SUBMISSION TO NODE21 CHALLENGE (TEAM MTEC)

Finn Behrendt *Marcel Bengs* *Alexander Schlaefer*

Institute of Medical Technology and Intelligent Systems, Hamburg University of Technology, Germany

1. INTRODUCTION

Automated detection and localization of lung nodules is a challenging task especially if only chest radiographs are available. Recently, convolutional neural networks have shown promising results for this task. Still, the task remains difficult. First, the available data for training is limited as annotated data is rare and expensive to obtain in the medical domain. Even though there exist large-scale data sets of chest x-ray radiographs [1, 2, 3, 4] only a small part of the data sets contain nodules and even less reliable annotations from domain experts are available. Second, imbalanced data sets are very challenging and especially larger models are prone to overfitting without proper rebalancing. Furthermore, class imbalances impede the training and generalization of deep learning methods especially for the task of object detection [5].

In the recent literature, various approaches and model architectures for the task of object detection are proposed [6]. These models are often compared on large-scale benchmark data sets such as the Common Objects in Context data set (COCO) [7]. However, due to the lack of a general benchmark data set for the medical domain and especially nodule detection, a fair comparison of the different approaches is hard to obtain. To collect and report state-of-the-art methods for automated nodule detection, Ecem Sogancioglu et al. host the Node21 Challenge where participants can submit individual solutions. The challenge has two tracks, nodule detection and generation. A data set, annotated and revised by radiologists is provided to the participants for the training of individual solutions. With this submission, we provide our solution for the detection track of Node21 and systematically address each of the aforementioned limitations. First, to address the class imbalance, we generate artificial nodules and add them to the training data to obtain a balanced training set. Second, to address limited data we apply transfer learning by using pre-trained models, trained on a similar task and only fine-tune the model parameters with the provided data. Thereby, we enrich the experience of our models and counteract overfitting. Lastly, with several methods available for object detection and their individual advantages, we leverage a selection of methods to obtain a rich and well generalizing algorithm for our final submission.

2. METHODS

Table 1. Summary of the data sets with their respective number of images and fraction of images containing nodules. After sampling the additional test set, we apply a 5-fold cross-validation to the data set. For the training- and validation set, representative numbers of one fold are reported.

Data Set	Number of Images	Fraction Positives
Training Set	3626	23%
Validation Set	906	23%
Add. Test Set	350	50%
Exp. Test Set	281	59%
Final Test Set	N/A	N/A

2.1. Data Set

The data set of the Node21 challenge consists of 4882 frontal chest radiographs from four different public data sets (JRST [1], PadChest [3], Chestx-ray14 [2], Open-I [4]). All images are revised and annotated by radiologists. The majority of the radiographs (N=3748) are free of nodules while 1134 radiographs show at least one nodule (1476 nodules in total). For the evaluation of the challenge submissions, two additional test sets are used. One experimental test set (N=281) to test intermediate results for the participants and one final test set to evaluate the final submission. In contrast to the training set, the experimental test set shows a smaller class imbalance. For the final test set, no further details are provided.

To evaluate our models and methods we partition the provided data set into a training set (N=4532) and an additional test set (N=350). We use this test set beside the experimental test, which only has limited evaluations. To achieve a balanced test set, which represents a high variance, we take an equal amount of images with nodules and images without nodules from each public available data set. For JRST, PadChest and Chestx-ray we use 50 images from each class respectively while for the Open-I data set we only use 25 images due to the low number of images with nodules. Furthermore, we apply a 5-Fold cross-validation to the training set. To ensure a similar class distribution across the folds we sample the individual training and validation sets in a stratified fashion grouped by

patients. Having tuned our algorithms, we merge the additional test set to our training set to train the final models.

2.2. Pre-Processing

We follow the pre-processing strategy provided by the challenge organizers. No additional pre-processing steps are applied. The images are provided with a resolution of 1024×1024px. For models with a required input resolution other than 1024×1024px, we resize the images to the desired input resolution by linear interpolation and scale back the detected bounding boxes to match the input resolution for evaluation.

2.3. General approach

Our general approach is to use an ensemble of high-performing state-of-the-art methods to leverage all individual benefits and to build a well-generalizing model. We include four different models to our experiments, namely Faster-R-CNN [8], RetinaNet [8], EfficientDet-D2 [9] and Yolov5 [10]. To account for the limited training data, except for RetinaNet, all models utilize pre-trained weights as is described in section 2.6. To tackle the class imbalance in the data, we generate images with nodules and add them to the training set. Thereby, we replace nodule-free images in the training set with images with generated nodules (for details, see Section 2.4. For better generalization, we apply 5-fold cross-validation.
Afterwards we consider each model from the individual folds and ensemble the predictions with weighted box fusion [11] to obtain a combined prediction. In the following, details for the used architectures are provided. All models are implemented using Pytorch with Pytorch-lightning, except for Yolov5 where the original pipeline is adapted as it comes with excellent pre- and post-processing steps.

