FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 45 (2020) No. 3
ISSN 0867-6356
DOI: 10.2478/fcds-2020-0010 e-ISSN 2300-3405

Transfer Learning Methods as a New Approach in Computer Vision Tasks with Small Datasets

Andrzej Brodzicki[1], Michal Piekarski[1,2], Dariusz Kucharski[1],
Joanna Jaworek-Korjakowska[1], Marek Gorgon[1]

Abstract. Deep learning methods, used in machine vision challenges, often face the problem of the amount and quality of data. To address this issue, we investigate the transfer learning method. In this study, we briefly describe the idea and introduce two main strategies of transfer learning. We also present the widely-used neural network models, that in recent years performed best in ImageNet classification challenges. Furthermore, we shortly describe three different experiments from computer vision field, that confirm the developed algorithms ability to classify images with overall accuracy 87.2-95%. Achieved numbers are state-of-the-art results in melanoma thickness prediction, anomaly detection and *Clostridium difficile* cytotoxicity classification problems.

Keywords: Deep neural networks, Transfer learning, Signal processing, Image analysis, Anomaly detection

1. Introduction to Transfer Learning Methods

The fundamental problem of artificial intelligence including machine learning and deep learning methods is the amount and quality of data. While the ideal AI scenarios highlight the technology's incredible computational power and offer promising results, the practical applications begin with raw, mostly unbalanced data classes. In some cases, the process of gathering the data might be unexpectedly expensive or even impossible. There is no single reasons for that. Sometimes an experiment can not be repeated or data gathering can be dangerous or harmful.

[1] Department of Automatic Control and Robotics, AGH University of Science and Technology, Krakow, Poland, e-mail: jaworek@agh.edu.pl

[2] SOLARIS National Synchrotron Radiation Centre, Jagiellonian University, Krakow, Poland, e-mail: piekarski@agh.edu.pl

There are many ways to artificially create more data samples (oversampling). Simple image transformations can produce slightly different images. More sophisticated methods, like SMOTE (Synthetic Minority Over-sampling Technique), creates new data that match specific class criteria [1]. Sometimes however, no matter how we augment the data it may not be enough to successfully train a network from scratch. Modern neural networks are trained on millions of images, while in practical problems there are merely hundreds. Furthermore, artificially created data inherit some features of their source. Trained model can therefore fail to represent the whole diversity of a class. Last, but not least, even if we have access to a large dataset, training a network on millions of data requires great computation power and a lot of time.

One of the answers to this problem is an idea of sharing, not the data itself, but the neural network model which actually has already "seen" similar or even different data before. This kind of approach is called transfer learning. It refers to a process where a model is first trained on a problem similar to the problem that is being solved (although on very different samples) and later used in another task. The idea of sharing knowledge between machine learning models was known even before the onset of modern deep learning [16]. However, it only became popular after convolutional neural networks (ConvNets) started to beat other algorithms.

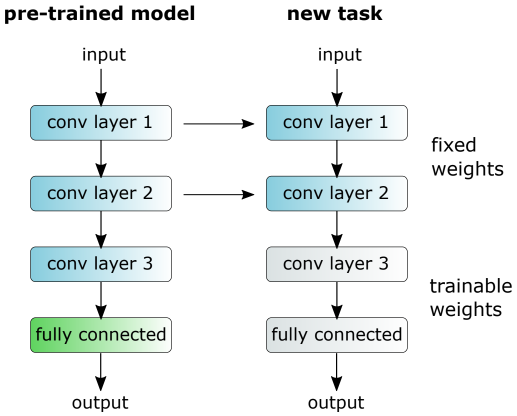


Figure 1. Schematic view of transfer learning idea.

