Efficient object detection in 3D CT scans

Jannis van Kersbergen, Eindhoven University of Technology, February 2021

***Abstract*—With the advancement of neural network object detectors within the field of computer vision, the technology has become mature enough to be applied to real-world medical problems, such as rib fracture detection in 3D CT scans. Most research on object detectors has been done on 2D images of common objects, such as cars, dogs or trees. The aim of this study was to expand this research by adding an additional spatial dimension and switching the image domain to CT scans. To achieve this, we developed a neural network object detection architecture by adjusting the recently published neural network object detection architecture called EfficientDet. To be of use in a medical setting, prediction speed is important, therefore the computational complexity was constrained to a minimum, while achieving a strong performance was still necessary. As a proof of concept, the 'object chest X-ray' dataset was analysed with promising results. Main experiments were performed on the large real-world dataset 'RibFrac' containing 3D torso CT scans. To compare classification accuracy, the state-of-the-art neural network classifier InceptionNet was used as a benchmark. We found that our pipeline based on EfficientDet was able to achieve a higher performance than InceptionNet in the classification task, while additionally providing localization information. With a false positive rate of five false positives per scan, 0.76 of all fractures could be detected and scans containing rib fractures could be identified with an AUC of 0.976.**

I. Introduction

With the increasing use of neural networks in recent years, much progress has been made in the field of computer vision. Deep learning algorithms showed their potential by improving a multitude of computer vision tasks such as image classification [11], segmentation [25] or hand writing recognition [5]. With a gradual improvement over the last years, neural networks reached a maturity that allowed them to be used for solving real world problems outside of classic or academic computer vision tasks.

With the constant advancement of medical imaging acquisition technologies, the amount of image data that is used in health care has increased drastically in recent years [17]. Due to their complexity, interpretation of medical images is mainly limited to trained and experienced experts. To keep up with the increasing demand of medical image analysis and to support clinicians, sophisticated machine learning methods are needed.

One important field within medical image processing is object detection, which entails localizing and classifying salient image parts. Use cases include quality control by finding artifacts or supporting diagnoses with pathology detection. Traditional machine learning approaches tend to rely on handcrafted features based on medically trained experts, whose availability is limited for broad scale research. Moreover, the experience of experts is not always easily translated into explicit clear cut guidelines for feature crafting. This gap can be filled by deep learning. Not only has deep learning proven to surpass traditional machine learning methods in many other fields, such as image classification or segmentation, but it is less dependent on specific domain knowledge. Useful features are generally found automatically by the deep learning algorithm, as long as large enough datasets are available. Of course, domain- or dataset-specific knowledge can improve the algorithm.

A major challenge in this field is that medical datasets tend to be much smaller than datasets for natural images, due to patient privacy but also because annotations cannot be crowd sourced and only be made by medically trained experts, making the generation of large medical datasets time intensive and expensive. Small datasets in turn can impede the deep learning algorithm.

A second challenge is that much of the recent research on deep learning based object detection has been focused on common objects [8], [18], [4], [30] in RGB-images, such as finding pedestrians, cars and traffic lights in a crowded street scene. Since the domain of medical images can be quite different from natural images, it is not obvious that the same performance can be achieved without modifications. This is an active field of research, with open challenges, such as the MIDL2020 [2], CAMELYON16 [7] or MICCAI2020 [3], which are based on medical images of different modalities with the aim of finding pathologies or foreign objects.

In the medical praxis machine learning systems tend to be used for decision support, where the final decision and responsibility lies in the hands of the medical practitioner. Therefore, pixel-perfect predictions of the machine learning solution at all costs should not be the singular goal. While a high performance is the minimum requirement in a clinical setting, a short inference time is imperative to actually provide support instead of distractions. Also, with the limited resources available in many hospitals and other health care providers, the computational complexity should be kept as low as possible to make them more widely available.

Computerized tomography (CT) is a widely used medical imaging methodology, for example for diagnosing physical trauma patients [12]. In sever injury cases, diagnosis speed can have a large impact on survival outcome. CT scans are obtained by combining a series of X-ray images taken from different angles to create 3D volume of bones, blood vessels and tissues inside a body or body part. Voxel intensities correspond to values on the Hounsfield scale, which is a quantitative scale that describes radiodensity. The basic unit is the Houndsfield Unit (HU), which is low for low density fluids like water or blood and increases for denser substances like cartilage or bone. Potential machine learning algorithm therefore have to fulfil several requirements. On the one hand they must be able to make predictions with a short inference time and be computationally simple enough so that they are widely affordable for hospitals and other health care providers.

On the other hand they must be accurate and find relevant objects without a large false positive rate. Ideally the algorithm should also be generic enough to be applied to wider range of object detection tasks. For this study we chose to base our algorithm on the architecture 'EfficientDet', because of its impressive trade-off between inference time and performance measured on a widely used object detection benchmark.

Given these highly advanced techniques with excellent performance in benchmarks, the research question of this study is, how can rib fractures be found in 3D CT scans in a fast and reliable way using object detectors.

To investigate this, two datasets were used. First, a dataset of 2D chest X-rays, with foreign objects present in some images. X-ray images are widely used because of their low cost and high acquisition speed, but diagnosis and further processing can be hindered by foreign objects. This dataset is used as a proof of concept, since its objects tend to be large and clearly delineated from the surrounding tissue. Second, a dataset of 3D chest CT scans containing rib fractures was used. Rib fractures are much smaller relative to the image size and can be much harder to identify with the naked eye.

The contributions of this study are threefold. First, our study is a feasibility study which explores to which extent a state-of-the-art object detector (EfficientDet, originally developed for 2D images of common objects) can be applied to 3D world medical datasets, with a constraint on computational complexity.

Second, we present a complete data analysis pipeline, including extensive postprocessing of the object detection output, that is able to provide classification and localization information on multiple scales.

Third, we introduce a new evaluation metric, which is not based on the intersection over union (IoU), as is typically used in evaluating object detection algorithms. Our aim was to provide a more intuitive and simply metric, which might be more suited for algorithms designed to support rather than replace medical practitioners.

A. Related Research

Image object detection aims to localize and classify objects contained in an image. Object detectors based on neural networks can roughly be divided into two classes: two-stage detectors and one-stage detectors.

Two-stage detectors such as the 'Faster R-CNN' [24] or the 'Mask R-CNN' [8] are large models focusing on high performance rather than on fast inference. During the first stage, category independent bounding boxes are generated, which can be seen as candidate image patches encompassing an object. In the second stage those bounding boxes are processed further. The location is refined and the object class is predicted. To make sure all objects within in image are found, the first stage generally generates a large amount of so called region proposals, which can be rejected in the second stage. To give room for the possibility that some region proposals do not contain an object, the second stage calculates a confidence score, which represents the predicted probability of the region containing an object of interest at all. By thresholding low confidence predictions, false positives can be filtered out easily.

One-stage detectors such as SSD [21], Yolo [4], RetinaNet [18] or EfficientDet [30] focus on achieving high inference speeds by reducing the complexity, which initially led to worse performance than two-stage detectors. In one-stage detectors bounding boxes around objects are predicted in one go without first generating region proposals. First, the input is fed into the backbone network, which acts as a basic feature extractor. It is usually a deep CNN such as ResNet [9] or AmoebaNet [22] but depending on the application requirements, it can also be a lightweight network such as SqueezeNet [13] or MobileNet [10].

