A preliminary study on deep transfer learning applied to image classification for small datasets

M. Á. Molina[1], G. Asencio-Cortés[1], J. C. Riquelme[2], and F. Martínez-Álvarez[1]

[1]Data Science & Big Data Lab, Pablo de Olavide University, ES-41013 Seville, Spain
mamolcab@alu.upo.es, {guaasecor,fmaralv}@upo.es
[2]Department of Computer Science, University of Seville, Spain
riquelme@us.es

Abstract. A new transfer learning strategy is proposed for image classification in this work, based on an 8-layer convolutional neural network. The transfer learning process consists in a training phase of the neural network on a source dataset of images. Then, the last two layers are retrained using a different small target dataset of images. A preliminary study was conducted to train and test the transfer learning proposal on Malaria cell images for a binary classification problem. The methodology proposed has provided a 6.76% of improvement with respect to other three different strategies of training non-transfer learning models. The results achieved are quite promising and encourage to conduct further research in this field.

Keywords: Transfer learning · deep learning · classification · pattern recognition.

1 Introduction

Deep learning has become quite popular in the field of big data and, in particular, in some applications such as remote sensing [1] or time series [2,3]. Transfer learning is a discipline suitable in situations in which there is a small amount of data to be mined (target data). The adequate training of deep neural network typically requires many data and much time. Nonetheless, a vast majority of real-world problems are not characterized by such amount of data and, therefore, models are not as accurate as expected. The integration of deep learning with transfer learning is called deep transfer learning and it makes the most of both paradigms. Thus, deep learning is used to model problems within big data contexts and, afterwards, re-purposed to transfer the knowledge to models with insufficient data [4]. There is a major flaw in transfer learning, which is the lack of interpretability of its models because pretrained models are applied to the new data without any prior information or understanding of the model [5].

A new transfer learning strategy is proposed in this work, based on the application of a convolutional neural network (CNN). In particular, a 8-layer CNN is trained with the source dataset. Then, the last two layers are retrained

with a training set from the target dataset. Different training sets, as explained in Section 3, are created in order to validate the robustness of the method. In short, New methodologies have been used, such as the differential study in four experiments and the analysis of similarities between source and target subsets through dendrograms.

To assess the performance of the proposal, the malaria cell images dataset [6], available along with the work in [7], is tested. This dataset is formulated as a binary classification problem, in which cells are either parasited or uninfected. Three additional strategies are also evaluated to compare the performance in terms of accuracy. The results achieved are quite promising.

The rest of the paper is structured as follows. Section 2 overviews recent and relevant papers in the field of deep transfer learning and its application to image classification. Section 3 describes the proposed methodology and how deep transfer learning can be applied to improve classification performance in datasets with few samples. Section 4 reports the results achieved and discusses their goodness. Finally, Section 5 summarizes the conclusions drawn from this work.

2 Related works

Deep transfer learning is becoming one of the research fields in which much effort is being put into [8]. In fact, many applications can be found in the literature currently. Thus, Talo et al. [9] proposed a novel approach based on deep transfer learning to automatically classify normal and abnormal brain magnetic resonance images. Data augmentation, optimal learning rate finder or fine-tuning were the strategies used to infer the model.

A wide variety of applications in remote sensing problems are also available. In 2017, Zhao et al. [10] proposed a transfer learning model with fully pretrained deep convolution networks for land-use classification of high spatial resolution images. The authors claimed that the method accelerated the training process convergence with no loss of accuracy, as shown in the comparative analysis they report. The classification of Synthetic Aperture Radar (SAR) images through deep transfer learning was proposed in [11]. Given that labelling SAR images is quite challenging, the authors proposed to transfer learning from the electro-optical domain and used a deep neural network as classifier.

Another approach to range underwater source was recently introduced in [12]. In this case, the source domain was a set of large synthetic historical environmental data, which was transferred to the source domain (a deep-sea area). which migrates the predictive ability obtained from synthetic environment (source domain) into an experimental sea area (target domain). Reported results outperformed those of CNNs.

Another deep neural network model was proposed in [13] for plant classification. In particular, four different deep transfer learning models were applied to four public datasets, improving the performance of other methods.