Transfer learning is usually applied to topics like computer vision, signal processing, and natural language analysis. In the first of that fields, state-of-the-art results are recently achieved by deep convolutional neural networks. It is possible thanks to the fact that the lower layers of a ConvNet typically detect common patterns like lines and edges, the middle ones learn filters that detect parts of objects, while the last layers learn to recognize full objects, in different shapes and positions. The knowledge gained may be reused in the similar problem domain [17]. The concept of transfer learning is to use most of the layers from a pre-trained model and retrain only a final few for a new, different tasks (Fig. 1). The definition of transfer learning is described in terms of domain and task and presented in paper [11]: given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$. Reusing parts of an existing model, which we described here, is only one of many methods defined as transfer learning. A more detailed description, including division of those methods into four categories can be found in [15]. In the category of network-based transfer learning, there are two main strategies:

- **Feature Extraction** – training a new classifier on top of the pre-trained base model. In this method we leave the weights learned by convolution layers unchanged and train only the last, fully connected layer. It is a fast, simple, but still quite effective way to use ready-made architecture.
- **Fine-tuning** – retraining not only the fully connected layer, but also adjusting one or more convolution layers. In this solution we unlock some layers of a base model and train both the newly-added classifier and the last few layers of the base model. The weights from the original training are treated as the starting point. Unlocked convolution layers are not trained from the beginning, but only tuned to a new task. This method can improve model's performance, but can sometimes lead to overfitting. It is also more time-consuming.

The effectiveness of convolutional neural networks has been proven in many computer vision problems due to their powerful feature representation. Complete algorithms, used in many of our researches, require only simple data preprocessing and augmentation. It is then followed by re-training final layers of existing model, according to transfer learning methodology. Finally, results are tested using several metrics, like accuracy, precision, recall or ROC curve analysis. For a proper verification, we split data into train, test and validation subsets as well as use cross-validation. These steps are presented in Figure 2.

In the next section we briefly describe the history and architectures of most popular convolutional neural networks that achieved best results in many challenges. In section three *Small datasets classification problems in machine vision* we present the most interesting research projects which have been carried out by our research team in the field of transfer learning methods.

2. Deep Transfer Learning Models

One fair statement is that "accuracy not only depends on the network but also on the amount of data available for training". It has been widely proved that for traditional machine learning algorithms, performance grows according to a power law and then reaches a plateau, while deep learning performance scales with increasing data size. One of the large visual databases is the ImageNet project, which currently has 14,197,122 images from 21,841 different categories [12]. It is designed for use in visual object recognition software research. Since 2010 ImageNet runs annual challenge

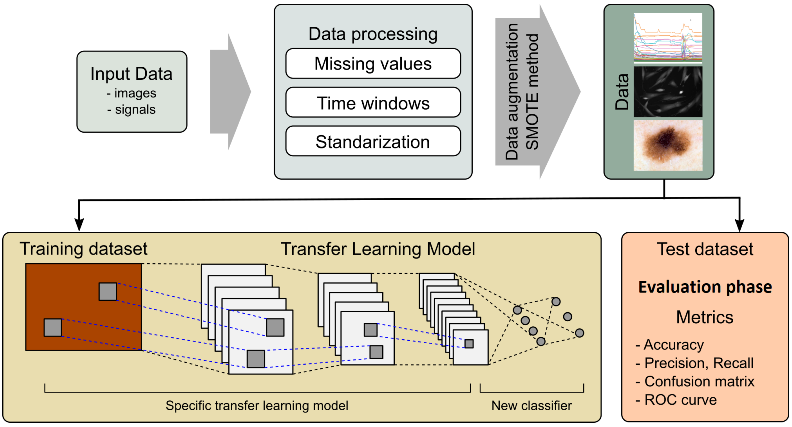


Figure 2. Transfer learning diagram for computer vision tasks including following steps: data processing, data augmentation, training and test data division, training process of the classification layer, and evaluation phase.

called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Original task was object localization – the dataset of 1000 non-overlapping object categories. Now, different algorithms compete to correctly classify and detect objects and scenes both on images and videos. Thanks to GPU resources and ImageNet database, quality an explosion of rapid development could have been noticed in the field of deep learning. It caused a huge production of many models such as AlexNet, Inception. ResNet or DenseNet, which we shall describe shortly.

