Hydranet: Data Augmentation for Regression Neural Networks

Florian Dubost\textsuperscript{1,2,3}, Gerda Bortsova\textsuperscript{1,2,3}, Hieab Adams\textsuperscript{3,4}, M. Arfan Ikram\textsuperscript{3,4,5}, Wiro Niessen\textsuperscript{1,2,3,6}, Meike Vernooij\textsuperscript{3,4}, and Marleen de Bruijne\textsuperscript{1,2,3,7}

\textsuperscript{1} Biomedical Imaging Group Rotterdam, Erasmus MC, Rotterdam, The Netherlands
\textsuperscript{2} Department of Medical Informatics, Erasmus MC, Rotterdam, The Netherlands
\textsuperscript{3} Department of Radiology and Nuclear Medicine, Erasmus MC, Rotterdam, The Netherlands
\textsuperscript{4} Department of Epidemiology, Erasmus MC, Rotterdam, The Netherlands
\textsuperscript{5} Department of Neurology, Erasmus MC, Rotterdam, The Netherlands
\textsuperscript{6} Imaging Physics, Faculty of Applied Sciences, TU Delft, The Netherlands
\textsuperscript{7} Department of Computer Science, University of Copenhagen, Denmark

Abstract. Deep learning techniques are often criticised to be heavily dependent on a large quantity of labeled data. This problem is even more challenging in medical image analysis where the annotator expertise is often scarce. We propose a novel data-augmentation method to regularize neural network regressors that learn from a single global label per image. The principle of the method is to create new samples by recombining existing ones. We demonstrate the performance of our algorithm on two tasks: estimation of the number of enlarged perivascular spaces in the basal ganglia, and estimation of white matter hyperintensities volume. We show that the proposed method improves the performance over more basic data augmentation. The proposed method reached an intraclass correlation coefficient between ground truth and network predictions of 0.73 on the first task and 0.84 on the second task, only using between 25 and 30 scans with a single global label per scan for training. With the same number of training scans, more conventional data augmentation methods could only reach intraclass correlation coefficients of 0.68 on the first task, and 0.79 on the second task.

1 Introduction

Deep learning techniques are getting increasingly popular for image analysis but are often dependent on a large quantity of labeled data. In case of medical images, this problem is even stronger as data acquisition is administratively and technically more complex, as data sharing is more restricted, and as the annotator expertise is scarce.

To address biomarker (e.g. number or volume of lesions) quantification, many methods propose to optimize first a segmentation problem and then derive the target quantity with simpler methods. These approaches require expensive voxel-wise annotations. In this work, we circumvent the segmentation problem by optimizing our method to directly regress the target quantity \cite{234}. Therefore we
need only a single label per image instead of voxel-wise annotations. Our main contribution is that we push this limit even further by proposing a data augmentation method to reduce the number of training images required to optimize the regressors. The proposed method takes advantage of the fact that the original global labels represent a countable quantity. Its principle is to combine real training samples to construct many more virtual training samples. During training, our model takes as input random sets of images and is optimized to predict a single label for each of these sets that denotes the sum of the labels of all images of the set. This is motivated by the idea that adding a large quantity of virtual samples with weaker labels may reduce the over-fitting to training samples and improve the generalization to unseen data. During training, in order for the network to process several images simultaneously, we replicate the full network as many times as there are images in the set. The parameters of each replicated network are then shared and their predictions are summed. Once the model is trained, the replicated networks can be removed, and the original architecture can be used for prediction.

We evaluate our approach on two different tasks: quantification of enlarged perivascular spaces (PVS) in the basal ganglia, and of white matter hyperintensities (WMH) in the brain. For the PVS, we evaluate our method on 1977 visually scored MRI scans from a single scanner. For the WMH, we used the training dataset from the WMH challenge [5]. This dataset contains 60 MRI scans from three different centers.

