Deep Learning-based Classification of Musculoskeletal Radiographs - Learn to Zoom from Network Response

Stephan Huschauer
InIT
ZHAW
Zurich, Switzerland
huschste@students.zhaw.ch

***Abstract*—We present a method to identify and focus on a finite, selected region of interest in a heterogeneous set of images based on network activity allowing to crop this spot without extensive preprocessing and without training a region proposal network. The method was tested on musculoskeletal radiographs obtained from the Stanford University MURA challenge [7].**

***Index Terms*—CNN activity, region of interest, high resolution image classification, medical image processing, pattern recognition.**

I. Introduction

The challenge of recognizing and locating a certain object within an image is characterized by finding an algorithm that reliably detects an object whilst having the presence of the object also classified. Hence we propose a novel method to select regions of interest from (high resolution) x-ray images using the activation of a trained network and choosing VGG19 as a baseline. The region of interest or object to detect for, are implants in elbow, finger, forearm, hand, humerus, shoulder, and wrist. The different studies, 14‘656 in total, were labelled either positive (so they show an abnormality) or negative (they don’t show an abnormality). Here, in this paper and context, abnormalities are implants such as bridges, pins, plates or others as well as broken bones and represent the object of interest. The labelling, i.e. the assignment to either positive or negative, was done manually. No pixel masks or bounding boxes are given by the annotation. The data is provided as an open-source training and test set. A validation set is kept by the Stanford University for the purpose of evaluation and ranking of the published solutions from the contestants. The overall goal of the contest is to look for a computational algorithm that reliably identifies and classifies x-ray images in direct competition to experienced human radiologists. At the time of publication of this work, the human radiologists are already beaten. All of the top ranked approaches are using ensemble based methods.

The main goal of this work is to develop a basis for a new, semi- or unsupervised approach to zoom into high resolution images. Participation in the MURA competition is given less importance than the exploration and testing of new methods and techniques for solving the described problem.

II. Related Work

A. MURA Baseline

The MURA Baseline is based on DenseNet169. The dataset consists of x-ray images [4]. The quality of the images in respect to resolution, exposition, and object orientation, varies tremendously. This marks the limit for any method used in the classification and/or detection of region proposals. Additionally, ”identification marks” (flags) used on x-ray images either as Left-Right position identifiers or displaying patient identification letters, also show to disturb classification and detection of targeted objects (see also section IV-B2 and Fig. 5, 6).

B. Mask-R-CNN

Mask-R-CNN is a state-of-the-art instance semantic segmentation method [2]. We apply the implementation ”maskrcnn-benchmark” described in [3] which is more powerful than the original Detectron described by He et al. For example, Fu et al. [5] is using it as detector for predicting segmentation masks in single-shot manner like detectors as YOLO [6] do. In [5] a overall training-time of 40-50 hrs is reported when using a server with 4 GTX1080Ti graphic cards for training on the COCO 2014 data [11]. This framework also supports Faster-RCNN, where only bounding boxes instead of pixelwise classification (instance semantic segmentation) is provided. Due to the small amount of training data, the training in our case was running for only approximately 12 hours when using Faster-RCNN only. As backbone the sleek ResNet-50-FPN was used, in respect to the small amount of annotated data. The model could be used to identify the disturbing ”identification marks” for removing them from the image at a later stage. For the training of the RCNN, the

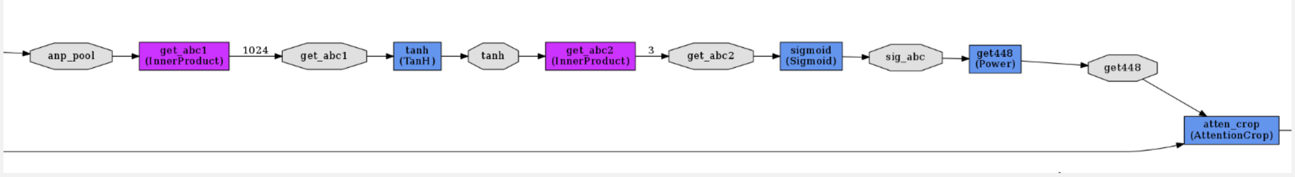


Fig. 1: RA-CNN Attention Proposal Sub network. The attention crop layer is custom designed and not part of the Caffe [13] distribution.

bounding boxes of ”flags” and implants were annotated in approximately 6000 images and prepared as coco-style [11] annotations.