The next module in the processing pipeline is the neck network. While the backbone network is a bottom-up feature extractor, the neck network acts as a top-down feature aggregator, which combines the feature maps of the backbone network at different resolutions by means of a CNN, see Fig. 5 It can be as simple as a feature pyramid network (FPN), which consists of a single top-down path of convolutional layers with cross connections from the backbone network or as complex as the bi-directional feature pyramid network (BiFPN), see Fig. 5, where information is processed and combined in an additional bottom-up path with skip connections from the backbone nodes.

The last module is the head, which usually consists of a shallow CNN. In the head network, the actual bounding boxes around objects are predicted. Per predicted bounding box, the output values are the location, usually indicated by the coordinates of the top left and bottom right corner of the rectangular box, a probability distribution over the possible classes of objects and a confidence score, indicating the estimated likelihood of an object being present. Most one-stage object detectors predict bounding boxes on feature maps of different resolutions to better find objects of different sizes and scales. In Fig. 5, the head network would be applied to all five outgoing edges of the BiFPN.

Although initially prioritising speed over performance, modern one-stage detectors such as Yolo v4 [4] and EfficientDet [30] reach similar performance on commonly used benchmarks. Since for medical applications both speed and accuracy are important, this study utilizes one-stage detectors.

A common benchmark for object detection algorithms is the Common Object in Context (COCO) dataset [32], with the performance metric average precision (AP), which combines classification and localization evaluation. Fig. 1 shows a comparison of several of the mentioned object detectors, plotting the performance (COCO AP) against the computational complexity (FLOPs). The different data points belonging to the same curve represent different versions of the same basic architecture. As can be seen, EfficientDet achieves a strong performance with minimal computational complexity.

Since objects within a dataset do not come in all shapes and sizes, the quality of region proposals can be increased by using so called anchors. An anchor is a bounding box with a predefined size and aspect ratio, which is used as a prior in the first stage during the region proposal generation. The optimal anchor sizes and aspect ratios depend on the particular dataset

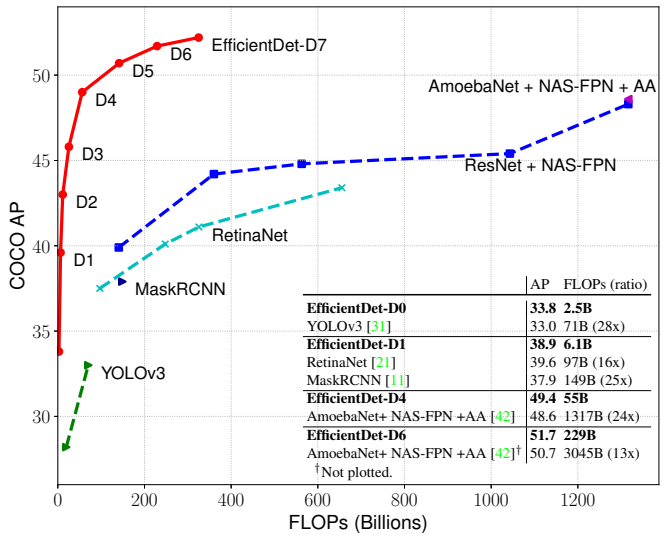

Fig. 1. Comparison of current state-of-the-art deep learning object detection architectures in terms of speed (FLOPs) and precision (COCO AP) on the COCO benchmark. Colored curves show the same base architecture at different scales, increasing performance but also computational complexity. EfficientDet shows that high performance can be reached with relatively small number of operations. Illustration reproduced from [15]

and can be calculated in a variety of ways such as k-means clustering [23], or differential evolution [34].

II. Data

For this study, two open datasets were used: the 'object chest X-ray' (OCXR) dataset [2] and the rib fracture (RibFrac) dataset [16].

A. OCXR

The OCXR dataset [2], which is part of the MIDL2020 object detection challenge, consists of 10 000 chest X-ray images collected from around 300 township hospitals in China. 50% of the images contain foreign images, 50% do not contain any foreign objects. Fig. 2 shows an example image with annotations. The goal of the OCXR challenge is to improve quality control for X-ray images, which are commonly used for pulmonary and heart disease diagnoses. Foreign objects constitute artifacts, which may occlude pathologies and hinder further processing in general, leading increasing false negative and false positive rates. The data stems from township hospitals and especially in rural and remote locations, standard image acquisition guidelines are not followed strictly enough leading to nearly a third of X-ray images not being usable for diagnosis [2]. Automating the detection of foreign objects could reduce cost and save radiologists' time to focus on other aspects of patient care. The images were annotated by 12 medically trained radiologists with 1 to 3 years of experience. The annotations can take three different shapes: rectangular bounding boxes, bounding ellipses and bounding polygons. The shape does not carry additional information and is only dependent on the annotators preferences.

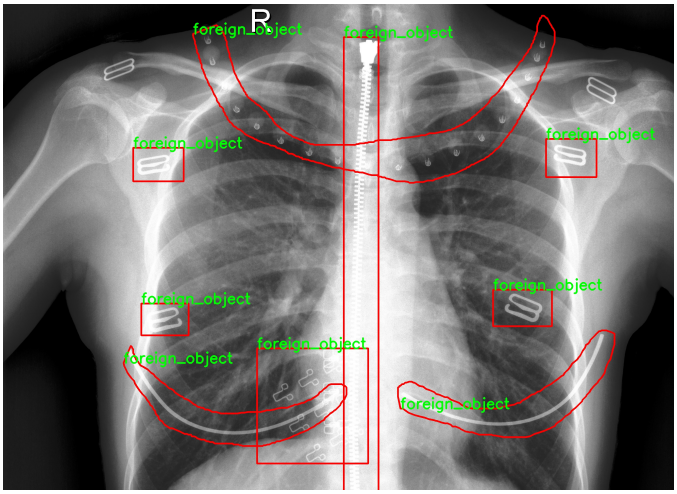

Fig. 2. Example of an X-ray image of the OCXR dataset with foreign objects being present. Red outlined regions represent the ground truth of foreign objects, which have to be detected by the machine learning algorithm

The dataset was randomly partioned into a training set (8 000 images), a validation set (1 000 images) and a test set (1 000 images). The ratio of images with and without foreign objects is 1:1 for all three data partitions. The image width varied between 1089 and 4096 pixels, the image height varied between 975 and 4932 pixels. Raw images had varying orientation with all multiples of 90° being present. Furthermore, several images had an inverse intensity, resulting in for example bones being represented as dark instead of light pixels.

B. RibFrac

The RibFrac dataset [16] is part of the MICCAI2020 conference and is published as an open challenge. Rib fracture detection is an important and common task in clinical practice, for which little research focusing on automatic machine learning methods has been done [16]. The original dataset consists of 660 3D CT scans, containing around 5 000 rib fractures. However, ground truth annotations are only openly available for the predefined training and validation set, making a proper crossfold validation impossible and only allowing for a test set performance analysis with a limited number of metrics. Therefore, the training and validation set of the original dataset was re-partioned into a training set consisting of 385 scans, a validation set of 25 scans and a test set of 90 scans for a threefold cross validation. Twenty scans contained no fractures, all other 480 scans contained at least one fracture. The new validation and test sets were selected in a way such that no scan was in multiple validation or test sets. In addition, all 20 volumes containing no fractures were always in the test set. Since the network inputs are transversal slices and in all volumes, the majority of slices do not contain fractures, the network was still fed with a sufficient number of slices of the healthy class. To calculate proper statistics on a volume level, the number of scans containing no fractures should be as high as possible.

