps overcoming training data shortage by adapting trained models to a different but related target task. We investigate the transfer learning performance of pre-trained CNN models on variably sized training datasets for binary classification problems, which resemble the discrimination between relevant and irrelevant content within a restricted context. This often plays a role in data triage applications such as screening seized storage devices for means of evidence. The evaluation of our work shows that even with a small number of training examples, the models can achieve promising performances of up to 96% accuracy. We apply those transferred models to data triage by using the softmax outputs of the models to rank unseen images according to their assigned probability of relevance. This provides a tremendous advantage in many application scenarios where large unordered datasets have to be screened for certain content.

Introduction

With the increasing amount of produced data, the search for relevant information amongst vast and unordered datasets becomes more important than ever before. In the context of digital images this means that, depending on the application scenario, images with a certain content have to be extracted from large collections of potentially irrelevant images. For example, finding supernovae within images of distant galaxies [1] as well as medical [2], [3] and military applications [4]. In such scenarios the datasets are commonly restricted to a certain context (e.g. images of star clusters, x-ray images of human body parts or satellite images), and exhibit a large amount of irrelevant content. Thus, when searching for content of interest, the problem of finding the needle in a haystack occurs.

Two possible retrieval scenarios can be derived from that situation: data triage and content-based image retrieval (CBIR). In the triage scenario, a binary classifier which discriminates between relevant and irrelevant content can be used to sort the images according to relevance. Triage is helpful whenever large datasets have to be screened for certain content, e.g. when seized storage devices have to be screened for child pornographic imagery [5] or violent content [6] has to be removed from social media profiles. As false positive or false negative decisions can have fatal consequences in some scenarios, the final decision about the relevance of an image is often left to human inspectors. However, an algorithm which automatically ranks the the images in a dataset according to some probability of relevance, can support the screening process by allowing a faster detection of the relevant images.

In the CBIR scenario, on the other hand, the relevant images in the dataset might be similar to few images that are already known. Using those known images as queries would allow an automatic ranking of the images in the dataset based on similarity to the query. One possible application scenario is the detection of contrabandism at border control. Once a suspicious good of a certain kind (e.g. antiques) is discovered among a traveller's luggage, the customs official in duty could query a snapshot of the unkown object against an image collection of similar objects which are tagged with their import restriction status. In both scenarios, ranking the images according to their relevance reduces the cost for retrieving the content of interest from the dataset. In the following we focus on the triage application scenario.

The automatic detection of image content is realized with computer vision algorithms. Since 2012, when AlexNet [7] won the ILSVRC challenge, convolutional neural networks (CNNs) have become very popular in the field of computer vision. Today, CNNs represent the state of the art in most computer vision tasks such as image classification and object recognition.

Training a generalizable CNN model, usually requires large and diverse datasets. In some cases, however, for example where expert knowledge is required to tag the images, the collection of such large training (and evaluation) datasets is expensive and often not practicable. Scenarios like counterterrorism, medical emergencies or police investigative work are often time-critical which makes a profound data collection and tagging impossible. In such cases, transfer learning (TL), i.e. adapting a pre-trained model to a small target dataset, can help overcoming the problem of scarce training data and allow building classification models within a short time. To achieve a model adaption by TL, a certain fraction of the hidden layers of the CNN is frozen, which means that their weights do not receive any further updates during TL. Only the layers behind those frozen layers are updated in order to adapt the model to the new dataset. As Girshick et al. [8] pointed out, TL helps increasing the classification accuracy for tasks in which training data is scarce. Furthermore, TL is helpful when training has to be executed quickly, since the training time increases linearly with the amount of training examples.

The idea of our approach is to adapt pre-trained CNN models to binary classification tasks via TL, which resemble the separation of relevant and irrelevant content in a dataset restricted to a certain context, such as satellite or x-ray images. Once the networks have learned to discriminate between (ir)relevant content within that context, their softmax outputs can be used to rank the images in the dataset according to probability of relevance. Since the result of a learning process depends on the dataset as well as on the the chosen model, we want to figure out differences and commonalities among different CNN architectures and datasets that allow some insights about the training effect of data scarcity. Furthermore, we investigate the dependance of the performance of a triage application on the training dataset size. To this end, we

evaluate our approach on a varying number of training examples.