2.1. AlexNet

Designed by Alex Krizhevsky, this architecture was one of the very first networks to push ImageNet classification accuracy by a significant stride compared to traditional methodologies. Introduced during ILSVRC in 2012, it outperformed previous state-of-the-art solution. It took advantage of GPU implementation by making convolution operation faster and more efficient [7]. As computing resources were the main limitation, the architecture has been optimized to use two GPUs available, in which calculations were performed in parallel (see Fig. 3).

AlexNet is a simple model, composed of 5 convolutional layers followed by max-pooling layers, used together for feature extraction part. For the classification process, the network uses 3 fully connected layers with Softmax activation. Non-saturating ReLU activation functions are used for a better training performance. Total number of parameters is 60 million and the number of neurons reaches 650 000. AlexNet has been an inspiration for researchers to use GPU resources to train their architectures.

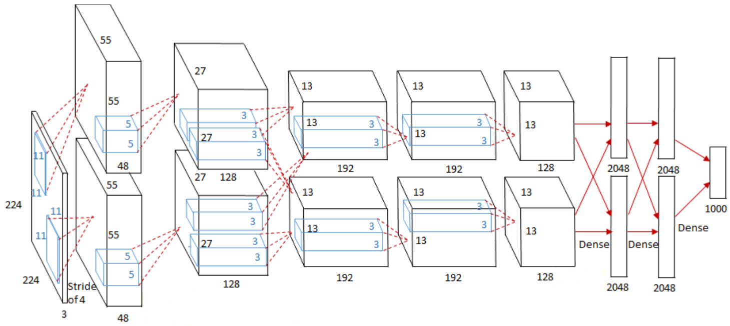


Figure 3. Architecture of AlexNet. One GPU runs the top layers and the second one the bottom layers. The GPUs communicate only at certain layers [7, 10]

2.2. VGG-16 and VGG-19

In 2012, K. Simonyan and A. Zisserman from Visual Geometry Group (University of Oxford) submitted for ILSVRC [13]. They presented two similar networks architectures: VGG-16 and VGG-19 and took the first and second places in the localisation and classification tracks respectively. Novelty of those models was to use only small convolution filters (3 × 3) which, in combination with the power of the GPU cluster, allowed to increase the depth of the network up to 16 and 19 layers. Nowadays VGG model is considered to be one of the best for transfer learning in image recognition tasks because of its simple architecture and high generalization ability. The VGG-16 model has roughly 134 million parameters and contains 16 trainable layers including convolutional as well as fully connected, max pooling, and dropout layers. The VGG-19 version has 144 million parameters and 19 trainable layers (see Fig. 4c).

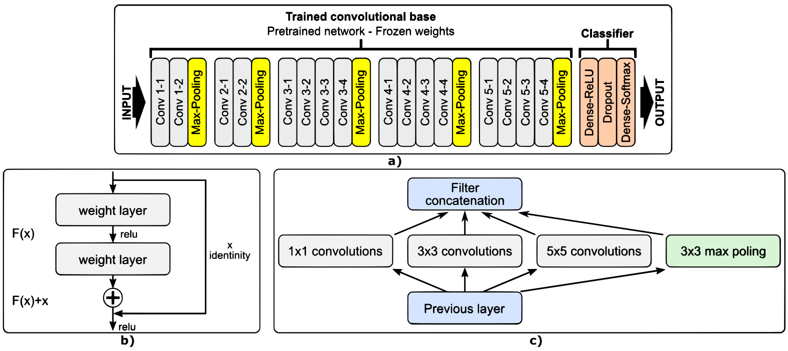


Figure 4. Transfer learning models: a) schematic overview of the VGG-19 network architecture, b) residual learning building block, c) an Inception block example [14]

