A two-stage model to classify joint damage in radiographs

Michael Stadler[1] and Chenfu Shi[1]

[1]Centre for Genetics and Genomics Versus Arthritis. Division of Musculoskeletal and Dermatological Sciences, School of Biological Sciences, Faculty of Biology, Medicine and Health, The University of Manchester, UK

Abstract

Joint damage in rheumatoid arthritis (RA) is assessed by manually inspecting and grading radiographs of hands, and feet. This is highly complex, and requires trained experts, whose subjective assessment leads to low inter-rater agreement. Here we develop a machine learning pipeline, powered by deep convolutional neural networks, that automates this procedure.

Landmarks on the joints in and and feet radiographs of 367 patients were manually labelled, before training a neural network to learn and predict landmarks in unseen images. Based on these predicted landmarks, joints were extracted from images, and used to train models to predict joint damage. This prediction happens in two stages, to deal with the highly imbalanced nature of the data, where a vast majority of joints show no damage. In the first stage, a filter model is trained to predict whether or not damage is present. For joints that are likely to be damaged, a second classifier is then further used to classify the damage grade.

Though the models show promising performance for some outcomes, they fail to generalize properly for rare outcomes. Bigger datasets, and gold standard labels will be required to improve models, to eventually get them ready to be used in clinical practice

1. Introduction

Rheumatoid arthritis (RA) is a common inflammatory autoimmune disease that can lead to joint damage in the form of joint space narrowing, and joint erosion [1]. The extent of this joint damage can be assessed from radiographs using the established and validated Sharp/van der Heijde (SvH) method [1]. However, this requires experienced radiologists to manually inspect and grade the images. This is not only very time consuming, but also highly subjective, ultimately leading to low inter-rater reliability, where even trained experts often disagree on the final score [2]. The goal of this challenge is to automate this process, by training a computer to automatically score the joints present in a radiograph.

The SvH method assigns different narrowing (ranging from 0-4) and erosion scores (ranging from 0-5) to joints in the hands, and feet. Only feet erosion differs, where both sides of the joint are scored independently, leading to a final score range of 0-10. All the individual score are then added up to create one overall score. The challenge consists of three sub-challenges: the first sub-challenge is to simply predict the overall score, and sub-challenges 2 & 3 require individual joint-level predictions for narrowing and erosion scores. Sets of radiographs, consisting of four images (left and right, hand & foot), from two studies [3; 4], were provided for 367 patients, with the appropriate scores per joint (foot erosion scores are

only provided as sums, and not per side of the joint). Trained models are then evaluated on a hidden set of patients, and their performance is assessed based on a weighted root mean square error (RMSE) metric. Models were able to be evaluated on a leaderboard test set, with at most 3 uploads per week, before lastly being evaluated on a different, final independent test dataset.

Our solution to this problem has three key elements. First, we train a detector model, to find joints in the images. This allows us to train joint level models, which increases the number of available samples for training considerably. Because the wrist joints are very different from the other joints in the hand, they are kept separately. Narrowing, and erosion are kept separately, leading to two models, for hand, feet, and wrist joints each. Secondly, we use a two stage approach to classify each joint: the data is highly imbalanced, with most of the joints showing no damage. We found that this often lead to models that performed well on one category, but not on the other. Specifically, resampling the data to ensure the model sees more of the damaged joints often had a negative impact on the highly important non damaged joints. We therefore decided to train filter models, which predict whether a joint is damaged or not. Models that are predicted to not be damaged, are simply predicted to have 0 damage, and others are passed along to a different model which then predicts the final damage. Lastly, we found it best to train separate models for hands and feet. However, we found that retaining a small amount of the other category (e.g.: keep some feet joints when training the hands) greatly improved performance. The next section details our models, as well as the training process.

2. Methods

This section describes the different machine learning models trained to automate the joint damage prediction. All models described are convolutional neural networks (CNN), with ReLU nonlinearity [5], and batch normalization (BatchNorm) [6] in all hidden layers. Differfent models were trained for narrowing, and erosion, in both hands and feet. The models were initially developed and evaluated with a fixed, 25% holdout, created early on into the development process. Only final submissions to the leaderboard phase, as well as the final submission, were trained with the full dataset. On top of the provided challenge dataset, we used the RSNA boneage dataset [7], as well as the NIH chest x-ray dataset [8], for different purposes during model development.