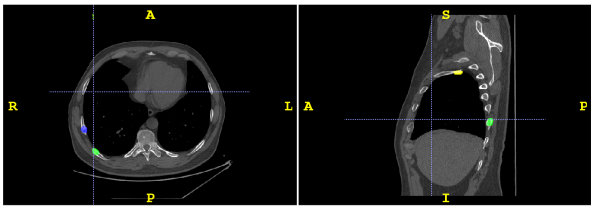

Fig. 3. Example of a scan of the RibFrac dataset from different perspectives. Left: transversal plane, right: sagittal plane. Rib fracture ground truth annotations are marked in bright red, other colors represent predictions aggregated into clusters.

An example scan is displayed in Fig. 3. Note that objects, that is rib fractures, are much smaller in this dataset than in the OCXR dataset.

Fractures are classified into five categories:

- displaced - 684 occurrences
- non-displaced - 630 occurrences
- buckle - 321 occurrences
- segmental - 209 occurrences
- other - 2576 occurrences

The 'other'-type was assigned if the fracture was ambiguous or difficult to diagnose.

The dimensions of the scans are: 512x512xN voxels, where N can vary from 74 to 649.

Ground truth annotations consist of voxel-level masks for regions containing rib fractures.

III. Methods

In this section, the complete data analysis pipeline is presented, which is able to output classification and localization information of objects in medical images. A flowchart of the pipeline is displayed in Fig. 4.

A. Model architecture - EfficientDet

The deep learning baseline and starting point for this study was the one-stage detector EfficientDet [30], which reaches state-of-the-art object detection performance for natural images of common objects and is easily scalable. Like most object detectors, EfficientDet consists of three parts: the backbone, the neck and the head. A schematic is shown in Fig. 5.

1) Backbone - EfficientNet: The backbone architecture is EfficientNet [31]. EfficientNet is a fully convolutional CNN, which makes use of skip connections, where layers are not only connected to the immediately following layer but also to layers deeper in the network, skipping several layers. Skip connections help to combat the problem of vanishing gradients of deep neural networks and since they also improve general performance, they have become a standard in many modern CNN architectures. The main idea behind EfficientNet is to balance the network's parameters in a way that the available computational power is used optimally. The three key scaling options are network depth (i.e. amount of layers), network width (i.e. amount of channels per layer) and the input resolution. Increasing any of the three factors can improve the network performance, but the performance gain will at some point saturate. For examples, the performance of ResNet with 100 or 1000 layers is almost identical [31], while the computational complexity is scaled with a factor of 10. The developers of EfficientNet developed eight empirically tested versions of the base architecture at different scales, which find a good compromise between the three scaling factors.

2) Neck - BiFPN: The neck of EfficientDet is the BiFPN displayed in Fig. 5 The white nodes on the right-hand side represent the feature maps of the backbone network at various resolutions - a higher P index correspond to half of the resolution as the next lower P index. With the BiFPN, several adjustments to the regular FPN are proposed. First, nodes with only a single outgoing edge (second column, top most and bottom most node) are removed to reach a high amount of cross-resolution feature fusion without increasing the computational complexity too much. Nodes that now end up with only one outgoing edge receive an additional outgoing edge, with either a skip connection or a cross resolution edge.

Second, the neck network can consist of multiple BiFPN layers, increasing the amount of feature fusion further. In Fig. 5 an example is shown with three BiFPN blocks.

Third, BiFPN uses an advanced method of combining features of different resolutions. To combine feature maps from different resolutions, commonly feature maps are resized to the same size and simply added up. The BiFPN feature maps are first multiplied with a learnable weight before they are added up to give the network the ability to learn, which feature maps are more important than others.

3) Head: The head network consists of two separate subnetworks, see Fig. 5: the box prediction network, which predicts the bounding box coordinates and the class prediction network. Both networks consist of several convolutional layers, each followed by batch normalization [14] and a nonlinear activation, which is standard practice. The amount of convolutional layers in both networks depends on the scale of the network and can vary between three and six. All five of the feature maps from the BiFPN, generated at the final five output nodes, are fed into both the box and class prediction network, such that final predictions are made on feature maps of all available resolutions.

The number of bounding boxes that are predicted for each image follows the following equation and depends on the input resolution, which in turn is determined by the scale of EfficientDet.

$$num\_bbox = \sum_{i=1}^{5} \left(\frac{input\_size}{8 * i} \right)^2 * num\_anchors \quad (1)$$

For example, for EfficientDet-D2 (third smallest scale) 110,484 bounding boxes are predicted on each image, where the output per bounding box are a confidence score and the location, represented by the coordinates of the upper left corner, the width and the height of the bounding box. The loss used for the gradient descent consist of two parts: the binary cross entropy of the classification prediction and L1

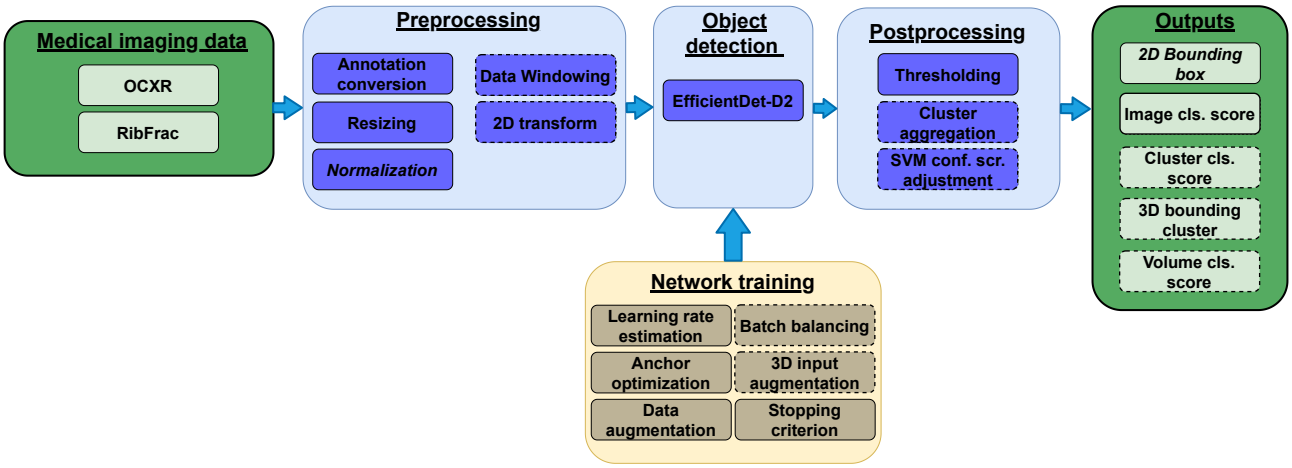

Fig. 4. Flowchart of the proposed analysis chain. Dashed boxes are only used for the RibFrac dataset, italics are only used for OCXR, other modules are used in both cases. The third smallest EfficientDet version is used (D2). Classification (cls) scores are predicted on three levels - image, cluster and volume and consist of a confidence score estimating the probability of an object being present.

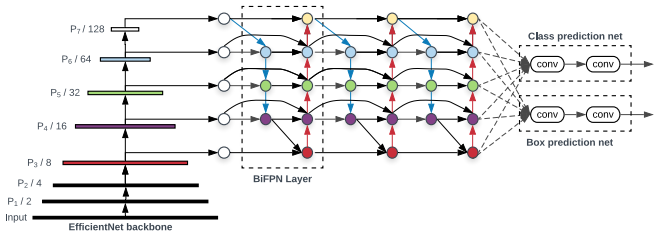

Fig. 5. Schematic of EfficientDet architecture consisting of the backbone (EfficientNet), the neck (bi-directional feature pyramid network) and the head module (class and box prediction net). Illustration reproduced from [30]

regression loss for the difference between the predicted and ground truth bounding box. Both losses are weighted equally.

