Radboud University Nijmegen

Faculty of Science

Automated Medical Report Generation

using object detection for chest x-ray scans

Thesis MSc Computing Science | Data Science

Supervisor:
Prof. Elena Marchiori

Author:
Denise Meerkerk

Daily Supervisor:
Zaheer ud Din Babar

Second reader:
Dr. Twan van Laarhoven

February 2023

Contents

Abstract

X-rays are increasingly used as a diagnostic tool. To alleviate the radiologist's workload, more research is being done to automate (parts of) the reporting process.

In this work we apply a method from the field of regular image captioning, to generate reports when given an image from the Indiana University chest X-ray dataset. The applied model uses feature vectors from region proposals and incorporates information about the spatial relationship between detected objects. The region proposals are obtained by an Object Detector trained on the VinBig abnormality dataset.

Quantitative and qualitative analyses of the results show very little variation in the generated reports. We speculate this is due to a mismatch between abnormal detected objects and predominantly positive reports. By using an Object Detector trained on finding normal organs, this problem could possibly be resolved.

1 Introduction

X-rays are an important diagnostic tool and are increasingly used, partly due to the aging world population and partly due to more sophisticated medical imaging systems which lead to more images per examination [1, 2]. From 1998 to 2010 the workload of radiologists has increased by 26% [2]. Analyzing these images is already a demanding task [3] for which radiologists require training and experience to develop more confident and accurate reports [4]. So, this increase in workload increases the likelihood of medical errors, which is not as rare as one would hope [5]. Because the referring physicians mainly consider the medical report, as opposed to the original X-ray image, these errors propagate [6].

A tool that helps produce higher-quality reports in less time could alleviate this problem. As it is a challenging task for trained professionals, it is also a challenging task for an automated system to classify findings of interest within chest radiographs [7]. It is even more challenging to train a model to draft up medical reports based on chest X-ray images. The reports are long compared to regular image captions and contain mainly information about the normal characteristics of the chest. Also, the data shows high variability in the type of abnormalities, and relatively few abnormalities are described. Research is already being done in this field specifically [4, 8, 9, 10].

As shown by [11], the characteristics of the data justify the fact that standard encoder-decoder systems do not perform better than a model that always outputs the same report, obtained by extracting highly frequent sentences from the training data. This could mean that the encoder part of the model is not powerful enough, which is why current research focuses on attention mechanisms for improving report generation, and image captioning in general. In particular, a successful approach in image captioning uses feature vectors extracted from the region proposals obtained by an object detector.

In this thesis, we investigate such an approach in the specific context of radiology report generation. We implement and test a recently introduced method [12] which utilizes information about the spatial relationships between the detected objects. In our context, relative position and size could be relevant to better identify malformations of organs and abnormalities from X-ray images of the chest.

In this work, we aim to investigate if an architecture based on [12] could improve on the current state-of-the-art report generation models ([13, 14, 8]). As a proof of concept, we will investigate the effect of geometry and appearance features on the training of a report generating model, with the Indiana University (IU) Chest X-ray dataset. To accomplish this, we will train two models, one with and one without bounding box

information, and compare the results based on the most common captioning metrics. Furthermore, we will compare the results to other studies also using the IU X-ray dataset.

One of the problems to overcome in this project is the skewed distribution between normal (36%) and all 129 abnormalities in subjects, ranging from a 16% occurrence (lungs[1]) to a 0.026% occurrence (osteoporosis). Such a distribution is typical in chest X-ray datasets [8]. A third model will be trained in which the abnormal X-rays are upsampled. Meaning, that some of the images, in this case, X-ray images containing abnormalities, will be duplicated N times. Again the results of this model will be compared to the other two models and studies from the literature.

Finally, a qualitative analysis will be made for four images from the test set and their generated captions. We also inspect the variation of the captions generated by our models.

2 Related Work

2.1 Image Captioning

The automatic generation of medical reports is largely built on image captioning techniques. Image captioning developed from a uniform grid of image regions as model inputs to attention at the object level in images [15].

Just like those image captioning projects we make use of an object detector. During this project, we used the YOLO(v3) [16] as the object detector. Other popular object detectors include a Convolutional Neural Network (CNN) [17] or Faster R-CNN [18, 12, 15] for regular image captioning. The information from the object detector can either be internalized in the model [17], or existing features can be used [12, 15]. In the first case, the object detector is internalized as the encoder part of the encoder-decoder system. In the latter case, the training of the object detector is done separately from the training of the captioning model.

Finally, the image or features of the image need to be translated into written words. Encoder-decoder systems, originated in translation models [19], also turned out to be a useful technique in the image captioning field. In [18] the encoder and decoder are fully connected to each other. In [15] instead of an encoder-decoder system, an LSTM layer is used as a language model to caption the image. In [17] the decoder consists of a simple RNN structure. In [12] the same decoder is used as proposed by [19] for a machine translation project. In this decoder, each of the (six) decoder layers is connected to the previous decoder layer.

In this project, the same encoder-decoder is used as in [12] which was proposed for image captioning and was built on [19] which was proposed as a machine translation model.

2.2 Radiology Report Generation

In [4] a deep learning model to automatically generate radiology reports given a chest X-ray image is proposed called CDGPT2. Before full medical reports are generated, specific tags from the image are predicted and weighted semantic features are calculated from the predicted tag's pre-trained embeddings. Their model surpassed most non-hierarchical recurrent models and transformer-based models in quantitative metrics, especially the semantic similarity metrics, while being considerably faster to train. A qualitative analysis from a radiologist is included to see how well the generated reports are written. It turned out that their model performed well on normal images, with an

[1] Obviously, having lungs is not abnormal in itself, at least if you identify as a mammal, bird, reptile or adult amphibian.

accuracy score of 99%, while the model performed worse on the abnormal instances with an accuracy score of 36.5%. The same Indiana University X-ray dataset was used.

The memory-driven Transformer model proposed by [9] uses a relational memory (RM) to use the information from similar reports previously seen by the model and a memory-driven conditional layer normalization (MCLN) to incorporate the memory into the decoder of their Transformer. Their experimental results show that their proposed approach outperforms previous models with respect to both language generation metrics and clinical evaluations. They also demonstrate that their approach is able to generate long reports with necessary medical terms as well as meaningful image-text attention mappings. Two X-ray datasets were used, one of which was the Indiana University X-ray dataset.

The paper by [8] proposes to separate abnormal and normal sentence generation, by using a dual word LSTM in a hierarchical LSTM model. They start by extracting the feature embeddings from their MTI tag prediction (CNN). These embeddings are used to generate a topic or whether the model should stop (LSTM). Based on the generated topic they predict whether the topic is for an abnormal or normal sentence, before constructing said sentence by a specialized LSTM for (ab)normal sentences. Their final model is not only selected on the BLUE-4 metric but also on the number of distinct sentences. By focusing on isolating the abnormal and normal sentences, they hope to increase the variability of the generated paragraphs. They conduct an analysis on the distinctiveness of generated sentences compared to the BLEU score, which increases (undesirably) when less distinct reports are generated. A way of selecting a model that generates more distinctive sentences is proposed. They hope their findings will help to encourage the development of new metrics to better verify methods of automatic medical report generation. The same Indiana University X-ray dataset was used.

In [14] an encoder-decoder based framework that can automatically generate radiology reports from medical images is proposed. They specifically use a Convolutional Neural Network as an encoder coupled with a multi-stage Stacked Long Short-Term Memory as a decoder to generate reports. Their experimental results, split into abnormal and normal images, show the effectiveness of their model. Just like in this project, the Indiana University dataset was used.

The [13] paper presents a domain-aware automatic chest X-ray radiology report generation algorithm. It learns core findings and fine-grained descriptions of findings from images and uses their pattern of occurrences to retrieve and customize similar reports from a large report database. They also develop an automatic labeling algorithm for assigning such descriptors to images and build a novel deep-learning network that recognizes both coarse and fine-grained descriptions of findings. During post-processing, mentioned findings whose evidence is absent in the predicted label pattern will be removed. Their model is trained and validated on the MIMIC-4 and NIH datasets, not on the IU dataset. They did test their model on the IU dataset. The resulting report generation algorithm significantly outperforms the state-of-the-art using established metrics.

3 Method

For the most part, we follow the approach of [12]. The main differences are the use of a different Object Detector (OD) model and the nature of the datasets. They used a Faster R-CNN as their OD, while we use YOLOv3 [16], suggested as an improvement

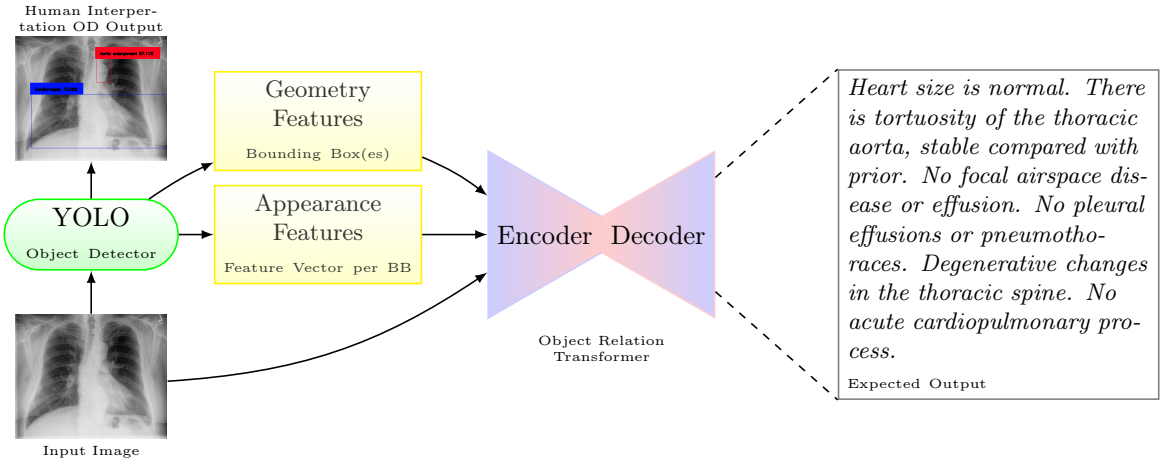


FIGURE 1: Schematic illustration of the proposed model. First a trained object detector provides bounding box coordinates (Geometry Features) and corresponding feature vector(s) (Appearance Features). This information is encoded by the Encoder to a latent vector space. The decoder uses this latent vector space to generate a caption.

by [15]. The dataset they used, MSCOCO, contains regular images, while our research is focused on medical chest X-ray images.