C. RA-CNN

The 2017 CVPR publication ”Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition” [1] provides classification of high resolution images. Based on a learned attention, the model zooms in the original full size image and crops the selected region. The classification is then executed once again on this newly selected image part: In total 3 of such consecutive runs of selection-classification were executed and the result of the classifications from all 3 levels are combined. This approach is very similar to the one we are using but differs in the measurement of the attention. In our makeup the attention is calculated from the network activations, whereas in RA-CNN the attention is learned in a semi-supervised manner instead and is called Attention-Proposal-Subnetwork (APN, refer to Fig. **??**). The reported gain in accuracy lays between 3.3% and 3.8%.

D. Feature extraction

For comparison the popular transfer learning approach of feature extraction is considered and tested. The pretrained network will be trained for a few iterations on the new dataset. After this step, the classificator will be replaced by a SVM. Purpose of using the SVM: The SVM is based on a convex optimization and therefore always finds the global minimum. If the data is not separable in a given space, perhaps in a higher dimensional space, however, the data is separable nevertheless. If this is the case, only the inner product of this space needs to be known. This technique is often used when the amount of labelled data is small, for instance in [12]. Further details are explained in section IV-B2.

III. Methods used

A. Preprocessing

The images were preprocessed before feeding them into a network. The images were resized and cropped to fit to the input size of the network and they were normalized to obtain numerical more stable results. When ”guided back propagation” (GBP, [9]) is used, the normalization differs in order to prevent vanishing of gradients due to this step.

Normalization for GBP:

$$img_{normalized} = 0.5 + 0.5 * \frac{(img - mean(img))}{std(img)} \tag{1}$$

where:

$$mean(x) := \frac{1}{N}\sum_{i}^{N} x_i \tag{2}$$

and:

$$std(x) := \sqrt{mean((x - mean(x))^2)} \tag{3}$$

B. Baseline ZHAW

The baseline developed at the ZHAW for the MURA challenge:

- VGG19: For a baseline, we use VGG19, a convolutional neural network that is 19 layers deep and has the strength to classify images into 1000 object categories. The default image input size for this model is 224x224.
- Trained on MURA: We use the pre trained VGG19 and modify only the last layer since we only have to classify 2 classes (positive and negative). In Table I the official ranking in the MURA challenge is listed [7]. Tweaking the network with preprocessing the images by shrinking/expanding the intensity exposure, a small gain in the kappa-cohen score could be achieved (also shown in Table I).

The kappa-cohen score is given by the formula (the score is symmetric, so p_o and p_e can be swapped without changing the score, $\kappa = 1.0$ is the theoretical best score):

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{4}$$

TABLE I: MURA official Performance

Method Used	Performance Measure	
	Rank	***κ Score***
VGG19	23	0.744
VGG19 + Intensity Rescaling	18	0.754

C. Zoom Mechanism

The sequence for the process reads (see Fig. 2 Zoom Train Schemata):

a) Scale image to network input size, normalize and predict image (forward-pass for classification).

b) Use Activations to see ”where networks looks”, the method ”guided back propagation” (GBP) is used.

c) Compute Entropy:

$$entropy := \sum_i p_i * log_2(p_i) \tag{5}$$

Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image.

d) Use threshold to select region of interest in the original image and crop (refer to Fig. 4). If the proposed region is smaller than the network input size (here 224x224 used), the region will be extended to match.

e) Crop the selected region out of the high resolution image.

f) If terminal step: predict the ”new” image that was cropped before and return final classification result, otherwise continue at (a)

This process is used once (terminate at (f)); for larger-scale images, this process will be executed recurrently.

IV. Experiments

A. Hardware Selection

The experiments were executed using a Titan Xp graphic card with 12GB of memory in single or dual configuration, a GTX1080Ti graphic card with 11GB, and a RTX2080Ti graphic card with 11GB memory also in single or dual configuration (via NVLINK). The latter card has some known quality issues and one of them failed within a few weeks. For the continuation of the experiments, no homologous substitute was available due to delivery delays and shortcomings on the market.