B. Model architecture - Inception Network

Since the task for the RibFrac dataset is classification task - i.e. classifying images or complete scans containing rib fractures - we chose the state-of-the-art classification network InceptionNet as a benchmark for comparison. The Inception architecture quickly became popular after winning the ImageNet Large-Scale Visual Recognition Challenge 2014 [28] and is still one of the state-of-the-art neural network architectures for image classification. The hallmark of this architecture is the efficient computation of convolutional filters, which was obtained by reducing the number of channels with a 1x1 convolution beforehand combined with the feature combination at different scales. In this way, the number of larger computationally more expensive convolutional filters can be reduced. The 1x1 convolutions are followed by a non linear activation and therefore serve a double purpose. Another benefit of this module structure is that feature maps on different scales are produced by the filters of varying size and type (convolutions and max pooling).

In the third iteration of the architecture - Inception v3 [29] - the convolutional filters are further optimized by replacing them with 1xN and Nx1 convolutions in parallel or series depending on the filter size, which reduces the number of learnable parameters further.

Nine of these inception modules connected in series make up the core of the inception network, preceded by a shallow convolutional sub-network that serves as a first feature extractor and downsampler. At the end of the inception modules, the final predictions are made with a fully connected layer followed by softmax activation.

To counter the problem of vanishing gradient, which occurs in deep neural networks and makes them difficult to train, auxiliary network branches were added to the fourth and seventh inception module. These auxiliary branches consist of a pooling layer, a convolutional layer and two fully connected layers and predict a preliminary image classification. In this way, the loss gradient is injected at different stages of the network and can reach the earlier layers more easily.

C. Implementation

The basis of the implementation of EfficienDet used for this study is the GitHub repository 'Zylo117' [1], which ported the original developers implementation from TensorFlow to the PyTorch deep learning framework and achieves the same benchmark performance as the original implementation [1].

Generally, a larger version of EfficientDet achieves a higher performance at the cost of the inference speed. The performance difference is especially noticeable for the smaller versions, while for the larger versions the performance starts to saturate. For both the OCXR and the RibFrac dataset, a medium sized version of EfficientDet was used - EfficientDet-D2. For real-time applications, inference speed is important and EfficientDet-D2 is the largest version, which achieved real-time[1] inference on the COCO object detection benchmark. Furthermore, since the object detector will be applied to a 3D dataset, an efficient 2D baseline detector is crucial, because adding another dimension will increase the computational complexity.

[1]Real-time inference is here defined as 30 FPS with a Tesla V100 graphics card

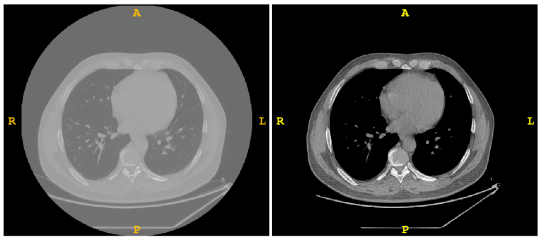

Fig. 6. Comparison of a transversal slice with and without windowing. Left: original image. Right: windowing applied to emphasize bone structures, leaving only values between -300 to 1000 HU.

1) Preprocessing: 2D transformation: Since EfficientDet was developed for 2D image analysis, the 3D RibFrac dataset was subdivided into a stack of 2D images. The most natural way to subdivide the scans is along the transversal axis, as it is also the axis along which image acquisition takes place.

Annotation conversion: Much like most current object detectors, EfficientDet was developed for rectangular bounding boxes. All other shapes of ground truth annotation such as ellipsoid and polygonal and pixel-wise annotations were converted into rectangular bounding boxes: the smallest rectangular bounding box, which encompasses the complete annotation with sides parallel to the image borders (i.e. no rotation). Furthermore, classes were merged together, such that the task became a binary classification task.

Normalization: In the case of the OCXR dataset, all images were normalized with respect to the global intensity statistics. Slice-wise, the global intensity mean was subtracted and the result was divided by the global intensity standard deviation.

Data windowing: In the case of the RibFrac dataset, the data was preprocessed by using an intensity window transform. The intensity resolution of CT scans can be very high, which results in a lot of detail in image parts that are not relevant for the task. For example, in this dataset, fractures can only be found on the ribs, therefore tissue details in the lung away from the ribs are unnecessary and could possible impede proper training. A bone window transform was applied which emphasises bone structures and deemphasises soft tissues. As can be seen in Fig. 6, on the right, the ribs are much easier to discern and less detail can be seen in areas where clearly no bones are located. All HU values lower than -300 and higher than 1000 are set to zero, values between this range are linearly mapped to the interval between 0 and 1.

Resizing: As the input size was fixed by the scale of EfficientDet, all images were resampled to a size of 768x768 using bilinear interpolation. In the case of the Inception architecture, the input size was fixed to 299x299 voxels. To allow for a fair comparison, each transversal slice is subdivided into four equally large quarters, which were then resized to the necessary resolution. Every quarter is analysed separately leading to a total image resolution of 598x598.

D. Network training

Before predicting the bounding boxes for the test set, the network had to be trained on the training data set. During inference, this pipeline block was not used.

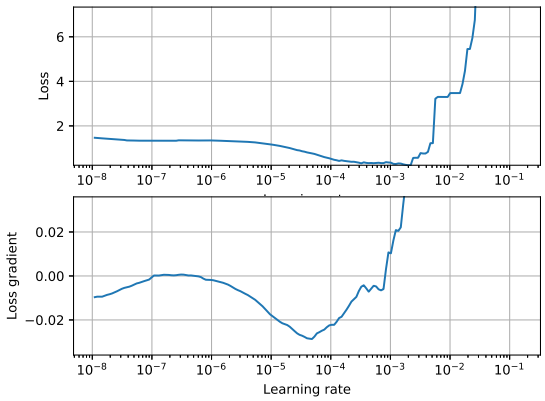

Fig. 7. Typical curve of the optimal learning rate algorithm. The optimum is found as the minimum of the loss gradient.

Pretraining: EfficientDet was pretrained on the Microsoft COCO dataset [19]. This dataset is a large dataset of natural images of common objects, such as dogs, cars and trees. While the medical domain has different pixel statistics than the domain of natural images of common objects, the pretraining will give a better starting point than a random initialization, since especially low level features like edges and corners are likely similar. InceptionNet was pretrained on the ImageNet dataset, which is a large dataset commonly used for image classification [6].

Learning rate estimation: An important parameter to tune is the learning rate. A too high learning rate can lead to overshooting optimal weight values, while a too low learning rate will lead to a slow training. The optimal learning rate can be estimated with the method of Smith [27]: first the learning rate is set to a very small value, for example 10^{-8}. Network training is started and after every mini batch, the learning rate is multiplied with 1.1, exponentially increasing the learning rate over time. The recorded loss against the learning rate will have a typical shape, see Fig. 7: at first the loss will barely decrease, since the learning rate is too small, next the decrease in loss will speed up, indicating a learning rate close to the optimum and finally steeply increase, indicating an overshoot. The estimated optimal learning rate is the learning rate for which the gradient of the loss is minimal - i.e. the inflection point. To reduce the influence of the weight initialization and batch selection, this procedure was repeated ten times and the average of the estimated optimal values was used. Both for estimating the optimal learning rate and actual training, the Adam optimizer was used.

Anchor optimization: In total, nine anchors were used with three different aspect ratios at three different scales. The optimal anchors were calculated with an evolutionary differential method [34].