2.1 Landmark detection

The first task was to locate joints in images. To this end, we manually labelled joints in all training images with landmarks using labelme [9]. The wrists were labelled by three key positions on the wrist. We additionally labelled a small portion of the RSNA images (91 images), to improve the generalizability of our model. We opted for landmarks over a segmentation approach, because the challenge required us to submit predictions for each specific joint, which means we need to not just know where a joint is, but also which specific joint it is. Figure 1 shows examples of landmarks predicted on the RSNA dataset.

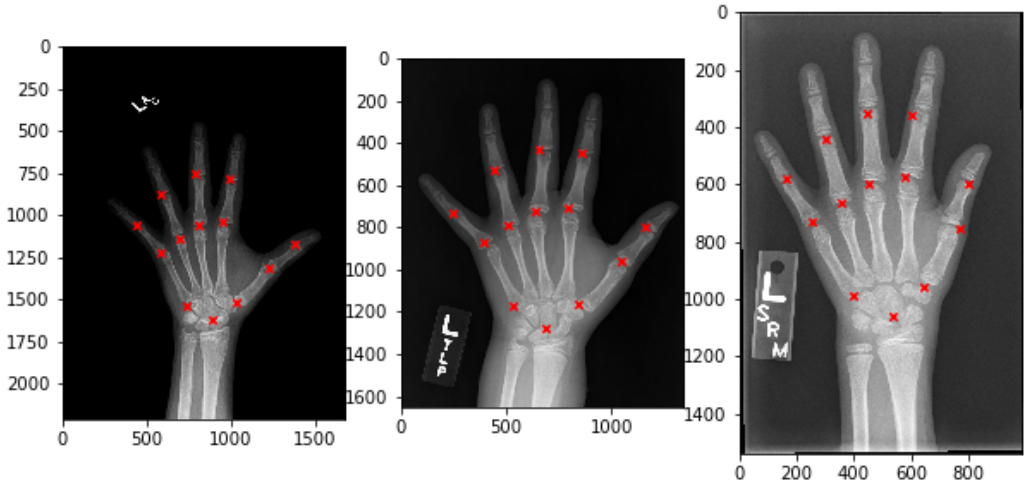

Figure 1: Examples of predicted landmarks on samples from the RSNA dataset

We then pretrained a VGG inspired CNN with 8 convolutional blocks and one dense layer, on the publicly available face landmark dataset celeba dataset [10], using a mean square error (MSE) loss. Those models are then re-trained to predict the position of hands and feet landmarks respectively. We then used those trained models as new pretrains for the respective other category, where the initial hand model was used as pretrain for the feet, and vice versa. During training, and later for extraction, images of the right hand and foot are flipped, to allow combination of all images into one model. These models were trained with the Adam optimizer [11], a batch size of 64, and with data augmentations. At training time, images were randomly augmented by cropping and zooming (up to 80%), small rotations (up to 15%), and random contrast and brightness adjustments, before downsizing to a common image size of 512x512.

The final predictions are achieved from a cascade of two models. After our initial models, we retrained a second model as described above, with slightly more parameters. While performance improved, we also found that some images would always lead invalid landmarks, outside the image dimensions. Therefore, if the bigger model fails, the smaller model redoes the predictions. Additionally, we added a fallback that if both predictors fail for an image, its joints are ignored in the later prediction stages, and their score is simply replaced by the mean from the training data. During the leaderboard phase of the challenge, only a single image failed the first predictor, and no image failed both stages.

2.2 Joint damage prediction

After predicting the joint landmarks in an image, joints are extracted by cropping out a rectangular area of the image, centered on the predicted landmark. The dimensions of the area depend on the original image dimensions, and can vary between joints (e.g.: joints on the big toe are extracted as a bigger box). These dimensions were fine-tuned manually, to try and create uniform pictures across the different joint types. For the wrists the extracted area is calculated based on the the three wrist landmarks, to ensure that all of the wrist

joints are contained in the extracted image. After extraction, augmentations are applied, before resizing the images to a common image size of 224x224. We chose those dimensions, to make models compatible with ImageNet pretrains, though we ultimately decided not to use it.