The remainder of this paper is structured as follows: In the *Related Work* section we provide an overview of related work; section *Datasets and Models* briefly introduces the datasets and CNN models that we used in our approach. Section *Transfer Learning* describes the transfer learning (TL) experiments conducted to obtain information about suitable parameter settings when training data is scarce. The triage approach and its evaluation is described in section *Triage Experiments*, which is followed by the *Conclusion* section.

Related Work

When it comes to the choice of the dataset on which to pre-train a model prior to TL, the well-known ImageNet[1] dataset with 1.2 million images distributed over one thousand object classes is commonly used in TL. As Huh et al. [9] pointed out, thus far, no other approach could outperform pre-training on ImageNet for learning general-purpose deep features. The authors state that the dataset size of ImageNet itself might not be as important as is often assumed. But even with a reduced set of 500 images per class, pre-training on ImageNet did not lead to much worse TL performance. The authors also found out that reducing the number of classes could increase TL performance, at least for some target datasets. Additionally, they showed that learning fine-grained features during pre-training is not necessary for a good TL performance. However, as only a single CNN architecture (AlexNet [7]) was examined, it is not clear, whether their observations are in fact architecture-dependent or not.

Surprisingly, Agrawal et al. showed that regularization mechanisms such as early stopping, which aim to prevent a model from overfitting, are counterproductive for TL, when applied during the pre-training phase of a transfer learning task [10]. Simply increasing the number of training examples of the pre-training dataset does not lead to better TL performance, as highlighted by Joulin et al. [11], where the YFCC100m [12] dataset of 100 million Flickr images was used for pre-training. However, the TL accuracy did not improve substantially in comparison to pre-training on ImageNet.

A study about the optimization of TL parameters was conducted by Azizpour et al. in [13]. However, in contrast to our work, they focused on different parameters. They used the ImageNet dataset to pre-train the AlexNet and OverFeat [14] CNN architectures. For TL different datasets were used, which were divided into five distinct target recognition tasks (ordered by distance to the source task, i.e. ImageNet classification): image classification, attribute detection, fine-grained recognition, compositional semantic recognition and instance retrieval. Instead of reusing the complete CNN architecture, Azizpour et al. extracted features from intermediate layers and performed the TL step by using a linear Support Vector Machine (SVM) for classification tasks, and the Euclidean distance function for retrieval tasks, on those features. By making the distinction between the five different recognition tasks, they observed that the evaluated TL parameters depend on the distance between source and target task. By varying the number of parameters per layer Azizpour et al. constructed architectures of different complexity. Comparing TL performance on those different architectures revealed that for target tasks similar to the source task, complex architectures with hundreds of millions of parameters achieve the highest TL performance. On the contrary, for more distant target tasks (e.g. instance retrieval) medium-sized architectures (about 60 million parameters) achieved the best results. Increasing the network depth did not lead to any substanial performance decrease. Additionally, the higher the distance between target task and source task was, the more effective were the lower layers as feature sources for TL [13]. However, Azizpour et al. state that a good trade-off for any target task is using the features of the first fully-connected layer. They also state that in order to preserve the features learned during pre-training, the learning rate in the TL step should be lower than in the pre-training step.

Yosinsky et al. discovered in [15] that TL performance decreases when the pre-training and target datasets are dissimilar. They also mentioned that the layer index where to split the pre-trained model into frozen and non-frozen layers should be chosen carefully, since the transferability of the model can decrease when that split is set between co-adapted layers.

Regarding the mini-batch size, which is a frequently discussed hyperparameter in deep learning, Keskar et al. [16] recommend small mini-batches. Although large mini-batches require less training time due to a better parallelizability, they also yield a worse generalization (generalization gap) and tend to overfitting.

