Master thesis
Computer Science

Radboud University

Leveraging radiology reports for automatic lesion detection in CT scans

Author:
Manuela Bergau
s4543645

First supervisor/assessor:
Prof.dr.ir. Arjen P. de Vries
arjen@cs.ru.nl

Second assessor:
Bram van Ginneken
bram.vanginneken@radboudumc.nl

January 20, 2023

Abstract

Currently, the interpretation of medical images needs to be done by trained specialists, due to the required domain knowledge of the task. Considering the amount of data and the time limitations of the specialists automating this task is highly valuable. Therefore, automatic lesion detection is a popular research task as it would free the resources of medical experts for other tasks. To train lesion detection models a lot of data is needed. The data collection and annotation poses an especially difficult problem within the medical context. Due to the personal nature of medical data, it is rarely publicly available and CT scans are a relative costly medical image procedure. For the annotation of the data we need to have medical experts do the labeling which adds to the costs and whose time is already bound by the daily work in the hospital.

In this thesis we investigate if the medical reports that radiologists write in the clinical context can be leveraged to automate the annotation process. To incorporate the text information within the visual lesion detection task a multimodal model is needed.

We adapt the MDETR model [1] to handle medical data. Although the performance of the proposed model is below the state of the art, we give insights that can be used to build systems supporting medical staff in their daily work and providing annotated data for machine learning projects.

Contents

Chapter 1

Introduction

Currently, the interpretation of medical images needs to be done by trained specialists, due to the difficulty of the task. Considering the amount of data and the time limitations of the specialists automating this task is highly valuable. In this thesis we focus on lesion detection.

Lesion is a broad term for any disruption of tissue.[1] This damaged tissue is visible in medical images (such as X-ray or CT), and needs to be tracked and possibly investigated to learn whether a threat - such as a tumor - is present. CT imaging is one possible modality to determine lesion locations as they give a detailed 3D image of the inside of the body. This level of detail given by a CT scan is an advantage and disadvantage at the same time. The lesions in question are often only a few millimeters in size, making them hard to distinguish within the large number of single images (slices) of a CT scan. This makes the interpretation of the scans time-consuming. Therefore, automatic lesion detection is a popular research topic as it would free the resources of medical experts for other tasks.

To train such models a lot of data is needed. The data collection and annotation poses an especially difficult problem within the medical context. Due to the personal nature of medical data, it is rarely publicly available and CT scans are a relative costly medical image procedure. The cost is not only quantified in terms of the equipment and time needed to set up a scan but also the high radiation load on patients. Therefore CT scans are only done when necessary.

Annotation of the data poses another problem as we need to have medical experts do the labeling which adds to the costs and whose time is already bound by the daily work in the hospital. At the moment the quality of the available annotations at the Radboudumc is poor. Improvements of the annotations will be highly useful for other research projects. This problem is not limited to the Radboudumc. New ways to generate annotated data will be highly beneficial for further research in the field of automatic lesion

[1] See: `https://medical-dictionary.thefreedictionary.com/lesion`

detection in general.

In this thesis, we therefore investigate if the medical reports that radiologists write in the clinical context can be leveraged to automate the annotation process. We explore if the addition of textual data has a positive influence on the lesion detection task. Also, we test how using image data from different sources (hospitals) influences the detection sensitivity.

As medical reports are already produced, using this data would not add to the workload of a radiologist. Most information that would be needed to annotate a scan is included in the report, alas in a free-text format. Therefore some form of processing is needed to structure the information. For that we use a Named Entity Recognition (NER) tagger which is able to tag the information in the reports that is important for this task.

Our model is based on MDETR [1] which addresses the problem of object detection and labeling for general object detection by combining image data with the captions describing the image. To match this approach with our problem we split the reports in sentences and the CT scans in slices. The reports already contain information regarding which slices of the scan the radiologist is referring to. With this information we can filter the relevant slices and thus reduce the amount of data we need to process, as well as reduce the false detection rate (as we only select slices where a lesion is present). Also, we reduce the original 3D CT data to 2D slices.

Using this approach enables us to combine the two modalities and address the detection problem as well as the labeling of the CT slices using a single model. The resulting pipeline produces bounding box annotations that can be used for further research. The automatic annotations are easy and quick to validate by hand compared to annotating completely by hand.

The main contributions of this thesis are:

1. Adjusting an existing multimodal model to work within a medical context
2. Building of an annotation pipeline
3. Analyzing the importance of the data origin
4. Comparing the influence of the used loss functions used for object detection and lesion detection
5. Investigating the benefit of using a multimodal model over a classic detection model for lesion detection

Chapter 2

Related Work

As our problem consists of multiple aspects we look at each of them individually in this chapter. We look at the state of automatic lesion detection in medical imaging, multimodal models within a general context and the medical domain, analyze the advantages and disadvantages of end-to-end models and consider difficulties that are particularly important when labeling medical data.

In Section 2.6 we lay out the concepts of the literature that form the basics of our model.

2.1 Lesion Detection

The object detection task is defined as the detection of object instances from several classes in a given image [2]. In cases where objects from multiple classes are detected it is often combined with the classification of the detected object (discussed in Section 2.2). Object detection in general images and photographs is a task that is extensively researched, however when transferring those methods to the medical domain special care is needed. Lesions are any tissue disruptions. Those lesions can be for example tumors or wounds that need to be observed or the result of any other condition that leads to damaged tissue. Most lesions are only a few millimeter in size. This is one of the biggest differences to general object detection, where the majority of objects cover large parts of the image (as they are the focus of the image when it is created). Another considerable difference is the visual properties of the images. Normal images are usually RGB images while medical images use different systems depending on the modality.

Many research projects are done on computer tomography (CT) images as it is the most universally applicable method of detecting lesions. However, other methods such as PET-CT, X-rays for lungs, MRI for the prostate or the brain are used as well. CT images are made by measuring the density of bones and tissues by using the same technology as X-ray images. In contrast

to 2D X-ray images, during a CT scan the X-ray source is rotated around the body. With each rotation an image slice is constructed; the slices are then stacked together to create a 3D view.[1]

As reliable automatic lesion detection can have a great impact on the healthcare system, there are numerous projects researching different aspects of the problem.

CT scans are 3-dimensional, but 3D models are complex, one aspect of research is the incorporation of 3D context into the model. In the following part we discuss some approaches that incorporate context information without processing the whole 3D scan.