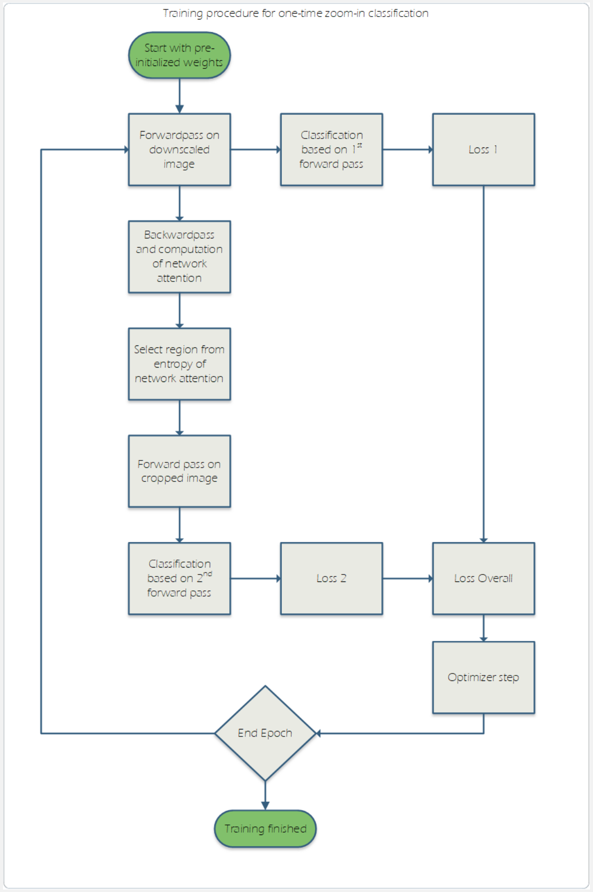

Training procedure for one-time zoom-in classification
Start with pre-initialized weights
Forwardpass on downscaled image
Classification based on 1st forward pass
Loss 1
Backwardpass and computation of network attention
Select region from entropy of network attention
Forward pass on cropped image
Classification based on 2nd forward pass
Loss 2
Loss Overall
Optimizer step
End Epoch
Training finished

Fig. 2: Zoom Train Schemata: The execution flow of the training sequence used in the experiments. The weights are initialized with the trained VGG19 baseline. The first forward pass generates a loss on the output, and the attention from the GBP generates under application of the local entropy a region to crop. The cropped region is used to do another forward pass and output a second loss. Both losses[1]were used to update the weights of the network.

B. Proposed Method

1) First evaluation: First experiments with the proposed method show, that the attention of the network is often focused on the ”flags” in the x-ray images rather than on the objects of interest. These ”flags” are used to identify left and right, and sometimes also display/contain patient identification letters. In Fig. 4 some responses and the corresponding entropy based proposal of a cropping region are shown. In Fig. 3 the comparison of grad-cam [8] and guided back propagation is illustrated. Grad-cam is the state-of-the-art method to visualize network attention by its responses, it is based on GBP that is used in our method.

2) Flag removal: To remove the ”flags”, some of the images were annotated manually (refer to Fig. 6 for examples

[1] Due to a memory leak when using pytorch 1.0 for the implementation, the backward pass of both losses was realized separately.

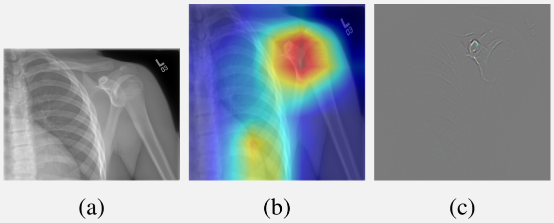
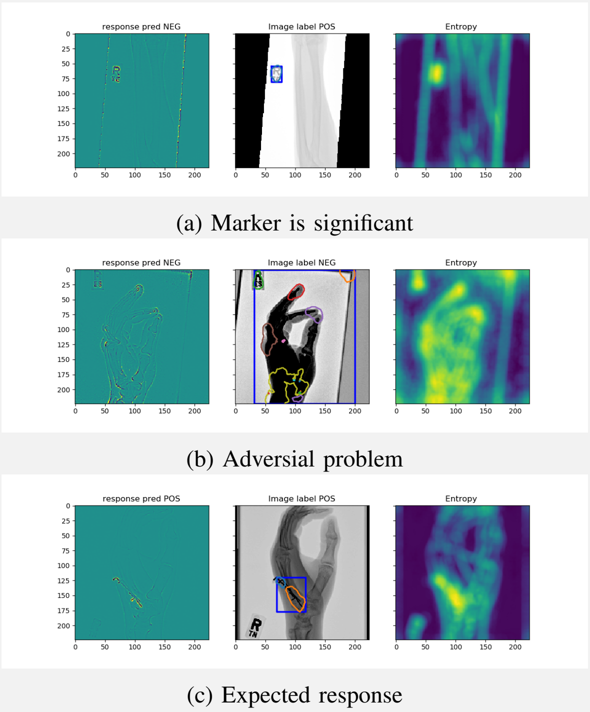


