10.31341/jios.46.2.2
Open Access
UDC 004.032.26:004.8:005.92
Preliminary Communication

Office Documents Classification under Limited Sample. A Case of Table Detection Inside Court Files

Paweł Baranowski *pawel.baranowski@ieef.lodz.pl*
Department of Economics and Sociology
University of Lodz, Lodz, Poland, and
Section of Economic Cybercime
Institute of Economic and Financial Research, Lodz, Poland

Adrian Stepniak *adrstepniak@gmail.com*
Department of Economics and Sociology
University of Lodz, Lodz, Poland

Abstract

Deep convolutional neural networks (CNNs) became an industry standard in image processing. However, in order to keep their high efficiency, a large annotated sample is required in the case of supervised learning. In this paper we apply the techniques specific for relatively small sample to a court files dataset. Specifically, we propose transfer learning and semisupervised learning to classify scanned page as having a table or not. We use four CNNs architectures established in the literature and find that transfer learning improves the classification performance, compared to the fully supervised learning. This result is especially evident in the scenarios where only a part of convolutioanl layers are transferred. The gains from semisupervised learning are ambiguous, as the results vary over CNNs architectures. Overall, our results show that office documents classification can achieve high accuracy when transferring initial convolutional layers is applied.
Keywords: Convolutions, Deep learning, Document processing, Image classification, Office documents, Table detection, Transfer learning

1. Introduction

Business litigation and business crime cases often require analysis of financial statements, bank documents and accounting records. Such data make up only a small portion of the documents (court files) and at the same time, this information is contained mostly in tables. Given the size of court files (thousands of pages), we find automated table detection appealing. Specifically, we perform a binary image-level classification—each image (scanned page) is classified according to whether it contains a table. Motivated by previous studies, we utilize deep convolutional neural networks, which proved to be efficient in a variety of image processing tasks.

However, one of the most important concerns related to CNNs (and other computer vision models) is the growing complexity of the models. Due to the rapid technological development, e.g. widespread use of high-performance graphics processing units, hardware is no longer a limitation. Nevertheless, to ensure the generalization of the results, the size of the sample should grow with the complexity of the models. In this case, traditional supervised learning may be cumbersome, especially for specific issues when there the sample needs to be manually annotated from the beginning. In the paper, we address the issue of a limited sample by re-using the convolutional layers trained over a large sample, either from another domain (transfer learning) or from an unannotated sample of office documents (semi supervised learning). These two approaches are confronted with a standard supervised classification. Such empirical comparison allows to verify the efficiency of transfer learning or semisupervised learning in table detection. Besides, we perform several experiments to compare the predictive accuracy of the models with partial transfer learning, hence find an optimal scope of the parameter transfer.

The paper is organized as follows: in Section 2, we provide an overview of the literature as well as the motivation for the study. Section 3 presents the research methods, including neural networks architecture (sec. 3.1), approaches to machine learning (sec. 3.2) and data used in the study (sec. 3.3). Then we present the results of the empirical research, in terms of out-of-sample classification performance, and the discussion (Section 4). We conclude our paper in Section 5.

2. Literature

A vast amount of literature has used deep neural networks to solve computer visions problems. The wide group of image recognition methods are often aimed at automating human works (image recognition and classification, object detection, image segmentation). In this study, we rely on convolutional neural networks (CNNs)—a special type of artificial neural networks designed specifically for image analysis [15, 16]. The main building blocks of a convolutional network are convolutional layers—composed of filters, with each filter activated by a specific pattern. In addition, the pooling layers, following one or more convolutional layers, transform the image in the way that subsequent convolutional layers respond to larger fragments of a source image. The main advantage of CNNs is the ability to detect local patterns, no matter where a given shape is in the image (as opposed to traditional densely connected neural networks). Their high efficiency has been widely confirmed (see [13], [28], [23], [24], [21]).

One of the first state-of-the-art CNNs to exploit this insight was the VGG model [27], which used blocks that consisted of consecutively stacked convolutional layers and a pooling layer at the end. In addition, the convolutional layers use filters of smaller size than in earlier, shallower architectures (this is based on the observation that 5x5 or 11x11 filter can be replaced by the product of an appropriate number of 3x3 filters).

A further step in the development of CNNs was GoogleNet architecture (now more commonly known as Inception-V1) [29]. The key idea was to use spatial

separable convolution, where convolutional layers were stacked parallelly rather than serially (as in VGG and earlier CNNs). The novel 'inception' block is aimed to split the input data into several parallel branches with convolutions with filters of different sizes, and finally aggregate the signals from these branches.