Datasets and Models

We used three distinct image datasets of different domains to evaluate TL and triage. The datasets comprised different numbers of images belonging to two classes, respectively. As mentioned in the introduction, this work is focused on triage applications with restricted context. The datasets used are described in the following:

- "Knives Images Database"[2], which consists of images taken at various indoor scenes including or excluding knives. We refer to this dataset by ***knives***.
- "Ships in Satellite Imagery"[3], provided by Kaggle, which consists of satellite images of waters surfaces including or excluding ships. We refer to this dataset by ***ships***.
- A self-compiled balanced dataset consisting of two thousand selfies and similar images showing adult people from Reddit.com including or excluding raw nudity, which we collected for a previous paper [5]. Due to the nature of the content of this dataset we did not relase it. We refer to this dataset by ***nsfw***.

All images in the respective datasets either contained an object of interest (e.g. a knife, ship or nsfw content) or not. Thus, we considered the class containing the objects of interest as positive (*P*) and the other class as negative (*N*). For *knives* all images containing a knife are labeled as *P*, for *ships* the images containing a ship are labeled as *P*, and for *nsfw* the images containing raw nudity are labeled as *P*.

All images of the three datasets were rescaled to a fixed size 200 by 200 pixels in order to provide constant input sizes to the evaluated networks. As we conducted two successive experiments with the datasets, TL and retrieval, we first split the datasets into

[1] http://www.image-net.org

[2] `http://sit4.me/knivesdatabase`

[3] `http://sit4.me/kaggleships`

Table 1. *Dataset splits for TL and data triage. P denotes the positive class, whereas N denotes the negative class for each split.*

Dataset name	Original		Transfer Learning				Triage	
			Train		Validation			
	P	N	P	N	P	N	P	N
knives	3,559	9,340	200	200	200	200	3,159	8,940
ships	700	2,100	200	200	200	200	300	1,700
nsfw	1,000	1,000	200	200	200	200	600	600

disjunct subsets for the two tasks. The subset for retrieval did not have to be split into train and test sets, as this part does not require any further training of the models. Table 1 provides a complete overview of the datset split for each task. The subsets for TL were further divided into balanced subsets for training and validation, which are listed in Table 2.

For the evaluation of our approach we compared three distinct popular CNN architectures: InceptionV3 [17], ResNet50 [18] and VGG16 [19]. Regarding the number of weight parameters, VGG16 is the most complex architecture (138 million parameters), followed by ResNet50 (25 million parameters) and InceptionV3 (23 million parameters).

Transfer Learning

Like performed in other works before [9], [20], we decided to use ImageNet for pre-training the three CNN models for our TL experiments, as ImageNet was shown to be a good pre-training dataset for many TL application scenarios [9] and yields diverse classification models due to its large number of distinct classes. With our TL experiments we aimed for answering the following questions:

- **How many training examples are necessary in order to achieve a satisfying classification performance?** Usually, deep CNN architectures require a large amount of training data, as they incorporate vast amounts of parameters to tune. Nevertheless, in some scenarios, when the generation of enough training data is too expensive or simply impossible, it would be beneficial, knowing some minimally necessary amount of training data in order to achieve a reliable prediction model.
- **Is any of the pre-trained CNN models more suitable for adaption to a binary classification problem with scarce training data than the others?** Due to their different architectures and complexity, it is interesting to investigate, whether there is one model that outperforms the others on all datasets or whether different datasets are best handled by distinct models.

As often mentioned in the literature, TL performs best when applying smaller learning rates than those used for pre-training [14]. However, as we discovered in preliminary experiments, when the training data is scarce, it is beneficial to start with a high learning rate which gradually decays to lower values. The effect is a broader exploration of the loss function in the beginning, which might reduce the risk of the learning process getting stuck in nearby local minima. However, decaying is necessary in order to prevent the model from bouncing back and forth on the loss function. So, we set the initial learning rate to 0.1 and applied a linear decay to 0.001, using SGD optimization which is a commonly used optimization technique in deep learning [16]. As we trained on datasets with different sizes, we chose a relative mini-batch size of 10% of the training data. This way, we ensure the same amount of iterations for a certain number of epochs for all dataset sizes.