Li et al. [14] linked emotions during conversations, by means of acoustic signals to behaviors through deep transfer learning as well, in an effort of explicitly quantifying the existing relationship. Hybrid architectures of both convolutional and recurrent neural networks to achieve the goal were explored.

3 Methodology

3.1 Image preprocessing

The first step in the image preprocessing is to rescale all the images to the same dimensions, because it is necessary to have the same number of input pixels passed to the neural network. For the image rescaling process, the function *resize*() of the OpenCV library [15] was applied using a bilinear interpolation algorithm. The second step of the preprocessing is to encode the image labels, in order to have as many outputs of the neural network as image labels. Thus, a predicted probability is returned for each label.

3.2 Creation of source and target subsets

Disjoint source and target subsets of images were extracted from the original set of images. The source subset is the dataset from which the initial model was trained. The target subset is the dataset used both to update such model (transfer process) and to test the updated model.

To extract the source and target subsets, it has been tried that such datasets were as different as possible. Additionally, the source subset is larger than the target one. The idea underlying this strategy is to check if the transfer learning is effective when the source and target subsets contain dissimilar images and the target set is a smaller one.

For this purpose, each image was firstly encoded using the Google Inception V3 deep neural network trained on ImageNet [16], which is available online. As a result of this process, for each image a vector of 2,048 real values were obtained from the weights of the last layer of such network. This process has been conducted using the Orange's Image Embedding node from the Orange-ImageAnalytics package (version 0.4.1) [17].

A table formed by the 2,048 real values of the image along with its label were generated for all the original images. Next, a hierarchical clustering was applied to each different label using such table as input. A dendrogram was generated for each label (U and P) after applying the hierarchical clustering.

Finally, the first two nodes of the first level of each dendrogram were selected. As example, for binary classification, two dendrograms were generated and four nodes were selected from them (U_A,U_B,P_A,P_B,). The node which contains the largest number of images for each label was added to the source subset. Similarly, the node which contains the smallest number of images for each label was added to the target subset.

It can be concluded that source and target subsets of images were generated in such a way that they contain dissimilar images and the source subset is larger than the target subset.

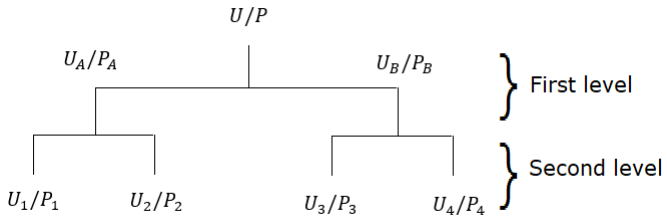

Fig. 1: Dendrogram scheme for each label U and P

3.3 Deep neural network architecture

The next step consists in training a convolutional neural network and testing it using the subsets described in the previous section. The way these subsets are divided to validate the methodology will be explained in the next subsection.

The deep convolutional neural network is composed of three layers of 2D-based convolution using a kernel of size $3x3$ and performing 32, 32 and 64 filters, respectively. Moreover, two layers of MaxPooling were added to the network, with a $2x2$ size for the two of them. Finally, two dense flatten and fully-connected layers were added as the last layers of the network. The neural network proposed has $848,226$ parameters to be adjusted. The detailed network used is shown in Table 1. To implement the neural network architecture, Keras 2.2.4 over TensorFlow 1.14 was used [18].

Table 1: Deep neural network architecture used for transfer learning.