The Object Detector finds the interesting parts of the X-ray images, which is described in 3.1. Using this information, the Object Relation Transformer can caption a given X-ray image, explained further in 3.2. In Figure 1 the schematic of the algorithm is shown.

3.1 Object Detection with YOLOv3

We used YOLOv3 [16] as the object detector model. We trained it on the VinBig data set and applied transfer learning with pre-trained weights of the ImageNet data set, to train faster. The training stopped after 17 epochs when the early stopping conditions were met.

The trained model is used to extract bounding box proposals from the X-ray images of the IU X-ray data set and to get a feature map for each bounding box proposal. Overlapping bounding box proposals with an intersection-over-union (IoU) exceeding a threshold of 0.7 are discarded since they probably describe the same abnormality. Each generated bounding box contains a potential abnormality in the X-ray image. The encoder expects a feature vector for every bounding box, but YOLO only generates features for the whole image. Therefore the image within every bounding box is put through the YOLO model to extract the feature map of the bounding box. This is done at the conv_80 layer of the YOLO model because its shape is most similar to the feature map as used in [12]. We further discard all bounding boxes where the class prediction probability is below a threshold of 0.5, which is the standard value of the YOLO model.

We hypothesize that YOLO will also yield images without a bounding box since there are images in the dataset without any abnormalities. If no relevant bounding boxes are found, a bounding box over the full image is returned.

3.2 Object Relation Transformer

Originally encoder-decoder systems were designed for translation jobs. The central idea is that when a human brain translates a sentence, the meaning of the sentence will

first be conceptualized in the brain before being translated into the second language. Analogously the encoder will be trained to conceptualize a sentence to a latent space, while the decoder is trained to use that latent space to form a translation.

According to the same principle, images can also be conceptualized. Instead of translating this is called captioning.

The Object Relation Transformer (ORT) [12] is an encoder-decoder system. It encodes the X-ray image together with the bounding box coordinates and corresponding feature vector(s) obtained by the object detector. This information is used to find a relation between the different objects in an X-ray image and saves it to a latent vector, taking into account appearance-based and geometry-based attention. To accommodate a varying number of bounding boxes and corresponding feature vectors the ORT uses, just like the original Transformer, multi-head self-attention. Specific to the ORT, the appearance and geometry weights are combined in an NxN matrix Ω, where N is the number of bounding boxes. Each head calculates an output based on the input values and the combined appearance and geometry weights. This means that the ORT can handle a number of bounding boxes ranging from 1 to infinity, computing limits aside.

The decoder is trained to generate a caption using the latent vector as input. We call this version of the model the vanilla model or $\text{ORT}_{\text{vanilla}}$.

3.2.1 Without Bounding Boxes

To test the effectiveness of the method proposed by Herdade [12] in the context of medical X-ray images, we also train a model without bounding box information. To limit the number of changes to make to the code and original model, we effectively give each image one bounding box stretching over the full image. The feature vector of the whole image is used. As mentioned in section 3.1, the same is done for images for which no bounding boxes are found by the YOLO model. Effectively indicating that the whole image is important.

The results of this method will be compared to the vanilla model with the standard metrics. This version of the model is called $\text{ORT}_{\text{no-bb}}$.

3.2.2 Upsampling

As mentioned in [20, 8], the annotations are imbalanced. For a large number of patients in the IU dataset, nothing abnormal is found in the X-ray image by the radiologist. This is a frequent problem in medical datasets because X-rays are often made to rule out certain issues. The dataset is also considered too small for deep learning models [21]. That is why we decided to experiment with the upsampling of patients with findings in their X-rays.

This was done by filtering through the "Problems" column of the annotation and filtering out all 'normal' instances. All other "Problems" are considered not normal and duplicated nine times. This means that each normal image occurs only once in the dataset during training, and an image containing at least one abnormality occurs ten times in the dataset during training.

If the original image was part of the training set, all its potential duplicates are also part of the training set. Both the validation and test set are not upsampled. The test set is not upsampled because inherent to a test set, we do not have the information to make a distinction between normal/abnormal images. Secondly, we need to ensure comparability between the results of the different model versions and other models from the literature. The validation set is not upsampled, because we use it to mimic the conditions at the moment of testing the model.

The results of this method will be compared to the vanilla model with the standard metrics. This version of the model is called ORT_{up10}.

4 Experimental Set-Up

After discussing the structure of the dataset and the used metrics (section 4.1), the implementation of the experiments is discussed in detail in section 4.2.

4.1 Datasets and Metrics

For this project, two different X-ray datasets were used. For the training of the Object Detector, the VinBig dataset [21] was used. The Indiana University (IU) dataset [22] was used for the training of the Object Relation Transformer.

4.1.1 VinBig and Bounding Box Metrics

First, we will discuss the data used to train the Object Detector and the metrics used to assess the quality of the generated bounding boxes. Originally we only wanted to work with the IU dataset because it is annotated by radiologists and was one of the more recently published datasets when we started this project. However, to train the object detector we needed a dataset with bounding box annotation, which the IU dataset didn't provide. For this purpose, the VinBig dataset got utilized, which was downloaded from Kaggle[2].

This version of the VinBig data set consists of 15 thousand dicom images of chest X-rays with bounding box annotation and class labels for 14 (local) abnormalities and images without any abnormal findings. The abnormalities are listed in Table 1. Approximately 70% of the images have the global label of 'no finding'. The remaining 30% of images contain one or more of the listed abnormalities. For the full distribution, we refer to [21]. The dataset does not provide medical report annotation.

We don't need to do a lot of preprocessing for the images. For more efficient training and to increase the similarity between the two data sets we needed to make the VinBig images smaller. The width and height of the images were decreased by a factor of 4, meaning that the resulting images decreased in size by a factor of 16. After resizing, the images are on average 602 by 697 pixels. Finally, the images are converted from dicom to png format using the 'bone' colormap.

The bounding box annotations also had to be preprocessed. Due to the resizing of the images, the bounding box coordinates would not align with the abnormality. The same resizing factor is used to calculate the new correct bounding box coordinates corresponding to the new image size.

All 14 abnormalities occur in the IU dataset as well, albeit not one on one in all cases. The cardiomegaly abnormality does occur in both datasets in the same way. However, the aortic enlargement abnormality is more specific in the VinBig dataset, while the IU dataset only mentions the 'aorta'. The other way around happens as well, the VinBig dataset has an abnormality called 'nodule/mass', which is split into two separate abnormalities for the IU dataset. Furthermore, the IU dataset has about 130 unique abnormalities while the VinBig dataset has 14 abnormalities.

To determine the correctness of the generated bounding boxes we use the Mean Average Precision ((m)AP) metric. First, the AP is calculated for each separate class; in our case the VinBig abnormalities. To obtain the AP first the Intersection over

[2] https://www.kaggle.com/competitions/vinbigdata-chest-xray-abnormalities-detection/data

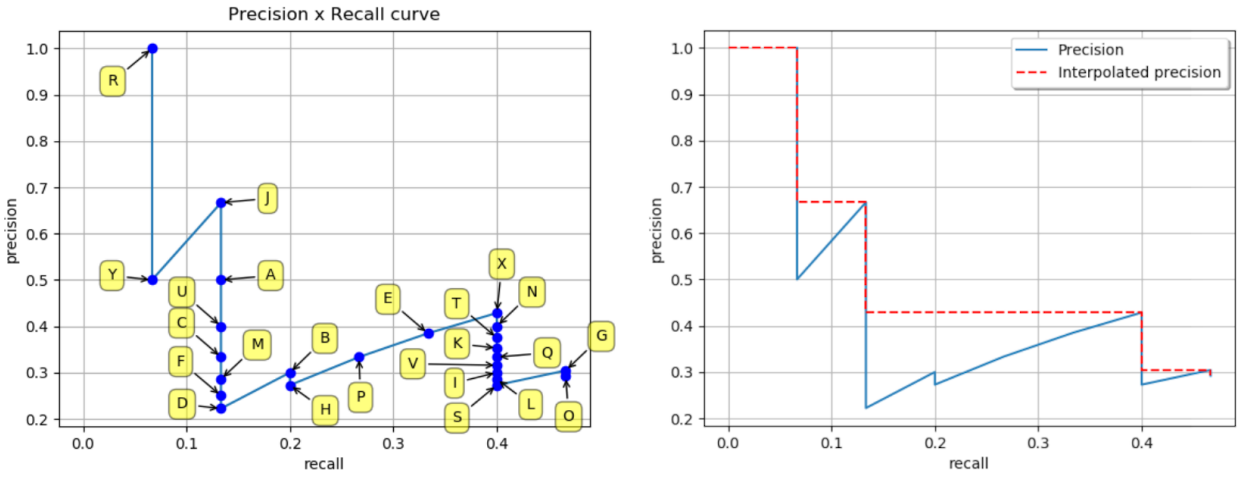
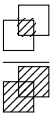


FIGURE 2: On the left an example of a zig-zag pattern from a recall precision curve. On the right image is illustrated how the interpolated precision is determined. Both figures are obtained from [23].

Union (IoU) is calculated for every predicted bounding box (bb_p) in an image, for every overlapping ground truth bounding box (bb_{gt}) according to Equation 1.

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \quad (1)$$



If the IoU value exceeds a certain threshold, we count this bb_p as a True Positive (TP), otherwise as a False Positive (FP). All bb_ps of a class are ranked according to their predicted confidence level in descending order and the accumulated precision (P_T) and recall R_T is calculated from top to bottom, according to Equations 2 and 3.