In order to investigate the performance on different training dataset sizes, we performed TL for a duration of 150 epochs on increasing balanced subsets of the training data. Table 2 shows the numbers of examples per class that we used in our training subsets. A balanced set of 400 samples was held out for validation. Figure 1 shows the validation accuracy plots per epoch for the datasets *knives*, *ships* and *nsfw* and each of the CNN architectures InceptionV3, ResNet50 and VGG16, repsectively. Additionally, the final validation accuracy values are given in Table 3.

Table 2. *Different training data subsets for TL. The first row shows the subset index, the second row contains the corresponding number of training examples per class in the subset.*

Subset:	1	2	3	4	5	6	7	8
ex./class:	25	50	75	100	125	150	175	200

Table 3. *Validation accuracy on training subsets for InceptionV3 (I), ResNet50 (R) and VGG16 (V). CNN architectures are indicated by the first letter of their name (I, R, V) as subscript to the corresponding dataset. The first row indicates the number of examples per class present in the respective training subset.*

	25	50	75	100	125	150	175	200
$knives_I$	.67	.50	.77	.70	.75	.81	.81	.79
$knives_R$	.82	.84	.84	.86	.89	.91	.91	.90
$knives_V$	.50	.50	.50	.87	.88	.89	.90	.89
$ship_I$	.50	.80	.50	.88	.87	.50	.87	.87
$ship_R$	.81	.85	.87	.87	.86	.87	.88	.90
$ship_V$	.50	.50	.50	.95	.94	.50	.96	.96
$nsfw_I$	.50	.74	.66	.73	.50	.75	.75	.77
$nsfw_R$	.81	.82	.81	.83	.82	.83	.84	.87
$nsfw_V$	.50	.82	.79	.50	.81	.50	.81	.82

From the graphs in Figure 1 we can infer the following conclusions and assumptions: While InceptionV3 and VGG16 learn faster than ResNet50, ResNet50 and VGG16 achieve higher accuracy than InceptionV3 on all datasets. Furthermore, for Incep-