Yan et al. [3] use the slice below and above the slice of interest to mimic the RGB structure. This makes it possible to use models designed for RGB images as backbone. The feature maps are then concatenated and used together with the output of the region proposal in the position-sensitive region of interest pooling. The feature maps provide a 3D context to the proposed regions.

Cai et al. [4] follow a similar approach of using three slices where the middle slice is the target image. To fuse the feature maps, a series of convolution layers is used. At each convolution stage the feature maps are concatenated and upsampled for the final detection.

In contrast to these two approaches Lung et al. [5] do not include 3D context. The proposed ROSNet aims to extract robust features by adapting a pyramid scheme of upsampling and downsampling. These high- and lowlevel features are concatenated before the detection stage. This nested structure enables the detection of small lesions.

Yan et al. [6] apply the feature pyramide scheme as well as the three-channel 3D context applied by [3].

Another problem when working with medical images are noisy images. Due to the general small size of lesions, denoising images without loosing lesion information is challenging. Additionally, measuring the quality of denoised medical images is cumbersome. To deal with these problems Chen et al. [7] proposes to connect the denoising task with the detection task. This ensures that the denoising network's ability to preserve important features is improved.

2.2 Joint Learning vs. Separate Training

In cases where both detection and classification of objects in images is required, for example lesion detection combined with lesion type classification, two main strategies are used. The traditional method trains a detection model, of which the output is fed into the classification model ([8; 9; 10; 11]).

[1] `https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct`

With increasing computing power, researchers often use end-to-end models that perform both tasks simultaneously ([12; 13]).

End-to-end models have the advantage that the performance of classification is not completely dependent on the performance of the detection task, as in a two stage model where the classifier learns from the output of the detection model. The end-to-end model can also increase the performance of both tasks as it can identify information in the data that is important for both tasks. For example, features such as shape and color are not only important to find the object's position in the picture, but for the label as well.

However, training two separate models is not as resource intensive as end-to-end models, as end-to-end models are generally more complex. With increasing complexity more data is needed to train end-to-end models [14]. The growing need for data poses a challenge in itself as the data annotation can be difficult to scale.

We use an end-to-end model in our experiments, the details can be found in Section 2.6.5.

2.3 The medical label problem

With the growing use of deep models within the medical domain, larger labeled datasets are needed. Labeling medical data by hand poses a challenge as medical experts are expensive and hard to come by for this task. The process itself is not scalable to huge datasets. Secondly, classification problems often have to deal with the long tail problem. This means that there are many rare cases, and for those rare cases only limited examples are available for training a machine learning model.

To increase the dataset sizes and heterogeneity while reducing the annotation costs, automating the process is necessary. To automate the labeling process, different approaches exists to include other information that is generated within the medical context. Such data can be annotations on medical images (used by Yan et al. [15] for DeepLesion) or radiology reports that are written by radiologist for the purpose of the diagnosis (used by Yan et al. [16] for LesaNet). The two approaches differ not only in the data used but also in the objective. Yan et al. [15] created DeepLesion to investigate the lesion detection task with a large database. Therefore, it has only bounding box annotations. LesaNet gathers not only bounding box annotations, but class labels as well. The following outlines both labeling approaches in more detail.

Deep Lesion is a public dataset created by the National Institutes of Health's Clinical Center. The CT images in the dataset come from 4,400 unique patients and have ∼32,000 annotated lesions [15]. The dataset provides annotation on a single key slice only, but provides 30mm extra slices

above and below the annotated slice. The bounding box labels of DeepLesion where gathered using bookmarked images. These bookmark annotations are created by the radiologists routinely to mark important findings (e.g. as reference points). Those bookmarks include the position and diameter of the lesions. These measurements are taken along the longest diameter of the lesion and the diameter perpendicular to the longest diameter, along the measured plane. The measurements are then transformed to bounding box coordinates by enclosing the measurements with a rectangular box including a 20 pixel padding in each direction.

LesaNet [16] tackles the annotation problem by defining a hierarchical ontology based on RadLex [17] which is a radiology lexicon. The defined ontology contains the categories 'body parts', 'types' and 'attributes' for lesions. The reports the authors are working with contain bookmarks for lesions. Bookmarks are hyperlinks that are inserted by the radiologist in the report linking to the image of interest. Such sentences, containing one or more bookmarks in combination with the ontology, are then used to mine relevant labels for a given lesion. Given a lesion image, the LesaNet model is able to predict those labels even for rare cases and retrieve lesions from the database with the same labels (but not necessarily the same appearance).

It is important to note that neither the DeepLesion nor the LesaNet approach produce fully annotated images, as in both cases it relies on the annotations that are made within the diagnosis process. The goal of the annotator in this case is to mark interesting lesions and not all lesions. This limitation needs to be kept in mind when analysing negative predictions made by a model.

2.4 Deep Lesion state of the art

Due to the amount of data, DeepLesion is a popular dataset for research used for example by [6; 5; 3] previously mentioned. The performance of these models is hard to compare, as the metrics used are not directly comparable. For this reason we only name the metrics of models we can (approximately) compare.

The MULAN model by Yan et al. [6] achieves an average sensitivity of 86.12 % at the detection task on the test set of DeepLesion. The 3DCE model by Yan et al. [3] measures sensitivity at various false positives (FPs) per image and compares it to several baseline models. Their best performing model achieves 80.7 % sensitivity at 2 FPs per image using 27 images for context. Li et al. [18] MVP-Net improves 3DCEs results to 87.6 % sensitivity at 2 FPs per image, while only using 9 slices for context.

2.5 Multimodal detection

In their survey on multimodal learning with transformers, Xu et al. [19] state that transformers have many advantages, such as scalability in modelling different modalities and tasks. These advantages make transformer architectures a prominent choice for multimodal learning for numerous applications. Each modality has its own properties and may carry different parts of information, combining these can lead to a more robust model. Various examples exist that show the capability of multimodal learning for different tasks [20]. As there seems to be no clear definition of the term 'multimodal learning', we limit ourselves to research that combines text and image modalities as input for the purpose of a joint embedding. This strategy can be utilized for various downstream tasks. Among the most popular tasks are (zero-shot) classification e.g. [21], image captioning [22] and (visual) question answering [23].

Within the non-medical domain multimodal learning has shown its strength for zero-shot classification [21]. Rare diseases are often not present at all during training (for the lack of training data), so this is especially interesting in a medical domain, where the variety and scarcity of training data is often a challenge (see Section 2.3).

Naturally, multimodal learning is explored for multiple tasks within the medical domain as well with promising results ([24], [25], [26]).