Data augmentation: A common method to artificially increase the sample size of a dataset and improve the generalizability of the neural network is data augmentation [15]. Since especially the OCXR dataset is rather small, data aug-

mentations are a valuable tool. Data augmentations artificially increase the sample size by copying and then distorting the original image in certain ways. In the original implementation only one data augmentation method was used, whereby the image was mirrored along the vertical axis. For this study, five additional data augmentation methods were implemented: mirror, crop, intensity inversion, rotation, brightness change and blur.

Cropping was done by randomly selecting an image patch comprising at least 40 percent of the original image and stretching it back to the original resolution. During intensity inversion the current value of each pixel was subtracted from the maximal intensity value 255. Rotation was done by randomly rotating the image to 90°, 180° or 270°. For the brightness change, all intensity values of the an image were randomly multiplied by 0.8 or 1.2. The image was blurred with a Gaussian filter with a standard deviation of 2 pixels. With a probability of 25% a single randomly chosen data augmentation was used, in 75% the original image was fed into the network.

3D context: Since rib fractures extend over multiple transversal slices, there are some clear spatial dependencies between neighboring slices. To make use of these dependencies, the EfficientDet pipeline was adjusted. EfficientDet was built as an object detector for RGB images, thus using an input with three color channels. CT-images on the otherhand are greyscale images. The left over two color channels therefore can be filled in with neighboring transversal slices and provide the network with more spatial context. Note that bounding box predictions were still made only solely on the center slice.

Balanced batches: Within the RibFrac data, a large class imbalance is present. Only roughly every fourth transversal slice contains a rib fracture. Sampling batches randomly will lead to the network training on much more healthy images, which can impact networks ability to detect fractures. Therefore the probabilities to draw samples from each class are adjusted, such that probabilities to draw from each class are equal.

Stopping criterion: To minimize the risk of overtraining and maximize the generalizability of the model, two different stopping criteria were tested.

If overtraining during training was detected, the validation loss was calculated every 300 training iterations. If the validation loss did not reach a new minimum within 5 epochs or a maximum of 50 training epochs elapsed, the training was stopped and the performance was evaluated with the network weights achieving the lowest validation loss.

If no overtraining was detected, but loss convergence is seen on the validation set in a preliminary training run, the validation set was combined with the training set into a larger training set. The advantage of this method is that more samples are available for training, which can improve the network generalizability and therefore test performance.

E. Postprocessing

Thresholding: For the EfficientDet-D2 scale, 110,484 bounding box are predicted on each analysed image. Most of the bounding boxes do not encompass an object and ideally should have a low confidence score. To reduce the false positive rate, all predicted bounding boxes with a confidence score lower than 0.05 were discarded.

Cluster aggregation: To exploit the 3D spatial context of the data, the 2D bounding boxes were aggregated into 3D-clusters. For example, since the resolution along the transversal axis is high, relative to rib fracture sizes, rib fractures were extended over multiple transversal slices. By combining bounding box prediction from multiple slices, false positives could be detected and gaps could be filled in. Bounding boxes were grouped together if there was at least one pixel overlap between bounding boxes in the same or neighboring transversal slice. For an example, see Fig. 9. The colored lines are the individual bounding boxes, which are grouped together into clustered indicated by the same color.

SVM confidence score adjustment: From the aggregated 3D clusters, cluster statistics were calculated and used with a linear support vector machine (SVM) to generate adjusted confidence scores on the cluster level. The following seven statistics were used: cluster size, cluster extension along the transversal axis, maximum, mean and standard deviation of the bounding box's confidence scores and the standard deviation of the bounding box centers in x- and y-direction. Since the ground truth annotation of rib fractures are rather cube shaped than irregular "smeared out" clusters, we assumed that a low standard deviation of bounding box centers could be a valuable predictor to identify true positive clusters. The output of the SVM is a confidence score estimating the probability of a rib fracture being present in the image.

While neural networks have been shown to be superior to SVMs in solving many problems in the field of computer vision, we expected the SVM to be able to refine the results of the neural network, because information about the 3D context has been added in the cluster aggregation, which was not available to the network. Furthermore an SVM instead of an additional neural network was chosen, since the amount of data points is greatly reduced after transforming bounding boxes into clusters and neural network tend to perform best with large datasets, which can inhibit neural network training. Also linear SVM predictions are fast and computationally much simpler and therefore better comply with the computational complexity constraint of this study.

F. Performance metrics

FROC: For the OCXR dataset, the Free-Response Operating Characteristic (FROC) is used to measure the localization ability of the network. For this, all predicted bounding boxes were sorted according to their confidence score and evaluated whether the bounding box had an overlap with a ground truth bounding box. A predicted bounding box counted as correctly localized if the center of the predicted bounding box lied within the ground truth annotation. Next the cumulative true positive rate and false positive rate of the ordered bounding box list was calculated. At checkpoints of average false positive rates of 0.125, 0.25, 0.5, 1, 2, 4, 8 the sensitivity was calculated. The final FROC score was the average of the seven sensitivity scores.

AUC: The main performance metric was the area under the receiver operating curve (AUC) calculated on three different levels, with varying localization information: cluster, image, volume.

The probability score for an aggregated cluster is the output from the SVM. A common way to assess whether a bounding box accurately located an object is the intersection over union (IoU) [19] between the predicted bounding box and the ground truth annotation. A prediction is counted as correct, if a threshold is reached. In the case of images the intersection and union are calculated in pixels. This metric can easily be expanded to 3D by counting voxels instead of pixels for the intersection and union. This metric combines two aspects, namely how much of the ground truth object was found and how large the prediction is relative to the ground truth size. For example, a prediction that is smaller than the object but is completely contained by it can result in the same IoU as a large prediction that only covers a part of the ground truth. For clinical support, the first case is more useful, since it can more easily focus the attention on the pathology. Furthermore it seems that IoU values are difficult to estimate for human observers [26], which makes it not ideal for a support system for humans.

Hit metric - intersection: For this study, we instead simply used the fraction of the ground truth object that is covered by the prediction relative to the object's size with a threshold of 0.1. An easy exploit of this metric would be to increase the predicted cluster size, since the metric is blind to no overlapping voxels. However, during network training the the L1 loss between predicted and ground truth bounding box was used as loss and therefore overly large bounding boxes were penalized. Additionally, to ensure that the cluster size was not the driving factor behind a good performance, the average predicted and the ground truth object are reported. To be of use in a clinical setting, not only the AUC is important but also an actual operating point has to be selected. With a too high false positive rate the algorithm becomes distracting rather than supportive. For this study the fraction of found objects are reported at an ope srating point of five false positives per volume.

The inception architecture has a designated output to indicate the estimated probability of an object being present per transversal slice. EfficientDet on the other hand produces only bounding box predicitions. To estimate the probability of an object being present in the image, the bounding box probability scores within an image were used. The highest probability score of all bounding boxes were assigned as image probability score. If no bounding box was predicted on the image, zero was assigned.

The volume score is derived from either the cluster or the image score. In the case of EfficientDet, the volume score is the highest probability of a cluster that a volume contains. In the case of InceptionNet, the highest probability score of the images contained in the volume is used. Another metric to evaluate how well fractures in transversal slices per volume were found is the correlation coefficient between the probability score per image and the binary ground truth per image. This is a more fine grained metric, which takes predictions of every image into account instead of instead of only cluster prediction scores.