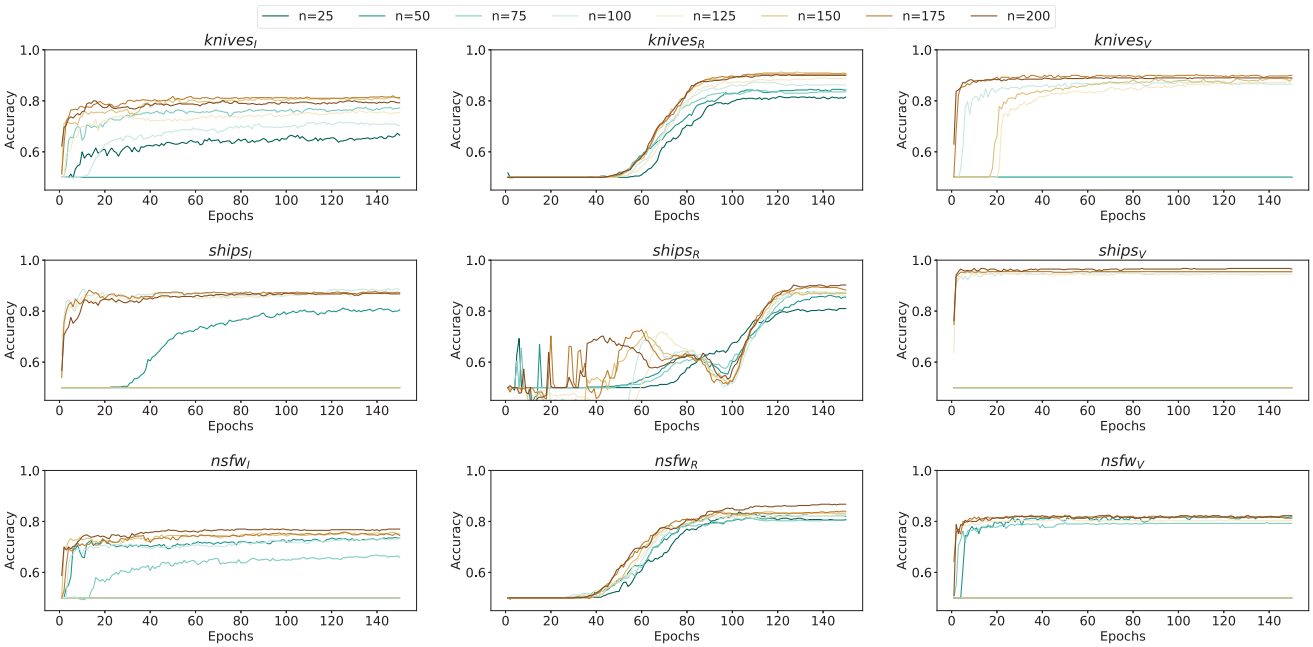

Figure 1. *Validation accuracies of TL with varying training set size. Each row shows the results for* knives, ships *and* nsfw, *respectively. Each column shows the results for InceptionV3 (I), ResNet50 (R) and VGG16 (V), respectively. Each subplot is additionally entitled with the corresponding dataset name and a subscript indicator of the corresponding CNN architecture.*

tionV3 dataset size seems to affect accuracy much more than for ResNet50 and VGG16. VGG16 seems to combine fast learning of InceptionV3 and dataset size independece of ResNet50, which we assume to result from its high complexity in terms of parameters.

Regarding the different datasets, the following conclusions can be drawn: On the *ships* dataset the achieved accuracy is generally slightly higher than on the *knives* and *nsfw* datasets. Satellite images of waters surfaces do not contain a lot of background noise, which makes the detection of ships (that usually have a characteristic shape, seen from above) relatively easy.

On the *knives* dataset, the achieved accuracy is generally slightly higher than on *nsfw* dataset. This can be explained by the fact that ImageNet already contains a class *'letter opener, paper knife, paperknife'*, i.e. features describing the concept of a knife are already known to the pre-trained models. On the contrary, *nsfw* content is not present in ImageNet. However, the *nsfw* dataset requires less training data for a model to achieve high performance. The reason might be that in the *nsfw* dataset the relevant features (naked human bodies) cover a much larger part of the individual images, than an average knife in the *knives* dataset or most ships in the *ships* dataset, where the part of the images covered by background is much larger, in general.

Another phenomenon that arose is that for some subsets the models were not able to adapt to the target data. This is reflected by 0.50 accuracy values throughout the TL process. This behavior might be explainable if only the smallest subsets would be affected. But, as in our case, the larger datasets always comprised the smaller ones, the reason for performance drops for increasing the dataset size is harder to explain. One assumption is, that due to the relatively small number of training examples in general, the addition of new examples can deteriorate the model when a certain amount of those examples contains poor features (features that are also present in many examples of the opposite class and are thus not characteristic for the target class). This way, more data can misguide the model in wrong directions.

Triage Experiments

In order to use the models which were trained in the previous section, we used the output values of their softmax layers. The softmax output values reflect probabilities that the model assigns to an image for the presence of the respective classes. All softmax output values add up to one.

For each dataset we considered the positive class P as the one of interest, so that we only extracted the softmax value of class P for each image. According to those values, which were assigned to every image in the triage datasets, we created a ranked list, sorted by that value in descending order. As highly imbalanced datasets in favor of the negative class N are much more challenging in triage scenarios, we limited the percentage of positive examples in the respective triage datasets to 5%. We measured the triage performance by precision-recall (PR) curves and the corresponding mean average precision (MAP), as provided in Figure 2.