The base network is again a VGG inspired network, and an extension of the model used by [12]. The model has 6 convolutional blocks, where each block consist of two groups of Conv -> ReLU -> BatchNorm, followed by a max-pooling operation, to halve the image dimensions. The initial convolutions have 32 filters, which are doubled after every second block, up to 128 filters. The latter part of the network, varies between the filter and the prediction models. The filter networks end with a global average pooling (GAP) layer, and a single dense layer, with a sigmoid activation function, to predict the probability of a damaged joint. The prediction models on the other hand flatten the feature pixels into a single vector, which are then fed through 3 fully connected blocks, with the structure FC -> ReLU -> BatchNorm -> Dropout (0.5) [13]. The fully connected layers have 1024, 512, and 256 neurons, and are followed by a final dense layer for the linear regression output. The filter networks have roughly 600K parameters, whereas the final prediction networks have about 2M parameters.

All models were pretrained using the publicly available RSNA boneage dataset [7]. Due to the limited sample size for wrists, the wrists models were further pretrained with the NIH chest dataset [8] (for joint models we found worse performance if they were also pretrained on the chest data). Similarly, we found that filter models for the wrists performed poorly, which is why they were only trained and used for the joint predictions. Otherwise, the wrist models are equal to the joint models, except they have 6 final, linear, output layers that represent the predictions for the 6 different output joints in the wrist.

The subsections below detail the training procedures for the pretrain as well as for the filter, and damage prediction models. The filter and damage prediction models were trained using the Adam optimizer with decoupled weight decay regularization (AdamW) [14]. The used open source implementation of AdamW is available on github[1].

2.2.1 Pretrain models

Joint models are pretrained on the RSNA dataset, by predicting the landmarks in the RSNA data, and extracting the joints as described above. The models are then pretrained to predict the boneage, the patient's sex, as well as the specific joint type. Unfortunately the RSNA dataset consists of only hands, but the same pretrain is used for both the hand and feet models, regardless. For the wrist models, the model is first pretrained on the NIH chest dataset, to predict the age, sex, as well as the different disease labels available in the dataset. Similar to the joints, the model is then trained on wrists extracted from the RSNA dataset, to predict the sex, and the boneage of the image. All pretrains use the Adam optimizier, with a learning rate of 3×10^{-4}, with the same augments used by the landmark train. On top of that, a small amount of Gaussian noise is randomly added to 30% of the images, and joint images (not wrist images) are additionally randomly flipped vertically, and horizontally, with a chance of 50% respectively.

1. https://github.com/OverLordGoldDragon/keras-adamw

2.2.2 Filter models

Filter models are trained to predict whether a joint has either a narrowing or erosion score of greater than 0. To this end, the usual damage scores are turned into binary labels, where 1 indicates the presence of joint damage. All filter models were trained with AdamW, using a cosine decay learning rate schedule without restarts, and normalized weight decay, as introduced by [14]. The models were trained for 75 epochs, a batch size of 128, with an initial learning rate of 3×10^{-4}, and a base weight decay of 1×10^{-6}. The training process used the same augments as introduced for the RSNA pretrain.

Because the dataset consists of mostly easily classifiable negative samples (no damage), and some easy classified positives (very high damage), we decided to use the focal loss [15] to implicitly down-weight these easily classified samples. The focal loss is an extension of the balanced binary cross-entropy loss:

$$-\alpha_t(1-p_t)^\gamma \log(p_t) \tag{1}$$

,

where p_t is equal to the classifier's assigned probability p, for positive samples, and equal to $1-p$ for negative samples. The focal term $(1-p_t)^\gamma$ becomes smaller, as the classifier's confidence in a sample grows (p approaches 1 for positive, and 0 for negative samples). This way, the loss for easy samples with an already high confidence has less influence on the overall loss, than the loss for more difficult to classify samples. Here, γ is a hyper parameter that influences how fast the focal loss should degrade. We found that both, $\gamma == 2$ (default) and $\gamma == 1$ work reasonably well. α_t is a second hyper parameter, that additionally weighs samples based on their class. Equal to its role in the original balanced cross-entropy, the idea is to assign higher waits to samples of the rare class. In the binary case, α_t is equal to α for the rare positive samples, and set to $1-\alpha$ for negative samples. To reduce the number of parameter to fine-tune, we set α for the positive samples equal to their inverse class frequency: $\frac{n_{negatives}}{N}$. Lastly, we initialize the bias of the outcome neuron to $\ln \frac{n_{positives}}{n_{negatives}}$. This way the network reflects the scarcity of positive samples right from the start, considerably reducing the initially incurred loss.