1.1 Related Work

Data augmentation and other regularizers. Data augmentation can act as a regularizer and improve the generalization performance of neural networks. This has been demonstrated by Simard et al. [12] with transformation applied on MNIST [16]. The authors define a manifold representation for training samples, and propose to compute transformations corresponding to vectors in the tangent plane. Simple data-augmentations such as rotation, translation and flipping are a special case of such transformations, and are often used to reduce over-fitting of deep learning algorithms, as they are simple to implement and can be very effective [17]. The authors of Unet [14], probably the most used segmentation network in medical imaging, stress that random elastic deformations of training samples significantly improved the performance of their model. In case of medical images, these transformations can mimic the possible movements of biological tissues. Generative adversarial networks (GAN) [18] can also be used to generate samples for the network during training, and hence reduce the over-fitting [19]. Perez et al. [17] compare different data augmentation methods and propose to learn augmentations that best improve the performance of the model. To learn augmentations, the authors include in their network an augmentation branch taking as input other images of the same class as the input image. This branch is optimized to generate images similar to the input image. The authors explore different similarity metrics describing content or style.
Recently, data augmentation methods using combinations of training samples have been published. Zhang et al. [1] proposed to construct virtual training samples by computing a linear combination of pairs of real training samples. The corresponding one-hot labels are summed with the same coefficients. The authors evaluated their method on classification datasets from computer vision [22] and on a speech dataset, and demonstrate that their method improves the generalization of state-of-the-art neural networks, such as DenseNet-BC-190 [23] or ResNet-101 [24]. In some experiments, the authors achieve better results using both their method and Dropout [9] instead of Dropout alone. Simultaneously, Inoue et al. [11] reached similar conclusions by averaging pairs of training samples, after standard data augmentations including random extractions and flipping. Tokozume et al. [21] also used similar methods to improve performance for sound classification, and validated their method on three datasets. However, in case of grayscale volumetric inputs, summing image intensity values could overlay the target structures, confuse discriminative shapes, and thus harm the performance of the network. For instance one of the most discriminative feature of PVS is their tubular shape, averaging several close PVS could create a new shape, which would probably not and should not be detected as PVS by the network. With our method, training samples can be combined without overlaying the intensity values. The other difference with the above-mentioned approaches is that our method is also not designed for classification, but for regression of global labels, such as volume or count in an image. With the proposed combination of samples, our method computes plausible augmentation.

Note that for these methods, the amount of regularization introduced can be modulated by at least one parameter, for instance the degree of rotation applied to the input image, or the percentage of neurons dropped in Dropout [9]. With most methods, the more regularization is applied, the longer the optimization takes until convergence. Introducing too much regularization can make the optimization unreasonably long, and force stopping before convergence of the validation loss, which leads to a sub-optimal performance. Choosing the right amount of regularization is therefore a key factor. In the proposed method, the regularization effect can be controlled by varying the average number of samples used to create combinations.

Biomarker extraction with regression networks. Recently, methods employing neural network regressors trained with global labels have been proposed for biomarker extraction [2,3,4]. Cole et al. [2] predict brain age from 3D images of grey matter density computed from MRI scans. Dubost et al. [3] predict the number of enlarged perivascular spaces in a 3D region of interest in the brain extracted from a MRI scan. And Espinosa et al. [4] predict Agatston scores to quantify coronary artery calcifications from 3D non-contrast non-ECG gated chest CT scans. These networks needed hundreds training samples to be trained accurately. With our method, we can achieve the level of interrater agreement only using 25 training scans, which could for instance only require one to two hours of labeling from an expert rater.
2 Methods

The principle of our data augmentation method is to create many new (and weaker) training samples by combining existing ones (see Figure 1). In the remainder, the original samples are called real samples, and the newly created samples are called virtual samples.

2.1 Proposed Data Augmentation.

During training, the model is not optimized on single real samples $I$ with label $y$, but on sets $S$ of $n$ random samples $I_1, I_2, ..., I_n$ with label $y_s = \sum_{i=1}^{n} y_i$, with $y_i$ the label of sample $I_i$. These sets $S$ with labels $y_s$ are the virtual samples. Consequently, the loss function $L$ is computed directly on these virtual samples $S$ and not anymore the individual real samples $I_i$. This approach is designed for labels describing a quantitative element in the samples, such as volume or count in an image.

To create the sets $S$, the samples $I_i$ are drawn without replacement from the training set at each epoch. To create more combinations of samples, and to allow the model to use the real samples for its optimization, the size of the sets $S$ can randomly vary in $\{1, n\}$ during training.