Fig. 3: Attention of ResNet-152 Network with (a) input image (not cropped to 224x224), (b) guided grad-cam, (c) guided back propagation.

of hand-annotated bounding boxes) to obtain a dataset for the training of a detector. Classical pattern matching methods were not applicable because the ”flags” differ in size, angle, and composition of letters and character style, therefore exceeding the applicability of these methods (refer to Fig. 6 that displays examples of extracted flags). However, a first attempt with mask-RCNN shows that implants were often, falsely, detected as ”flags”. Additional implementation of implant annotations improved the result, but failed to completely solve the issue with the misclassification. Applying Faster-RCNN instead of Mask-RCNN improved the detection because no pixel masks become annotated. (Remark: Training Mask-RCNN using bounding boxes only, showed even better results than expected). For rectangular ”flag” compositions however, this annotation seemed to be sufficient. Contrary, for implants the bounding box is not sufficient, but interestingly, after some iterations the predicted mask was closer to the ground truth than expected. The attempt of using segmentation instead of detection is restricted to the problematic of ”flag” positioning in the x-ray image. The samples in Fig. 5 clearly demonstrate the problematic of a robust detection and also illustrate the problem of marker positioning when a proper detection given by a bounding box exists. Issues with detection base upon (a) false detection of implants as ”flag” (marker on x-ray image) or (b) a problematic positioning of the flag in relation to the object that may cause wrong or bad flag removal as shown in pictures (c) and (f). Pictures (d) and (e) demonstrate a correct detection of ”flag” (marker) and implant. Already the processing of only a small set of images showed that blanking of detected bounding boxes does not result in the proper removal of ”flags”. Instead, it will be necessary to have pixel-mask and the blanking needs being replaced by a filling with the mean value of the surrounding to overcome this drawback. Thus a method which does not need removal of the ”flags” is strongly desired.

3) Learning from response: Fig. 2 shows an approach with training directly on the cropped region proposed by the attention. The time per batch is very high, since the prototype uses a lot of computation on the host (CPU). The training was not converging, neither with ADAM nor with SGD optimizer. The underlying idea is to force the network to ”focus” on regions that are representative for the classification problem. But the realization used seems to be far too naive and does not work. An interesting observation is that, even if the prediction is correct, no clear attention region is present, comparable with the result of an adversial attack.