Whereas the Incepction used branches of equal depths, in terms of the number of convolutional layers, [10] used skip connections to pass some layers. More specifically, the input signal was split into two branches, where one was processed by convolutional layers and the other directly forwarded to the next layers (i.e. skips some convolutions). This architecture, called ResNet, allowed to prevent the vanishing gradient problem [6] and thus, the use of much deeper networks.

Xception (Extreme Inception) architecture [2] provided a generalisation of spatial separable convolutions. A novel technique was the use of depthwise separable convolution, in which a single convolutional filter for each input channel is applied, followed by a pointwise (1x1) convolution.

These four model architectures are applied in our empirical exercise, mainly due to three reasons. Firstly, their architectures are diverse, which allows a broad set of CNNs to be tested using transfer learning. Secondly, all these models are strongly represented both in the state-of-the-art academic literature as well as in business applications. Finally, despite the development of new variants of CNNs (see [13] for a survey) these models are effective in standard image processing tasks, like classification.

As already mentioned in the introduction, we focus on the realistic case with a limited annotated sample (target sample). Specifically, we confront fully supervised VGG, ResNet, Inception and Xception models with the analogue transfer learning and semisupervised models.

The earliest literature on image-based tables detection proposed rules based on graphical features such as: vertical/horizontal lines, lines crossings or regular distances between the lines etc. (see [3] for an overview). Another strand of the literature use textual content of the pages, such as assessing the regularity of white spaces between the words inside a table [4] or textual content of captions [8]. We do not follow this approach, as processing the text is especially vulnerable to OCR processing errors. Nevertheless, an extended method based on feature extraction as well as their localization within the image, was successfully applied to detect tables in websites [14]. Finally, vast majority of recent studies (e.g. [7], [5], [26]) apply deep convolutional neural networks—a method typically used in other fields of image analysis.

We argue that the classification of pages based on the unstructured images (scanned documents) to find the pages containing at least one table provides an interesting case of office automation and consequently may increase workers performance. In addition to the gains from process automation, the approach taken in our study may further reduce the amount of manual work, as we deal with a limited sample of labelled (annotated) pages. Since the usefulness of transfer learning or semisupervised learning in a wide range of computer vision applications has been proven (in bioinformatics, medical diagnostics, transportation, recommendation

systems, etc) [19, 31, 32], for office document classification this technique has not received much attention.

A few papers successfully applied these techniques for a number of issues related to table processing. First, [18] used semisupervised learning based on VGG architecture. Recently, [20] performed iterative transfer learning to deal both with table segmentation as well the recognition of the table type (borderless or bordered). Finally, [25] proposed feature detection based on transfer learning and VGG or ResNet architectures. These papers, however, focused on table segmentation rather than classification based on the presence of the table or tables. Our study adds to the literature by (1) considering various architectures, (2) comparing transfer learning from very general domain and semisupervised learning, (3) dealing with the real-life court files rather than documents from text editors (i.e. documents in our dataset were originally in paper form; therefore, the scans are not perfectly aligned horizontally and may contain minor impurities).

3. Data and methods

3.1. Model Architectures

As already mentioned, the main goal of the research is to check if transfer learning or semisupervised learning may be successfully applied to the case of table detection. Therefore, we tend to be agnostic with respect to the specific CNN architecture. Consequently, we employ four model architectures:

1. VGG-16 [27],
2. ResNet50 V2 [11],
3. Inception V3 [30],
4. Xception [2].

Table 1 provides a brief comparison of the models used in the study and report the size of the models (number of trainable parameters and convolution filters) as well as the architectural features. As a benchmark scenario, we use slightly modified architectures of the supervised models taken from the papers. To make the benchmark comparable, instead of using the original set of fully connected layers, we enforce identical set of top layers for all models: 10% dropout, a dense layer with 512 neurons and ReLU activation function, 10% dropout and final output neuron with a sigmoid activation function.

Architecture	Parameters	Convolutional layers	Branches	Skipconnections
VGG	15M	13	1	No
ResNet	26M	182	up tp 2	Yes
Inception	24M	94	up tp 4	No
Xception	23M	111	up tp 2	Yes

Table 1. Overview of classification models (supervised learning)

3.2. Approaches to machine learning

Each of the models presented in previous section is trained using different machine learning techniques. The techniques of our interests are transfer learning (TL) and semisupervised learning (SSL). Both TL and SLL are designed to deal with the case when the number of labelled cases is small, as it reduces the effort to manually annotate the content. Furthermore, both are based on a two-step procedure.

