Neural Networks for Tool Image Classification

Boas Bamberger [1], Oliver Erlenkaemper[2] and Fabian Wolf[3]

***Abstract*— Augmented reality solutions for field workers are an emerging market. [1]–[5] An augmented reality solution for field workers requires software perceiving the environment of field workers. A sub-task of that perception is the classification of different tools. We seek to determine the best-performing neural network for tool image classification. Furthermore, we introduce a novel dataset for tool image classification (TIC Dataset). To determine the best-performing neural network for tool image classification we train and evaluate neural networks on the TIC Dataset. We select the neural networks for the experiment based on a literature review which we conducted. We found that, in general, not one particular but several neural networks are suitable for tool image classification, especially neural networks using convolutional layers and skip connections.**

I. Introduction

Augmented reality solutions for field workers are an emerging market. [1]–[5] An augmented reality solution for field workers requires software perceiving the environment of field workers. A sub-task of that perception is the classification of different tools. For example, a software displaying step-by-step instructions in the field of vision of a field worker needs to distinguish a screwdriver from a wrench when telling the field worker to tighten the screw with the wrench lying on the ground next to him instead of the screwdriver in his hand. This task is called tool image classification. We seek to determine the best-performing neural network for tool image classification. Furthermore, we introduce a novel dataset for tool image classification the TIC Dataset. To determine the best-performing neural network for tool image classification we train and evaluate neural networks on the TIC Dataset. We select the neural networks for the experiment based on a literature review which we conducted. Our time and resources were limited. For this reason, we exclusively train the neural networks supervised without auxiliaries. On that account, we exclude metalearning, especially neural architecture search, non-neural machine learning models, other computer vision tasks, unsupervised learning, semi-supervised learning, transfer learning, adversarial training, data augmentation, input normalization, weight decay, and multi-task learning. The following sections of this paper are structured as follows. Section II introduces related work and explains fundamentals required to understand this paper. Section III defines the methodology followed to determine the best-performing neural network for tool image classification. Section IV reports the results of the experiment conducted by this paper. Section V discusses the results and reflects on the limitations of our work. Section VI summarizes the contributions of this paper and proposes future work.

[1]Scientific Supervisor bamberger@uni-mannheim.de
[2]Business Supervisor oliver.erlenkaemper@honeywell.com
[3]Authors172298@student.dhbw-mannheim.de

II. Related Work

We seek to determine the best-performing neural network for tool image classification. To the best of our knowledge our paper is the first paper on tool image classification. Tool image classification is a subtask of image classification which is an advanced field and mostly solved by neural networks. Neural networks achieve accuracies of $96.08\% - 98.66\%$ [6], [7] on , CIFAR [8], MNIST [9], SVHN [10], STL [11] and 88.61% [6] on ImageNet Dataset [12] which comprises 1000 classes and 3.2 million images. Especially convolutional neural networks (CNNs) [13], residual neural networks [14], inception networks [15] and dense neural networks [16] have advanced the field.

III. Method

We determine the best-performing neural network for tool image classification in the course of an experiment on the TIC Dataset. Optimally, we would conduct a neural network architecture search and hyperparameter search for the experiment, but since our computational resources are limited we select the neural networks for the experiment based on a literature review which we conducted.

A. Dataset Construction

The TIC Dataset is a dataset for tool image classification. Thus, we created images consisting of exactly one tool of different classes. We chose the classes based on the tools which were freely available to us. The resulting classes are drill, hammer, pliers, saw, screwdriver and wrench. In real world scenarios, tools are presented from various angles and with various backgrounds. Due to this, the created images are taken from arbitrarily chosen angles and with arbitrarily chosen backgrounds. We took all images of one class before taking images of another class and stored them in different folders. The resulting folder structure is comprised by a root folder containing one folder for each image class. Consequently the images are inherently labeled by the folder structure. To create an image we placed a tool in an arbitrarily chosen position and took a series of close-up images. During the series, the camera was moved around to create arbitrary angles. For the next series, the tool, the background, and/or the position of the tool was changed.

B. Model Selection

We selected the neural networks based on a literature review. The literature review was conducted according to the method of [17] to examine the state of the art of image classification in regard to neural network architectures. For each architecture we selected one neural network for the experiment. Since the best-performing neural network on the ImageNet Dataset for each architecture uses more RAM than available to our limited hardware [6], [18], [19], we selected the neural network from the paper originally proposing the architecture.

C. Experiment

We split the TIC Dataset $60\%/20\%/20\%$ into training development and test dataset. We train each neural network on the training dataset, tune its hyperparameters on the development dataset and evaluate it only once on the test dataset to prevent overfitting the test data. To conduct the experiment, we were provided 200 hours on an Amazon Web Service g4dn.xlarge instance running a Deep Learning AMI (Ubuntu 18.04). [20] The g4dn.xlarge instance provides 4 vCPUs, 16GB RAM, 125GB storage, and a NVIDIA T4 GPU. [21] We implemented the experiment in Python 3 [22] using Keras 2.2.4.2 [23], Tensorflow 2.1.0 [24], CUDA 10.1 [25], and cuDNN 7.5.0 [26].