Table 4. *Mean Average Precision (MAP) of the models trained with 200 examples per class in the TL section on each training dataset, applied to the corresponding triage datasets.*

Dataset	InceptionV3	ResNet50	VGG16
$knives_{tr}$	.46	.63	.59
$ships_{tr}$	.28	.19	.81
$nsfw_{tr}$	.24	.59	.57

The observations made for TL, are well reflected in the eval-

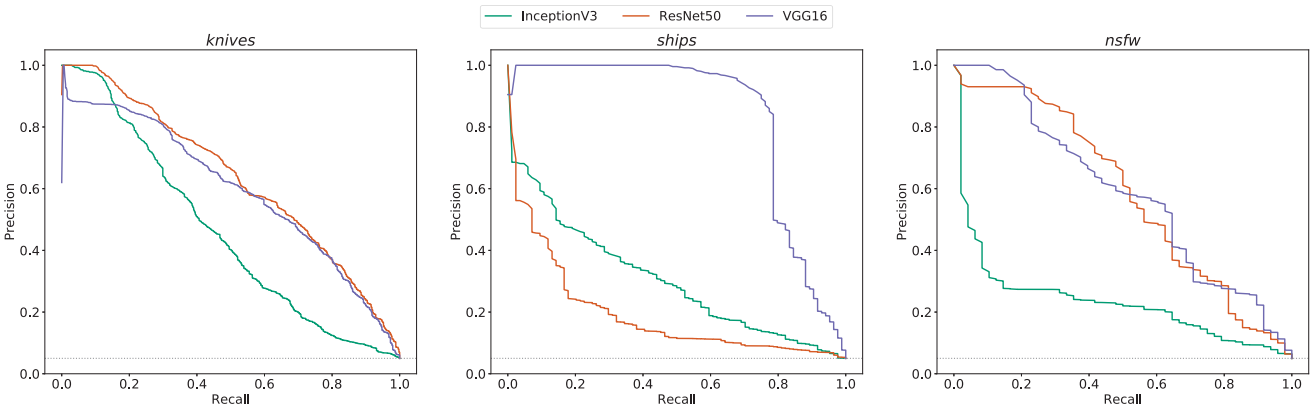

Figure 2. *Precision-recall (PR) curves of InceptionV3 (I), ResNet50 (R) and VGG16 (V). The additional grayish dotted line represents the triage performance on a randomly sorted dataset, when no ranking is applied prior to the data screening.*

uation of the triage task. The higher TL performance achieved on the *ships* dataset leads to better PR curves than for the other two datasets. Also, the fact that with the *ships* dataset, VGG16 performed much better than the other two models in the TL task, can be immediately seen in the PR curves. Likewise, the observation that for the two other datasets, *knives* and *nsfw*, ResNet50 and VGG16 achieved similar performance, reflects the TL results. The affirmation of the TL results on the held out triage datasets can be seen as a further reliability test for the learned models. Nevertheless, an evaluation on completely different datasets with the same context would allow much more profound assertions about the generalizability of the models.

Considering practical application scenarios as mentioned in the introduction of this paper (e.g. the screening of large datasets for means of evidence), the PR curves from Figure 2 and their corresponding MAP values given in Table 4 indicate the efficiency of our approach. On average, a random ordering (regarding image content) of the dataset can be assumed. With a percentage 5% of positive examples, this only allows a constant precision of 0.05 at every stage in the triage process, which requires the investigator to screen a very large amount of data if a high recall is desired. The ranked list according to our approach, on the other hand, achieves both, high precision and high recall at much earlier stages.

Conclusion

In this work we examined the performance of transfer learning (TL) with scarce training data and its application to triage problems. Each of the three evaluated CNN models was able to adapt to the new target tasks. Due to the nature of the restricted context of the binary classification datasets, InceptionV3 and VGG16 adapt very quickly to the new task after just few epochs, while the accuracy of ResNet50 starts increasing later (after epoch 50) and usually slower. Once, the accuracy curve has saturated, ResNet50 and VGG16 exhibit less noise in validation accuracy than InceptionV3, which might be an indicator that their learned models are more reliable. The evaluation of the triage task reflected the observations which were made in the TL section of this work. This is a reliability indicator for the learned models.