2.3. Inception

During ImageNet Challenge in 2014 Szegedy et al. presented Inception architecture. This particular instance was called GoogLeNet. [14]. It outperformed VGG-19 significantly while having 12 times fewer parameters. The main goal (and the most obvious way to improve the network's performance) was to increase its size, both in depth and width. However this simple solution came with major drawbacks like risk of overfitting or, more importantly, dramatic increase in computational resources needed. Proposed approach was to move from fully connected to sparsely connected architectures. As a result, the main idea behind the Inception model is to connect several layers parallelly in a kind of block instead of stacking up one on another (Fig. 4a). It was assumed that a network utilizing such an approach will choose the most useful layers rising its weights, while decreasing useless layers at the same time (based on the Hebbian principle) [14]. Moreover, 1×1 convolution has been introduced, which helped reducing the feature-map dimension and global average pooling [8].

2.4. ResNet

Over the years many ImageNet challanges have shown that the depth of the network is a key factor and many non-trivial tasks benefited from very deep models. However, when model increases in depth it becomes more and more difficult to train. Problems that commonly occurred were degradation and vanishing gradients (at some point accuracy saturates and than degrades rapidly). In deep learning networks, a residual learning framework helps to preserve good results through a neural network with many layers. The deep residual network deals with problems mentioned earlier by using residual blocks, which take advantage of residual mapping to preserve inputs (Fig. 4b). A Residual Network (ResNet) consists of a set of layers stacked one on another. The characteristic feature of this architecture is a shortcut at each layer to directly connect the input with the output [4]. Layer together with this shortcut is called a residual block. This network won the ImageNet classification task in 2015, presenting 152-layer model. Despite being eight times deeper than VGG-19 it still had lower complexity. Other challenges (i.e COCO - Common Objects in Context [9]) shown that residual learning principle is generic and useful to other problems [4].

2.5. Xception

Over time, more research groups began to think about improving the concept of Inception architecture. Chollet et al. started studying different versions of Inception - based models, like InceptionV1, InceptionV3, GoogLeNet and Inception-ResNet in order to find a way to increase performance not by capacity but rather by more efficient use of model parameters [2]. Proposed architecture was called *extreme Inception* - Xception for short. Novelty of this approach was to introduce depthwise separable convolution layers to the underlying Inception model (see Fig. 5). Separable convolution in deep learning frameworks, consists in a depthwise convolution, i.e. a spatial convolution performed independently over each channel of an input [2]. The Xception architecture has 36 convolutional layers grouped in 14 modules whereas data flow consists of three steps: entry flow, middle flow (repeated 8 times) and finally, exit flow. Compared to Inception V3, Xception shows slightly better performance on the ImageNet dataset.

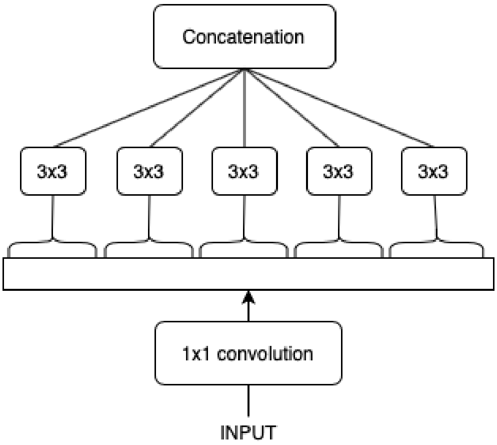


Figure 5. An *extreme* version of Inception module

2.6. DenseNet

Previous works have proven that convolutional networks can be deeper, more accurate and train efficiently if they contain short connections in their architecture. Residual Networks (ResNets) with this key characteristic in topology broke down the barrier of 100 layers. Huang et al. has gone one step further with this concept and introduced the Dense Convolutional Network – DenseNet for short [5]. It connects each layer to every other layer in a feed-forward fashion. Figure 6 illustrates this architecture schematically. It has been shown that such a model has a lot of advantages. Firstly, better parameter efficiency: DenseNet with 20 million parameters achieved comparable results as ResNet with 40 millions. Secondly, DenseNet has improved flow of information and gradients which contributes to easier training, strong feature propagation and vanishing-gradient problem elimination. Finally, it is also worth mentioning that dense connections have regularizing effect which reduces overfitting. DenseNet architecture significantly outperformed current state-of-the-art results in most challenges (ImageNet, CIFAR etc.).