Training and hyperparameter tuning was executed exactly as in the paper originally proposing the neural network, except for dataaugmentation, weight decay and dropout which we excluded in this paper, see Section I. Furthermore, we needed to adept the batchsize to the size of the available GPU RAM. We evaluated the performance of each neural network on the test dataset in accuracy.

IV. Evaluation

The accuracy for each neural network on the TIC Dataset is reported in Table I. For comparison we also report the original accuracy for each neural network on the ImageNet Dataset. Among these neural networks, DenseNet-264 performs best for the TIC Dataset. ResNet-152, ResNeXt-101, and DenseNet-264 perform rather similar. Therefore, we conclude that several neural networks are suited for tool image classification. EfficientNet-B7 achieved an accuracy of 16.67% meaning it did not learn at all.

TABLE I
Experiment Results

Neural Network	TIC Dataset	ImageNet
VGG-19	72.40	76.3 [27]
ResNet-152	92.89	78.57 [14]
ResNeXt-101	94.66	79.6 [28]
DenseNet-264	**97.45**	77.85 [16]
EfficientNet-B7	16.67	84.4 [29]

V. Discussion

We found that, in general, several neural networks are suitable for tool image classification. This is in accordance with the state of the art of image classification, as image classification leaderboards show that different neural networks perform rather similarly for different image classification tasks. [8]–[12], [30]–[43]

Especially, we found, that ResNet-152, ResNeXt-101, and DenseNet-264 are suitable for tool image classification. This also was to be expected since, ResNet-152, ResNeXt-101, and DenseNet-264 are suitable for image classification in general. [14], [16], [28] Despite differing in structure, each of these neural networks uses convolutional layers and skip connections. [14], [16], [28] This might be the cause of the similar performance. In comparison to these neural networks, VGG-19 performs less well. VGG-19 does not use skip connections. [27] Thus, we conclude, not using these connections might be the cause of the lower performance of VGG-19.

In comparison to the ImageNet Dataset ResNet-152, ResNeXt-101, and DenseNet-264 perform better on the TIC Dataset, this might be due to the different complexity of the datasets. ImageNet Dataset has 1000, while the TIC Dataset has 6. Furtheremore the classes are completely different. Accordingly the complexity of classification is different for each class.

EfficientNet-B7 did not learn at all. This might be, because, due to the limited GPU RAM, we were only able to train EfficientNet-B7 with a batch size of 1. A batch size of 1 causes the loss function to fluctuate heavily. This impairs convergence to the optimum and thus learning. On that account, the accuracy of EfficientNet-B7 reported in Section IV cannot be interpreted as the performance of EfficientNet-B7 for tool image classification, but simply as that EfficientNet-B7 was not able to learn in the course of the experiment. We excluded metalearning, especially neural architecture search, non-neural machine learning models, other computer vision tasks, unsupervised learning, semi-supervised learning, transfer learning, adversarial training, data augmentation, input normalization, weight decay, and multi-task learning. On that account, it is likely, that even higher accuracies can be achieved using these techniques in addition to our approach.

VI. Conclusion

We conclude that, for tool image classification convolutional neural networks using skip connections are suited. For industry aiming to create augmented reality solutions for field workers our implementation can be used as basic module of software perceiving the environment of field workers. For future work, we propose to improve our work with more performant hardware and additional training data to implement metalearning, semi-supervised learning, transfer learning, adversarial training, data augmentation, input normalization, weight decay, and multi-task learning, since these techniques were shown to improve performance. [44]–[46] Furthermore, we propose future work on the implementation

of a holistic augmented reality solution for field workers, i.e. further computer vision tasks, a software framework, and a hardware framework for implementation.

REFERENCES

[1] Ernst & Young, Stop talking about the future of work (2019).
URL https://assets.ey.com/content/dam/ey-sites/ey-com/en_au/topics/campaigns/future-of-work/stop-talking-about-future-of-work.pdf

[2] Ernst & Young, The future of work: the changing skills landscape for miners (2019).
URL https://minerals.org.au/sites/default/files/190214%20The%20Future%20of%20Work%20the%20Changing%20Skills%20Landscape%20for%20Miners.pdf

[3] C. Detzel, A. Kumar, S. Iyer, V. Taneja, Rolling out augmented reality in the field (2018).
URL https://www.bcg.com/de-de/publications/2018/rolling-out-augmented-reality-field.aspx

[4] E. Shook, M. Knickrehm, Harnessing revolution creating the future workforce (2017).
URL https://www.accenture.com/_acnmedia/pdf-40/accenture-strategy-harnessing-revolution-pov.pdf

[5] B. Guy, M. Brett, P. Mitesh, R. Vaibhaw, The coming evolution of field operations (2019).
URL https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/Operations/Our%20Insights/The%20coming%20evolution%20of%20field%20operations/The-coming-evolution-of-field-operations.ashx

[6] P. Foret, A. Kleiner, H. Mobahi, B. Neyshabur, Sharpness-aware minimization for efficiently improving generalization (2020). arXiv:2010.01412.