Layer (type)	Output Shape	Params	Updateable
Conv2D	(None, 48, 48, 32)	896	No
Conv2D	(None, 46, 46, 32)	9,248	No
MaxPooling2D	(None, 23, 23, 32)	0	No
Conv2D	(None, 21, 21, 64)	18,496	No
MaxPooling2D	(None, 10, 10, 64)	0	No
Flatten	(None, 6400)	0	No
Dense	(None, 128)	819,328	Yes
Dense	(None, 2)	258	Yes

3.4 Four validation schemes

The target subset is randomly divided into two parts: training (70%) and test (30%). Freezing the same test part (30%) of the target subset, for a fair comparison, four different validation schemes have been proposed:

1. The model is generated using the training part (70%) of the target subset, and it is tested by evaluating its predictions over the test part (30%) of the target subset.
2. The model is generated using the whole source subset, and it is tested by evaluating its predictions over the test part (30%) of the target subset.
3. The model is generated using the whole source subset along with the training part (70%) of the target subset, and it is tested by evaluating its predictions over the test part (30%) of the target subset.
4. In this scheme the transfer learning procedure is carried out. The steps are the following:
   - The model is trained using the whole source subset.
   - Then, such model is updated using the training part (70%) of the target subset. This updating process only optimizes the weights within the two last layers of the neural network, maintaining the rest of its layers without changes.
   - The updated model is tested by evaluating its predictions over the test part (30%) of the target subset.

For each scheme, the methodology has been tested up to 10 times, having each execution a different random distribution of samples.

3.5 Source-Target similarity analysis

In order to check how the similarity between source and target subset images affects to the transfer learning effectiveness, the creation of these subsets was extended including the first two levels (instead of only the first level) of the dendrograms extracted from the hierarchical clustering process described in Section 3.2.

Extracting the images from each node of the second level of dendrograms, more combinations are available. Specifically, since dendrograms used are binary trees, there were 4 nodes taken from the second level of each dendrogram (U_1,U_2,U_3,U_4,P_1,P_2,P_3,P_4). As example, for the image binary classification, $4 + 4 = 8$ nodes were extracted, as it can be seen in Figure 1.

To carry out the source-target similarity analysis, all combinations among extracted nodes are proved and the effectiveness achieved by transfer learning was analyzed in Section 4.

3.6 Class imbalance analysis

Finally, an analysis has been conducted to prove how the effectiveness of the proposed transfer learning methodology varies depending on the ratio between image classes (labels) in source and target subsets.

For such purpose, both source and target subsets derived from dendrograms were ranked according to the ratio between the minority and majority classes. Such ratio was expressed by a percentage and it was ranged from 50% (the number of images labeled with the minority class is half of the number of images labeled with the majority class) to 100% (same number of images for each class).

4 Experimentation and results

4.1 Image dataset

The set of images used to perform the methodology explained in previous section have been taken from Kaggle. Exactly, the chosen dataset is a set of images of cells which can be infected by the Malaria parasite or not. The challenge of these mages is to provide a complete data set of images in order to reduce the burden from microscopists in resource-constrained regions and improve diagnostic accuracy. The original source of images can be consulted in [6].

A set of 5000 images were randomly selected from the 13780 of the Kaggle challenge for each label; in total 10000 images were used to train and test our methodology. In order to work with similar type of images, all of them have been rescaled. This rescaling was to 50x50 pixeles.

4.2 Evaluation metrics

In order to quantify the effectiveness of the methodology proposed, Binary Cross Entropy and Accuracy were computed.

The Binary Cross Entropy is a loss function that is applicable for binary classification. This is the most common loss function when working with this type of data sets. In this case, the output layer has one node. The typical activation function is a sigmoid and the formula is the following: $CE = -(y_i \cdot \log \hat{y_i}) + (1 - y_i) log(1 - \hat{y_i})$

The metric used for Accuracy is the Binary Accuracy, which calculates the mean accuracy rate across all predictions for binary classification problems. The formula is: $Acc = \frac{1}{n} \sum_{i=1}^{n} y_i = \hat{y}_i$

4.3 Experimental settings

The experimental settings established to execute the experiments were:

Batch size: with a value of 128, it defines the number of samples that will be propagated through the network.

Epochs: one epoch is when an entire data set is passed forward and backward through the neural network only once. The number of epochs used was 5.

Optimizer: The optimizer used is the RMSprop. This optimizer recommends to leave the parameters at their default values, except the learning rate, which, in this case, has been set to $e_r = 1 \cdot 10^{-4}$.

4.4 Results and discussion

The results obtained applying the methodology through the four validation schemes described in the previous section are shown in Table 2.