As outlined, all models are trained with additional adversarial samples from the other joint type, mixed in. This mean that 20% of each batch used for training the hand models, consists of foot joints, and vice versa. Additionally, we trained one fully combined model for both erosion and narrowing, which is trained to predict both hand and foot joints. Our final filter models are ensembles of three filters: two 20% mixed models, with $\gamma == 2$ and $\gamma == 1$ respectively, and one fully combined model. If a filter reaches a specific classification threshold, we considered a joint to be potentially damaged, meaning it gets passed on to the damage prediction models. The different cutoffs were intentionally kept very stringent, to ensure a high recall of positives, regardless of precision, since the follow up models were still trained to predict non damaged joints as well. The cutoffs were fine-tuned on our development holdout, and we found that 0.35 worked well for the narrowing models, but the erosion models required a slightly lower cutoff 0.3. Table 1 shows the performance for these ensembles, with the specified cutoffs, for the development holdout.

Model \Metric	AUC	Recall (Negatives)	Recall (Positives)
Hands Narrowing (classification threshold: 0.35)	94.43	91.29%	87.17%
Feet Narrowing (classification threshold: 0.35)	94.82	90.23%	82.72%
Hands Erosion (classification threshold: 0.30)	90.22	90.64%	75.16%
Feet Erosion (classification threshold: 0.30)	88.94	78.46%	82.47%

Table 1: Performance of the filter models on the development holdout

2.2.3 Damage prediction models

The damage prediction models were trained using a traditional MSE loss, and AdamW with normalized weight decay and a cosine decay learning rate schedule without restarts. All models were trained for 300 epochs, with a base weight decay of 1×10^{-6}. The base learning rate was again 3×10^{-4}, except for the feet erosion models, where a higher learning rate of 1×10^{-3} was required. As with the filter models, the batch size was 128, and 20% of each batch consisted of adversarial examples of the opposite joint type. To mix feet and hand erosion, we re-scaled the scores from the adversarial samples to be in line with the scores of the main joint type. This means that feet scores mixed in with the hands were halved, and hands mixed in with the feet were doubled. This was not required for narrowing scores, because they are already on the same scale across hands and feet. Contrary to the filter models, we found no advantage of fully combining hands and feet for these continuous outcomes. Augments were the same as used for the RSNA pretrain, and the filter models.

To deal with the high imbalance of the data, we used resampling to ensure each batch contains an even amount of samples for each class. The already introduced random augments help alleviate overfitting under these aggressive resampling conditions. Despite this, the problem can not be solved altogether, and models still end up overfitting onto the majority class. However, we found our model performance much improved after resampling. Wrist models are re-sampled slightly differently, such that half of each batch is made up of non damaged wrists (all 6 scores are 0), and the other half is made up of wrists with at least some damage.

The final predictions were clipped to 0 and the maximal possible score for the specific outcome, to ensure only valid predictions are made. For the final predictions, two models are ensembled, to reduce issues of train to train variation, caused by random order and mixing of mini-batches. Table 2 shows the model performance for the different joint models on the development holdout (wrist models were omitted for the sake of brevity, since they each contain 6 outcomes).

2.2.4 Final predictions

For the final predictions, each model is used to classify the extracted joint, as well as 50 randomly augmented versions of it (using the same augments as used during the training). The final prediction is then taken as the robust mean of all predictions. To this end, all predictions are sorted, and the lowest and highest 10% of the predictions are discarded, to reduce the influence of outliers. The final prediction is then taken as the mean of the 80%

Model \Metric	MSE	MAE	RMSE	Filter-RMSE
Hands Narrowing	0.1432 (±0.5227)	0.1348 (±0.3535)	0.3784	0.7889
Feet Narrowing	0.2634 (±0.9302)	0.1908 (±0.4764)	0.5132	1.0806
Hands Erosion	0.3728 (±1.8027)	0.1859 (±0.5816)	0.6106	1.9364
Feet Erosion	0.9582 (±4.8916)	0.3584 (±0.9109)	0.9789	2.3097

Table 2: Performance metrics on the development holdout for joint models. Filter-RMSE denotes the RMSE reduced to samples with non 0 score

of the data, centered on the median. This is done for both the filters, and the damage prediction models.