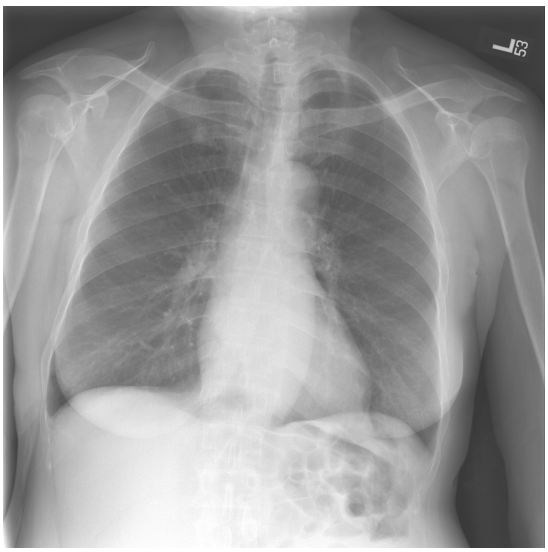
$$P_T = \frac{\sum_{t=0}^{T} TP_t}{\sum_{t=0}^{T} TP_t + FP_t} = \frac{\sum_{t=0}^{T} TP_t}{\sum_{t=0}^{T} \text{all detections}_t} \quad (2)$$

$$R_T = \frac{\sum_{t=0}^{T} TP_t}{\sum_{t=0}^{T} TP_t + FN_t} = \frac{\sum_{t=0}^{T} TP_t}{\text{all ground truths}} \quad (3)$$

$$(4)$$

Where T is the number of seen bb_ps up until that step, ranging from 0 to the number of predicted bounding boxes. FN represents the false negatives, in this case, all bb_{gt} that are not discovered as bb_p (yet). Since the sum of TP and FN is always equal to the total amount of bb_{gt}, we don't have to count the number of FN.

By calculating the precision and recall this way, the precision will fluctuate and the recall will either stay the same or increase at each step. If the precision and recall are plotted against each other, a zig-zag pattern would be expected. The interpolated precision can be obtained by getting the upcoming maximum precision as illustrated by Figure 2. The area under the interpolated precision plotted against the recall is the AP of that specific class and a higher score indicates more overlap between bb_ps and bb_{gt}s. To get the mAP score, the APs of each class are simply averaged. Because we use an IoU threshold of 50%, the metrics are more accurately denoted as mAP_{50} and AP_{50}. For a more detailed explanation and example of the (m)AP metric, we refer to [23].



Problems: *normal*
Findings: *The cardiac silhouette and mediastinum size are within normal limits. There is no pulmonary edema. There is no focal consolidation. There are no XXXX of pleural effusion. There is no evidence of pneumothorax.*
Impression: *No acute abnormality.*

FIGURE 3: The X-ray image and corresponding problems, findings, and impression of patient 1280. No abnormalities are identified on this X-ray.

4.1.2 IU Chest X-Ray and Captioning Metrics

In this section, we will discuss the IU X-ray dataset used to train the Object Relation Transformer and the metrics necessary to assess the models.

The IU X-ray dataset consists of (multiple) frontal and/or lateral X-ray chest images of 3851 patients annotated by radiologists. In this project, only the frontal images are taken into account. The annotations consist of the different parts of a radiologist report like the indication, findings, and impression. In addition, the reports are manually annotated with medical labels. About 36% of the images are considered normal, the remaining 64% contain one or more abnormalities. The 130 abnormalities are not uniformly distributed. Some of the abnormalities only occur once in the whole dataset, while others like -one of the most frequently occurring abnormality- cardiomegaly occurs in 9% of the images. All abnormalities from the VinBig dataset are represented in this dataset as well, however not always in exactly the same format. This dataset is open source and all images are de-identified to protect patient privacy.

The goal of this project is to construct the findings and impressions from the information the model gets from the corresponding X-ray image. Unfortunately, 512 of the patients do not have an annotation in the findings column. The images of these patients are omitted from the project.

In patients without abnormalities, the findings of an X-ray image are an enumeration of things that are normal or not wrong in the patient, for an example see Figure 3. In patients with abnormalities the enumeration starts with things that are wrong and ends with things that are normal or not wrong, for example, see Figure 4.

This dataset does not have a standard train and test split. In this project, the same data split is used as for the Kaggle competition[3] from which it was downloaded. The validation set is constructed from the train split, by shuffling the images randomly and assigning the last 10% of the image to the validation set. For the purposes of reproduction, a seed value was set.

Multiple caption metrics are used to compare the resulting report from the model

[3] https://www.kaggle.com/datasets/raddar/chest-xrays-indiana-university

Problems: *Pulmonary Disease, Chronic Obstructive; Bullous Emphysema; Pulmonary Fibrosis; Cicatrix; Opacity; Opacity; Opacity; Opacity*
Findings: *There are diffuse bilateral interstitial and alveolar opacities consistent with chronic obstructive lung disease and bullous emphysema. There are irregular opacities in the left lung apex, that could represent a cavitary lesion in the left lung apex. There are streaky opacities in the right upper lobe, XXXX scarring. The cardiomediastinal silhouette is normal in size and contour. There is no pneumothorax or large pleural effusion.*
Impression: *1. Bullous emphysema and interstitial fibrosis. 2. Probably scarring in the left apex, although difficult to exclude a cavitary lesion. 3. Opacities in the bilateral upper lobes could represent scarring, however the absence of comparison exam, recommend short interval followup radiograph or CT thorax to document resolution.*

FIGURE 4: The X-ray image and corresponding problems, findings, and impression of patient 4. Multiple abnormalities are identified on this X-ray.

to the reference annotation. BLEU (Bilingual Evaluation Understudy) [24], originally developed for translation tasks, can be used to score the constructed candidate caption compared to a reference caption. The number of matching n-grams is counted, and the more matches between the constructed sentence and the reference the higher the score.

For example, BLEU-n is calculated by determining the modified n-gram precision (p_n, Equation 5) and the brevity penalty (BP, Equation 6).

$$p_n = \frac{\#\text{ overlapping } n\text{-grams}}{\#\text{ candidate } n\text{-grams}} \tag{5}$$

In other words, to calculate the p_n, the number of overlapping n-grams between the candidate and reference sentence is divided by the number of candidate n-grams. Each n-gram from the reference sentence can only be matched once, to prevent candidates with repeating words from getting high scores.

To prevent short sentences with high overlap from obtaining high scores, all candidate sentences that are shorter than the reference sentence get a penalty.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-\frac{r}{c}} & \text{if } c \leq r \end{cases} \tag{6}$$

Where c is the length of the candidate caption and r the length of the reference caption. Finally, the BLEU score can be calculated by Equation 7.

$$\text{BLEU}_n = BP * p_n \tag{7}$$

Note that this is a simplified case since we only have one reference sentence for each image and we calculate a BLEU score for each n-gram separately instead of using weights. For a more in-depth explanation of the general calculation of BLUE scores, we refer to [24].

METEOR (Metric for Evaluation of Translation with Explicit ORdering) is another metric used to compare the difference between sentences. It is specially designed to overcome some of the weaknesses of BLEU and NIST. Compared to BLUE, METEOR has a higher correlation with human judgment on the corpus level [25].

Similar to BLEU, METEOR calculates the uni-gram precision according to Equation 5, where $n = 1$. METEOR also calculates a recall score (Equation 8), the difference being the division by the number of reference unigrams instead of candidate unigrams. Next, the F_{mean} is calculated by combining the precision and recall using a harmonic-mean (with ratio $1p : 9r$), see Equation 9.

$$r = \frac{\#\text{ overlapping uni-grams}}{\#\text{ reference uni-grams}} \tag{8}$$

$$F_{mean} = \frac{10pr}{r + 9p} \tag{9}$$

To take into account longer matches, METEOR applies a penalty if only lower-level n-grams are matched, see Equation 10. This is done by counting the fewest possible number of chunks. A chunk is defined as an n-gram from the candidate sentence that matches the reference sentence completely. In the edge case where the candidate sentence is exactly the same as the reference sentence, the number of chunks is one, which leads to a small penalty depending on the number of overlapping uni-grams. If there are no bi-gram or longer matches, then the number of chunks is equal to the number of overlapping uni-grams, which equals a Penalty of 0.5. So, the longer the n-grams, the lower the Penalty.

By combining the F_{mean} and Penalty the METEOR score can be calculated according to Equation 11.

$$\text{Penalty} = 0.5 * \left(\frac{\#\text{ chunks}}{\#\text{ overlapping uni-grams}} \right)^3 \tag{10}$$

$$\textsc{Meteor} = F_{mean} * (1 - \text{Penalty}) \tag{11}$$

The final metric we use is ROUGE (Recall-Oriented Understudy for Gisting Evaluation). Specifically, ROUGE-L is used, which scores the similarity between two sentences based on the Longest Common Subsequence (LCS) [26].

To calculate the ROUGE-L score, the recall-LCS, and precision-LCS need to be obtained, see Equations 12 and 13 for reference sentence 'Ref' with length r and candidate sentence 'Can' with length c.

$$R_{lcs} = \frac{LCS(\text{Ref}, \text{Can})}{r} \tag{12}$$

$$P_{lcs} = \frac{LCS(\text{Ref}, \text{Can})}{c} \tag{13}$$

Where the function $LCS(A, B)$ outputs the length of the longest common subsequence between two sequences.

Using the R_{lcs} and P_{lcs} the ROUGE-L score, based on the F-measure, can be calculated according to Equation 14. The ROUGE-L score ranges from zero, when no overlap is found between the candidate and reference sentence, to one, when the candidate sentence matches the reference sentence perfectly.

$$\textsc{Rouge-l} = \frac{\left(1 + \left(\frac{P_{lcs}}{R_{lcs}}\right)^2\right) R_{lcs} P_{lcs}}{R_{lcs} + \left(\frac{P_{lcs}}{R_{lcs}}\right)^2 P_{lcs}} \tag{14}$$

4.2 Implementation

To try to ensure reproducibility, this section will discuss the implementation of our experiments. First, the YOLO implementation will be discussed in section 4.2.1. Second, the implementation of the ORT will be discussed in section 4.2.2. All experiments were run on a shared NVIDIA-SMI Driver (Version: 510.60.02). For the GPU environment, CUDA version 11.6 was used.