Li et al. [24] explores the existing vision and language models LXMERT, VisualBERT, UNIER and PixelBERT and their ability to adapt to chest X-ray findings classification. The authors find an improvement compared to CNN-RNN models, as well as an advantage of the joint text-image embedding compared to only text embedding.

Zhang et al. [25] propose a pre-training method for image and text encoders that aligns the encoded representation of a given image-text pair. This results in the representation of a related image-text pair being closer together than a unrelated one. Resnet50 is used as the image encoder and BERT is used as the text encoder. The alignment is achieved via a contrastive loss that is computed for both modalities to preserve mutual information. The trained encoders are tested on image classification, image-image retrieval and text-image retrieval. The results show an improvement compared to the baseline models among all tasks, suggesting that the encoders benefit from the alignment with a different modality. It is interesting to note, that after pre-training both encoders can be used separately in downstream tasks and achieve a better performance compared to the single-modality trained encoder.

Müller et al. [26] propose a mathematical framework to align text and image representation of X-rays. They use their proposed framework to introduce a pre-training method 'Localized representation learning from Vision and Text (LoVT)' that extends ConVIRT [25]. The authors show that their

proposed method enables obtaining results comparable to previous work with less training data, if the reports are used. For the alignment, local encodings are used to compute a global encoding. An attention mechanism is used to align the text and image encoding.

2.6 Preliminaries

In this thesis we want to train a model combining textual information with images. We used a model that consists of a text backbone and an image backbone, whose output is merged together for the final prediction (see Figure 2.2). We explored different options for the two backbones. This section introduces the basics of the architectures and concepts we adopted for our purpose.

2.6.1 ResNet

After its introduction in 2016 [27], ResNets gained popularity due to their performance, even with deep architectures. The residual connections that skip one or more layers allow for stacking more layers without losing performance due to the vanishing gradient problem [28]. Therefore ResNets are a popular choice as image backbones.

2.6.2 VGG

VGG, introduced by Simonyan and Zisserman [29], is a convolutional neural network. It is designed to be as simple as possible while having a deep architecture. VGG uses only convolutional layers with a single filter size, together with max pooling. This allows architecture depth of 16-19 layers where the VGG16 variant is a popular choice for image backbones ([16], [30]).

2.6.3 Transformer

The transformer, a neural network architecture introduced by Vaswani et al. [31], has gained increasing popularity due to its performance with various complex problems such as Natural Language Processing (NLP). In contrast to previous approaches, transformers are able to pay attention to dependencies even in long input sequences. This makes transformer models a great choice for all sequence-to-sequence tasks, especially translation. A transformer has an encoder and decoder stage: first encoding the input token by token before computing the output in the decoder stage. Since the introduction there were many variants developed using the same principle and transformer attention building blocks (see Figure 2.1). Among them are models of the BERT family.

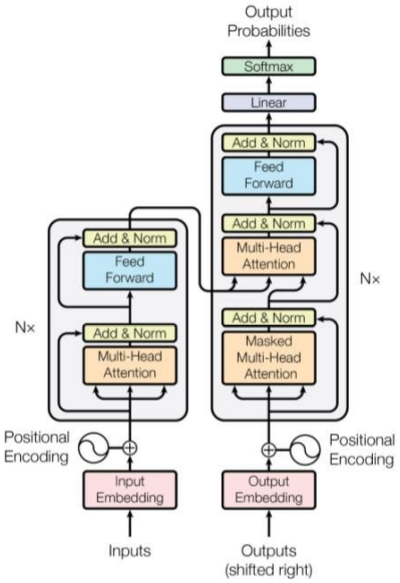


Figure 2.1: The attention block as it was introduced by Vaswani et al. [31]

2.6.3.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) by Devlin et al. [32] uses transformer attention heads to improve over earlier work on various NLP tasks. While the BERT architecture uses the attention mechanism of the original transformers, the input sequence is processed as a whole and not token by token. Furthermore, only the encoder is used.

There are different variants of the original BERT architecture, modifying the number of encoder layers, hidden dimensions and attention heads.

Because the pretraining of BERT models requires not only large datasets but also considerable computation power, there are various pretrained models released to be used in further research projects. This saves resources. In the following we name a few pretrained models that are relevant for our research.

Bertje [33] and RobBERT [34] are two variants of BERT that were trained on the Dutch language. They are set apart by the data and BERT architecture used. Bertje uses a 12GB corpus that is selected based on quality considerations of the content (for example the removal of Twitter data)[33]. RobBERT uses the Dutch OSCAR corpus which is 39GB large [34] and is based on RoBERTa [35] a variant of BERT that refined the pretraining procedure to improve the performance of BERT models.

BioBERT [36] is trained on general English texts retrieved from the English Wikipedia and books as well as English biomedical text corpora collected from PubMed abstracts and PMC full-text articles. The authors

aim to improve performance of biomedical text mining tasks, as biomedical texts contain many domain specific nouns that are not well understood by a general purpose BERT model.

2.6.4 DETR

The DEtection TRansformer (DETR) by Carion et al. [37] is an end-to-end object detection and labeling model for images, using the transformer architecture. Due to the use of Hungarian bipartite matching loss, which assigns each prediction to a ground truth, prediction for multiple objects can be done in parallel. Each predicted bounding box is assigned one class label of a fixed set of labels.

The bipartite matching computes the best pairwise matching of a list of bounding box predictions and ground truths, that minimizes a pairwise cost function $\mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)})$ for a permutation of N elements $\sigma \in \mathcal{G}_N$.

$$\hat{\sigma} = \arg\min_{\sigma \in \mathcal{G}_N} \sum_{i}^{N} \mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)})$$

DETR eliminates the need for special components that encode prior knowledge. Instead a ResNet backbone is used to extract image features (any other CNN can be used as backbone as well) followed by a transformer for the actual labeling and prediction output.

2.6.5 MDETR

We base our own model on MDETR, a multi-modal model for bounding box object detection and labeling by Kamath et al. [1]. In contrast to DETR the labels are not assigned based on a fixed set of labels, but using a caption that describes the input image. Therefore the labeling is not limited to a fixed set of class labels. Only the objects that are mentioned in the caption are detected. How we adapt the base model to work with the medical data can be found in Section 3.1.