2.3.1. Faster-R-CNN

We use an implementation of Faster-R-CNN that is similar to the baseline algorithm provided by the challenge hosts. We utilize the torchivison implementation of Faster-R-CNN with a pre-trained ResNet-50 as a backbone network. All model-specific parameters are kept unchanged. However, for transfer learning, we keep all model parameters trainable.

2.3.2. RetinaNet

For RetinaNet, similar to Faster-R-CNN we make use of the torchvision implementation with a ResNet-50 as a backbone.

2.3.3. EfficientDet-D2

We implement EfficientDet-D2 with a EfficientNet-B2 backbone based on [12]. We allow a maximum of 100 predicted bounding boxes per image and drop all predictions with a predicted score below 0.01. In addition, we freeze all batch normalization layers. This is motivated by two reasons. First, the retraining of the networks by the challenge hosts requires the models to run with a small batch size as the GPU memory is limited to 16GB. However, reasonable batch size is required to produce a meaningful estimate of the population parameters. Therefore, we freeze the batch normalization parameters, to avoid undesirable training configurations. Second, with frozen parameters we observed less overfitting of our models when training with imbalanced data sets.

2.3.4. Yolov5

For Yolov5 we use the original implementation from [10]. For our experiments, we chose Yolov5x as model architecture. As for EfficientDet-D2, we drop all predictions with a prediction score below 0.01. Furthermore, on top of oversampling, the positive classes are weighted by a factor of 1.27. For Yolov5, we use two different input resolutions of 640×640px and 1024×1024px and merge the predictions for both resolutions in the final ensemble model. The size of anchor proposals is automatically chosen by k-means clustering based on the bounding box sizes before training.

2.4. Imbalanced Sampling and Nodule Generation

As a first step to address the class imbalance, we oversample the minority class of our training set, to rebalance the mini-batches. During validation, we undersample the minority class to enable a proper validation. Thereby, we achieve a balanced dataset for both training and validation. As a second step, we consider generating artificial nodules. To this end, we make use of the provided baseline algorithm from the generation track of the node21 challenge. We randomly sample 1000 images from the training data and use the generation algorithm to place one or more nodules in the healthy image. By this, we achieve a balanced data set for the training and evaluation of our models. Note, that the nodule generation is done offline beforehand. As a result, we achieve a balanced data set for training and evaluation without the need for oversampling and thus with a reduced risk for overfitting.

2.5. Data Augmentation

We use common data augmentation strategies provided by the albumentations library [13] for training. For Faster-R-CNN, RetinaNet and EfficientDet-D2 we crop or pad the images randomly by a maximum of 50 pixels to add robustness for different fields of view. Also, Horizontal flipping (p=0.5) and random rotation by a maximum of 5 degrees (p=0.6) are applied. Furthermore, we blur the images (p = 0.5) and apply cutout augmentation (p=0.5). For Yolov5 the augmentation

Table 2. Hyperparameters for Training. For gradient clipping, the gradients' global norm is clipped to the reported values.

Parameter	Yolov5-large	Yolov5-small	Faster-R-CNN	RetinaNet	EfficientDet-D2
Learning Rate	8.94e-3	1.15e-2	1.0e-4	1.0e-4	1.0e-4
Optimizer	SGD	SGD	Adam	Adam	Adam
Batch Size	8	16	16	16	16
Epochs	20	20	25	60	20
SWA Start	N/A	N/A	20	45	15
SWA Annealing epochs	N/A	N/A	5	15	5
Warmup Epochs	2.5	2.8	5.0	5.0	5.0
Gradient Clipping Value	N/A	N/A	3.0	3.0	3.0

strategies of the original pipeline are used[1]. During the evaluation, no augmentation is applied, except for Yolov5, where test time augmentation (TTA) is applied. Hereby, each image is evaluated multiple times for flipped and scaled versions of the image. The predictions are then merged before applying non-maximum-suppression.

2.6. Transfer Learning

For Yolov5, Faster-R-CNN and EfficientDet-D2, we make use of pre-trained model weights to account for the limited training data. The models are pre-trained on the VinDR data set [14] and the model checkpoints originate from the VinBigData Chest X-ray Abnormalities Detection Challenge. Thus, the pre-trained weights are well suited to be used as starting point for fine-tuning the models on the nodule data set. Across all models, we keep all layers trainable except for batch norm parameters.

2.7. Training parameters

For all our models we individually tune hyperparameters coarsely based on the validation set performance of our cross-validation approach. Therefore, the parameters differ for each model. For Faster-R-CNN, RetinaNet and EfficientDet-D2, we train a fixed number of epochs, apply stochastic weight averaging (SWA) [15] and use the last checkpoint for the final prediction. For Yolov5, we validate our models every epoch and early stopping is applied based on the validation set. Here, training is stopped, if the weighted combination of Precision, Recall and mean average precision does not improve for 12 epochs. Finally, the best model is used for the final prediction. A summary of all training parameters is provided in Table 2. For all models, cosine annealing [16] is used as learning rate schedule. Training of our models is performed on NVIDIA RTX 3090 and NVIDIA V100 (32GB) graphics cards depending on the model size.