If the training set contains $m$ samples, with our method, we can create $\sum_{i=1}^{n} \binom{m}{i}$ possible different combinations, because (1) the order of the samples $I_i$ in $S$ has
no effect on the optimization, (2) the samples are drawn without replacement within an epoch, (3) the size of the set $S$ can vary.

**Difference between the proposed method and mini-batch stochastic gradient descent (SGD).** In mini-batch SGD, the model is also optimized on sets of random samples, but the loss function $L$ is computed individually for each sample of the batch, and then summed (averaged). For the proposed method, the predictions are first summed, and the loss function is then computed a single time. For non-linear loss functions, this is not equivalent: $\sum_{i=1}^{n} L(\hat{y}_i, y_i) \neq L(\sum_{i=1}^{n} \hat{y}_i, \sum_{i=1}^{n} y_i)$, with $\hat{y}_i$ the model’s prediction for sample $I_i$.

### 2.2 Implementation.

There are at least two possible implementations of the proposed method. The first implementation could consist of modifying the computation of the loss function across samples in a mini-batch, and provide mini-batches of random size. Alternatively the models architecture could be adapted to receive the set of images. Depending on ones deep learning library, one of these implementation might be easier to implement. We opted for the second approach, and present it below in more details. We optimize a regression neural network with an 3D image for input, and global label representing a volume or count for output.

**Base Regressor.** In Figure 2 left, we detail the architecture of our regression neural network. This architecture is similar to that of a 3D, shallow Resnet [10], where skip connections concatenate features maps instead of summing them. It is both simple (196 418 parameters) and flexible to allow fast prototyping. There is no activation function after the last layer. The output $\hat{y}$ can therefore span $\mathbb{R}$ and the network is optimized with the mean squared error (MSE). We call this regression network $f$, such that $f(x) = \hat{y}$, with $x$ the input image.

**Hydranet.** To process several images simultaneously, we replicate $n$ times the regressor $f$ during training (Figure 2 right). This results in $n$ different branches (or "heads", hence the name "Hydranet") $f_1, f_2, ..., f_n$ which can receive the corresponding images $I_1, I_2, ..., I_n$. The weights of each head $f_i$ are shared such that $f_i = f$. A new network $g$ is constructed by summing the predictions of each individual branch $f_i$:

$$g(S) = g(I_1, I_2, ..., I_n) = \sum_{i=1}^{n} f_i(I_i) = \sum_{i=1}^{n} f(I_i) = \sum_{i=1}^{n} \hat{y}_i. \quad (1)$$

In Section 2.1, we explained that it is desirable that the size of the sets $S$ can randomly vary in $\{1, n\}$ during training. To allow this feature, each element of $S$ has a chance $p$ to be a black image $B$ of zero intensities only (Figure 1 right column). With $f(B) = 0$, the following situation becomes possible:

$$g(S) = f(I_j) + \sum_{i=1,i\neq j}^{n} f_i(B) = f(I_j) + (n-1) f(B) = f(I_j). \quad (2)$$
Fig. 2: Architectures. On the left, architecture of the base regressor $f$. 'Conv' stands for 3D convolutions, followed by the number of feature maps, and the kernel size. After each convolution, there is a ReLU activation. The round arrows are skip connections with concatenated feature maps. GAP stands for Global Average Pooling layer, and FC for Fully Connected layer. On the right, example of our data augmentation method with $n = 4$ replications. Each replication $f_i$ is a copy of the base regressor $f$ on the left. Once the training is done, all $f_i$ but one can be removed, and the evaluation is performed using the original architecture.

Finally, for this implementation, the batch size $b$ has to be a multiple of the number of heads $n$. In our experiments, we chose $b = n$.

3 Experiments

In this section, we provide details about the datasets and experimental settings, and present results on both tasks: estimation of number of PVS in the basal ganglia, and estimation of WMH volume. We compare the optimization and performance of our method to that of more standard methods (base regressor $f$ and dropout) and for different sizes of training set.