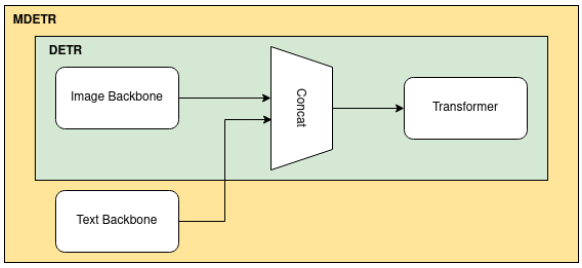


Figure 2.2: The DETR and MDETR model

MDETR builds on DETR [37]. DETR consists of an image backbone with a transformer. MDETR adds a textual backbone to retrieve features from both modalities. See Figure 2.2 for an overview of the components and how MDETR extends the DETR model. The features from both backbones are transposed to a shared dimension after which they are concatenated and input in the transformer. The final output consists of 100 bounding box coordinates, together with a probability distribution per box over the caption to determine the label assigned to the box. To be able to detect less than 100 objects the input caption is appended a token representing 'no object'.

The model is trained using the L1, GIoU (see Section 2.6.6) and contrastive align loss inspired by InfoNCE [38]. The goal of the contrastive align loss is the alignment of the encoding of the two modalities. So that visual tokens and their textual counterpart are closer together than unrelated tokens. Given that L is the maximum number of tokens and N is the maximum number of objects, the loss functions are defined as follows:

$$l_o = \sum_{i=0}^{N-1} \frac{1}{|T_i^+|} \sum_{j \in T_i^+} -\log\left(\frac{\exp(o_i^\top t_j/\tau)}{\sum_{k=0}^{L-1} \exp(o_i^\top t_k/\tau)}\right)$$

$$l_t = \sum_{i=0}^{L-1} \frac{1}{|O_i^+|} \sum_{j \in O_i^+} -\log\left(\frac{\exp(t_i^\top o_j/\tau)}{\sum_{k=0}^{N-1} \exp(t_i^\top o_k/\tau)}\right)$$

T_i^+ is the set of textual tokes that object o_i needs to align. Similarly, O_i^+ is the set of objects that token t_i needs to align with. The temperature parameter τ is set to 0.07. The final contrastive align loss value is computed by taking the average of l_o and l_t. To determine which predictions correspond best to which target the same bipartite matching loss used in DETR is used here as well [1].

2.6.6 Generalized Intersection over Union

Generalized Intersection over Union (GIoU) was first introduced by Rezatofighi et al. [39] and is defined as:

$$GIoU = \frac{|A \cap B|}{|A \cup B|} - \frac{|C \setminus (A \cup B)|}{|C|} = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

It is frequently used as evaluation metric for object detection. The GIoU is an update of the intersection over union (IoU) in an effort to better reflect the difference between non overlapping predictions. The IoU is 0 in all cases where there is no overlap. The GIoU metric has values between -1 and 1,

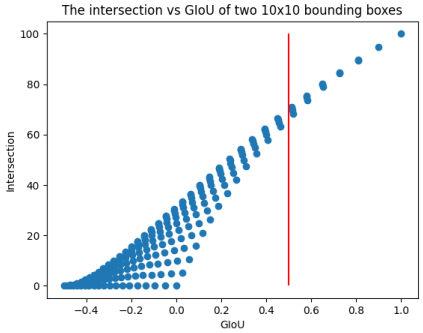


Figure 2.3: A comparison of the intersection and GIoU of two boxes with the same size (10x10). The red line is the GIoU threshold value of 0.5.

where non overlapping bounding boxes have a GIoU value < 0 (note that overlapping boxes can have a negative GIoU value as well).

Nonetheless, the GIoU has its weaknesses. As Figure 2.3 shows there are cases with (nearly) similar intersection area but with a GIoU above and below the threshold. This does not make the metric less useful but we decided to include another metric that is not area based to get a more complete impression of the performance of our models. We discuss this 'center distance' metric in the next section

2.6.7 Center distance

As previously discussed, one disadvantages of the GIoU metric is that it is only considering the overlapping and non-overlapping areas. Therefore, predictions that are equally close to the ground truth can have a different GIoU value. To consider the closeness of prediction and ground truth in our evaluation as well, we also introduce the center distance metric. Which we define as:

$$C\_dist = ln(\frac{|center_{gt} - center_{pred}|}{size_{gt\_lesion} * 0.5} + 1)$$

Considering an example of two bounding boxes of the same size, we can see in Figure 2.4 how the GIoU and center distance change compared to each other. We can clearly see cases where the GIoU is below the threshold but the center distance is the same as cases above the threshold.

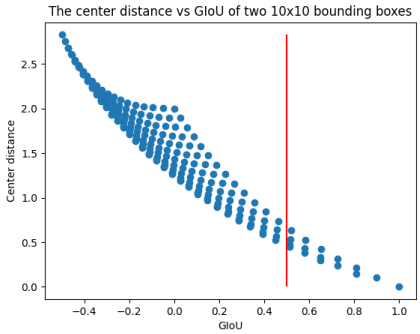


Figure 2.4: Comparison of the GIoU and center distance metric using a 10x10 bounding box.

Chapter 3

Method

In this Chapter we describe the details of our method. We performed multiple experiments where we test various configurations of text and image backbones for the DETR and MDETR models. In Section 3.1 we show the details of the adjustments we made to the base model MDETR [1] to suite our medical data.

We provide the details of the datasets which were used for finetuning. Part of the data is collected from real world medical data and therefore needs to be pre-processed to fit our needs. The details about the pre-processing can be found in Section 3.2. In Section 3.3 we describe the details of the pretraining and training of the models.

3.1 Model Adjustments

We made multiple adjustments to the MDETR and DETR models to better suit our medical data. The original model outputs 100 predictions, which is disproportionate to our data where a maximum of three lesions per slice is present. In theory a slice can contain more than three lesions, but we found that in our datasets one to two lesions is the norm with three lesion being the upper limit (see Tables 3.2 to 3.4 for the number of lesions in each dataset). For this reason we reduced the model output to three predictions.

For the DETR model we had to remove the class loss (cross-entropy loss) as we do not have a finite set of class labels for all our data. This does mean that during pretraining our model will not learn features combining both modalities, which could result in a lower performance compared to the original DETR model. However, after fine-tuning the MDETR model, this problem should be solved.

As mentioned before, there is a considerable difference between medical images and general images that (M)DETR is trained on. To enable the model to better learn specialized features, we only use medical images for training. For the same reason, we adapted the backbones of the model after

testing different options for the image and the text backbone. The image backbone is pretrained using the Deep Lesion dataset instead of (M)DETRs model weights. The details of the pretraining can be found in Section 3.3. For the textual backbone, we use pretrained BERT models that were trained on general Dutch data or English biomedical data.