In Table 2, the four proposal schemes can be observed. In each sub table, $SelectedImages$ explains how the training and test subsets are built. $Executios$ indicated the number of executions of each scheme. $Loss$ and $Accuracy$ are

the metrics used to evaluate the results of each scheme and which have been defined previously. *Average* and *SD* are the average and standard deviation of the accuracy of the ten executions

Table 2: Effectiveness achieved for each validation scheme with no transfer learning (Schemes 1, 2 and 3) and with our transfer learning proposal (Scheme 4).

Scheme 1					
Selected images	**Execution**	**Loss**	**Accuracy**	**Average**	**SD**
Train set: 70% Target Domain / **Test set: 30% Target Domain**	**1**	0.5650	71.34%	**69.06%**	0.04
	2	0.5826	73.04%		
	3	0.5617	75.21%		
	4	0.5782	71.73%		
	5	0.6076	65.92%		
	6	0.6135	64.45%		
	7	0.6046	66.46%		
	8	0.5722	73.66%		
	9	0.6311	63.21%		
	10	0.6083	65.61%		

Scheme 2					
Selected images	**Execution**	**Loss**	**Accuracy**	**Average**	**SD**
Train set: Source Domain / **Test set: 30% Target Domain**	**1**	0.5994	65.61%	**60.58%**	0.04
	2	0.5997	69.48%		
	3	0.6132	61.74%		
	4	0.6322	60.73%		
	5	0.6378	60.81%		
	6	0.6733	56.24%		
	7	0.6379	59.49%		
	8	0.6316	59.18%		
	9	0.6963	55.92%		
	10	0.6702	56.55%		

Scheme 3					
Selected images	**Execution**	**Loss**	**Accuracy**	**Average**	**SD**
Train set: Source Domain and 70% Target Domain / **Test set: 30% Target Domain**	**1**	0.6736	58.33%	**63.83%**	0.10
	2	0.5622	72.19%		
	3	0.5371	72.73%		
	4	0.7207	53.29%		
	5	0.8145	53.29%		
	6	0.5489	74.13%		
	7	0.5259	73.04%		
	8	0.8140	53.29%		
	9	0.6597	53.29%		
	10	0.4900	74.75%		

Scheme 4					
Selected images	**Execution**	**Loss**	**Accuracy**	**Average**	**SD**
Train set: Source Domain and Retrain with the 70% Target Domain / **Test set: 30% Target Domain**	**1**	0.5065	76.92%	**75.82%**	0.02
	2	0.5153	75.45%		
	3	0.4701	78.23%		
	4	0.5157	76.14%		
	5	0.5065	78.23%		
	6	0.5184	77.38%		
	7	0.5818	69.95%		
	8	0.5290	75.76%		
	9	0.5345	71.88%		
	10	0.4908	78.23%		

As it can be seen in Table 2, the fourth scheme, which is the transfer learning one, is the scheme with the best results of all of them obtaining a better average accuracy, with an improvement of 6.76%, which is a very remarkable performance.

Another important feature is the robustness that the transfer learning technique brings to the results. The standard deviation of the transfer learning (Scheme 4) is smaller than the other schemes. Such result demonstrates that, with this technique, the learning is more robust and the dependence of the random train and test subsets is lower.

For the Source-Target similarity analysis, the four clusters obtained by the second level of the dendrogram for each class (image label) are used in order to make different combinations for constructing the source and the target subsets. The number of images of the second level for the class Uninfected are U1: 1883, U2: 2213, U3: 408 and U4: 496 images ($total = (U1 + U2) + (U3 + U4) = (1883 + 2213) + (408 + 496) = 4096 + 904 = 5000$ images). The number of images of the second level for the class Parasitized are P1: 377, P2: 2041, P3: 936 and P4: 1646 images ($total = (P1 + P2) + (P3 + P4) = (377 + 2041) + (936 + 1646) = 2418 + 2582 = 5000$ images).