[7] H. M. D. Kabir, M. Abdar, S. M. J. Jalali, A. Khosravi, A. F. Atiya, S. Nahavandi, D. Srinivasan, Spinalnet: Deep neural network with gradual input (2020). arXiv:2007.03347.

[8] A. Krizhevsky, Learning multiple layers of features from tiny images, University of Toronto.

[9] Y. LeCun, C. Cortes, C. Burges, Mnist handwritten digit database, ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist 2.

[10] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, 2011.
URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[11] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 215–223.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[13] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.
URL https://www.nature.com/articles/nature14539

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE (Ed.), 29th IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Piscataway, NJ, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE (Ed.), 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 1–9. doi:10.1109/CVPR.2015.7298594.

[16] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: IEEE (Ed.), 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 2261–2269. doi:10.1109/CVPR.2017.243.

[17] J. Webster, R. T. Watson, Analyzing the past to prepare for the future: Writing a literature review, MIS Quarterly 26 (2) (2002) xiii–xxiii.
URL http://www.jstor.org/stable/4132319

[18] A. I. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, N. Houlsby, Large scale learning of general visual representations for transfer, ArXiv abs/1912.11370.

[19] H. Touvron, A. Vedaldi, M. Douze, H. Jegou, Fixing the train-test resolution discrepancy, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc, 2019, pp. 8252–8262.
URL http://papers.nips.cc/paper/9035-fixing-the-train-test-resolution-discrepancy.pdf

[20] Amazon Web Services, Inc., Aws deep learning ami (ubuntu 18.04) (2020).
URL https://aws.amazon.com/marketplace/pp/Amazon-Web-Services-AWS-Deep-Learning-AMI-Ubuntu-1/B07Y43P7X5

[21] Amazon Web Services, Inc., Amazon ec2 g4 instances (2020).
URL https://aws.amazon.com/de/ec2/instance-types/g4/

[22] G. Van Rossum, F. L. Drake, Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, 2009.

[23] F. Chollet, et al., Keras (2015).
URL https://github.com/fchollet/keras

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/

[25] NVIDEA, Cuda technology (2007).
URL https://developer.nvidia.com/cuda-toolkit

[26] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, E. Shelhamer, cudnn: Efficient primitives for deep learning, ArXiv abs/1410.0759.

[27] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556.
URL https://arxiv.org/abs/1409.1556

[28] S. Xie, R. Girshick, P. Dollar, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[29] M. Tan, Q. Le V, Efficientnet: Rethinking model scaling for convolutional neural networks, in: ICML (Ed.), 2019 International Conference on Machine Learning (ICML), 2019.
URL https://arxiv.org/pdf/1905.11946v3.pdf

[30] Z. Liu, P. Luo, S. Qiu, X. Wang, X. Tang, Deepfashion: Powering robust clothes recognition and retrieval with rich annotations, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[31] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, CoRR abs/1708.07747. arXiv:1708.07747.
URL http://arxiv.org/abs/1708.07747

[32] L. N. Darlow, E. Crowley, A. Antoniou, A. J. Storkey, Cinic-10 is not imagenet or cifar-10, ArXiv abs/1810.03505.

[33] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing, 2008.

[34] L. Bossard, M. Guillaumin, L. Van Gool, Food-101 – mining discriminative components with random forests, in: European Conference on Computer Vision, 2014.

[35] G. van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The inaturalist species classification and detection dataset, in: IEEE (Ed.), The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[36] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.

[37] G. Cohen, S. Afshar, J. Tapson, A. V. Schaik, Emnist: Extending mnist to handwritten letters, 2017 International Joint Conference on Neural Networks (IJCNN)doi:10.1109/ijcnn.2017.7966217.

[38] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, D. Ha, Deep learning for classical japanese literature, CoRR abs/1812.01718. arXiv:1812.01718.
URL http://arxiv.org/abs/1812.01718

[39] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011).

[40] S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc, 2017, pp. 3856–3866.
URL http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf

[41] M. Combalia, N. C. F. Codella, V. Rotemberg, B. Helba, V. Vilaplana, O. Reiter, C. Carrera, A. Barreiro, A. C. Halpern, S. Puig, J. Malvehy, Bcn20000: Dermoscopic lesions in the wild (2019). arXiv:1908.02288.

[42] N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, A. Halpern, Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic), in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), 2018, pp. 168–172.

[43] P. Tschandl, C. Rosendahl, H. Kittler, The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, Scientific data 5 (2018) 180161. doi:10.1038/sdata.2018.161.

[44] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359. doi:10.1109/TKDE.2009.191.
URL https://www.cse.ust.hk/~qyang/Docs/2009/tkde_transfer_learning.pdf

[45] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, R. Fergus, Intriguing properties of neural networks, CoRR abs/1312.6199.

[46] H. El-Amir, M. Hamdy, Deep Learning Pipeline: Building a Deep Learning Model with TensorFlow, 1st Edition, 2020.
URL https://doi.org/10.1007/978-1-4842-5349-6