3.1.1 Image backbone

For the image backbone, we tested multiple ResNet variants and VGG16. Both architectures are frequently used by related work, but their relative effectiveness is unclear and the literature inconclusive.

Yan et al. [15] tested multiple models as backbone that vary in complexity such as VGG16, ResNet50, DenseNet-121 and AlexNet for their model. They found that VGG16 lead to the highest accuracy on the validation set.

Carion et al. [37] reports results of DETR using a ResNet-50 and ResNet-101 backbone. Based on these results the authors of the MDETR model selected ResNet-101 as their backbone, but also tested EfficientNetB3 and EfficientB5 where the performance difference between backbones is small.

Lee et al. [30] used a modified VGG16 in their Single Shot MultiBox Detector for focal liver lesion detection.

Chen et al. [7] used ResNet50 in their Lesion-Inspired Denoising Network in the detection part of the network to extract features before the region proposal.

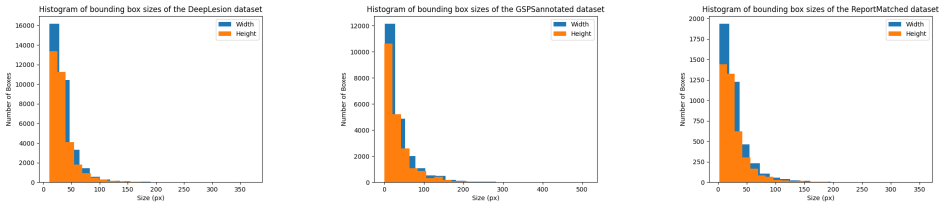
As there seems to be no clear preference, we will investigate empirically the difference between the effectifeness of these architectures for our problem. Additionally, we looked at the difference between networks initialized with pre-trained ImageNet [40] weights compared to random initialization. As Raghu et al. [41] suggest, using ImageNet weights has no significant advantage over random initialisation. However, this does not hold for small datasets where the ImageNet weights are beneficial mostly for larger networks. In contrast to Raghu et al. [41], Zhang et al. [25] found that ImageNet initialisation results in better learned image representations for medical images. Again, we will explore this issue empirically in our experiments in Chapter 4.

3.1.2 Text Backbone

For the text backbone we used BioBERT by Lee et al. [36], RobBERT by Delobelle et al. [34] and Bertje by de Vries et al. [33]. All three models are used in the NER tagger that was developed to process the radiology reports (see Section 3.2.2.2 for details). They all have a similar overall performance but different strengths when it comes to the accuracy of the single tag types, we cannot conclude from the results which model is better suited for our data. Therefore we wanted to test the same models on this task as well.

Dataset name	Source	Number of instances	Number of patients	Modality
DeepLesion	NIH	31972	4400	image only
$\mathcal{D}_{\text{report}}$	UMC	4041	1617	image + text
$\mathcal{D}_{\text{GSPS}}$	UMC	15044	3106	image only

Table 3.1: An overview of the different datasets used.



(a) DeepLesion (b) $\mathcal{D}_{\text{GSPS}}$ (c) $\mathcal{D}_{\text{report}}$

Figure 3.1: A comparison of the bounding box sizes within the three different datasets.

As previously mentioned, BioBERT is pretrained on English biomedical text corpora. While our data is Dutch, we found similarities between the biomedical terms in both languages. During development of the NER tagger (see Section 3.2.2.2) we found only slight differences between the performance of the English model and the two Dutch models RobBERT and Bertje. For the NER tagger we used an ensemble of all three models.

3.2 Data

We first use the Deep Lesion dataset [15], for pre-training (see Sections 3.2.1 and 3.3). We also collected our own data, from anonymized medical data collected at the Radboudumc. We compile two different datasets: The $\mathcal{D}_{\text{GSPS}}$ dataset uses Grayscale Softcopy Presentation State Storage objects (explained in Section 3.2.2.1) to annotate slides from CT scans. The $\mathcal{D}_{\text{report}}$ dataset links a part of the annotated slices with a sentence from the radiology report using a NER tagger.

3.2.1 Deep Lesion

Deep Lesion, as previously mentioned, is a public dataset by the National Institutes of Health's Clinical Center. The CT images in the dataset come from 4400 unique patients and have ∼32,000 annotated lesions [15] (see Table 3.1). As the reports for the CT scans are not publicly available, we

split	number of images	Number of images with 1 lesion	2 lesions	3 lesions	total lesions
train	22390	21981	399	10	22809
val	4777	4684	91	2	4872
test	4805	4711	93	1	4900

Table 3.2: The number of images in each split and the number of lesions on each image in DeepLesion.

split	number of images	Number of images with 1 lesion	2 lesions	3 lesions	total lesions
train	2847	2798	42	1	2885
val	649	635	14	0	663
test	543	536	7	0	550

Table 3.3: The number of images in each split and the number of lesions on each image in $\mathcal{D}_{\text{report}}$.

are unable to use this dataset for the training of the whole model. However, the large number of annotated images makes it perfect to train the visual component of the backbone. The Deep Lesion dataset provides annotation on a single key slice only, but provides 30mm context slices above and below the annotated slice. We sample the slice above and below the annotated slice to get a three channel input data similar to RGB images (see Figure 3.2). This will provide the model with some context of the lesion.

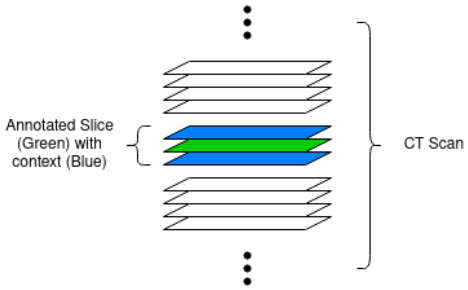


Figure 3.2: Illustration on how we sample the annotated slice together with context.

split	number of images	Number of images with 1 lesion	2 lesions	3 lesions	total lesions
train	9349	9349	0	0	9349
val	2899	2899	0	0	2899
test	2796	2796	0	0	2796

Table 3.4: The number of images in each split and the number of lesions on each image in $\mathcal{D}_{\text{GSPS}}$.