During the first step, a 'source' model is trained using a large sample of images. Such a model is able to detect a number of generic image features (shapes). In the TL approach, the sample comes from another domain than the target dataset although is labelled, therefore a supervised learning approach is used. On the other hand, in SSL the images are from a similar domain but are unlabeled (or even may be a superset of the target data, which represents the case of a small part of the dataset being labelled; this approach was employed in our study). Consequently, unsupervised learning is applied at the first step of SSL. The key notion behind this step is that these models can recognize generic shapes [31, 32] and at the same time the major part of the whole model (in terms of the number of the parameters) is responsible for detection of these features. In the case of SSL, we use a convolutional autoencoder. This architecture uses the same input as the output, but the model is trained to provide a good representation of the images, specifically images summarization by using a small number of elements of the latent vector, representing generic features of the image (bottleneck) - see Fig. 3. In the second step, the model is reshaped to comply with the binary output and then is trained using a relatively small labelled dataset. However, the first layers are not trained (freezing of the parameters).[1] The signal stemming from the detected patterns are then passed to the densely connected layers, responsible for classification. Importantly, only the densely connected layers are subject to the training on the target sample. Additionally, we include a TL scenario that freezes only a part of convolutional layers (partial transfer learning). As the consecutive layers represent a pattern of different size, this may be especially effective when only larger/smaller features are useful for the classification. All in all, the schemes for TL and SSL are presented on Fig. 2 and 3.

3.3. Datasets

As for the target dataset, we use the dataset containing ca. 11,000 scanned pages, annotated with respect to table presence (of which 32% contains a table, hence the output is relatively well balanced). We follow a conventional 80%/20% random sample split so that 9,076 observation are used for training and the remaining 2,190 for out-of-sample testing. The dataset consists of various types of tables, mostly bordered or semi-bordered. They also differ in size and placement on the page (see Fig. 1).

[1] In the case of the autoencoder the whole part of the model following the bottleneck is replaced by densely connected layers.

Figure 1. Examples of images including tables (a part of the content has been anonymized, due to Polish legal requirements with regard to court files)

Additionally, the documents are scanned from paper form, therefore suffer from minor scanning errors such as skewness or discoloration. Moreover, the original documents occasionally contain handwritten notes, signatures or stamps. Consequently, our table detection may be more challenging compared to analysis of datasets including only tables extracted from electronic files such as PDFs or text editors. These images are in grayscale and have the size of 192x192 pixels (the resolution of the image is relatively low, though it does not constrain the results). In addition, to prevent overfitting the models, we use several data augmentation methods: image rotation (by up to 10 degrees), horizontal and vertical shift (up to 10% of the image size), zoom (up to 20%), and horizontal mirroring. Each of these transformations is randomly applied to the images in the training set; the test set, however, remains constant throughout each stage of learning.

In the TL scenario, we use models pre-trained on the ImageNet [22] dataset. This dataset represents the general domain (food, furniture, electronics, animals, clothing, people) and is used for image recognizing over 1,000 categories of items. Each model was tested in an analogous manner. Parameters (weights) in convolutional layers were copied from the pre-trained model, then the same layers with the same parameters as in benchmark scenario were added sequentially: global average pooling, dropout, dense layer with ReLU activation, dropout and final output neuron with a sigmoid activation function. In addition, the number of layers whose weights were frozen varied, all convolutional layers or selected part of the first layers were frozen. A summary containing the number of layers and trainable parameters in each model is presented in the Table 2. The number of frozen layers was chosen so that the following number of all model parameters were not further trained:

1. approximately 50% of the parameters (Medium TL scenario),
2. approximately 75% of the parameters (Large TL scenario),
3. all parameters except those in dense layers (Full TL scenario).

In the case of SSL, we use the dataset of office documents, that is from similar domain, as opposed to TL scenario. This dataset consists of over 150,000 pages collected from the court cases examined by our institution (IEEF) and 350,000 pages

from RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing) dataset [9]. From the IEEF dataset, we include only the images with mean pixel intensity below 99%, consequently, blank pages are omitted. The testing procedure was analogous to the TL approach, except that in this case only the parameters from the encoder part were copied, the sequence of the final layers was identical.