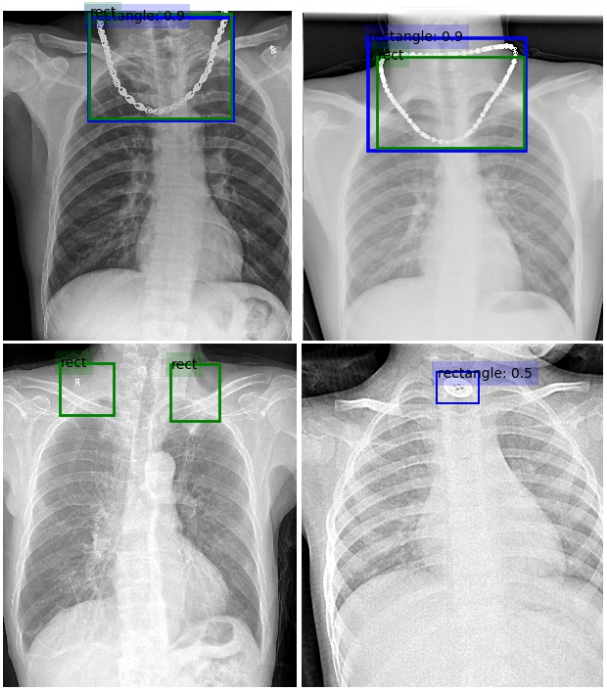

Fig. 8. Correct and incorrect detection. Predictions in blue, ground truth in green. The confidence score is shown above the predicted box.

IV. Experiments & Results

In this section, the experiments and results of the analysis on both the OCXR and the RibFrac dataset are presented.

A. OCXR

The experiment performed on the OCXR dataset used EfficientDet-D2 to investigate how easily an object detector developed for common object in color image can be used on X-ray images to find foreign objects. Since rib fractures are smaller and blend more easily into the background and are therefore harder to find, a good performance on the OCXR is necessary for EfficientDet to be considered as a potential candidate for rib fracture detection. Following the learning rate estimation protocol from section III-D, the learning rate was set to 10^{-3}. The empirically found anchor sizes were 1:1, 1.4:0.7 and 0.7:1.4 with sizes 0.660, 1.047 and 1.662. During training overtraining was noticed, therefore the validation dataset was used to find the optimal moment to stop training, which was reached after 46 epochs.

The training resulted in an AUC of 0.931 and an FROC of 0.762, which put EfficientDet's performance in the middle of the challenge leaderboard. Several example predictions are shown in Fig. 8.

B. RibFrac

A baseline experiment and three additional experiments were performed on the RibFrac dataset, to investigate the effect of certain aspects of the training procedure. For the

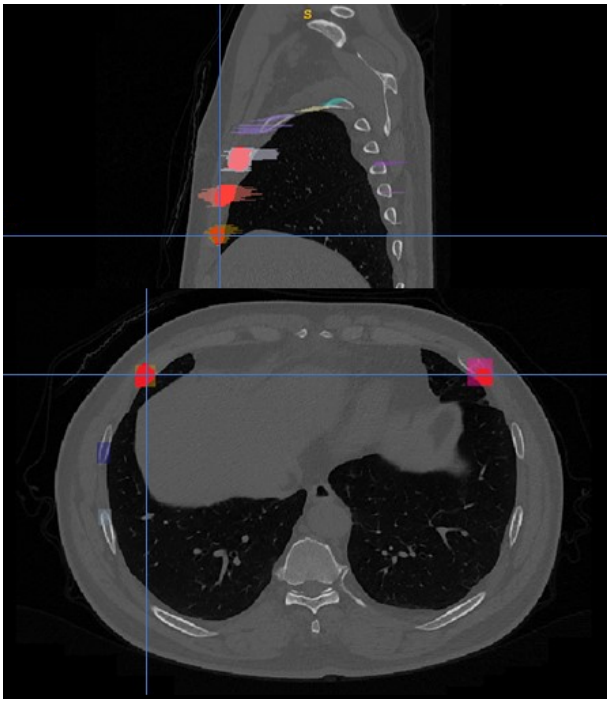

Fig. 9. Aggregation of individual bounding boxes into 3D clusters. Top - sagittal plane, bottom - transversal plane. Ground truth rib fractures are shown in bright red, other colors represent predictions.

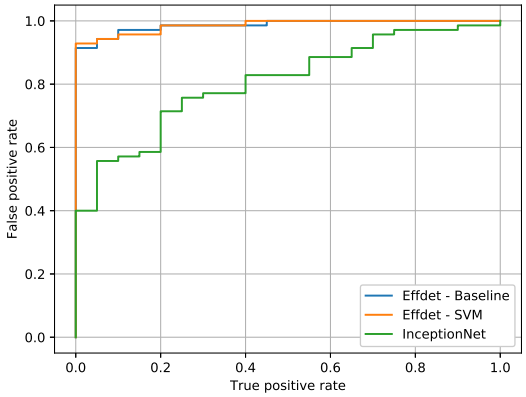

Fig. 10. AUC comparison for the baseline experiments for EfficientDet with (orange) and without SVM (blue) and InceptionNet (green).

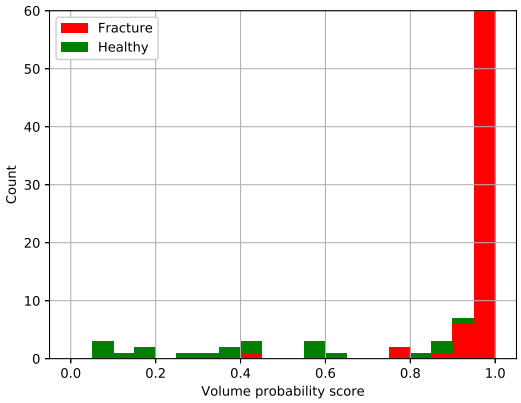

Fig. 11. Histogram of probability scores for volumes based on SVM predictions

baseline experiment, the general methodology from Fig. 4 was followed with the exception of batch balancing and 3D input augmentation. Furthermore the optimal early stopping point was found using a validation dataset. The additional experiment investigated those three factors (batch balancing, 3D input augmentation and combining the training and validation set together and stop training after a preset amount of epochs).

The estimated optimal learning rate for all experiments done with EfficientDet was found to be 10^{-3} coinciding with the learning rate for the OCXR dataset. The empirically found anchor sizes were 1:1, 1.4:0.6, 0.6:1.4 and 0.7:1.4 with sizes 0.4, 0.498, 0.635. As expected the optimal anchor sizes for the RibFrac dataset are smaller then for the OCXR dataset. The batch size during training was set to 32.

Similar experiments were performed with InceptionNet on the RibFrac dataset, the validation loss did not converge but resulted in overtraining for longer training runs. Since without a validation set, the optimal training point before overtraining occurs could not be found, this experiment was skipped for InceptionNet. The learning rate was found to be $5x10^{-5}$ and the batch size was set to 16.

The results can be seen in Table I. For EfficientDet, two values are given: first the result using an SVM during post-processing, second without. All metrics are averaged over all three crossfolds. Fig. 10 shows a comparison between the ROC curves for the volume predictions of the baseline experiment for EfficientDet (with and without SVM) and InceptionNet.

An example output of aggregated bounding boxes can be seen in Fig. 9.

Figure 11 and 12 shows a histogram over the volume probability scores with and without using an SVM respectively of one crossfold of the baseline experiment. Scans containing a fracture are displayed in red, stacked on top are healthy scans displayed in green.

Fig. 13 displays EfficientDet's performance on a cluster level. The five curves show how many fractures of a particular class were found relative to the class occurrence, depending on the probability score threshold on a cluster level. Decreasing the threshold will naturally lead to more fracture being found on the cost of a higher false positive rate. The dashed vertical line represents the operating point of five false positive per volume.