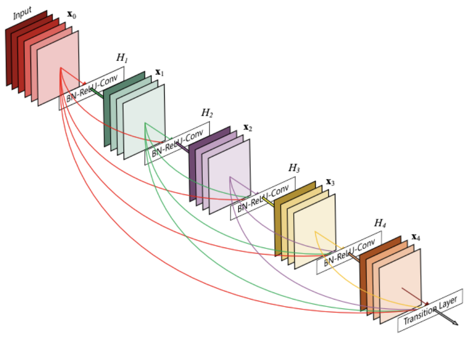


Figure 6. DenseNet net model architecture [5].

3. Small datasets classification problems in machine vision

To successfully train a deep neural network in machine vision tasks a diverse dataset is needed. The question arises: *can transfer learning be the answer to this issue*? We present three examples where we solved the problem of small datasets by using a dedicated transfer learning method. Table 1 summarizes the described projects in terms of dataset samples and achieved accuracy.

Table 1. Summary of the described research projects using transfer learning methods.

Title	Dataset (samples)	Accuracy [%]
Thickness prediction	244	87
Anomaly detection	10,000(imbalanced)	95
Cell classification	369	93

3.1. Melanoma Thickness prediction

Thickness is one of the most important factor in melanoma prognosis. To address this problem, we have implemented an effective computer-vision based deep learning tool that can perform the preoperative evaluation [6]. The novelty of our approach is that we directly predict the thickness into one of three classes: less than 0.75 mm, 0.75-1.5 mm, and greater that 1.5 mm, based solely on dermoscopic images (see Fig. 7). We have used transfer learning of the pre-trained, adapted to our application, VGG-19 convolutional neural network (CNN) with an adjusted densely-connected classifier. Our database contained only 244 dermoscopic images. Experiments confirm the developed algorithms ability to classify skin lesion thickness with 87.2% overall accuracy what is a state-of-the-art result in melanoma thickness prediction.

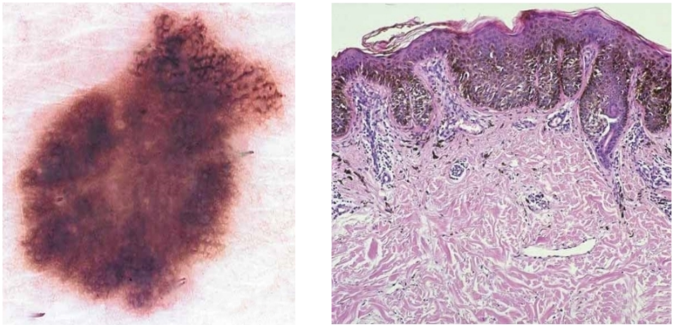


Figure 7. Melanoma thickness prediction

3.2. Anomaly detection solution

Deep convolutional pre-trained neural network VGG-19 was used to detect abnormal situations in multivariate diagnostic signals. Motivation of this research was to increase synchrotron's beam stability by detecting anomalies in different subsystems of the machine. Currently, only slow-changing anomalies are detectable with considerable attention and experience of the operator. The general idea is shown in Figure 10b. The input signals were pressure readings from the storage ring. Due to the fact that anomalies are rare, problem with the predominant number of samples in non-anomaly dataset occurred. To solve the problem of unbalanced classes SMOTE algorithm (Synthetic Minority Over-sampling Technique) has been used, which generates new, synthetic data. Finally, the dataset contained 9898 training samples and 1644 test samples, where the data were balanced and each class was equally represented. The validation set was created by randomly choosing 20% of samples from training dataset. In the proposed model, the classifier has been created specifically for the problem of recognizing two classes: anomaly and correct signal. The first layer is the Fully-Connected layer with ReLU activation function, the next is the Dropout layer with rate of 0.5. The classifier finishes the Fully-Connected output layer with two outputs and the Softmax activation function. As we were dealing with two class classification, the binary cross-entropy loss function has been applied, as an optimiser we have chosen the Adam optimization algorithm. The model achieved very good results, reaching accuracy of 95% by only 10,000 examples in the dataset. Mistakes were mostly false alarm (FP - false positive, see Table 2).