3.1 Data

The PVS dataset contains 2017 PD-weighted scans, from 2017 subjects, acquired from a 1.5T GE scanner. The original voxel size is $0.49 \times 0.49 \times 0.8$mm$^3$ and the images are $512 \times 512 \times 192$ voxels wide. The scans were visually scored by an expert rater who counted the PVS in the basal ganglia in a single slice: the one showing the anterior commissure. The WMH dataset is the training set of the MICCAI2017’s WMH challenge. It contains 2D multi-slice FLAIR-w and 3D T1-w scans from 60 participants from 3 centers and 3 vendors: 20 scans from Amsterdam (GE), 20 from Utrecht (Philips) and 20 from Singapore (Siemens). For simplicity, we only used the FLAIR-w scans. Although the ground truths of
Fig. 3: Examples of preprocessed MRI scans from the PVS and WMH datasets. Middle slice of 3D images. Lesions are circled in red. WMH in the centrum semiovale in FLAIR-w scans from Singapore (left) and Utrecht (middle). PVS in the basal ganglia in a PD-w scan (right).

the challenge are pixel-wise, we only used the number of WMH voxels as ground truth during our training.

3.2 Experimental Settings

Preprocessing. For the regression of PVS in the basal ganglia, we start by creating a mask of the basal ganglia from the 3D PD scans by using the subcortical segmentation algorithm from FreeSurfer [6]. Both masks and image are then registered to MNI space, using Elastix [8] with the mutual information as similarity measure and default settings. Before applying the mask, we smooth its border with a gaussian kernel (standard deviation of 2 voxels) and crop the images around the center of mass of the segmented basal ganglia. After cropping, the images are $168 \times 128 \times 84$ voxels-wide. For the WMH dataset, we only crop each image around its center of mass, weighted by the voxel intensities. The size of the cropped image is the same for all 3 centers: $100 \times 130 \times 18$ voxels-wide. For both tasks the intensities are then rescaled between 0 and 1. See Figure 3 for samples of preprocessed scans.

Training of the networks. For all methods, images are padded with zeros for the convolutions. The weights of the networks are initialized from a Gaussian distribution of 0.0 mean and 0.05 standard deviation. During training, the images are randomly augmented on-the-fly with standard methods. The possible augmentations are flipping in $x, y$ or $z$, 3D rotation from -0.2 to 0.2 radians and random translations in $x, y$ or $z$ from -2 to 2 voxels. This standard data augmentation is exactly the same for the proposed method and the base regressor $f$. Adadelta [7] is used as optimizer, with Keras' default parameters. For a fair comparison, both the proposed method and the base regressor $f$ are trained with the same batch-size $b = 4$. The architecture of the network for the proposed method has then four branches ($n = b = 4$). During an epoch, the proposed method gets as input $m/n$ different combinations of $n$ training samples, were $m$ is the total number of training images. During the same epoch, the base regressor $f$ simply gets the $m$ images separately (in batches of size $b = 4$). For the proposed method, the
probability $p$ to draw black image (Sec. 2) was set to 0.1. In some experiments we also included a dropout layer [9] after each convolution and after the global pooling layer. If not mentioned, no dropout was included. The code is written in Keras with Tensorflow as backend, and the experiments were run on a Nvidia GeForce GTX 1070 GPU.

**Training/testing splits.** For all the experiments, the datasets are split into training and testing sets. In the training set, a small subset is used to evaluate the overfitting: the validation set. For the PVS dataset, we experiment with varying size of training set, between 12 and 25 scans. The validation set always contains the same 5 scans. All methods were evaluated on the same separated test set of 1977 scans. For the WMH dataset, we split the set into 30 training scans and 30 testing scans. Six scan from the training set are used as validation scans. In both cases, the dataset is randomly (uniform distribution) split into training and testing sets. For the PVS dataset, once the dataset has been split into 30 training scans and 1977 testing scan, we manually sample scans to keep a pseudo-uniform distribution of the lesion count when decreasing the number of training scans.

**Evaluation Metrics.** To compare the automated predictions to visual scoring (for PVS) or volumes (for WMH), we use two evaluation metrics: the mean squared error (MSE), and the intraclass correlation coefficient (ICC).