Multiple python environments got used for different scripts. All are available on the GitHub[4] page of this project. In the bash job-file of each script, the right environment is called. If no bash job is available, environment `scriptie4` should be tried first.

4.2.1 YOLO

To estimate the performance of the YOLO Object Detector, the VinBig data was split in a train, validation, and test set. The total dataset consists of 18.000 images. We use the standard test split provided by Kaggle, consisting of 3000 images (16.7%). From the remaining 15000 images we construct a train and validation set by randomly selecting 3000 images for the validation split and using 12.000 images (66.7%) as a train set.

Unfortunately, we encountered an issue during the preprocessing of the test images, so in the end, only 1967 images were used as test data.

During training, the original YOLOv3 loss function is used to monitor the learning process and is used for the backpropagation step during training. This loss function is built on 4 sub-loss functions based on four object detection characteristics, see Equation 15. MSE and BCE stand for Mean Squared Error and Binary Cross Entropy, respectively.

$$\begin{aligned} \text{YOLO}_{loss} = & MSE(\text{bb coordinates}) + \\ & BCE(\text{objectness score}) + \\ & BCE(\text{no objectness score}) + \\ & BCE(\text{multi-class predictions score}) \end{aligned} \tag{15}$$

The standard batch size of 16 was adopted. We also used the same early stopping parameters as the original YOLO implementation, meaning that the loss was monitored with `min_delta` $= 0.01$ and `patience` $= 7$. We started learning with a learning rate of $1.00E-04$.

For the $\text{ORT}_{\text{no-bb}}$ experiment, `construct_feature_file.py` gets `no_box` parameter set to `True`. Every image gets one box and one feature vector based on the whole image. For $\text{ORT}_{\text{vanilla}}$ and ORT_{up10} the `no_box` parameter is set to `False`.

4.2.2 ORT

To evaluate the performance of the ORT captioning model, the IU dataset needs to be split in a train, validation, and test set. The original dataset consists of 3851 patients. Some patients have incomplete annotations, so those are omitted from this project. After preprocessing 2038 images remain. The dataset was already split into a train and test set. The test set consists of 181 images (9%). From the train set we randomly select 173 images (8%) to perform as the validation set. The remaining train set consists of 1684 images (83%).

In [12] the ORT was pre-trained for 30 epochs with softmax cross-entropy loss, after which they used self-critical training based on CIDEr metric for another 30 epochs.

[4] https://github.com/DeniseMeerkerk/ScriptieRepo

The different versions of our ORT model got trained using softmax cross-entropy loss and stopped training after 50 epochs. We didn't do self-critical training with CIDEr, because it uses built-in inverse document frequency, and medical words are not very well represented. In future work, other language metrics could be used. For now, it was outside the scope of this project.

The batch size was set to one. Larger batch sizes were met with memory issues on the used server.

The difference in implementation of the ORT model between the three experiments is mainly seen in the input files used for training (`json`, `h5`, and `features`). Now, we will discuss how the $\text{ORT}_{\text{no-bb}}$ and ORT_{up10} experiments differ from $\text{ORT}_{\text{vanilla}}$.

For the $\text{ORT}_{\text{no-bb}}$ the construction of the feature file doesn't construct bounding boxes according to the trained YOLO model, but one bounding box for the whole image as discussed in section 3.2.1. To achieve this, the `construct_feature_file.py` parameter `no_bb` is set to true. Consequently, the input files following from the features file differ from the vanilla experiment as well.

For experiment ORT_{up10}, we upsampled the dataset as described in section 3.2.2. First, all training images with abnormalities are identified and duplicated nine times in the input `json` file. So, images without abnormalities occur once, while every image with abnormalities occurs ten times (`upsample_with_findings.py`). After that, the vocabulary of the upsampled dataset is constructed and the captions are encoded to that vocabulary (`prepro_labels.py`). It is important that all training input files correspond to each other. To make sure that it does, the encoded captions are copied from one `json` file to another (`vocab_fix_json.py`).

5 Results

5.1 Object Detector

We do not have bounding box annotation for the IU dataset. Therefore, it is harder to interpret the results of the object detector. We will discuss the loss curve during the training of the YOLO model, the Mean Average Precision (mAP) score of the VinBig test set, and a comparison of the number of generated bounding boxes between images with and without abnormalities for the IU dataset.

During the training of the object detector the loss score was saved for the model, as seen in Figure 6. After approximately the 300^{th} iteration the loss doesn't decrease anymore. After approximately 1750 iterations the model stops training based on the early stopping parameters. Unfortunately, the validation loss was not saved in this process, therefore we can not compare the training loss against the validation loss.

Due to a missing plugin (the jpeg 2000 decoder plugin) for the `pydicom` package, only 1967 of the 3000 test images of the VinBig dataset were able to be pre-processed. It is unknown why this only affected around one-third of the test images. Of the remaining test images the average precision (AP) score was calculated for each of the 14 abnormalities present and annotated in the dataset, as well as the mean of all the abnormalities scores, as seen in Table 1. The scores of our YOLO object detector are very low. Most of the abnormalities have an AP score of zero, so the mAP score of 0.0542 is very low as well. For comparison, the top-scoring Kaggle submission has a mAP_{40} score of 0.314. Do note that we use a mAP_{50} score, so the comparison should be interpreted with care. The object detector seems to do reasonably well in detecting aortic enlargements and cardiomegalies, the two most frequently occurring abnormalities in the VinBig dataset [21]. For the other abnormalities, we hypothesize that the classification threshold was

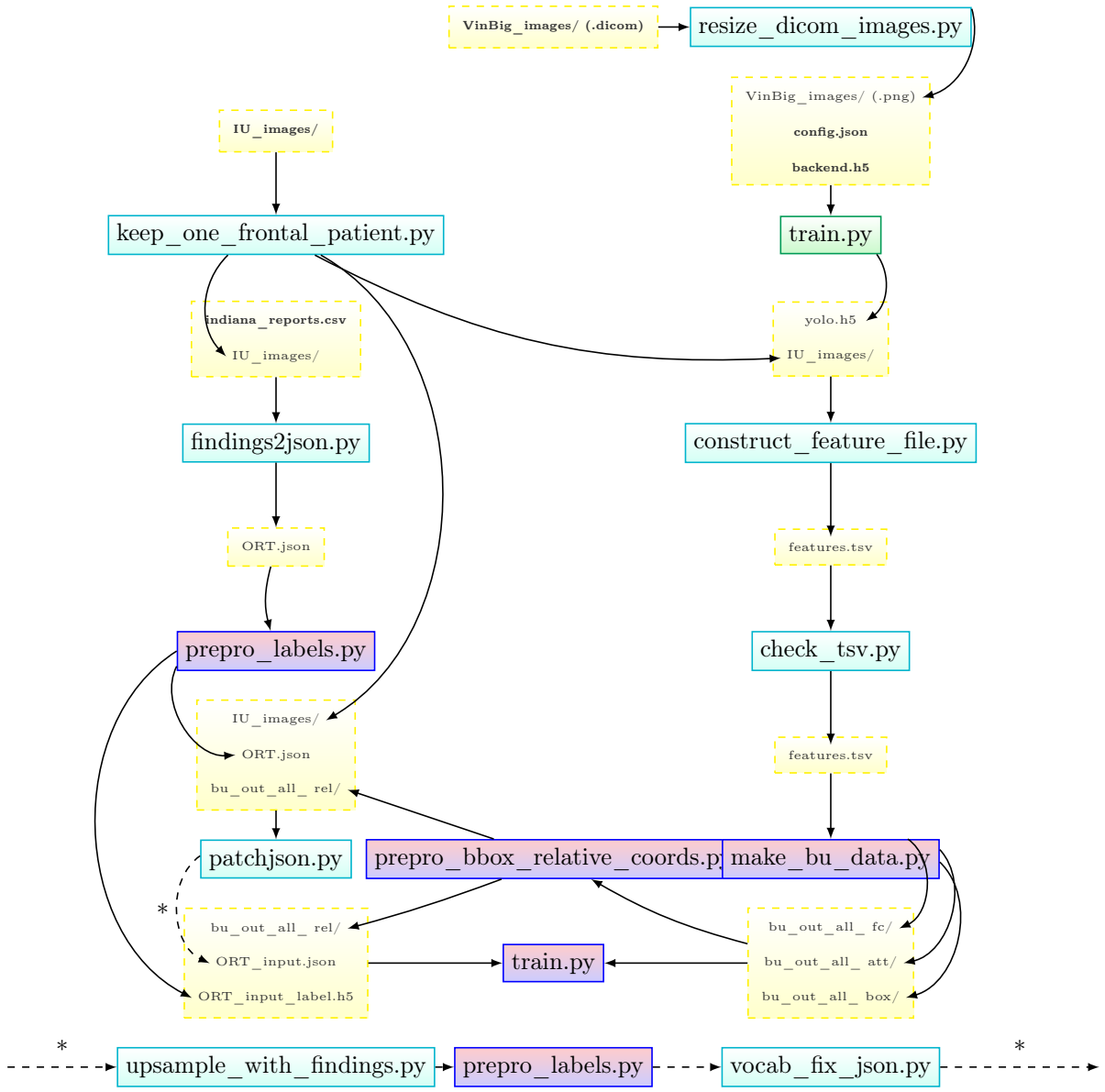


FIGURE 5: Flow-chart of all scripts (https://github.com/DeniseMeerkerk/ScriptieRepo) needed to train the ORT. In- and output of the scripts are indicated with yellow nodes. The bold inputs are a prerequisite. The red/blue nodes are scripts used from the original ORT repo (https://github.com/yahoo/object_relation_transformer). The green node is directly used from the YOLOv3 repo (https://github.com/experiencor/keras-yolo3. The turquoise nodes indicate scripts made for this project. To prepare for the upsampling experiment the dashed arrow with an * should be substituted with the scripts at the bottom of the figure.