Table 2. Confusion matrix in the validation process for the analyzed VGG-19 based model.

Actual class	**Predicted class**	
	positive	**negative**
positive	697	125
negative	20	802

This is a very desirable property for a system that detects anomalies and situations that are potentially dangerous for the infrastructure. Tests have also shown that, except for the learning part which can be up to couple of minutes, the classification of time windows is very fast. Building such a system was possible due to the use of the advantages of transfer learning, because the database used to teach the classifier was created by the Authors, which obviously limited its size.

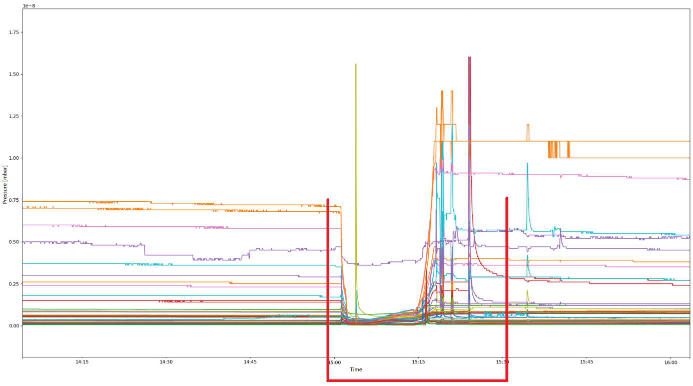


Figure 8. Anomaly detection.

3.3. Biomedical image classification

We proposed an algorithm for automatic classification of *Clostridium difficile* bacteria cytotoxicity. This infection is one of the most common contagious disease in hospitals. With antibiotics being one of the risk factors, novel forms of therapies are constantly developed [3]. Our algorithms help to speed up the process of testing and make the results more reliable than human subjective approach.

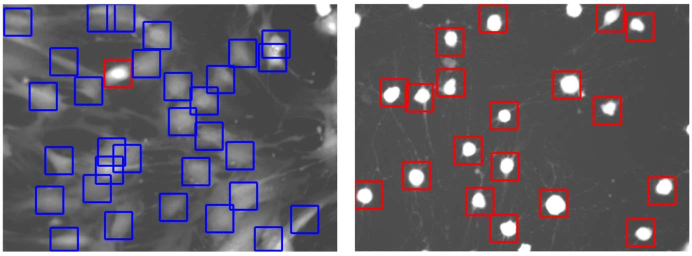


Figure 9. Clostridium difficile cytotoxicity classification. Blue frames mark living cells, red – dead cells

There were 369 fluorescent images available, depicting both dead and alive human cells (see Fig. 9). At first, we segmented them using classical image processing methods, such as adaptive binarisation and watershed transform. From this images we created and labelled a dataset of 6112 individual cells. To balance the number of samples in positive and negative class we applied simple image transformations, thus creating two classes with approximately 4000 samples each. We split them randomly into train (60%), validation (20%) and test (20%) subsets.

Table 3. Optimal parameters for DenseNet121 architecture, based on accuracy.