V. Discussion

A. OCXR

The results of the OCXR dataset are promising. Comparing the results to the official challenge leaderboard, EfficientDet has with an AUC score of 0.931 the 26th highest AUC (top placement: 0.963) and with an FROC score 0.760 the 27th highest FROC (top placement: 0.852) out of 40 contestants. Considering the computational complexity constraint, it shows

Network	AUC vol.	AUC img.	AUC clust.	obj. found	rel. cluster size	corr. coef.
Baseline EffDet	0.967 / 0.976	0.902	0.952 / 0.944	0.762 / 0.761	1.650 / 1.633	0.667
Baseline IncNet	0.810	0.800	-	-	-	0.436
3D-input EffDet	0.746 / 0.873	0.836	0.931 / 0,929	0.765 / 0.770	5.718 / 5.696	0.527
3D-input IncNet	0.715	0.763	-	-	-	0.396
Balanced EffDet	0.888 / 0.937	0.880	0,957 / 0,954	0.764 / 0.773	2.294 / 2.195	0.615
Balanced IncNet	0.678	0.762	-	-	-	0.509
Large EffDet	0.937 / 0.956	0,0.885	0.934 / 0.923	0.738 / 0.734	2.275 / 2.25	0.660

TABLE I

EXPERIMENTAL RESULTS. WHERE APPLICABLE, THE FIRST NUMBER SHOWS THE RESULT WITH SVM ADJUSTMENT, THE SECOND WITHOUT. AUC WAS MEASURED ON THREE LEVELS: VOLUME, IMAGE, CLUSTER. RELATIVE AMOUNT OF FRACTURES FOUND (OBJ. FOUND) WAS CALCULATED AT AN OPERATING POINT OF ON FIVE FALSE POSITVE PER SCAN ON AVERAGE. RELATIVE CLUSTER SIZE IS REPORTED TO SHOW THAT THE INTERSECTION HIT METRIC WAS NOT ABUSED.

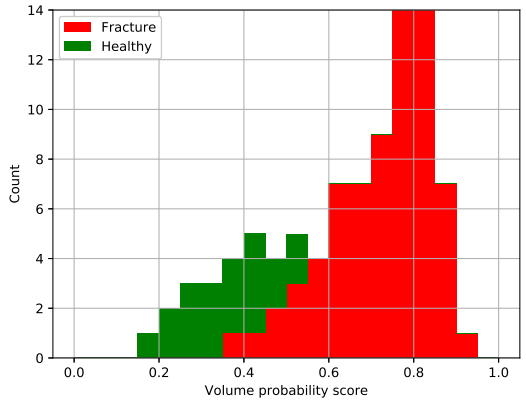

Fig. 12. Histogram of probability scores for volumes without using SVM predictions.

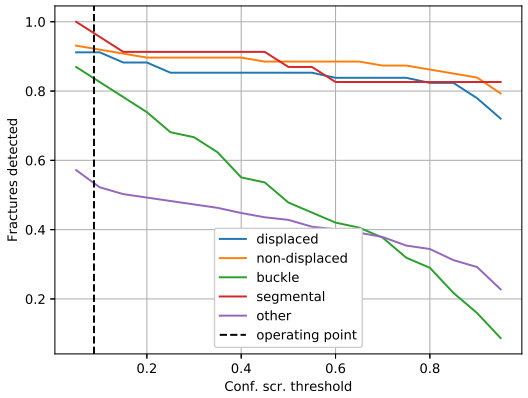

Fig. 13. Fraction of fractures correctly located depending on probability score threshold for the baseline experiment.

that EfficientDet is a promising candidate for object detection in CT scans.

During the error analysis, it became apparent that there are significant inconsistencies in the provided annotations impacting the performance. For examples, chains or buttons (bottom right image of Fig. 8) or clips close to the spine are not always annotated as foreign objects in the ground truth labels, even if they are clearly located within the lung field. Furthermore, in some cases only the lower part of a necklace is annotated in the ground truth labels and in other cases the complete necklace, including the parts that reach out of the lung field. Many of the false positive predictions with a confidence score above 0.3, belong to those inconsistent cases.

B. RibFrac

The current study aimed to use object detector neural networks to efficiently and reliably find objects on medical images. The performed experiments and results show that the object detector EfficientDet is a viable candidate for image pathology classification.

In this section, the result concerning the RibFrac dataset are discussed focussing on three topics: comparing the performance of EfficientDet and InceptionNet, analysing the impact of using an SVM to adjust the cluster classification scores and explanations are discussed on why the additional experiments were not able to improve baseline performance.

EfficientDet vs InceptionNet: Table I shows that EfficientDet achieves a significantly better performance than InceptionNet in all experiments and on all performance metrics. Even though EfficientDet contains, with roughly 8 million trainable parameters, only a third of the roughly InceptionNet's 24 million trainable parameters, EfficientDet is able to better identify 3D scans and 2D transversal slices containing rib fractures, showing the versatility of an object detector used as a binary classifier.

The small size of the rib fractures likely play a significant role in the performance difference. For example, in the widely used ImageNet databased, the class defining part of the image is relatively large, i.e. an image of the class dog, has a dog covering half of the image or more, meaning that a large part of the image contains important information. In the RibFrac dataset the opposite is true, rib fractures are small relative to the image size and therefore most of the image parts carry little information about the image class. InceptionNet only has one output to estimate the class of the entire image. EfficientDet on the other hand has roughly 100,000 outputs per image, where different outputs make predictions about different image parts. Therefore a much more detailed signal is used for each training iteration, which might lead to a better pattern extraction.

Furthermore, with the aggregated clusters, more local information is available using EfficientDet. With InceptionNet the smallest region on which predictions are made, are image quadrants, which represents an area of roughly 65,000 voxels of a scan. For EfficientDet the smallest region are clusters,

which in the baseline experiment are only roughly 10,000 voxels on average. In a medical setting, interpretability and explainability provided by clearer localization due to aggregated clusters are of utmost importance to promote trust in the human-machine collaboration.

We found that with increasing localization information, the classification accuracy decreases. This methodology could therefore be used in a top-down fashion. First, the scan as a whole would be evaluated, for which the accuracy is the highest to give a quick, but confident estimation on the nature of the complete scan. As a second step, localization information could be added on an image level, which could finally be refined with a cluster analysis. This hierarchy would allow for adding subsequently more localization information, while the diagnosis is in process. The same analysis used to evaluate an enitre scan could also be used on 3D parts of the scan.

Fig. 13 shows the classification performance per class. Displaced, non-displaced and segmental fractures were detected with greatest confidence. Around 90% of fractures belonging to those classes could be detected over a wide range of probability score thresholds. Those fractures are the easiest to recognize even for non-specialized since there is a clear gap visible in the bone. Buckle fractures are categorized by deformed bones without necessarily a clear break. Together with the 'other' class, those kind of fractures had a significantly lower detection rate. In total 0.76 of all fractures could be detected, while allowing no more than an average of five false positives per volume.

Confidence score adjustment with SVM: The SVM confidence score adjustment was not able to increase the AUC score significantly on any of the three scales. However, the probability score distribution on a volume level shows was drastically altered and can give new insights.

Fig. 11 shows the effect of the SVM confidence score adjustment, which pushes both classes to their respective end of the spectrum. Especially scans containing fractures have a much higher confidence score. With a conservative threshold of 0.7, all volumes containing fractures were classified correctly with the exception of one, at the cost of only four false positive. Healthy scans on the lower end were also better separated. Predictions score on either end of the spectrum can rather safely be trusted, while scores between 0.4 and 0.7 are less reliable and need additional analysis, for example on a cluster or image level or extra attention of a medical practitioner.