3.2.2 Radboudumc datasets

3.2.2.1 Raw data

We have access to ∼250.000 Dutch radiology reports together with the corresponding CT scans. The data was collected between 2000 and 2021. The reports are written in a free text format. Guidelines prescribe the contents of a report but it is up to the doctor how to write the report. Therefore, no useful assumptions on the structure of the report can be made.

The Grayscale Softcopy Presentation State Storage (GSPS) is a DICOM service that stores additional information that affects the display of an image[1]. We have access to GSPS objects for a subset of the CT scans that we could collect. Included in this information are graphical annotations that are made by the radiologist. There is no information available what is present at those coordinates. The list contains size information as well, however this is not fully reliable, as each GSPS object gets assigned the closest text label, but multiple GSPS objects can get assigned the same same text label. This means that lesions with different sizes can get assigned the same size extracted from the same text label.

To collect text information for our datasets, we process the radiology reports using an NER tagger. The details are described in Section 3.2.2.2. After that we are able to combine the information with the GSPS objects to retrieve our final dataset of image-text pairs ($\mathcal{D}_{\text{report}}$). The process is described in detail in Section 3.2.2.3. The GSPS objects that are not matched with a report sentence are used together with the DeepLesion dataset for pretraining ($\mathcal{D}_{\text{GSPS}}$). Both Radboudumc datasets are split on patient level, using the same split as DeepLesion (70% training, 15% validation, 15% test).

Figure 3.3: Examples of sentences labeld by the NER tagger

3.2.2.2 NER tagger

To process the information in the radiology reports, a NER tagger is developed that is able to tag which words contain which information [42][2]. The following tags are assigned:

- Type: The type of lesion
- Location: The organ or structure the lesion is located at
- Position: The position of the lesion. e.g right, left etc.
- MixedPosLoc: Words that are not explicit position or location information but a mix of both. e.g. para-aortaal
- Slice: The CT scan slice the lesion is visible
- Size: The size of the lesion
- Characteristics: Words describing the lesion

The model uses an ensemble of the pretrained transformer models Bertje, RobBERT and BioBERT which were fine tuned on 1000 hand-labeled sentences. As the models are not able to predict the relation between the tags, the choice was made to work on sentence level only. The same sentence could in theory contain information about multiple lesions, however in reality we saw a limited number (usually 1-3) of similar lesions described in one sentence. The finetuning dataset was sampled from sentences from the radiology reports, that contain slice and size information, using a simple regular expression to guarantee that at least one lesion is mentioned in the sentence. However, this limits the model to a specific sentence structure. As the model is only trained on those positive examples it is likely to be less accurate when presented with different sentences.

[1]See https://dicom.offis.de/dispcons.php.en.

[2]The tagger was developed during an internship, which is not publicly available. Details and the report can be made available on request.

3.2.2.3 $\mathcal{D}_{\text{report}}$ dataset

To create the dataset, we are especially interested in the 'Slice', 'Size' and 'Type' tags resulting from using the tagger discussed in the previous paragraph. The slice information consists of the CT scan series and one or multiple image numbers. The image numbers can be compared to the Z-coordinate of the GSPS objects to find candidates for the precise position of the lesion(s) mentioned in the sentence. If multiple possible candidates are found, the size information is compared to each other with an error margin of 0.5mm. We are able to match ∼4000 locations with sentences from the report for 1704 different patients (see Table 3.1). The sentences are filtered with the same regular expression as used with the NER tagger, to ensure the best performance. A random subset of the resulting matches are then hand checked for sanity. This automatic matching speeds up the labeling process significantly. Finally, we sample the slice above and below the found lesion location for context and to get a 3D input image similar to RGB images so no adjustments need to be made to the model in this regard.

3.2.2.4 $\mathcal{D}_{\text{GSPS}}$ dataset

The $\mathcal{D}_{\text{GSPS}}$ dataset consists of the scan slices that contain GSPS objects, but could not be annotated with a sentence from the accompanying report. This dataset contains 15044 scans from 3106 different patients (see Table 3.1), about half of the DeepLesion samples but from a similar number of unique patients.

3.3 Training

We want to create a model based on MDETR, but as mentioned previously, initial experiments have shown that the number of samples in the $\mathcal{D}_{\text{report}}$ dataset is not sufficient to train the whole model at once. Also finetuning the existing general object detection model is not an option for the same reason. In the following we discuss a number of approaches tho help overcome this situation. The goal is to utilize the datasets we have as efficient as possible. In this Section we describe the setup for each of the experiments.

3.3.1 Pretraining

We have two datasets without additional text information. To use them to pretrain our model we pretrain only the image component in our model. We refer to this setting as the DETR model as the resulting setup is similar to the model by Carion et al. [37]. As mentioned previously, however, in contrast to the original DETR model, we cannot use the class loss due to the missing class labels. As the second pretraining setting we only train

the image backbone without transformer. This way, we explore various image backbone options and select the best one for the DETR and MDETR experiments.

3.3.1.1 Image backbone only

As mentioned earlier, ResNet and VGG16 are popular choices for image backbones. Depending on the task some suggest that VGG16 achieves a better performance and some ResNet. Therefore, we decided to explore those architectures for our image backbone. Specifically, we test ResNet18, ResNet34, ResNet50, ResNet101 and VGG16. We use the DeepLesion dataset and perform bounding box regression with one output bounding box. We also experiment with the initialisation of the model with ImageNet weights. All backbones are trained using the L1 and generalized intersection over union (GIoU) loss, a learning rate of 10^{-5} with exponential decay and a batch size of 4. Based on the results of these experiments we chose ResNet101 as the backbone for our subsequent experiments (see Section 4.1.1.1).

We used the trained ResNet backbones to finetune the MDETR model with the $\mathcal{D}_{\text{report}}$ dataset. All of those models output identical bounding boxes regardless of the image input. The output bounding box positions change only slightly during training. None of the models was able to learn well enough with the limited data. This shows that pretraining only the image backbone is not enough. This problem particularly shows up with larger ResNet image backbones, which increase the complexity of the model. Therefore, we proceed with training the image backbone together with the classification transformer.

3.3.1.2 Image backbone + transformer (DETR)

Inspired by the DETR paper, which the MDETR model is based on, we pretrain the image backbone together with the transformer. This has the advantage that the transformer is already trained on the image features. However due to the absence of labels we cannot use the class loss that DETR uses and we reduce the transformer size to 2 encoders and decoders to compensate for the dataset size (see Section 4.1.2.1 for details about the transformer size).