(a) Marker is significant

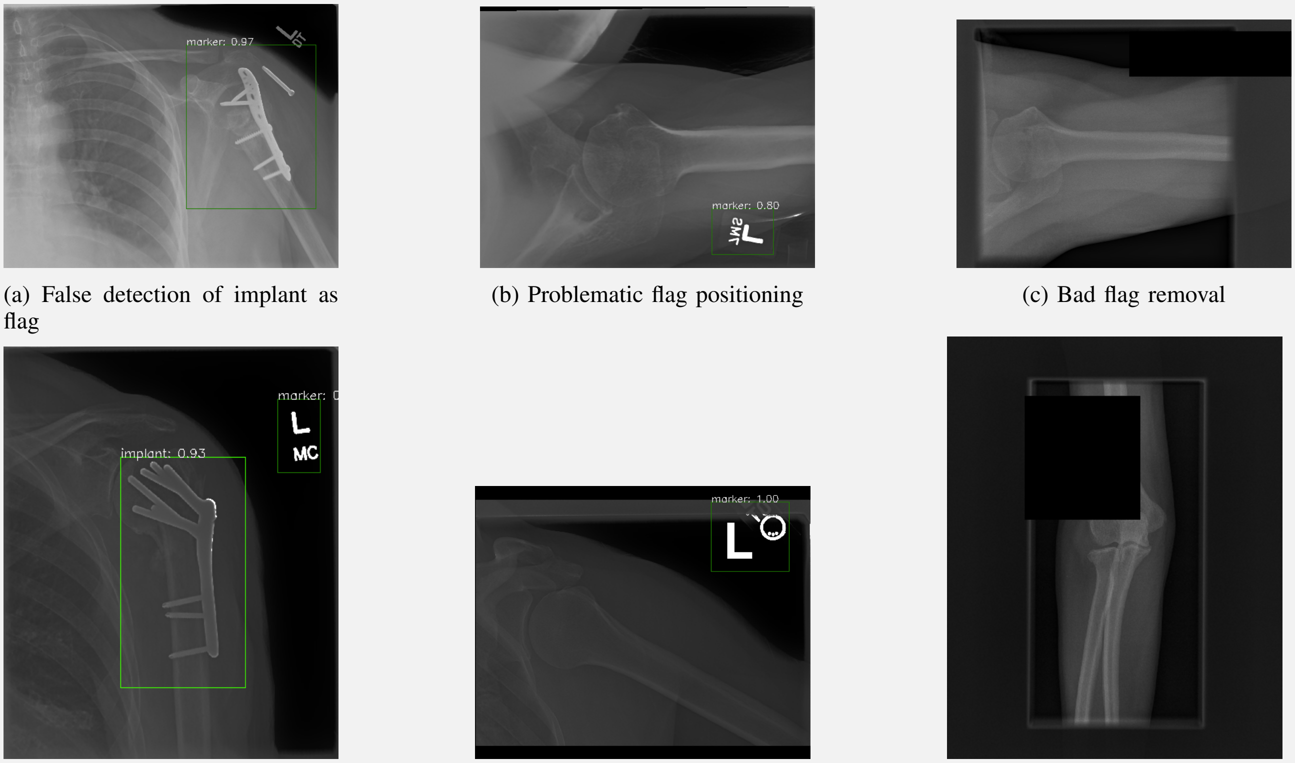
(b) Adversial problem

(c) Expected response

Fig. 4: Attention Crop Region Selection: The Figures illustrate the process of the proposed method. On the left of each sub figure the GBP is shown, in the middle the original image with over layered crop region in blue (the other contours representing candidates), and on the right the measured entropy of the BGP output.

C. Feature extraction

For comparison the popular transfer learning approach of feature extraction is considered and tested. The features come from ResNet-152 and are pre-trained for a few iterations on the MURA dataset. The features are extracted from the network and used by a SVM. A peculiar study was run for the different objects such as elbow, finger, forearm, hand, humerus, shoulder, and wrist. Since the SVM needs to see all data at once, this attempt caused the workstation in use (128GB Memory) running out of memory. To bypass this technical limitation, the training was speed-up by having the SVM used for each subset using a bagged classifier of 10 SVMs. The bagged classifiers were used in a one-vs-rest manner for prediction. This solution allows to train with the data in parallel. As kernel a polynomial kernel of degree 5 was used. The overall results on the MURA test set is given in Table II. Please note that not the same performance on the κ



(a) False detection of implant as flag

(b) Problematic flag positioning

(c) Bad flag removal

(d) Correct detection of flag and implant

(e) Correct detection of flag

(f) Bad flag removal

Fig. 5: Faster-RCNN detection results: (a) shows a false detection of a flag, this case is very crucial in case of blanking the flags bounding box, (b) problematic position of flag, this case would lead to side effects if region is blanked, (c) in the top right corner the flag is blanked, please note the side effect by the edges of the blanked region, (d) correct detection of both flag and implant, the flag could be removed easily by blanking the bounding box, (e) correct detection of a flag, (f) false detection and blanking of corresponding bounding box, this case is very crucial too.