### 3.3 Results

**Enlarged Perivascular Spaces (PVS).** Figure 4 compares the proposed method to the base regressor $f$ on the PVS datasets, and for an increasing number of training samples. Their performance is also compared to the average interrater agreement computed for the same problem and reported in [3]. With less than 30 training samples, only the proposed method reaches a performance similar

![Fig. 4: Comparison between the proposed method with $n = 4$ and the base regressor on the PVS dataset. MSE on the left, and ICC is displayed on the right. The proposed method significantly outperforms the base regressor in ICC (Williams’ test p-value $< 0.001$), when averaging the predictions of the methods across the four points of their learning curve. Note that all methods were optimized for MSE.](image-url)
to the interrater agreement, and significantly outperforms the base regressor in ICC (Williams’ test p-value < 0.001) when averaging the predictions of the methods across the four points of their learning curve. We also notice that for all training set sizes, the proposed method always reaches a better MSE than the conventional methods. Note that all methods were optimized for MSE (not ICC).

White Matter Hyperintensities (WMH). We conducted three series of experiments, and trained in total five neural networks (Table 1). When using small training sets, the proposed method outperforms the base network $f$, when optimized either for MSE or for mean absolute error. With larger training sets, the difference of performance reduces, and the base regressor performs slightly better on the ICC.

Table 1: Results on the WMH dataset. We conducted three series of experiments with different training set sizes and loss functions. In the two first rows, we repeated the experiments with three random initializations of the weights (on the same split), and report mean and standard deviation. MAE is an acronym for mean absolute error.

| Method            | Training scans | Testing scans | Loss | Performance (ICC) |
|-------------------|----------------|---------------|------|-------------------|
| Base Network $f$  | 30             | 30            | MSE  | 0.79 ± 0.12       |
| Proposed Method   | 30             | 30            | MSE  | **0.84 ± 0.02**   |
| Base Network $f$  | 30             | 30            | MAE  | 0.78              |
| Proposed Method   | 30             | 30            | MAE  | **0.87**          |
| Base Network $f$  | 40             | 20            | MSE  | 0.89              |
| Proposed Method   | 40             | 20            | MSE  | 0.86              |

4 Discussion

With the proposed data augmentation method, we could reach the inter-rater agreement performance on PVS quantification reported by Dubost et al. [3] with only 25 training scans, and without pretraining. The proposed method outperforms the base regressor $f$ on both datasets.

Dubost et al [3] also regressed the number of PVS in the basal ganglia with a neural network. We achieve a similar result (0.73 ICC) while training on 25 scans instead of 1000. Note that the authors reported interrater agreements with ICCs of 0.70 and 0.68, and intrarater ICC of 0.80 on their dataset. In our case, even the base regressor $f$ reaches a good performance with only 40 training samples.
(ICC of 0.68). For a similar performance, Dubost et al. [3] needed more than 200 training samples. We believe that this is due to the use of global pooling in our network architecture (instead of fully connected layers in [3]), which might have a regularization effect.

Training on such small datasets could drastically ease the use of deep learning regression methods in medical image analysis. For instance for PVS, creating global labels, for less than 30 scans, would generally only take between one and two hours for an expert human rater. This could make it affordable for different centers to create their in-house training sets.

With our data-augmentation strategy, the models are sometimes optimized with larger labels than those in real images. There can be controversy on whether only realistic transformations should be used during training [20]. Dubost et al. [3] showed on PVS quantification that although unrealistic transformations could make the training longer, they did not harm the performance of the model.

We noticed that the more variability in the combinations of training samples, the longer the convergence. This variability can for instance be partly estimated using the number of possible combinations as described in section 2.1, and increases with the number of training samples $m$, with the maximum size of the combination sets $n$, and with the average size of combination sets (as it can randomly vary in $\{1, n\}$). In our implementation, the average size of the combination sets can be regulated by $p$, the probability to input a black image in a given branch of the networks architecture. This variability seems to act as a degree of the regularization effect, similarly to e.g. the percentage of dropped neurons in Dropout [9]. The regularization effect can also be increased by drawing samples with replacement when creating combinations. In this work we opted for drawing samples without replacement for the simplicity of the implementation. Drawing with replacement increases the number of possible combinations, and could be beneficial for small training sets.

Zhang et al. [1] also proposed to combine training samples as a data augmentation method. Contrary to our approach, they evaluated their method on 2D images and use a large batch size. In their experiments, combining more than $n = 2$ images does not bring any improvement. With the proposed method, training with combinations of four images brought improvement over only using pairs of images, probably due to the small size of our training set. The smaller the training set is, the more regularization is needed. We did not experiment with values of $n$ larger than 4.