Activation function	Dropout	Optimizer	Batch Size	Epochs
sigmoid	0.5	AdaMax	256	30

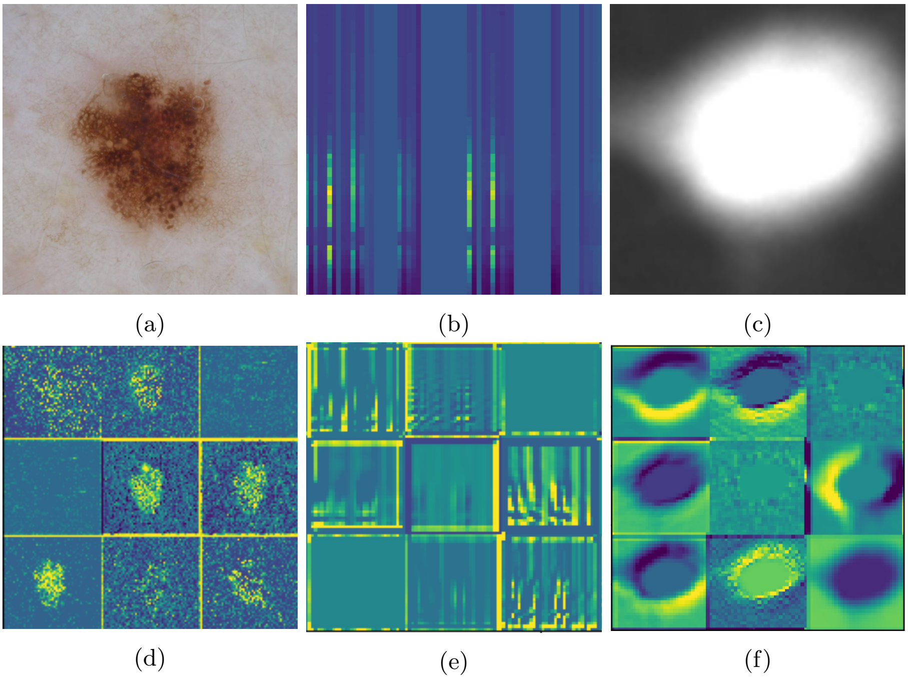
Then, for binary classification as either dead or alive, we trained and compared four convolutional neural network architectures (mentioned before in 2) – VGG19, ResNet50, Xception and DenseNet121. We used grid search optimisation to choose both model parameters (activation function in a densely connected layer, dropout rate) and learning hyperparameters (optimizer, batch size, number of epochs) for each of the four models. The best one, DenseNet121, achieved an average accuracy of 93% as well as 92% sensitivity and 94.5% specificity, with other three being only slightly worse. Confusion matrix for this best architecture can be found in Table 4, while used parameters are presented in Table 3.

Table 4. Best model (DenseNet121) prediction results on a test set.

Actual class	**Predicted class**	
	positive	**negative**
positive	745	66
negative	46	785

3.4. Results visualisation

Although the majority of layers in pre-train model have fixed weights, we can still look inside and observe network's behavior. This is done by visualising activations of subsequent layers, in response to different input images (see Fig. 10). By analysing this maps we can see which features are important. The deeper we go, the network reaches greater level of abstraction and the objects became less recognisable. Sometimes defining precisely how the network achieved a result is difficult. However, by this kind of visualization, artificial intelligence can be made more explainable. Another useful visualisation technique is creating heatmaps (see Fig. 11). In this technique, activation of final convolution layer is superimposed on top of the input image.



(a) (b) (c)

(d) (e) (f)

Figure 10. Activation visualisations of the first, convolutional layers (low level of abstraction) of pre-trained VGG-19 model in response to different input images from the mentioned examples. a) melanoma input image, b) standardized anomalies input window, c) cell input image, d) - f) activation visualisations for melanoma, anomalies and cells inputs, respectively

4. Discussion and conclusion

Presented examples show that transfer learning is a universal method, that may be applied to solve different challenging tasks. Medical applications prove that not only pictures similar to those in ImageNet dataset can be correctly classified. Moreover, the detection of synchrotron anomalies is an example of using image recognition methods in the case of non-image related problems. However, transfer learning methods have also many limitations. Currently, one of the biggest challenges of transfer learning is the problem of negative transfer. The distribution of the training data, which are used to pre-train the model, should not vary too much from the test data, and the data should not overfit the model.