[1] Specific data augmentation parameters are fine-tuned by hyperparameter evolving, provided by the Yolo framework

2.8. Final Submission and Ensembling

For our final submission, we build an ensemble of Faster-R-CNN, RetinaNet, EfficientDet-D2 and Yolov5. Additionally, we use two different input resolutions for the Yolov5 model, namely 640×640px and 1024×1024px. To leverage the whole data set, we fuse our additional test set to the training set for training the final models. For each model, we train 5 versions from different folds of the data set by performing a five-fold training approach. As inference time is limited, we only use one out of five folds of EfficientDet-D2 for our final model and replace it with one fold of yolov5-large. This results in 20 model checkpoints in total. For each model prediction we use non-maximum-suppression with an IoU threshold of 0.2 to suppress overlapping box predictions of the same model. Finally, we ensemble all model predictions by using weighted box fusing [11]. Here we skip boxes with a score below 0.1 and predicted boxes are merged if their IoU is above 0.2.

3. FINDINGS

In this section, we want to provide additional findings that arise during working on the challenge. First, we observe that the imbalanced data set is very challenging and especially larger models are prone to overfitting without proper rebalancing. Our experiments show that simple oversampling can mitigate the effect of the imbalanced training data. Furthermore, the generation of artificial nodules has shown small improvements over the oversampling approach. Thus, we decided to use the training set with generated nodules for the training of our final models. Further work could focus on an additional class weighting of the loss during training and also training sets with a larger proportion of images with nodules would be interesting to investigate.

Second, we investigate the use of detection transformer (DETR) [17]. Even though DETR outperforms all other models on our evaluation sets, it shows significant performance drops on the experimental test set. We believe, that a larger data set would be required to train this model properly as it did not even converge if trained from scratch.

Third, we investigate the use of TTA and mosaic augmenta-

tion for all models. However, for both techniques, only the yolov5 model shows benefits from their use.
Overall, we tuned our algorithms and all hyperparameters coarsely based on the validation- private- and experimental test set. In general, for tuning hyperparameters, we utilized our additional test set and for broader model decisions we also evaluated the algorithms on the experimental test set. We observed that the metrics on the different data sets are not always congruent which might be caused by the different degrees of imbalances. In cases where no congruent result was found, we chose the solution that works best on the experimental test set, as a smaller domain shift to the final test set is assumed.

4. REFERENCES

[1] Junji Shiraishi, Shigehiko Katsuragawa, Junpei Ikezoe, Tsuneo Matsumoto, Takeshi Kobayashi, Ken-ichi Komatsu, Mitate Matsui, Hiroshi Fujita, Yoshie Kodera, and Kunio Doi, "Development of a digital image database for chest radiographs with and without a lung nodule," *American Journal of Roentgenology*, vol. 174, no. 1, pp. 71–74, 2000, PMID: 10628457.

[2] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3462–3471.

[3] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá, "Padchest: A large chest x-ray image dataset with multi-label annotated reports," *Medical Image Analysis*, vol. 66, pp. 101797, 2020.

[4] "Design and development of a multimodal biomedical information retrieval system," *Journal of Computing Science and Engineering*, vol. 6, pp. 168–177, 2012.

[5] K. Oksuz, B. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 43, no. 10, pp. 3388–3415, oct 2021.

[6] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. PP, pp. 1–1, 09 2019.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, Cambridge, MA, USA, 2015, NIPS'15, p. 91–99, MIT Press.

[9] Mingxing Tan, Ruoming Pang, and Quoc V. Le, "Efficientdet: Scalable and efficient object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10778–10787.

[10] Glenn Jocher, "ultralytics/yolov5: v6.0," https://github.com/ultralytics/yolov5/tree/v5.0, Oct. 2021.

[11] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, pp. 1–6, 2021.

[12] Ross Wightman, "efficientdet-pytorch," https://github.com/rwightman/efficientdet-pytorch, 2021.

[13] E. Khvedchenya V. I. Iglovikov A. Buslaev, A. Parinov and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *ArXiv e-prints*, 2018.

[14] Ha Q. Nguyen, Khanh Lam, Linh T. Le, Hieu H. Pham, Dat Q. Tran, Dung B. Nguyen, Dung D. Le, Chi M. Pham, Hang T. T. Tong, Diep H. Dinh, Cuong D. Do, Luu T. Doan, Cuong N. Nguyen, Binh T. Nguyen, Que V. Nguyen, Au D. Hoang, Hien N. Phan, Anh T. Nguyen, Phuong H. Ho, Dat T. Ngo, Nghia T. Nguyen, Nhan T. Nguyen, Minh Dao, and Van Vu, "Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations," *ArXiv e-prints*, 2020.

[15] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson, "Averaging weights leads to wider optima and better generalization," *ArXiv e-prints*, 2019.

[16] Ilya Loshchilov and Frank Hutter, "SGDR: stochastic gradient descent with warm restarts," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017, OpenReview.net.

[17] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, Eds., Cham, 2020, pp. 213–229, Springer International Publishing.