Additionally, we performed a loss ablation analysis to compare to the analysis Carion et al. [37] did on their original DETR model. We compare our results with the Carion et al. [37] in Section 4.1.2.2.

Lastly, as we work with data from different sources the question arises how similar this data is and how well models will generalise across the different sources. We test different combinations of training data regarding the resulting sensitivity on the test data:

1. training on DeepLesion only

2. training on $\mathcal{D}_{\text{GSPS}}$ images only

3. mixing 20% of the $\mathcal{D}_{\text{GSPS}}$ images together with DeepLesion

4. finetune the DeepLesion only model on $\mathcal{D}_{\text{GSPS}}$ images

We tried higher percentages of the $\mathcal{D}_{\text{GSPS}}$ dataset, but the results did not change much while the computational power and training time needed increased greatly. We therefore restrict our final analyses reported here to the above mentioned experiments.

For all experiments we train for 300 epochs using a leaning rate of 10^{-4} and 10^{-5} for the image backbone with a learning rate drop after 200 epochs.

3.3.2 Finetuning

Both pretraining settings are fine-tuned using the $\mathcal{D}_{\text{report}}$ dataset. Similar to the image backbone we tested multiple pretrained BERT models as text backbone described in Section 2.6.3.1. We finetune for 20 epochs using a learning rate of 10^{-5}, 10^{-6} and 50^{-5} for the model, image backbone and text backbone respectively with an exponential decay.

3.4 Evaluation

To evaluate our models we use two metrics: Generalized Intersection (GIoU) over Union [39] and center distance. With each metric we focus the evaluation on a different aspect of the overall performance on the task. The GIoU is area based while the center distance looks at the distance between the center points of the bounding boxes. We use the same threshold of 0.5 for the GIoU, as commonly used in the literature. For the center distance we use a threshold of 1. We report the sensitivity for each model using each of the metrics.

Chapter 4

Experiments

In this chapter we present and analyse the results of our experiments as described in Section 3.3. First we explore the two ways of pretraining in Section 4.1. Next in Section 4.2 we analyse the effect of adding the text information to our model. All experiments where conducted on the cluster of the research group. The cluster consists of 25 machines. There are a number of different GPUs available (rtx2080i, gtx1080i, gtx1080, gtxtitanx, titanxp); each experiment gets assigned one machine.

4.1 Pretraining

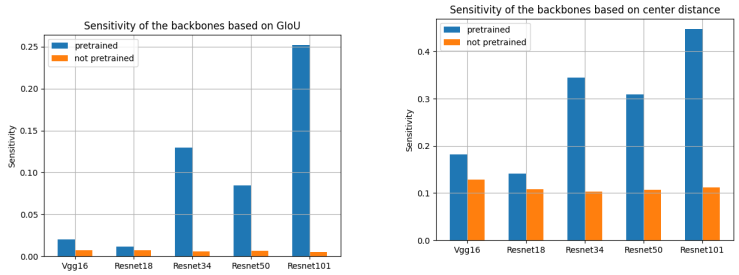
A central challenge with all deep learning models is gathering enough data. This is especially difficult within the medical domain, as gathering (and sharing) medical data always raises privacy concerns. As we were not able to gather as many text-image data pairs as the MDETR model [1] is originally trained on, we explore in this section how we can pretrain the model using image data only. Even with this strategy we still do not have a comparable amount of data. However, it does enable us to closely examine how the model learns.

In this section we present the results of pretraining. We test two settings: training only the image backbone and training it together with the transformer (DETR). We test image backbones from the ResNet family as well as VGG16 all pretrained on ImageNet as well as randomly initialised. Since DETR uses two loss functions, we perform an ablation analysis on the loss functions to gain insight into how the DETR model learns.

4.1.1 Image Backbone

4.1.1.1 Image Backbone only

As we can see from Figures 4.1a and 4.1b, using the image backbones initialised with pretrained ImageNet weights leads to a large increase in sensi-



(a) The sensitivity of the backbones based on the GIoU metric.

(b) The sensitivity of the backbones based on the center distance metric.

Figure 4.1: The sensitivity of the backbones.

tivity compared to random initialisation. These findings are in line with [41] and [25], in that initialisation with pretrained ImageNet weights is especially beneficial for larger networks and small target datasets. The impact of ImageNet initialisation on sensitivity is not as large for VGG16 and ResNet18 as it is for the larger networks. However, all models have an overall low sensitivity with the sensitivity increasing with the model complexity, with an exception being the performance of ResNet34 and ResNet50 where the ResNet34 performs better. This is only a small discrepancy but even with tweaking various hyperparameters we were not able to achieve a score similar to the ResNet34.

As we can see comparing both Figures 4.1a and 4.1b the performance difference manifests not only in terms of bounding box size and overlap as it is tested with the GIoU, but the distance between ground truth and prediction in a similar way. Nonetheless, a higher percentage of predictions that is not considered correct by the GIoU is close enough to the ground truth to be considered useful, where only the shape is still off (examples shown in Figure 4.2).

4.1.2 DETR

As only pretraining the image backbone does not yield satisfying results, we try the approach by Carion et al. [37], to train the transformer together with the image backbone. We analyse the influence of the two loss functions by doing a loss ablation analysis in Section 4.1.2.2. We investigate how the data source and target configuration influences model sensitivity in Section 4.1.2.3 using image data from the Radboudumc (the $\mathcal{D}_{\text{GSPS}}$ dataset that has the same source as the $\mathcal{D}_{\text{report}}$ dataset) as well as data from a different hospital (the DeepLesion dataset). The training time of one epoch of the DETR models took $30-50$ min, resulting in a total training time of $\sim 1-2$ weeks for 300 epochs. Finetuning DETR on a part of the $\mathcal{D}_{\text{GSPS}}$ dataset took 2,5