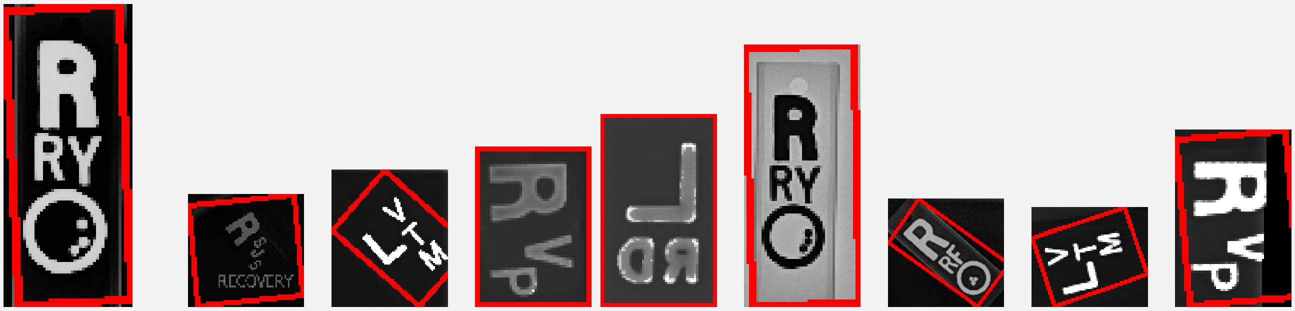


Fig. 6: Extracted flags: The red box is the hand-annotated bounding box of the flag. The orientation and size are preserved as in the corresponding image. The flags differ in size, orientation, texture, transparency, characters and font style. This are only randomly selected samples, there are more shapes in the dataset.

TABLE II: Table SVM Performance

Subset Used	Performance Measure					
	Accuracy	*κ Score*	*Training Time [s]*	*Train Samples*	*Test Samples*	*MURA κ Score*
Elbow	85.8	0.716	3.01	4931	465	0.710[a]
Finger	78.9	0.582	2.22	5106	461	0.389[a]
Forearm	80.6	0.601	0.45	1825	301	0.737[a]
Hand	79.1	0.550	2.54	5543	460	0.815[a]
Humerus	86.4	0.729	0.31	1272	287	0.600[a]
Shoulder	78.5	0.569	4.86	8389	563	0.729[a]
Wrist	86.1	0.717	4.96	9077	659	0.931[a]

[a] Evaluated on the un-released validation set.

score per subset is achieved as in the official MURA baseline [4]. Only on finger and humerus the MURA baseline could be outperformed significantly. There is a not yet identified substantial problem in the implementation.

D. Recurrent Attention CNN

The Recurrent Attention CNN (RA-CNN) is originally trained on the Caltech-UCSD Birds dataset (CUB-200-2011, [10]). The last layer of the network was changed to handle 2 classes (positive and negative studies). Initialized with the pre-trained model, the network was trained on the MURA dataset. The images where resized and padded to the size of 512x512. For the normalization the mean values over the dataset were computed. On the input layer the images were finally cropped to 448x448. The convergence is very slow due to the model size. On the GTX1080Ti with 11GB memory that has been used to carry out the computation, only a batch size of 8 for the training and 2 for the testing, respectively, could be used. Since the exact hyper parameters are not cited in the paper and because the files for the training are missing in the model and code provided by the authors, some default parameters had to be assumed. Parameters that are unknown:

Weight of loss per scale (1,1,2 used), optimizer (ADAM used), optimizer parameters, batch size, normalization (standard Caffe [13] normalization used).

Multiple runs over several hours (up to 9 hours[2]) did not result in reproducing the performance gain over the MURA dataset as achieved and outlined in the paper [1]. The training was not converging and was rippling around the same loss value. There is the possibility that the convergence is very slow, this could not be rejected; it might be, that for the first epochs the loss is flat and begins to decrease later. The CUB-200-2011 dataset consists of 11,788 from 200 different birds and the net was pre trained in ImageNet. The MURA dataset is larger than the CUB-200-2011, but the MURA data has much more defects and anomalies regarding to the image material. A retraining on ImageNet and CUB-200-2011 was not performed, but could be interesting to verify the RA-CNN accuracy. The results would be comparable since in the RA-CNN paper [1] also a VGG19 baseline is used for at least one dataset. In a future approach, the experiments could be redone using graphic cards with higher memory capacity to increase batch size and reduce training time.