Architecture	TL Scenario	Convolutional layers	Parameters	Transfer
VGG	Medium	13(3)	15.0 (7.3)	51%
	Large	13(1)	15.0 (2.6)	83%
	Full	13(0)	15.0 (0.3)	98%
VGG	Medium	182(22)	24.6 (9.9)	60%
	Large	182(9)	24.6 (5.5)	78%
	Full	182(0)	24.6 (1.0)	96%
Inception	Medium	94(8)	22.8 (12.1)	47%
	Large	94(9)	22.8 (7.1)	69%
	Full	94(0)	22.8 (1.0)	96%
Xception	Medium	111(31)	21.9 (11)	50%
	Large	111(6)	21.9 (5.8)	74%
	Full	111(0)	21.9 (1.0)	95%

Numbers of the parameters are given in millions; in the parentheses we report the number of unfreezed (trainable) convolutional layers or parameters in the target model.

Table 2. Summary of machine learning scenarios

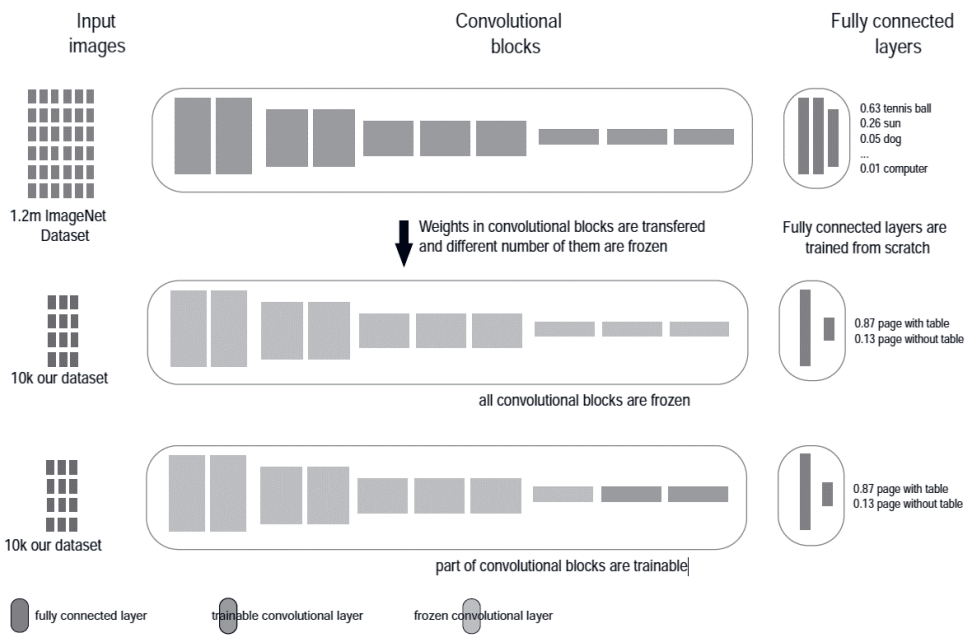

Figure 2. Transfer learning

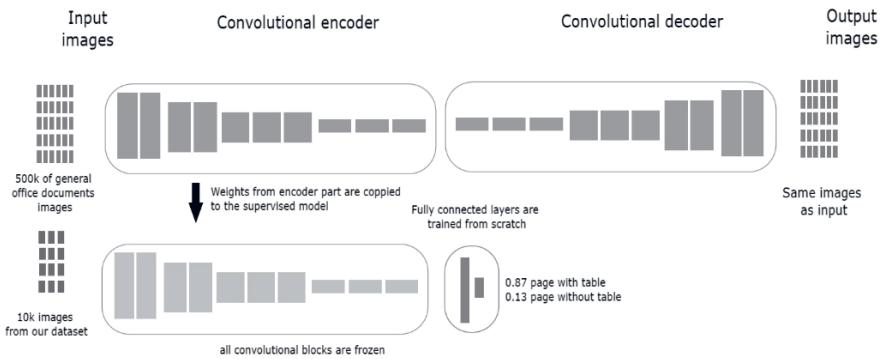

Figure 3. Semisupervised learning

4. Results

In Table 3 we show the performance of the models, as measured on the test sample (2,190 obs.). The set of scenarios include transfer learning models, either with different number of the convolutions being transferred from the general domain (see Table 2), as well as semisupervised learning models (autoencoders). Finally, we compare these with the benchmark, which is fully supervised models. We report two conventional classification metrics, that is accuracy of prediction ($Accuracy = \frac{TP+TN}{P+N}$) and recall ($Recall = \frac{TP}{P}$),[2] which express the percentage of correctly classified pages in the test sample and detected tables on the pages containing a table, respectively.