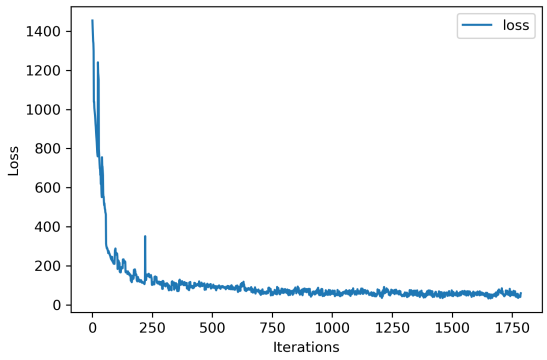


FIGURE 6: Loss score during the training of the YOLO object detector on the VinBig dataset.

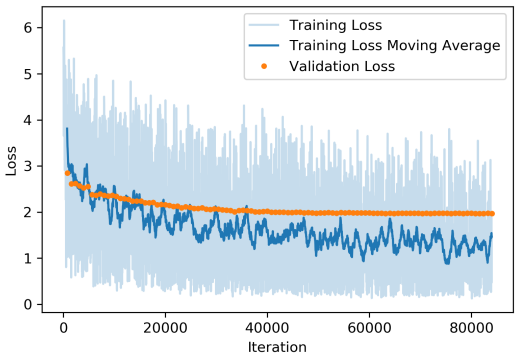


FIGURE 7: Training loss curve without bounding box information.

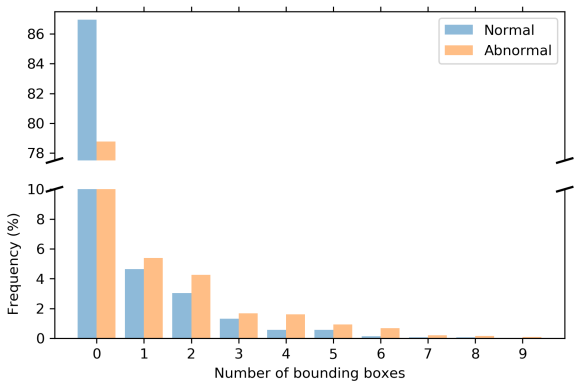


FIGURE 8: Distribution of the number of bounding boxes per image found by the YOLO object detector on the IU test set, split by normal/abnormal patients.

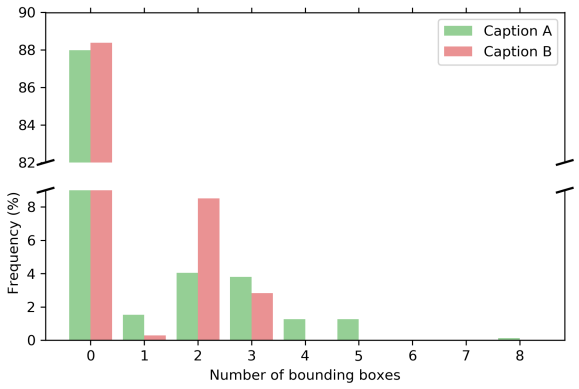


FIGURE 9: Distribution of the number of bounding boxes per image found by the YOLO object detector on the IU test set, split by caption A and B. See Table 6 for the exact generated caption corresponding to caption A and B.

set too high to generate any bounding boxes and that the occurrences in the dataset were too low.

In Figure 8, we observe that the distributions of the number of bounding boxes found per image of the normal and the bundled abnormal patients on the IU dataset are similar. We notice that the object detector returns no bounding box for a large part of both the normal and abnormal group of patients, which indicates that detecting abnormalities is difficult for this dataset. One would expect more bounding boxes for abnormal patients. To see whether the two samples could have been drawn from the same distribution, we perform a Kolmogorov–Smirnov test. According to the Kolmogorov–Smirnov test the number of bounding boxes is not significantly different between normal and abnormal samples ($p > 0.05$).

For a lot of images, no bounding box is generated, even for the images containing abnormalities (78%). More bounding boxes would mean more information for the ORT, which could lead to better results. It was tried to increase the number of bounding boxes by decreasing the threshold at which a bounding box is accepted from a 0.5 to 0.25 confidence score. However, during the training of an ORT model with this increased

TABLE 1: The AP scores (0.0-1.0) for different abnormalities on the VinBig train and test set after training the YOLO object detector.

Abnormality	**Val. AP**	**Test AP**
Aortic enlargement	0.4387	0.4388
Atelectasis	0	0
Calcification	0	0
Cardiomegaly	0.3198	0.3198
Consolidation	0	0
ILD	0	0
Infiltration	0	0
Lung Opacity	0	0
Nodule/Mass	0	0
Other lesion	0	0
Pleural effusion	0	0
Pleural thickening	0.0004	0.0004
Pneumothorax	0	0
Pulmonary fibrosis	0	0
mAP	0.0542	0.0542

number of bounding boxes, we encountered memory issues on the used server. Fixing these issues was outside the scope of this project.

There is no guarantee that more bounding boxes would improve the modelta because we still have the issue that there is a limited relation between the found abnormalities and the annotated caption, which will be discussed in more detail later on.

One of the caveats to note is that the object detector is trained on the VinBig dataset, which only has 14 different abnormalities, while the IU X-ray dataset has a total of 129 abnormalities. As mentioned before, all of the abnormalities in the VinBig dataset occur in the IU dataset as well.

5.2 Quantitative Analysis

In Figures 7, 10, and 11, the training and validation loss of the IU dataset is plotted against the number of iterations for all three ORT versions. Because the batch size was equal to one during training, due to a memory issue, the training loss fluctuates greatly. Therefore, a moving average of the training loss is plotted in addition.

At the start of the training of the $\text{ORT}_{\text{no-bb}}$ (Figure 7) the loss decreases fast and starts to plateau. The validation loss seems to plateau earlier in the process than the training loss. This can indicate a slight overfitting on the train set and/or that the validation set is not representative of the train set. After 84000 iterations or 50 epochs, the training is stopped. The train and validation loss of the $\text{ORT}_{\text{vanilla}}$ in Figure 10 shows a similar pattern.

During the ORT_{up10} training (Figure 11), the training loss seems to steadily decrease over the iterations. However, at the same time, the validation loss quickly starts to increase after its initial decrease. This was expected because the validation set is not upsampled, thus the validation set is not representative of the training set. This model uses more iterations than the other two models because the number of images per epoch is increased. All the versions of the model have trained for 50 epochs.

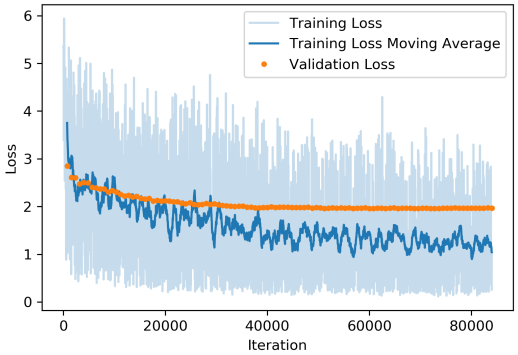


FIGURE 10: Training loss curve with bounding box information.

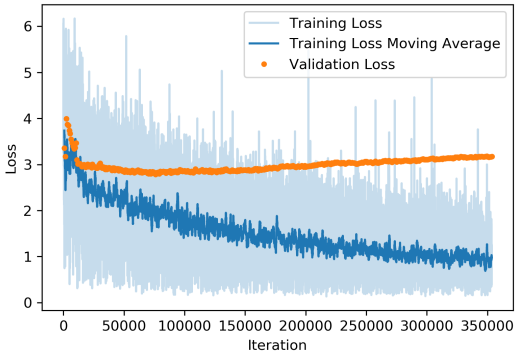


FIGURE 11: Training loss curve with upsampled dataset and bounding box information.

In Table 2, the BLUE-n, METEOR, and ROUGE metrics of models from the literature and our own are compared to each other. Be aware that all these results are directly cited from their respective papers, and that there is no standard test split, so the results should be interpreted with care. Overall, the FFL+CFL-based model [13] has the highest scores for all metrics.
This model is not trained on the IU dataset, but on two bigger datasets, therefore they suffer less from overfitting. On top of that, they use a carefully designed vector space to match the images to the captions (FFL). This design seems to work well and gives very explainable results. The drawback is that it requires more domain knowledge. The last big difference is the use of a database. Their model doesn't need to learn language from scratch because it uses the caption with the most similar vector and adjusts from there.

The second-highest METEOR score is earned by our own method. It's shared between $\text{ORT}_{\text{vanilla}}$ and $\text{ORT}_{\text{no-bb}}$. The METEOR score claims to be correlating with human judgment [25]. As will be further discussed in section 5.3, both the $\text{ORT}_{\text{vanilla}}$ and $\text{ORT}_{\text{no-bb}}$ only output a single caption independent from the inputted image. This caption occurs multiple times verbatim in the annotation of the dataset. Furthermore, METEOR uses a stemming function to match words with their plural/singular counterparts or other tenses. These characteristics explain why the METEOR score is relatively high compared to the other metrics.

Finally, we notice that the ORT_{up10} always performs slightly worse compared to the $\text{ORT}_{\text{vanilla}}$ and $\text{ORT}_{\text{no-bb}}$ without upsampling. Because the object detector is trained on a dataset that has abnormality bounding box annotation, but the objective of the ORT is to generate captions for a set of images that is considered 36% normal, while the X-rays with abnormalities also contain a lot of 'normal' medical phrases. The OD will only recognize abnormalities, e.g. it will recognize an abnormal heart, but not a normal heart. These two factors, the generated abnormal bounding boxes, and the frequent 'normal' medical phrases, possibly cause the ORT not to get the information it needs to properly annotate the IU dataset. It is hard for the ORT to generate text based on an absent bounding box, e.g. healthy heart. At the same time, it is hard to generate text for the present bounding box, e.g. enlarged heart, with a very rare occurrence in the dataset (9%). The ORT can not match both information sources. This could explain why the current local optima are found for all versions of the ORT model.