In future work, a cross-dataset evaluation of the models achieved via TL would allow a better examination of their generalizability. To investigate whether the small numbers of training examples observed in this work, which yield high performance, are generally sufficient when adapting complex CNN models to binary classification tasks, the conducted experiments should be repeated on more datasets in the future. One method to further increase the performance of the TL models might also be data augmentation.

Acknowledgments

This paper was funded by the German Federal Ministry of Education and Research (BMBF) as part of the research project ILLICID (funding ID: 13N13647). Additionally, we would like to thank Oren Halvani, Lukas Graner, Maximilian Li and Christoph Schüßler who supported us in various aspects of this paper.

References

[1] Cecilia Aragon and David Aragon. A Fast Contour Descriptor Algorithm for Supernova Image Classification. In *Proceedings of SPIE*, volume 6496, 2007.

[2] Yaniv Bar, Idit Diamant, Lior Wolf, Sivan Lieberman, Eli Konen, and Hayit Greenspan. Chest Pathology Detection Using Deep Learning with Non-medical Training. In *ISBI*, pages 294–297, 2015.

[3] Yuxi Dong, Yuchao Pan, Jun Zhang, and Wei Xu. Learning to Read Chest X-Ray Images from 16000+ Examples Using CNN. In *CHASE*, pages 51–57, 2017.

[4] Jagpal Singh, Jashanbir Singh Kaleka, and Reecha Sharma. Different Approaches of CBIR Techniques. In *International Journal of Computers and Distributed Systems*, pages 76–78, 2012.

[5] Felix Mayer and Martin Steinebach. Forensic Image Inspection Assisted by Deep Learning. In *ARES*, 2017.

[6] Qi Dai, Rui wei Zhao, Zuxan Wu, Xi Wang, Zichen Gu, Wenhai Wu, and Yu-Gang Jiang. Fudan-Huawei at MediaEval 2015: Detecting Violent Scenes and Affective Impact in Movies with Deep Learning. In *MediaEval*, 2015.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 2012.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, pages 580–587, 2014.

[9] Minyoung Huh, Pulkit Agrawal, and Alexei A. Efros. What Makes ImageNet Good for Transfer Learning? In *arXiv:1608.08614*, 2016.

[10] Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the Performance of Multilayer Neural Networks for Object Recognition. In *ECCV*, pages 329–344, 2014.

[11] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning Visual Features from Large Weakly Supervised Data. In *ECCV*, pages 67–84, 2016.

[12] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The New Data in Multimedia Research. In *CACM*, pages 64–73, 2016.

[13] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From Generic to Specific Deep Representations for Visual Recognition. In *CVPRW*, pages 36–45, 2015.

[14] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Robert Fergus, and Yann LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *ICLR*, 2014.

[15] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How Transferable are Features in Deep Neural Networks?

[16] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In *arXiv:1609.04836*, 2017.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *CVPR*, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, volume 770–778, 2015.

[19] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[20] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features Off-the-shelf: an Astounding Baseline for Recognition. In *CVPR*, 2014.

Author Biography

Felix Mayer *received his master's degree in computer science from Technische Universität Darmstadt, Germany in 2016. Since then he has worked in the Media Security and IT Forensics division at Fraunhofer Institute for Secure Information Technology (Fraunhofer SIT) in Darmstadt, Germany. His work is focused on object detection and data triage in the context of civil security research.*

Marcel Schäfer *graduated in mathematics at Bergische Universität Wuppertal in 2010. Since then he is a research fellow at Fraunhofer SIT in Darmstadt in the field of media security, civil security, big data and privacy. Besides he finished his PhD in computer science at Technische Universität Darmstadt in 2016.*

Dr. Martin Steinebach *is the manager of the Media Security and IT Forensics division at Fraunhofer SIT. From 2003 to 2007 he was the manager of the Media Security in IT division at Fraunhofer IPSI. In 2003 he received his PhD at the Technische Universität Darmstadt for his work on digital audio watermarking. Since 2016 he is honorary professor of Technische Universität Darmstadt.*