Supervised	VGG	Resnet	Inception	Xception
Accuracy	0.935	0.847	0.867	0.849
Recall	0.940	0.578	0.616	0.573
Transfer (Medium)	VGG	Resnet	Inception	Xception
Accuracy	0.966	0.965	0.963	0.973
Recall	0.962	0.922	0.938	0.934
Transfer (Large)	VGG	Resnet	Inception	Xception
Accuracy	0.948	0.952	0.948	0.952
Recall	0.936	0.901	0.908	0.894
Transfer (Full)	VGG	Resnet	Inception	Xception
Accuracy	0.868	0.931	0.927	0.913
Recall	0.640	0.846	0.855	0.797
Semisupervised	VGG	Resnet	Inception	Xception
Accuracy	0.939	0.801	0.913	0.926
Recall	0.856	0.545	0.774	0.959

Table 3. Classification performance metrics across the models and scenarios

[2] TP(TN) denote the number of pages correctly classified, with (without) a table. P(N) denote the number of pages with (without) a table.

The results presented in Table 3 reveal several interesting outcomes. Firstly, overall transfer learning tends to outperform the benchmark—supervised learning. The only exception is the VGG model, which performs well only in partial TL scenarios. In our opinion, this may be a result of relatively lower performance on the original task, that is classification into 1,000 categories on the ImageNet dataset (e.g., accuracy of the VGG was equal to 0.715, while for ResNet, Inception and Xception was 0.77 or higher, see [2]). However, the results of partial TL scenarios provide a stronger evidence for the gains from transfer learning in the case of table detection. Secondly, most TL results are stable across architectures, while the differences across the models in the benchmark scenario and SSL are larger. Thus, transfer learning is appealing not only in terms of predictive accuracy but also robustness. On the other hand, the results for SSL scenarios are mixed. Whereas Xception and Inception outperform the fully supervised benchmarks, VGG and ResNet are inferior or at most comparable to the benchmarks. Following [1], the performance of transfer learning from autoencoders depends on the consistency between original and target dataset. In SSL scenario we used a combination of court files dataset (30% of the images) and an external dataset containing various office documents. Experimenting with the choice of the source dataset may help to explain the differences between the SSL models. Yet another possibility to improve the SSL models could be using the first step convolutional layers' parameters only for the initialization of the training process (this idea was successfully applied by [12]). Thirdly, a comparison of all the scenarios indicate that transferring only part of the convolutions provides the best results (accuracy 95%-97%). This result holds for all CNNs architectures used in our study. Fourthly, the VGG benchmark significantly outperforms the remaining benchmark models, its performance is close to partial transfer learning models. This is not very surprising, given the fact that the number of parameters in VGG (15 million) and parameters re-trained in the partial TL approach are comparable (especially for ResNet and Inception).

5. Conclusion

This paper examines a predictive performance of deep convolutional neural networks (CNNs) applied for scanned document images processing, namely extracting the pages containing at least one table. To address the issue of the small annotated sample size, we apply transfer learning and semisupervised learning. Our results show that the transfer learning approach significantly outperforms the benchmark, supervised models trained on a small sample. Moreover, the gains from transfer learning are higher in the case when only a part of convolutional layers are transferred (Accuracy 95%). As noted by [17], yet another benefit from transfer learning may be lower energy consumption and carbon footprint for the model training. Overall, we find that transfer learning improves the classification and may be successfully utilized to automate table detection in large document collections such as court files.

Acknowledgment

The authors would like to thank Pawel Gajewski, Karol Korczak, Kamil Tylinski for commenting the first draft of the paper. We also acknowledge the comments gained from the participants of: the seminar at Institute of Computer Science, AGH University of Science and Technology, and 10th International Conference on Intelligent Information Processing. Any remaining errors are ours.

References

[1] Jorge Calvo-Zaragoza and Antonio-Javier Gallego. "A selectional auto-encoder approach for document image binarization". In: Pattern Recognition 86 (2019), pp. 37–47.

[2] François Chollet. "Xception: Deep learning with depthwise separable convolutions". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 1251–1258.