The ORT model was very successful on regular images [12]. There are a few possible reasons why the application on medical X-rays results in such low scores compared

TABLE 2: Quantitative comparison of the results (%) from the literature with our results. Both the **normal and abnormal** images are taken into account. Except for the ORT results, all scores are directly cited from the original paper. All results with # are obtained on the same test set. All results with @ are obtained on the same test set.

Methods \ Metrics	B1	B2	B3	B4	M	R
CDGPT2[4]	38.7	24.5	16.6	11.1	16.4	28.9
Base+RM+MCLN [9]	47.0	30.4	21.9	16.5	18.7	37.1
FFL+CFL-based [13]	**56**	**51**	**50**	**49**	**55**	**58**
HLSTM+att+Dual[8]	37.3	24.6	17.5	12.6	16.3	31.5
Baseline#[17, 14]	28.91	17.35	11.94	8.83	13.79	26.57
Depth=3, w/ RadGlove# [14]	37.40	22.41	15.27	10.99	16.35	30.76
$ORT_{no\text{-}bb}$@	13.12	3.23	1.20	0.50	19.27	18.51
$ORT_{vanilla}$@ [12]	13.12	3.23	1.20	0.50	19.27	18.51
ORT_{up10}@	12.49	3.05	1.14	0.46	17.49	16.84

to the application on regular images. Firstly, the IU dataset has longer captions for each image, but with less variety and on top of that a smaller dataset. This makes it a harder problem to solve for the model. Secondly, for the original application, they used self-critical learning based on the CIDEr language metric. This was not possible for our application as discussed in section 4.2.2. Another language metric could be used to perform self-critical learning, but the implementation was beyond the scope of this project.

We also look at the result split by normal/abnormal subjects. Not all discussed literature provide results with these splits, including the FFL+CFL-based method which had the highest overall scores. First, we will discuss the resulting metric scores for the normal images, see Table 3.

Out of the literature that does provide results for the normal split, the 'Depth=3, w/ RadGlove' model [14] obtained the highest scores, for all but the METEOR metric. The RadGlove model utilizes an encoder-decoder framework, consisting of Google's Inception-v3 model as an encoder and a 3-stage stacked LSTM as a decoder. The main difference is the use of medical pre-trained text embeddings obtained from 4.5 million radiology reports, for the decoder part of their setup. This form of transfer learning gives the decoder a head start during training. The highest METEOR score was obtained by our own model versions $ORT_{vanilla}$ and $ORT_{no\text{-}bb}$.

Again, the ORT_{up10} model performs slightly worse compared to the other two versions. This could be explained by the fact that the upsampling method is trained on relatively less normal images.

Finally, we will discuss the resulting metric scores for the abnormal images, see Table 4. For the abnormal images the 'HLSTM+att+Dual' model [8] performs best on all metrics. It is also the only model that has higher metric scores for abnormal images than normal images. Their model, after weighing whether a certain topic is normal or abnormal, uses two specialized LSTMs. One is specialized in constructing normal sentences, and the other one in abnormal sentences. This seems to be a good technique, especially for the generation of abnormal captions, which still contain a lot of normal phrases.

The best overall performing model, the 'FFL+CFL-based' model, is not available for comparison, because they didn't provide the metric scores split by (ab)normal images.

TABLE 3: Quantitative comparison of the results (%) from the literature with our results. Only the **normal** images are taken into account. Not all papers split their results into normal and abnormal images. Except for the ORT results, all scores are directly cited from the original paper. All results with # are obtained on the same test set. All results with @ are obtained on the same test set.

Methods \ Metrics	B1	B2	B3	B4	M	R
HLSTM+att+Dual[8]	35.7	23.3	16.5	11.8	15.6	31.3
Baseline#[17, 14]	38.17	24.09	16.79	12.30	17.47	32.54
Depth=3, w/ RadGlove# [14]	**43.64**	**28.89**	**21.41**	**16.68**	19.46	**35.56**
$ORT_{no\text{-}bb}$@	17.83	4.82	1.78	0.72	**24.03**	19.73
$ORT_{vanilla}$@ [12]	17.83	4.82	1.78	0.72	**24.03**	19.73
ORT_{up10}@	16.51	4.28	1.65	0.70	21.39	18.00

TABLE 4: Quantitative comparison of the results (%) from the literature with our results. Only the **abnormal** images are taken into account. Not all papers split their results into normal and abnormal images. Except for the ORT results, all scores are directly cited from the original paper. All results with # are obtained on the same test set. All results with @ are obtained on the same test set.

Methods \ Metrics	B1	B2	B3	B4	M	R
HLSTM+att+Dual [8]	**37.3**	**24.6**	**17.5**	**12.6**	**16.3**	**31.5**
Baseline#[17, 14]	9.57	4.55	2.58	1.66	7.58	17.95
Depth=3, w/ RadGlove# [14]	19.42	9.99	5.84	3.79	9.71	20.79
$ORT_{no\text{-}bb}$@	9.87	2.12	0.80	0.35	15.97	17.67
$ORT_{vanilla}$@ [12]	9.87	2.12	0.80	0.35	15.97	17.67
ORT_{up10}@	9.71	2.21	0.78	0.30	14.79	16.04

Again, the second highest METEOR score is earned by two model versions of our own: $ORT_{vanilla}$ and $ORT_{no\text{-}bb}$. As stated before the METEOR score is relatively high, because only one unique caption was generated, which occurs literally in the annotation of the dataset multiple times.

The upsampling model did not outperform the non-upsampling model versions, even though it was trained on relatively more abnormal examples. This is possibly due to the fact that for abnormal images a high portion of the caption text is dedicated to normal observations, as will be discussed further in section 5.3.

5.3 Qualitative Analysis

To further investigate our model, we perform qualitative analysis on four cases. Two normal images were randomly selected from the test set, one with and one without generated bounding boxes. And two abnormal ones, one with and one without generated bounding boxes. In Table 5 we compare the Ground Truth to the model-generated captions of four X-ray images with their possible bounding box information. The colors indicate overlapping medical phrases.

Firstly, we observe that the variation in the generated captions is very low. The $ORT_{vanilla}$ and $ORT_{no\text{-}bb}$ models only generate one unique report caption for the whole test set. The ORT_{up10} model generates two unique captions, as shown in Table 6.

The two generated captions seem to describe subjects without abnormalities well.

TABLE 5: Four example images of which two are considered normal and two abnormal. The output of the three versions of the model is shown below. The medically relevant text is color coded. Light green means normal lungs and dark green means some abnormality in the lungs. Yellow means normal heart and/or mediastinum. Blue stands for 'no acute/active disease'. Dark orange indicates scarring. Dark red indicates issues with the aorta. Purple means problems with the spine. Strike-through text indicates medically wrong information in the model-generated caption. Enlarged versions of the images can be found in Appendix C.

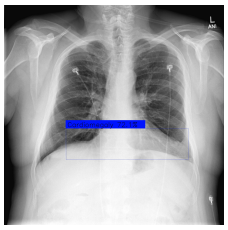
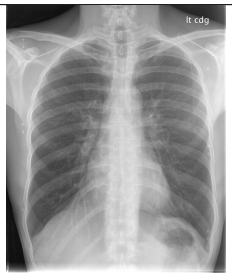
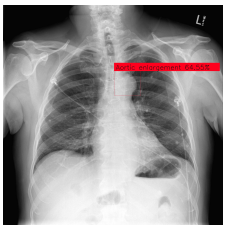
Image				
Problems	normal	normal	Lung ; Cicatrix ; Aorta, Thoracic	Calcinosis; Aorta ; Calcified Granuloma ; Thoracic Vertebrae
Ground Truth	The lungs are clear. There is no pleural effusion or pneumothorax. The heart and mediastinum are normal. The skeletal structures short thready changes of the spine. No acute pulmonary disease.	Heart size normal. Lungs are clear. XXXX are normal. No pneumonia effusions edema pneumothorax adenopathy nodules or masses. Normal chest	The lungs and pleural spaces show no acute abnormality . Lungs are hyperexpanded. Minimal XXXX scarring in both lower lobes. Heart size and pulmonary vascularity within normal limits. Stable mild tortuosity of the descending thoracic aorta. No acute pulmonary abnormality.	There are XXXX sternotomy XXXX identified. The heart is within normal limits in size. The aorta is calcified and tortuous. There are scattered calcified granulomas throughout both lungs. No focal infiltrate pleural effusion or pneumothorax. Mild degenerative changes of the thoracic spine. Stable appearance of the chest. No acute process.
$ORT_{vanilla}$ and **$ORT_{no\text{-}bb}$** model output	the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease	the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease	the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease	the heart is normal in size the mediastinum is unremarkable ~~the lungs are clear~~ no acute disease
ORT_{up10} model output	the heart and lungs have xxxx in the interval both lungs are clear and expanded heart and mediastinum normal no active disease	the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease	the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease	the heart and lungs have xxxx in the interval both ~~lungs are clear~~ and expanded heart and mediastinum normal no active disease

TABLE 6: Summary of the generated output by the three versions of the ORT model.

Label	Generated Caption	$\textbf{ORT}_{\textbf{no-bb}}$	$\textbf{ORT}_{\textbf{vanilla}}$	$\textbf{ORT}_{\textbf{up10}}$
A	the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease	100%	100%	69%
B	the heart and lungs have xxxx in the interval both lungs are clear and expanded heart and mediastinum normal no active disease	0%	0%	31%

This was also reflected by the quantitative results discussed in 5.2. However, none of the generated phrases describes anything abnormal. The first two examples in Table 5 are 'normal' patients, so the generated captions match the Ground Truth quite well. The last two examples do have some issues that are observable on the X-ray image, but in both cases, no phrase mentions something wrong with the aorta or any of the other indicated problems.

The four examples from Table 5, seem to indicate that when a bounding box is detected by the YOLO Object Detector the ORT_{up10} will construct Caption B. This prompted us to investigate further, by plotting the relative frequency of each of the two captions against the number of bounding boxes that were generated, see Figure 9. There doesn't seem to be a correlation between the number of bounding boxes and the generated captions.