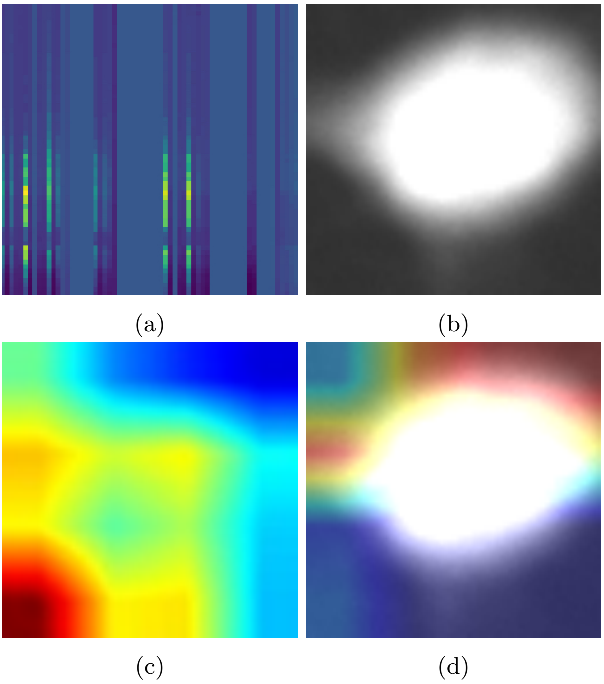


Figure 11. Activation of the last convolutional layer associated with a particular output class. It shows the importance of each part of the image for predicted class. a) standardized anomalies input window, b) cell input image, c) - d) heatmaps for anomalies and cells inputs, respectively

Acknowledgment

This scientific work was financially supported by AGH University of Science and Technology Statutory Funds on Decision No. 16.16.120.773.

References

[1] Bowyer K. W., Chawla N. V., Hall L. O., and Kegelmeyer W. P. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011.

[2] Chollet F. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[3] Garland M., Jaworek-Korjakowska J., Libal U., Bogyo M., and M. S. An automatic analysis system for high-throughput clostridium difficile toxin activity screening. *Applied Science*, 8(1512), 2018.

[4] He K., Zhang X., Ren S., and Sun J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] Huang G., Liu Z., van der Maaten L., and Weinberger K. Q. Densely connected convolutional networks, 2016.

[6] Jaworek-Korjakowska J., Kleczek P., and Gorgon M. Melanoma thickness prediction based on convolutional neural network with VGG-19 model transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[7] Krizhevsky A., Sutskever I., and Hinton G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[8] Lin M., Chen Q., and Yan S. Network in network. *International Conference on Learning Representations*, 2014.

[9] Lin T.-Y., Maire M., Belongie S., Bourdev L., Girshick R., Hays J., Perona P., Ramanan D., Zitnick C. L., and Dollár P. Microsoft coco: Common objects in context, 2014.

[10] Medium.com. Review: AlexNet, CaffeNet — winner of ILSVRC 2012 (image classification). https://medium.com/coinmonks/paper-review-of-alexnet-caffenet-winner-in-ilsvrc-2012-image-classification-b93598314160, 2018. [Online; accessed 20.06.2020].

[11] Pan S. J. and Yang Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010.

[12] Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A. C., and Fei-Fei L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[13] Simonyan K. and Zisserman A. Very deep convolutional networks for large-scale image recognition. *ArXiv:1409.1556*, 2014.

[14] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S. E., Anguelov D., Erhan D., Vanhoucke V., and Rabinovich A. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[15] Tan C., Sun F., Kong T., Zhang W., Yang C., and Liu C. A survey on deep transfer learning. *CoRR*, abs/1808.01974, 2018.

[16] Torrey L. and Shavlik J. Transfer learning. *Handbook of Research on Machine Learning Applications*, 01 2009.

[17] Yosinski J., Clune J., Bengio Y., and Lipson H. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

Received 29.02.2020, Accepted 29.07.2020