On both PVS and WMH datasets, using dropout [9] actually made the results worse when trained on very little data, even with low dropout rates such as 0.3.

In section 2.2, we mentioned two possible implementations of the proposed method: (1) changing the computation of the loss over mini-batches, (2) replicating the architecture of network. In this work we used the second approach, as it was simpler to implement with our library (Keras). However with this approach, all samples in the combination have to be simultaneously processed by the network, which can cause GPU memory overload in case of large 3D images or large values of $n$. The first approach does not suffer from this overload, as the
samples can be successively loaded, while only saving the individual scalar predictions in the GPU memory. In case of large 3D images, we would consequently recommend implementing the first approach.

5 Conclusion

We proposed a data-augmentation approach for neural network regressors trained with global labels, such as volumes or counts. The objective of the method is to train such regressors with small datasets. The method consists of combining training samples to create a larger training set. During training, samples are randomly grouped into sets, and their labels are summed. Using this method, we could train a neural network and reach inter-rater agreement performance on brain lesion quantification with only 25 training scans, with a single global label per scan, and without pretraining.

6 Acknowledgments

References

1. Zhang, H., Cisse, M., Dauphin, Y.N. and Lopez-Paz, D., 2017. mixup: Beyond empirical risk minimization. ICLR 2018.
2. Cole, J.H., Poudel, R.P., Tsagkrasoulis, D., Caan, M.W., Steves, C., Spector, T.D. and Montana, G., 2017. Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker. NeuroImage, 163, pp.115-124.
3. Dubost, F., Adams, H., Bortsova, G., Ikram, M.A., Niessen, W., Vernooij, M. and de Bruijne, M., 2019. 3D Regression Neural Network for the Quantification of Enlarged Perivascular Spaces in Brain MRI. Medical Image Analysis.
4. Espinosa, C.C., González G., Washko, G.R., Cazorla, M., Estpar, R.S.J., 2018. Automated Agatston score computation in non-ECG gated CT scans using deep learning. SPIE.
5. http://wmh.isi.uu.nl/
6. Desikan, R.S., Sgonne, F., Fischl, B., Quinn, B.T., Dickerson, B.C., Blacker, D., Buckner, R.L., Dale, A.M., Maguire, R.P., Hyman, B.T. and Albert, M.S., 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. Neuroimage, 31(3), pp.968-980.
7. Zeiler, M.D., 2012. ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.
8. S. Klein, M. Staring, K. Murphy, M.A. Viergever, J.P.W. Pluim, "elastix: a toolbox for intensity based medical image registration," TMI, vol. 29, no. 1, pp. 196 - 205, January 2010.
9. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), pp.1929-1958.
10. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
11. Inoue, H., 2018. Data augmentation by pairing samples for images classification. arXiv preprint [arXiv:1801.02929].
12. Simard, P.Y., LeCun, Y.A., Denker, J.S. and Victorri, B., 1998. Transformation invariance in pattern recognition: tangent distance and tangent propagation. In Neural networks: tricks of the trade (pp. 239-274). Springer, Berlin, Heidelberg.
13. Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, pp.321-357.
14. Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
15. Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint [arXiv:1502.03167].
16. LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.
17. Perez, L. and Wang, J., 2017. The effectiveness of data augmentation in image classification using deep learning. arXiv preprint [arXiv:1712.04621].
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).
19. Sixt, L., Wild, B. and Landgraf, T., 2018. Rendergan: Generating realistic labeled data. Frontiers in Robotics and AI, 5, p.66.
20. Ratner, A.J., Ehrenberg, H., Hussain, Z., Dunnmon, J. and R, C., 2017. Learning to compose domain-specific transformations for data augmentation. In Advances in neural information processing systems (pp. 3236-3246).
21. Tokozume, Y., Ushiku, Y. and Harada, T., 2017. Learning from between-class examples for deep sound recognition. arXiv preprint [arXiv:1711.10282].
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), pp.211-252.
23. Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017, July. Densely connected convolutional networks. In CVPR (Vol. 1, No. 2, p. 3).
24. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).