In Table 7 we check the number of occurrences of the phrases and full captions generated by the ORT models. We also look at the distribution between the occurrence of normal and abnormal instances. Both full captions A and B occur about fifty times in the total dataset, which makes them the two top occurring annotations.

Phrases A.1 to A.4 do occur a lot more -on average 2.7 times more- in the dataset compared to the phrases B.1 to B.4, explaining why caption A is found as a local optimum by the two $\text{ORT}_{\text{vanilla}}$ and $\text{ORT}_{\text{no-bb}}$ versions of the model, as opposed to caption B. Caption B does, however, have a higher abnormal occurrence percentage, meaning that during the upsampling caption B gets duplicated (8.5 times) more than caption A. After upsampling, phrases A.1 to A.4 still occur 2 times more often on average than B.1 to B.4. This could explain why both captions are generated by ORT_{up10}.

The final observation is that all occurrences of phrases and full captions are part of the annotation of abnormal images, even though they don't seem to indicate that anything is wrong with the patient. This illustrates an issue with the current upsampling setup. Every image that did not have the word 'normal' in the Problems column of the annotation was considered abnormal. However, a significant portion (92) of the problems were annotated with 'no indexing'. During the experimental phase of this project, it was unclear that this didn't have a medical meaning. It meant that during the construction of the dataset, something went wrong with the MeSH indexation[5]. This means that it is inconclusive whether these images contain abnormalities. A large part of these (50) has the same constructed captions as annotation B from Table 6 and 7. If one would repeat this upsampling experiment, the 'no indexing' images should not be upsampled.

A more elegant way to upsample the data would be to balance all problems. This is not straightforward, because one image can have multiple abnormalities. Another elegant way is to use a loss function that focuses training on the hard examples. Focal

[5]MEDLINE® Medical Subject Headings®

TABLE 7: The occurrence of all output phrases in the dataset is shown in the 'Total' column. The distribution of the occurrence between normal and abnormal instances is shown as well.

Label	Phrase	Total	Normal (%)	Abnormal (%)
A.2	the heart is normal in size	227	42.9	57.1
A.3	the mediastinum is unremarkable	134	45.9	54.1
A.4	the lungs are clear	667	56.9	43.1
A.5	no acute disease	129	46.1	53.9
A.	the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease	51	98.0	2.0
B.2	the heart and lungs have xxxx xxxx in the interval	66	1.5	98.5
B.3	both lungs are clear and expanded	106	34.3	65.7
B.4	heart and mediastinum normal	104	35.9	64.1
B.5	no active disease	149	43.2	56.8
B.	the heart and lungs have xxxx xxxx in the interval both lungs are clear and expanded heart and mediastinum normal no active disease	52	2.0	98.0

Loss is an example of such a function for object detection [27].

Additionally to upsampling, one could augment the images slightly while upsampling. For a problem with medical images, one should be a bit more selective with the possible augmentation techniques, as compared to a problem using regular pictures. For example, a common technique is to flip an image horizontally. If you do this with a healthy X-ray image, it would mean that the heart is not on the right side of the body anymore which is a rare condition that should be noted by a radiologist or the captioning model.

6 Conclusion

In conclusion, we modified the Object Relation Transformer as presented by [12] to write reports for medical images. A large amount of time was devoted to the modifications of the implemented Object Relation Transformer and computational memory constraints, so the empirical analysis of the model on this problem is not as extensive as we would have wished.

We hypothesized that the focus on abnormal areas in the image would aid the model in learning to write medical reports. The results of that hypothesis are inconclusive since the amount of generated bounding boxes was low. In order to fully test this hypothesis, the Object Detector (YOLO) should be adjusted to generate at least 2 bounding boxes per image to fully exploit the characteristics of the ORT algorithm. The current YOLO implementation doesn't have a parameter to set the minimum number of generated bounding boxes, only a maximum. This can be circumvented by adjusting the step

where the bounding boxes with a low confidence score are discarded, to make sure the two best bounding boxes are still selected.

Lastly, we are aware that the analysis of a method based on a single dataset is limited. Another dataset could also be used in addition, for instance, the MIMIC-CXR dataset [28]. This is a rather large dataset used in recent papers as a benchmark together with the IU X-ray dataset as used in this work.

For future endeavors, there are a number of potential solutions for the issue with low variety in generated reports. As discussed before, instead of training the OD on a dataset that only contains annotations for abnormalities, a dataset can be used that has annotations of objects in the X-ray images independently from their (ab)normality status. So it would have bounding box annotations for all hearts, instead of abnormal hearts only. To the best of our knowledge, such a dataset does not exist.

Currently, the ORT generates sentences word by word: 1-grams. There are also studies that use bigger n-grams or even separate the different phrases out of which the full annotation consists based on (some) medical knowledge [13, 8]. The little variation in the IU dataset annotation can be used to our advantage with this strategy. There is only a limited amount of ways to say some part of an X-ray is (ab)normal.

7 Acknowledgements

I am very grateful to Elena and Zaheer for providing this thesis subject. It has been an interesting journey and I can say I've learned a lot. Elena, without your patience during the implementation of the model, we wouldn't have come this far. Zaheer, thank you for pointing me in the right direction at the beginning of this project. I would like to thank Twan van Laarhoven in advance for agreeing to be the second reader of this thesis.

Secondly, I want to thank my fellow students in office M2.0.03. Although we all had very different thesis subjects, I feel like we could offer each other a sympathetic ear when things looked hard. And thanks for indulging my Pomodoro timer at the office.

Lastly, I would like to thank my family and friends for offering advice and distractions and lending me ears and shoulders. My favorite question has been *"Mag ik vragen hoe het met je scriptie gaat?"* Maaike, thanks for gently forcing me to work at the university for the last few months. Kas, thank you for all your feedback and support.

I would also like to thank my cat for all the relaxing snoring noises, emotional support, and above all, standing model for the first test with the Object Detection Model, see Figure 12.

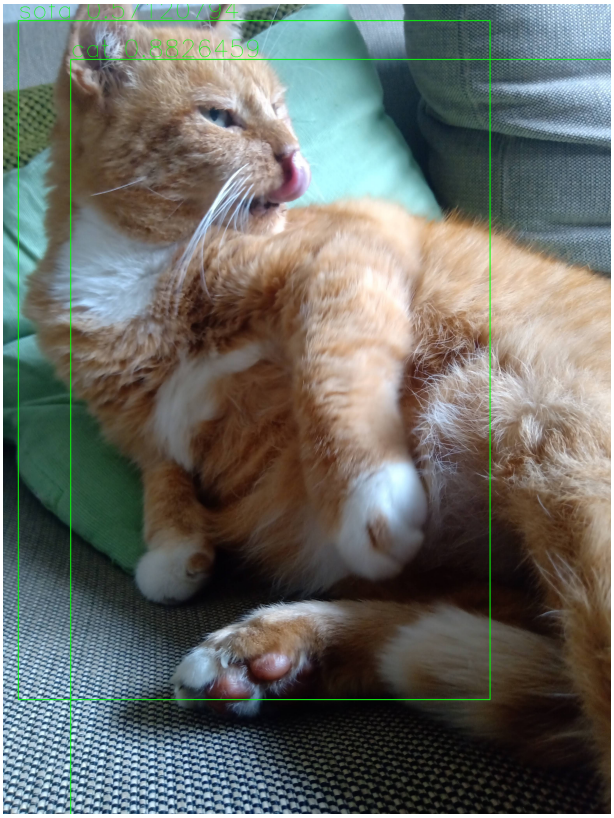


FIGURE 12: Thesis working from home support crew. Detected with YOLOv3 pre-trained ImageNet weights: cat 88.3%, sofa 57.1%.

A Metrics - Examples

In the following example the BLEU_1 and BLEU_4 scores are calculated. The matching uni-grams for BLEU_1 are matched by underlining. The 4-gram is matched with a yellow background.

Candidate: the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease

Reference: the lungs are clear there is no pleural effusion or pneumothorax the heart and mediastinum are normal the skeletal structures short thready changes of the spine no acute pulmonary disease.

$$p_1 = \frac{13}{17} \approx 0.76 \qquad BP = e^{1-\frac{30}{17}} \approx 0.47$$
$$BLEU_1 = BP * p_1 \approx 0.36$$

$$p_4 = \frac{1}{14} \approx 0.071 \qquad BP = e^{1-\frac{30}{17}} \approx 0.47$$
$$BLEU_4 = BP * p_4 \approx 0.033$$

In the following example the METEOR score is calculated for the same Reference and Candidate sentence as above. The matching uni-grams are matched by underlining, we omitted matching based on stemming the words to make the example more comprehensible. The larger n-gram ($n \geq 2$) chunks are matched with an orange, yellow or green background. The total number of chunks are the number of colored chunks summed with the remaining underlined uni-grams (so $\#chunks = 8$).

Candidate: the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease

Reference: the lungs are clear there is no pleural effusion or pneumothorax the heart and mediastinum are normal the skeletal structures short thready changes of the spine no acute pulmonary disease.

$$p_1 = \frac{13}{17} \approx 0.76$$
$$r_1 = \frac{13}{30} \approx 0.43$$
$$F_{mean} = \frac{10 * pr}{r + 9p} = 0.45$$
$$\text{Penalty} = 0.5 * \left(\frac{\#\text{ chunks}}{\#\text{ overlapping uni-grams}}\right)^3 \approx 0.15$$
$$\text{METEOR} = F_{mean} * (1 - \text{Penalty}) \approx 0.39$$

In the following example, the ROUGE-L score is calculated for the same Reference and Candidate sentence as before. The matching LCS is matched by underlining.