Of the seven properties used for the SVM training, the maximum, mean and the standard deviation of confidence scores from contained bounding boxes were the driving factors, followed by the cluster extension along the transversal axis. The confidence score statistics carried more information than the size and shape of the aggregated cluster. The transversal extensions or cluster thickness likely plays a role in filtering out spurious thin false positives, of which a lot occurred.

The disadvantage of using an SVM is the increased inference time. While applying an SVM is not very time intensive, the SVM operates on the cluster level and clusters first have to be aggregated. The aggregation time will scale linearly with the amount of predicted bounding boxes and the size of the volume.

Additional experiments: The three additional experiments were not able to improve the baseline performance. Using neighboring transversal slices to add more spatial context to the input resulted in more and much larger clusters. But even with larger clusters the AUC on all levels dropped. A reason might be that the scale of EfficientDet was too small and therefore was not complex enough to capture the relevant patterns. A similar method of 3D-enhancing the input images was also used by winners of the RibFrac challenge, who used a more computationally complex neural network [20]. Future research could investigate if other methods of combining information from neighboring slices could increase localization precision. For example, neighboring slices could be processed individually by the backbone network in parallel to combine feature maps at the neck network. To avoid a significant increase in computational complexity, the backbone network could share weights or the amount of channels could be reduced. Noticeably, false positive clusters are located on ribs, so the network seemed to be able to locate the possibly relevant image parts, namely the ribs, but was not able to find rib fractures more often.

The experiments with balanced batches also lead to a small decrease in performance as measured by the various AUC evaluations. This might be caused by the way that the performance metrics were calculated. The most important factor in determining the image, cluster or volume probability score was the maximum confidence score of contained bounding boxes. This metric is susceptible to a single high confidence false positive. Since there are usually many predicted bounding boxes in one image, cluster or volume, false positive suppression is more important than missing a true positive with one bounding box. By balancing the training batches, the network sees less images with no fractures, which might impair the networks ability recognize healthy scans.

Lastly, combining the validation and training set into a larger training set did not improve the performance. Since the validation was very small compared to the training set ($<10\%$), a large performance difference was not expected. While the validation loss curve did converge, it did so in a noisy fashion, which is common for gradient descend algorithms. From the results we conclude that the validation set has more value as a stopping criterion to find the best stopping point in the noisy gradient descend rather than by increasing the training set.

A direct comparison between the RibFrac challenge contestants and EfficientDet is not possible, since the winning architectures have not been made public yet. Only metrics not used in this study are reported and only short descriptions of the used methodology are currently available. From the reported metrics, one can infer that the challenge winners would achieve a higher performance using the metrics from this study. This assumed performance increase however comes at the cost of computational complexity. For example the challenge winner [33] used a cascade of three large neural network architectures: an object detector and a classifier in conjunction with a U-net type segmentation network.

If the task is not time critical, EfficientDet could be used as a first stage in an more complex pipeline. Since EfficientDet's false positive predictions are not randomly distributed over the entire image but focus on rib bones, the presented pipeline could be used as region proposals for further analysis.

VI. Conclusions

In this study, we presented a complete data analysis pipeline for analysing 3D CT scans, tested on two real-world medical datasets, showing that object detectors developed for common objects are promising candidates for object detection in medical 2D and 3D data. The discussed method was able to provide classification and localization information on three different scales and achieve a better performance than the state-of-the-art neural network classification benchmark InceptionNet. We also presented a new metric to evaluate object detection in a medical setting for decision support, focusing merely on the intersection of ground truth and prediction instead of the less intuitive IoU, which makes the performance evaluation easier to understand and interpret.

In conclusion EfficientDet is a promising lightweight architecture, prioritizing speed while still achieving good classification results. It is generic enough that it can be applied successfully to different datasets with drastically different objects to be detected.

By prioritizing speed and computational simplicity, the localization accuracy is limited and cannot compete with the MICCAI2020 RibFrac challenge winners. With the constant increase of computing power over time a complexity constraint could be considered only a delaying factor. However, since hospitals tend to lag behind in technological advancements and usually have limited resources, focussing on lightweight machine learning algorithms will likely stay relevant in the near future.

References

[1] "EfficientDet implemented in PyTorch," https://github.com/zylo117/Yet-Another-EfficientDet-Pytorch, accessed: 2020-08-01.

[2] "Website - Object chest X-ray," https://jfhealthcare.github.io/object-CXR/, accessed: 2020-10-07.

[3] "Website - MICCAI 2020 RibFrac Challenge: Rib Fracture Detection and Classification," https://ribfrac.grand-challenge.org, 2020.

[4] A. Bochkosvkiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[5] D. Cireşan and U. Meier, "Multi-column deep neural networks for offline handwritten Chinese character classification," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.

[7] B. Ehteshami Bejnordi, M. Veta, P. Johannes van Diest, B. van Ginneken, N. Karssemeijer, G. Litjens, and J. A. W. M. van der Laak, "Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer," *JAMA*, vol. 318, pp. 2199–2210, 2017. [Online]. Available: https://doi.org/10.1001/jama.2017.14585

[8] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR Proc.*, 2016.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[11] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[12] S. Huber-Wagner, R. Lefering, L.-M. Qvick, M. Körner, M. V. Kay, K.-J. Pfeifer, M. Reiser, W. Mutschler, K.-G. Kanz, W. G. on Polytrauma of the German Trauma Society *et al.*, "Effect of whole-body ct during trauma resuscitation on survival: a retrospective, multicentre study," *The Lancet*, 2009.

[13] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: AlexNet-level accuracy with 50x fewer parameters and 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2015.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[15] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, 2019. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2019.2939201

[16] L. Jin, J. Yang, K. Kuang, B. Ni, Y. Gao, Y. Sun, P. Gao, W. Ma, M. Tan, H. Kang, J. Chen, and M. Li, "Deep-learning-assisted detection and segmentation of rib fractures from CT scans: Development and validation of FracNet," *EBioMedicine*, 2020.

[17] A. Kesner, R. Laforest, R. Otazo, K. Jennifer, and T. Pan, "Medical imaging data in the digital innovation age," *Medical physics*, 2018.

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

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll'ar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[20] P. Liu, "Video stream:[MICCAI'20 RibFrac Challenge] Detection 2nd by Pengfei Liu," https://www.bilibili.com/video/BV1rk4y1C7zg/, 2020.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *European conference on computer vision*, 2016.

[22] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," *Proceedings of the aaai conference on artificial intelligence*, vol. 3, pp. 4780–4789, 2019.

[23] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, pp. 91–99, 2015.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[26] O. Russakovsky, L.-J. Li, and L. Fei-Fei, "Best of both worlds: human-machine collaboration for object annotation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[27] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. of the CVPR*, 2015.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR Proc.*, 2016.

[30] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," *CVPR Proc.*, 2020.

[31] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019.

[32] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-text: Dataset and benchmark for text detection and recognition in natural images," *arXiv preprint arXiv:1601.07140*, 2016.

[33] E. Youjun, "Video stream: [MICCAI'20 RibFrac Challenge] Detection 1st by Youjun E," https://www.bilibili.com/video/BV17a4y1L75z/, 2020.

[34] M. Zlocha, Q. Dou, and B. Glocker, "Improving RetinaNet for CT Lesion Detection with Dense Masks from Weak RECIST Labels," *arXiv preprint arXiv:1906.02283*, 2019.