With these clusters, the schemes 1 and 4 have been carried out again. The improvement for each group is shown in Table 3, where, the clusters obtained for the uninfected cells of Malaria set have been named as Target U and those obtained for the parasitized ones as Target P. The number of images obtained

from the sum of them from the two previous clusters is the Target Dim. The sum of the rest of clusters is the Source Dim. Scheme1 Acc and Scheme4 Acc show the accuracy obtained from each scheme. The column named Improvement shows the percentage of improvement using transfer learning techniques. Finally, Cosine Distance indicates the cosine distance between the source and target subsets, where values close to 0 indicate very similar data sets. The formula has the following expression:

$$Cos\theta = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\|\boldsymbol{a}\| \, \|\boldsymbol{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

To facilitate the understanding of the graphs, the values indicated in the table will be 1-Cosine Distance.

Figures 2 and 3 show the improvements caused by the transfer learning technique depending on the distance between source and target subsets (Figure 2) and the ratio between classes in source and target subset (Figure 3). In Figure 2, the relationship between the distance of the two subsets against the improvement using transfer learning can be observed. The distances obtained from the different combinations give a narrow range of values due to the own characteristics of the set of images. This causes that the improvements produced by transfer learning techniques are not noticeable. However, if the linear regression line of the curve obtained is drawn, a worsening of the results is observed as the distance between the two subsets is greater. In Figure 3, the X axis shows the ratio of the minority class in each subset, and the Y axis the improvement between scheme 1 and scheme 4. As the ratio of the minority class grows, an effectiveness improvement of the scheme 4 is observed (particularly with higher ratios of minority class in the source subset). Only in the two last cases this effect is not appreciable. These two cases are, precisely, those related with the two subsets with bigger distances between them. Other aspects to be studied in future works are the influences of the number of samples in each cluster in order to get more information to learn general behaviour. It is possible that some limitations in the results can be associated with these aspects besides the architecture of the neural network. Also, the linear regression line is drawn to show the trend of the transfer learning improvement.

5 Conclusions

In this paper the benefits of transfer learning have been empirically demonstrated using a dataset of images of cells parasitized, or uninfected, by the Malaria disease. First, comparing the fourth validation schemes proposed, the use of transfer learning techniques has provided a 6.76% of improvement with respect to different ways to train non-transfer learning models. Also, transfer learning has provided more robustness, reflected in the smaller standard deviations obtained, bringing more general knowledge of the treated data sets. According to the analysis of improvements, similarities of images and class imbalance ratios, no clear improvements have been observed. However, some relationship has been

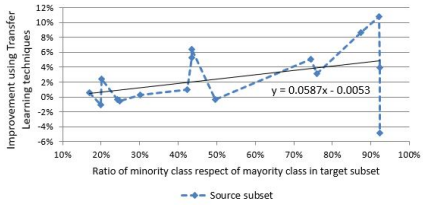

Fig. 2: Relationship between source-target subset distances and the transfer learning accuracy improvement.

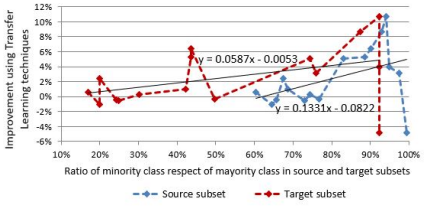

Fig. 3: Relationship between class imbalance ratio and the transfer learning accuracy improvement.

Table 3: Image classification accuracy achieved by transfer learning (Scheme 4).

Target U	Target P	Target Dim.	Source Dim.	1-Cosine distance	Scheme1 Acc.	Scheme4 Acc.	Improvement
U2	**P2**	4254	5746	0.1742	58.21%	68.96%	**10.75%**
U1	**P4**	3529	6471	0.1774	69.45%	78.13%	**8.68%**
U4	**P3**	1432	8568	0.2017	67.93%	74.30%	**6.37%**
U3	**P3**	1344	8656	0.2034	67.33%	72.60%	**5.27%**
U2	**P4**	3859	6141	0.1809	60.89%	65.98%	**5.09%**
U1	**P2**	3924	6076	0.1885	69.42%	73.33%	**3.91%**
U4	**P1**	873	9127	0.2118	88.24%	91.41%	**3.17%**
U1	**P1**	2260	7740	0.1980	86.99%	89.40%	**2.41%**
U2	**P3**	3149	6851	0.1774	80.31%	81.29%	**0.98%**
U2	**P1**	2590	7410	0.1749	92.68%	93.26%	**0.58%**
U4	**P4**	2142	7858	0.1909	74.66%	74.90%	**0.24%**
U1	**P3**	2819	7181	0.1853	89.34%	89.04%	**-0.30%**
U4	**P2**	2537	7463	0.1739	83.03%	82.64%	**-0.39%**
U3	**P4**	2054	7946	0.1859	80.75%	80.20%	**-0.55%**
U3	**P2**	2449	7551	0.1740	87.62%	86.57%	**-1.05%**
U3	**P1**	785	9215	0.2132	91.99%	87.12%	**-4.87%**