For the damage prediction models, we found that, while the augments improved predictions for the damaged joints, they also introduced some noise for the joints with a score of 0, negatively impacting the RMSE score. We therefore introduced a rounding cutoff of 0.3, and predictions ≤ 0.3 are simply rounded down to 0. The final overall damage scores for SC1 are calculated by simply summing up the final individual joint predictions. Table 3 shows the scoring metrics achieved for key submissions during the leaderboard phase.

Submission ID	SC1	SC2	SC3	Changes
9703882	0.4098	0.4529	0.4242	Final Model: Combined filter models, and adjusted cutoff
9703829	0.4184	0.4555	0.432	Ensembles, as well as fixed filter cutoffs
9703700	0.4491	0.4843	0.4855	Mixed models & GAP for filter models
9703328	0.4534	0.504	0.4823	Baseline

Table 3: Leaderboard scores for key submissions, and the respective changes made

3. Discussion

Classifying joint damage in radiographs is a time consuming task that requires highly trained experts. Here we develop a pipeline that automates this process. Joints are automatically located and extracted from radio graphs of hand & feet, and then automatically scored by a two stage classifier. The first stage classifier assesses whether the joint is damage or not, and if the joint is likely to be damaged, a second stage classifier predicts the degree of damage.

All classifiers employed are similarly structured convolutional neural networks (CNN), but theoretically any stage of the pipeline can be replaced and optimized independently, using different models and architectures. During the development we also tested a range of newer model architectures such as ResNet, DenseNet, and others. However, we found that there seemingly was not enough training data, even with ImageNet pretrains, to reliably train such deep networks. Smaller versions of these networks, with less parameters, performed better, but the simple VGG architecture remained the best performing one. Training these custom architectures meant that ImageNet pretrains were not readily available, and we instead opted to use different pretraining datasets, more aligned for the task of processing radiographs. We therefore settled on the RSNA dataset, which after extracting the different

joints from the images, provided us with sufficient data to pretrain our networks. Because only one wrist can be extracted from each RSNA sample (compared to 10 joints per image), we decided to further pretrain the wrist models using the NIH chest x-ray data. Joint localization and extraction was vital, to not only be able to more accurately analyze joints individually, but more importantly to increase the training data available to train the models.

To further alleviate overfitting, the different networks are additionally regularized by a variety of measures. Firstly, common augmentation techniques are applied to prevent the network from simply remembering the images. Additionally, decoupled weight decay is used to reduce the magnitude of weight updates at every iteration [14]. However, the strongest regularization, which incurred the biggest performance gain, is achieved by mixing in adversarial samples from the other joint types. We see this as a form of gradient noise, a well known regularization factor [16]. Joints between hands and feet are similar enough, but the networks perform better if it can learn just one or the other. Adding a small portion of these additional, different looking joints, introduces noise to the gradient updates, forcing the features to remain more general. We found this to work very well with the decoupled weight decay of AdamW, which incurs a similar effect when used in conjunction with batch normalization [17]. A stronger effect can likely be achieved by increasing the portion of adversarial samples in each batch, as training goes on.

The provided dataset represented the biggest limiting factor for model development. All models are limited by the small sample size, especially for rare outcomes. Generally the models for the feet images could be improved by finding a pretrain with feet x-rays. Furthermore, as outlined in the introduction, the feet erosion score is derived from scoring both sides of the joint individually, before adding them up. The main drawback of this is that very high scores become extremely rare, with only few samples existing with extensive damage on both sides (overall score ≥ 7). We therefore hypothesize that the feet erosion scores could easily be improved, by providing separate labels for both halves. Regardless of the modelling approach, with the same number of images, this would automatically double the number of available training samples. On top of that, the extremely rare classes would be subsumed by the more common smaller classes, making the classification easier. The wrist models are especially limited by the available sample size, performing significantly worse than the joint level models. However, there is no easy solution to this, as data collection is a laborious and expensive effort. The models presented here serve as promising starting points, but future efforts should focus on collecting additional data, especially from patient's with advanced joint damage. We expect this to be vital to create models ready to be used in clinical practice.