[3] David Doermann, Karl Tombre, et al. Handbook of document image processing and recognition. Springer, 2014.

[4] Jing Fang et al. "A table detection method for multipage PDF documents via visual seperators and tabular structures". In: 2011 International Conference on Document Analysis and Recognition. IEEE. 2011, pp. 779–783.

[5] Azka Gilani et al. "Table detection using deep learning". In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR). Vol. 1. IEEE. 2017, pp. 771–776.

[6] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[7] Leipeng Hao et al. "A table detection method for PDF documents based on convolutional neural networks". In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS). IEEE. 2016, pp. 287–292.

[8] Gaurav Harit and Anukriti Bansal. "Table detection in document images using header and trailer patterns". In: Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing. 2012, pp. 1–8.

[9] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. "Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval". In: International Conference on Document Analysis and Recognition (ICDAR).

[10] Kaiming He et al. “Deep residual learning for image recognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778.

[11] Kaiming He et al. “Identity mappings in deep residual networks”. In: European conference on computer vision. Springer. 2016, pp. 630–645.

[12] Jong-Hwan Jang, Tae Young Kim, and Dukyong Yoon. “Effectiveness of Transfer Learning for Deep Learning-Based Electrocardiogram Analysis”. In: Healthcare informatics research 27.1 (2021), pp. 19–28.

[13] Asifullah Khan et al. “A survey of the recent architectures of deep convolutional neural networks”. In: Artificial Intelligence Review 53.8 (2020), pp. 5455–5516.

[14] Jihu Kim and Hyoseok Hwang. “A Rule-Based Method for Table Detection in Website Images”. In: IEEE Access 8 (2020), pp. 81022–81033. DOI: 10.1109/ACCESS.2020.2990901.

[15] Yann Le Cun et al. “Handwritten digit recognition with a back-propagation network”. In: Proceedings of the 2nd International Conference on Neural Information Processing Systems. 1989, pp. 396–404.

[16] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: Proceedings of the IEEE 86.11 (1998), pp. 2278–2324.

[17] Jie Lucy Lu, Naveen Verma, and Niraj K Jha. “Convolutional Autoencoder-Based Transfer Learning for Multi-Task Image Inferences”. In: IEEE Transactions on Emerging Topics in Computing (2021).

[18] Shubham Singh Paliwal et al. “Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images”. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE. 2019, pp. 128–133.

[19] Sinno Jialin Pan and Qiang Yang. “A survey on transfer learning”. In: IEEE Transactions on knowledge and data engineering 22.10 (2009), pp. 1345–1359.

[20] Devashish Prasad et al. “CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020, pp. 572–573.

[21] Waseem Rawat and Zenghui Wang. “Deep convolutional neural networks for image classification: A comprehensive review”. In: Neural computation 29.9 (2017), pp. 2352–2449.

[22] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: International Journal of Computer Vision (IJCV) 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[23] G. Sapijaszko and W. B. Mikhael. “An Overview of Recent Convolutional Neural Network Algorithms for Image Recognition”. In: 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS). 2018, pp. 743–746. DOI: 10.1109/MWSCAS.2018.8623911.

[24] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: Neural networks 61 (2015), pp. 85–117.

[25] Muhammad Ali Shahzad et al. “Feature Engineering meets Deep Learning: A Case Study on Table Detection in Documents”. In: 2019 Digital Image Computing: Techniques and Applications (DICTA). IEEE. 2019, pp. 1–6.

[26] Shoaib Ahmed Siddiqui et al. “Decnt: Deep deformable CNN for table detection”. In: IEEE Access 6 (2018), pp. 74151–74161.

[27] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: arXiv preprint arXiv:1409.1556 (2014).

[28] V. Sze et al. “Efficient Processing of Deep Neural Networks: A Tutorial and Survey”. In: Proceedings of the IEEE 105.12 (2017), pp. 2295–2329. ISSN: 1558-2256. DOI: 10.1109/JPROC.2017.2761740.

[29] Christian Szegedy et al. “Going deeper with convolutions”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 1–9.

[30] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 2818–2826.

[31] Chuanqi Tan et al. “A survey on deep transfer learning”. In: International conference on artificial neural networks. Springer. 2018, pp. 270–279.

[32] Fuzhen Zhuang et al. “A comprehensive survey on transfer learning”. In: Proceedings of the IEEE 109.1 (2020), pp. 43–76.