Candidate: the heart is normal in size the mediastinum is unremarkable the lungs are clear no acute disease
Reference: the lungs are clear there is no pleural effusion or pneumothorax the heart and mediastinum are normal the skeletal structures short thready changes of the spine no acute pulmonary disease.

$$R_{lcs} = \frac{4}{30} \approx 0.13$$
$$P_{lcs} = \frac{4}{17} \approx 0.24$$
$$\text{ROUGE-L} = \frac{\left(1 + \left(\frac{P_{lcs}}{R_{lcs}}\right)^2\right) R_{lcs} P_{lcs}}{R_{lcs} + \left(\frac{P_{lcs}}{R_{lcs}}\right)^2 P_{lcs}} \approx 0.15$$

B Changes of Original Code

B.1 Object Detector

The most important script of the YOLO object detector is `train.py` in which code is supplied to train an object detector model. This script was designed to train on regular images but was easily adjusted to work on X-ray images with different classes. One minor adjustment was made in this specific script. In line 61-64 of the original code `None, None, None` is returned if some given label is not in the dataset, while all other cases 4 values are returned. When this function was called four output values were expected. We only encountered this bug because there was an issue with the labels in our `config.json` file at some point.
The `train.py` script relies on the `yolo3_one_file_to_detect_them_all.py` script and some changes were made in this script as well. During the preprocessing of the input all images are reshaped to a standard size. Due to rounding errors, some adjustments were made to the `preprocess_input` function. In theory it should still work as intended.

B.2 Object Relation Transformer

A lot of effort was put in making the ORT `train.py` script work. In an effort to increase the number of images per batch, the gc package got utilized unsuccessfully.

The main issue was with the loading of the data, which was done using `dataloader.py`. The `dataloader.py` is called from the `train.py` and `eval.py` scripts. After trial and error, the following changes were made. The image ids should have been strings instead of ints, this is changed throughout the script. *fc_feats, att_feats, boxes*

The `eval.py` and subscripts also had some issues. During training we turned the language evaluation off. Afterwards we did need to evaluate based on language, so we fixed it by providing a new language eval function in `eval_utils.py`. The old language eval function was based on the CocoEval package, for which we couldn't find the right version for our python environment.

schematic figure with all scripts and the order in which to be run, see Figure 5.

C Enlarged X-ray Images

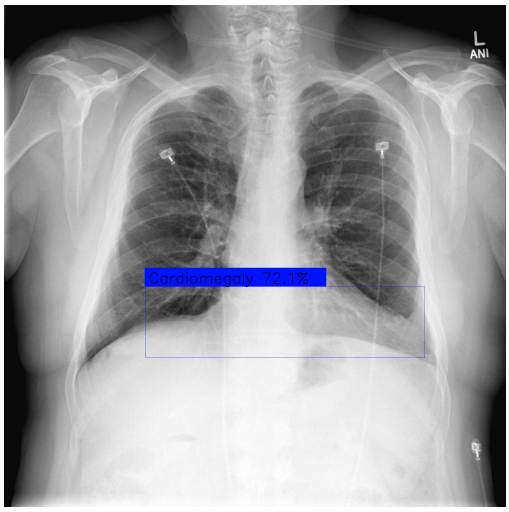


FIGURE 13: X-ray image of patient 1447. The object detector found signs of cardiomegaly (enlarged heart) with 72.1% certainty.

FIGURE 14: X-ray image of patient 2866. The object detector found no abnormalities.

FIGURE 15: X-ray image of patient 351. The object detector found no abnormalities.

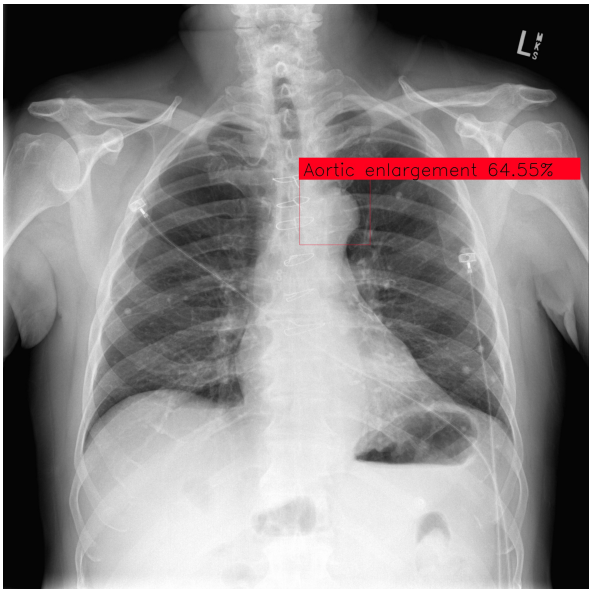


FIGURE 16: X-ray image of patient 1900. The object detector found signs of aortic enlargement with 64.55% certainty.

References

[1] A. P. Brady, "Error and discrepancy in radiology: inevitable or avoidable?," *Insights into Imaging*, vol. 8, pp. 171–182, Feb. 2017.

[2] F. H. Chokshi, D. R. Hughes, J. M. Wang, M. E. Mullins, C. M. Hawkins, and R. Duszak, "Diagnostic Radiology Resident and Fellow Workloads: A 12-Year Longitudinal Trend Analysis Using National Medicare Aggregate Claims Data," *Journal of the American College of Radiology*, vol. 12, pp. 664–669, July 2015.

[3] C. Qin, D. Yao, Y. Shi, and Z. Song, "Computer-aided detection in chest radiography based on artificial intelligence: a survey," *Biomedical engineering online*, vol. 17, no. 1, pp. 1–23, 2018. Publisher: BioMed Central.

[4] O. Alfarghaly, R. Khaled, A. Elkorany, M. Helal, and A. Fahmy, "Automated Radiology Report Generation using Conditioned Transformers," *Informatics in Medicine Unlocked*, p. 100557, Mar. 2021.

[5] L. Berlin, "Accuracy of Diagnostic Procedures: Has It Improved Over the Past Five Decades?," *American Journal of Roentgenology*, vol. 188, pp. 1173–1178, May 2007. Publisher: American Roentgen Ray Society.

[6] J. Pavlopoulos, V. Kougia, I. Androutsopoulos, and D. Papamichail, "Diagnostic captioning: a survey," *Knowledge and Information Systems*, vol. 64, no. 7, pp. 1691–1722, 2022.

[7] J. Rubin, D. Sanghavi, C. Zhao, K. Lee, A. Qadir, and M. Xu-Wilson, "Large Scale Automated Reading of Frontal and Lateral Chest X-Rays using Dual Convolutional Neural Networks," *arXiv:1804.07839 [cs, stat]*, Apr. 2018. arXiv: 1804.07839.

[8] P. Harzig, Y.-Y. Chen, F. Chen, and R. Lienhart, "Addressing Data Bias Problems for Chest X-ray Image Report Generation," *arXiv:1908.02123 [cs]*, Aug. 2019.

[9] Z. Chen, Y. Song, T.-H. Chang, and X. Wan, "Generating Radiology Reports via Memory-driven Transformer," *arXiv:2010.16056 [cs]*, Oct. 2020. arXiv: 2010.16056.

[10] F. Nooralahzadeh, N. P. Gonzalez, T. Frauenfelder, K. Fujimoto, and M. Krauthammer, "Progressive Transformer-Based Generation of Radiology Reports," *arXiv:2102.09777 [cs]*, Feb. 2021. arXiv: 2102.09777.

[11] Z. Babar, T. v. Laarhoven, and E. Marchiori, "Encoder-decoder models for chest X-ray report generation perform no better than unconditioned baselines," *PLOS ONE*, vol. 16, p. e0259639, Nov. 2021. Publisher: Public Library of Science.

[12] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, "Image captioning: Transforming objects into words," *Advances in neural information processing systems*, vol. 32, 2019.

[13] T. Syeda-Mahmood, K. C. L. Wong, Y. Gur, J. T. Wu, A. Jadhav, S. Kashyap, A. Karargyris, A. Pillai, A. Sharma, A. B. Syed, O. Boyko, and M. Moradi, "Chest X-Ray Report Generation Through Fine-Grained Label Learning," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020* (A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, eds.), Lecture Notes in Computer Science, (Cham), pp. 561–571, Springer International Publishing, 2020.

[14] S. Singh, S. Karimi, K. Ho-Shon, and L. Hamey, "From Chest X-Rays to Radiology Reports: A Multimodal Machine Learning Approach," in *2019 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, Dec. 2019.

[15] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.

[16] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Tech. Rep. arXiv:1804.02767, arXiv, Apr. 2018. arXiv:1804.02767 [cs] type: article.

[17] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[18] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-Memory Transformer for Image Captioning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10578–10587, 2020.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[20] Z. Babar, T. van Laarhoven, F. M. Zanzotto, and E. Marchiori, "Evaluating diagnostic content of AI-generated radiology reports of chest X-rays," *Artificial Intelligence in Medicine*, vol. 116, p. 102075, June 2021.

[21] H. Q. Nguyen, K. Lam, L. T. Le, H. H. Pham, D. Q. Tran, D. B. Nguyen, D. D. Le, C. M. Pham, H. T. T. Tong, D. H. Dinh, C. D. Do, L. T. Doan, C. N. Nguyen, B. T. Nguyen, Q. V. Nguyen, A. D. Hoang, H. N. Phan, A. T. Nguyen, P. H. Ho, D. T. Ngo, N. T. Nguyen, N. T. Nguyen, M. Dao, and V. Vu, "VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations," *Scientific Data*, vol. 9, p. 429, July 2022. Number: 1 Publisher: Nature Publishing Group.

[22] D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, S. E. Shooshan, L. Rodriguez, S. Antani, G. R. Thoma, and C. J. McDonald, "Preparing a collection of radiology examinations for distribution and retrieval," *Journal of the American Medical Informatics Association*, vol. 23, pp. 304–310, Mar. 2016.

[23] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 237–242, July 2020. ISSN: 2157-8702.

[24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[25] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

[26] C.-Y. Lin and F. J. Och, "Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 605–612, 2004.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

[28] A. E. W. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, R. G. Mark, and S. Horng, "MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports," *Scientific Data*, vol. 6, p. 317, Dec. 2019. Number: 1 Publisher: Nature Publishing Group.