found between the class ratio and the improvement of transfer learning, in such a way that more balanced datasets produce higher improvement using transfer learning. These works are a starting point to continue exploring the benefits and limitations of transfer learning, like the number of samples, distances and neural network structure. In future works, the results will be tested previously applied strategies with those being proposed here.

Acknowledgements

The authors would like to thank the Spanish Ministry of Economy and Competitiveness for the support under the project TIN2017-88209-C2-1-R.

References

1. D. T. Bui, N.-D. Hoang, F. Martínez-Álvarez, P.-T. T. Ngo, P. V. Hoa, T. D. Pham, P. Samui, and R. Costache. A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. *Science of The Total Environment*, 701:134413, 2020.

2. J. F. Torres, A. Galicia, A. Troncoso, and F Martínez-Álvarez. A scalable approach based on deep learning for big data time series forecasting. *Integrated Computer-Aided Engineering*, 25(4):335–348, 2018.
3. J. F. Torres, A. Troncoso, I. Koprinska, Z. Wang, and F. Martínez-Álvarez. Big data solar power forecasting based on deep learning and multiple data sources. *Expert Systems*, 36(4):e12394, 2019.
4. Z. Deng, J. Lu ad D. Wu, K. Choi, S. Sun, and Y. Nojima. New advances in deep-transfer learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(5):357–359, 2019.
5. D. Kim, W. Lim, M. Hong, and H. Kim. The structure of deep neural network for interpretable transfer learning. In *Proceedings of the IEEE International Conference on Big Data and Smart Computing*, pages 1–4, 2019.
6. R. Tatman. R vs. Python: The Kitchen Gadget Test, Version 1. `https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria`, 2017. Online – last accessed on January 29^{th}, 2020.
7. S. Rajaraman, S. K. Antani, M. Poostchi, K. Silamut, M. A. Hossain, R. J. Maude, S. Jaeger, and G. R. Thoma. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ*, 6:e4568, 2018.
8. C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A survey on deep transfer learning. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 270–279, 2018.
9. M. Talo, U. B. Baloglu, Ö. Yıldırım, and U. R. Acharya. Application of deep transfer learning for automated brain abnormality classification using MR images. *Cognitive Systems Research*, 54:176–188, 2019.
10. B. Zhao, B. Huang, and Y. Zhong. Transfer learning with fully pretrained deep convolution networks for land-use classification. *IEEE Geoscience and Remote Sensing Letters*, 14(9):1436–1440, 2017.
11. M. Rostami, S. Kolouri, E. Eaton, and K. Kim. Deep transfer learning for few-shot SAR image classification. *Remote Sensing*, 11(11):1374, 2019.
12. W. Wang, H. Ni, L. Su, T. Hu, Q. Ren, P. Gerstoft, and L. Ma. Deep transfer learning for source ranging: Deep-sea experiment results. *The Journal of the Acoustical Society of America*, 146:EL317, 2019.
13. A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan. Analysis of transfer learning for deep neural network based plant classification models. *Computers and Electronics in Agriculture*, 158:20–29, 2019.
14. H. Li, B. Baucom, and P. Georgiou. Linking emotions to behaviors through deep transfer learning. *Computers and Electronics in Agriculture*, 6:e246, 2020.
15. G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
16. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.
17. J. Demšar et al. Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research*, 14:2349–2353, 2013.
18. F. Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.