E. Early fusion

Since there are multiple x-ray images per study, the majority of them having at least most 3 different views, this set could be processed at once. An approach based on VGG19 with an input of 6 layers was tested (Fig. 7), except for the first layer, the network is shared across the views. There are only very few studies with more than 6 images; for those cases some views were discarded. Otherwise, in case there were less than 6 views, some images were repeated with different preprocessing (crop region, mirroring etc.). This approach did not lead to better results compared to the one in which every view was treated as a single sample. An approach with later fusion might lead to better results[3]; this attempt could be pursued in future works but is not outlined in this paper.

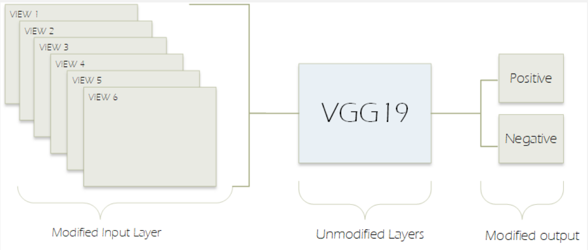


Fig. 7: Early Fusion Schemata: Fusion of the different views in the first layer of the network as used in the experiment.

[2] With the set-up in use, the processing of one batch takes 0.75 hours and the training was stopped after 12 batches.

[3] A common approach is to have a sub network for every view and at some point the data is concatenated.

V. Discussion and Conclusion

The experiments have shown that the network attention focuses on parts that are not representative for the classification problem. The approach that was chosen to remove the ”flags” in the training images in order to prevent the network from focusing on them, turned out to be not ideal. To consistently remove the ”flags” a robust segmentation would be necessary that needs a lot of annotated masks. As an alternative, one could use a different attention selection as in RA-CNN; in our case the RA-CNN was not working out of the box. A direct training using attention from back propagation, as stated within the experiment section, was not working with a naive approach. But it seems to be the most promising to learn directly from feature responses. Since the biggest effort had been invested in the zoom-in approach – set-up of the train schemata, implement the train schemata, and wait for results to be calculated (increase in training time) – a better working approach for the MURA challenge could not be developed and pursued further. Effectively, future approaches should consider using graphic cards with higher memory capacity to increase batch since and reduce training time and additionally, further attempts will have to test whether later fusion indeed have an improving effect on the results.

Classic approaches like Mask-RCNN and Faster-RCNN use Feature Proposal Networks (FPN’s) that are trained in supervised manner. RA-CNN provides an APN that is trained in a semi-supervised fashion. Our approach of ”Zoom-In by Network Attention” tries to generate proposals in an unsupervised-fashion.

References

[1] Fu, Jianlong, Heliang Zheng, und Tao Mei. ’Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition’. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4476–84. Honolulu, HI: IEEE, 2017. https://doi.org/10.1109/CVPR.2017.476.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, Mask-R-CNN, 2017, http://arxiv.org/abs/1703.06870

[3] mask-rcnn implementation, https://github.com/facebookresearch/maskrcnn-benchmark, last accessed 12.02.2019

[4] Rajpurkar at al., MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs, http://arxiv.org/abs/1712.06957

[5] Cheng-Yang Fu, Mykhailo Shvets, Alexander C. Berg, ’RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free’, 2019, http://arxiv.org/abs/1901.03353

[6] You Only look Once V3, https://pjreddie.com/darknet/yolo/, last accessed 14.02.2019

[7] https://stanfordmlgroup.github.io/competitions/mura/, last accessed 14.02.2018

[8] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, 2017

[9] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, Striving for Simplicity: The All Convolutional Net, ICLR 2015

[10] http://www.vision.caltech.edu/visipedia/CUB-200-2011.html, last accessed 15.02.2019

[11] http://cocodataset.org, last accessed 15.02.2019

[12] Sergio A. Serrano, Ricardo Benítez-Jimenez, Laura Nuñez-Rosas, Ma del Coro Arizmendi, Harold Greeney, Veronica Reyes-Meza, Eduardo Morales, Hugo Jair Escalante, Automated Detection of Hummingbirds in Images: A Deep Learning Approach, MCPR 2018

[13] http://caffe.berkeleyvision.org, last accessed 15.02.2019