One of the major hindrances for developing our models was the hidden evaluation metrics used to rank submissions. Often, developed models that improved traditional validation metrics across all outcomes (sometimes considerably), did not improve performance with respect to the hidden weighted outcome RMSE. Specifically, we noticed that despite the weighting efforts, performance on non damaged joints seemed to remain one of the main drivers of performance, which eventually lead to use developing the presented two stage approach. This problematic was gravely aggravated by the limited number of uploads available during the leaderboard phase. We strongly feel that, since there was a final hidden test set

anyway, uploads should not have been limited this severely. Ultimately, more uploads to be able to validate models would have allowed us to improve our models considerably, and we have no doubt that the same holds for the other teams as well.

Lastly, it is important to note that the models here were trained with labels derived from a single scorer. Future models, suited for clinical practice, will not just have to perform better, but also be trained from gold standard labels, derived from the agreement of multiple raters. Despite these limitations, we believe these results present a noteworthy step forward. With some additional tuning, the introduced filter models could already be used to lessen the work load in manual image annotation considerably, by automating the annotation for a large percentage of the joints. In the future, bigger datasets will hopefully reduce the issue of class imbalance, and also allow training of better performing more complex models.

References

[1] D. Van der Heijde, M. Van Leeuwen, P. Van Riel, and L. Van de Putte, "Radiographic progression on radiographs of hands and feet during the first 3 years of rheumatoid arthritis measured according to sharp's method (van der heijde modification)," 1995.

[2] A. A. Renshaw and E. W. Gould, "Comparison of disagreement and error rates for three types of interdepartmental consultations," *American journal of clinical pathology*, vol. 124, no. 6, pp. 878–882, 2005.

[3] S. L. Bridges Jr, Z. L. Causey, P. I. Burgos, B. Q. N. Huynh, L. B. Hughes, M. I. Danila, A. Van Everdingen, S. Ledbetter, D. L. Conn, A. Tamhane, *et al.*, "Radiographic severity of rheumatoid arthritis in african americans: results from a multicenter observational study," *Arthritis care & research*, vol. 62, no. 5, pp. 624–631, 2010.

[4] M. J. Ormseth, P. G. Yancey, J. F. Solus, S. L. Bridges Jr, J. R. Curtis, M. F. Linton, S. Fazio, S. S. Davies, L. J. Roberts, K. C. Vickers, *et al.*, "Effect of drug therapy on net cholesterol efflux capacity of high-density lipoprotein–enriched serum in rheumatoid arthritis," *Arthritis & rheumatology*, vol. 68, no. 9, pp. 2099–2105, 2016.

[5] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[6] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[7] S. S. Halabi, L. M. Prevedello, J. Kalpathy-Cramer, A. B. Mamonov, A. Bilbily, M. Cicero, I. Pan, L. A. Pereira, R. T. Sousa, N. Abdala, *et al.*, "The rsna pediatric bone age machine learning challenge," *Radiology*, vol. 290, no. 2, pp. 498–503, 2019.

[8] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

[9] K. Wada, "labelme: Image Polygonal Annotation with Python." `https://github.com/wkentaro/labelme`, 2016.

[10] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August*, vol. 15, p. 2018, 2018.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[12] J. Rohrbach, T. Reinhard, B. Sick, and O. Dürr, "Bone erosion scoring for rheumatoid arthritis with deep convolutional neural networks," *Computers & Electrical Engineering*, vol. 78, pp. 472–481, 2019.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[16] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," *arXiv preprint arXiv:1511.06807*, 2015.

[17] T. Van Laarhoven, "L2 regularization versus batch and weight normalization," *arXiv preprint arXiv:1706.05350*, 2017.

Authors Statement

All authors have contributed equally to all aspects of this work.

Acknowledgements

This work was jointly supported by the Medical Research Council (MRC) - grant code MR/N013751/1 [Stadler], and the Wellcome Trust (Wellcome) - grant code 215207/Z/19/Z [Shi].
Additionally, we would like to thank our respective PhD supervisory teams, for their support during this challenge.
Lastly, we would like to acknowledge the assistance given by IT Services and the use of the Computational Shared Facility at The University of Manchester.