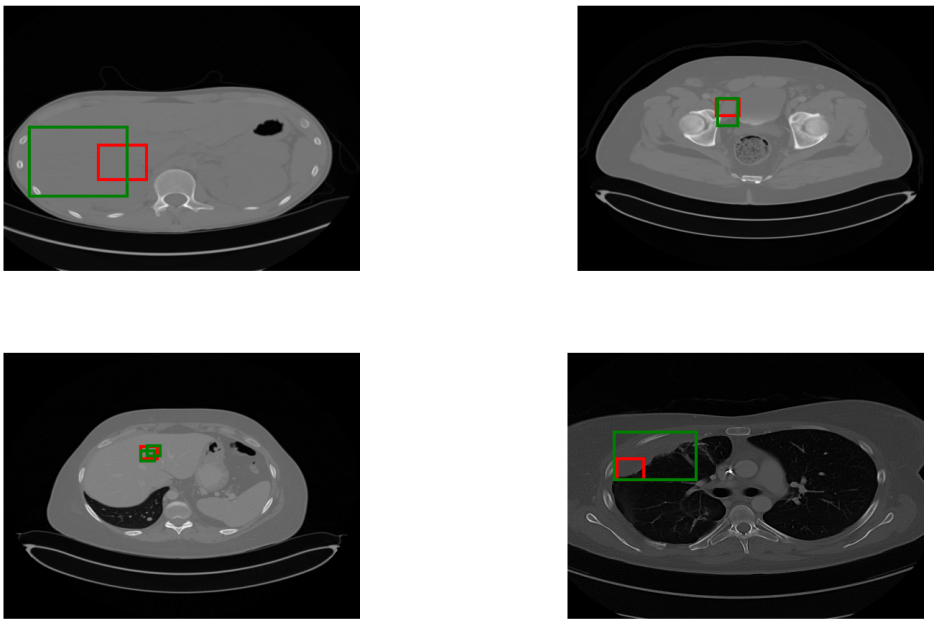


Figure 4.2: Examples of errors made by the pretrained ImageNet ResNet101 model trained on the DeepLesion dataset. Green boxes are the ground truth annotations, red boxes the predictions.

hours in total.

4.1.2.1 Transformer size

The original DETR and MDETR model use 6 encoder and decoder layers. Tests showed that reducing the size to 2 layers each is necessary to achieve a preliminary reasonable performance with our datasets. Models with a larger transformer size output the same bounding box prediction constantly, for every input image. This implies that we either need more data or need to reduce the complexity of the model. The COCO 2017 dataset [43], used to train DETR, has 118K training instances while DeepLesion has 32K instances and our $\mathcal{D}_{\text{GSPS}}$ dataset has no more than 15K instances. We are unable to collect additional data, so we have resorted to reducing the transformer depth to 2 encoder and decoder layers each in all DETR and MDETR experiments.

4.1.2.2 GIoU and L1 loss

The authors of the DETR model [37] report, in an ablation analysis with the two losses GIoU and L1, that the GIoU loss is the most informative. As

	GIoU based		Distance based	
Model	# correct predicted instances	sensitivity	# correct predicted instances	sensitivity
L1 loss only	309	**0.063**	858	**0.18**
GIoU loss only	51	0.01	226	0.05
Both	90	0.018	829	0.17

Table 4.1: the results of the loss ablation analysis using the two evaluation methods.

we can see in Table 4.1 we observe the opposite in our experiment, where the model trained on L1 only performs better than alternative models using GIoU or both.

There are multiple possible explanations for this. First of all, as we can see in Figure 3.1a, the variation in bounding box size is limited in our data and the bounding boxes cover only a small part of each slice. DETR is trained using the COCO dataset [43] where the objects present in the picture vary more by size (for example an elephant in a picture covers a larger portion of the picture than the human next to it). Furthermore, the authors of [37] say that the classification cross-entropy loss is crucial and can thus not be removed during training (as opposed to GIoU or L1 losses). However, we did not use a cross-entropy loss function as we do not have a fixed set of class labels due to the nature of our data and the sentence based labeling (see Section 3.1 for more information).

When considering the example of a picture with a human standing next to an elephant, the combination of class prediction and bounding box prediction can impact each other positively. Predicting the elephant class consequently means a larger bounding box and a larger bounding box has a higher chance of having a class label related to larger objects. The absence of the class prediction in our case could mean that the model needs to rely more on the location information than size which is enforced by the L1 loss. Another aspect to consider is how distinct classes are. In the COCO dataset [43] the objects are structurally more distinct (a human just looks different to an elephant) than in the $\mathcal{D}_{\text{report}}$ dataset.

As noted by Zheng et al. [44], predictions of models that are trained using the GIoU loss initially tend to increase in size, until the target is first touched. After reaching that point in the training process, the bounding boxes shrink to the right proportion, while moving to the correct location. This process results in slow convergence. We observe a similar initial behaviour here. The predicted bounding boxes of the GIoU-only model are quite large, which may very well indicate that more training epochs would

	GIoU based			Distance based		
	DeepLesion	$\mathcal{D}_{\text{GSPS}}$	$\mathcal{D}_{\text{report}}$ (without text)	DeepLesion	$\mathcal{D}_{\text{GSPS}}$	$\mathcal{D}_{\text{report}}$ (without text)
DeepLesion only	0.057	0.006	0	0.225	0.028	0.0288
DeepLesion + 20% $\mathcal{D}_{\text{GSPS}}$ mixed	**0.241**	0.029	0	**0.3**	0.095	0
DeepLesion only finetuned on 20% $\mathcal{D}_{\text{GSPS}}$	0.015	**0.072**	0	0.096	**0.165**	0.0396
$\mathcal{D}_{\text{GSPS}}$ only	0.014	0.004	**0.011**	0.078	0.052	**0.043**

Table 4.2: The sensitivity across the various test sets of DETR trained on numerous data settings

	GIoU based			Distance based		
	DeepLesion	$\mathcal{D}_{\text{GSPS}}$	$\mathcal{D}_{\text{report}}$ (without text)	DeepLesion	$\mathcal{D}_{\text{GSPS}}$	$\mathcal{D}_{\text{report}}$ (without text)
DeepLesion only	0.252	0.001	0	**0.447**	0.015	0
DeepLesion + 20% $\mathcal{D}_{\text{GSPS}}$ mixed	**0.256**	**0.049**	0	0.443	**0.169**	0
DeepLesion only finetuned on 20% $\mathcal{D}_{\text{GSPS}}$	0.025	0.048	0	0.152	0.160	0
$\mathcal{D}_{\text{GSPS}}$ only	0.004	0.003	0	0.083	0.059	0

Table 4.3: The sensitivity across the various test sets of ResNet101 trained on numerous data settings.

be needed for an optimal performance.

Looking at our performance metrics we see that the location which is trained with the L1 loss is usually correct. However, the bounding box size is often incorrect (low GIoU). This supports our observation that the model has trouble working out the bounding boxes. Adding the GIoU based loss only slows down training but does not improve the overall quality of predictions. Therefor we conclude that in our setting it is not necessary to use the GIoU loss.

4.1.2.3 Combining datasets

As Table 4.2 shows there is a large difference in performance across the different test sets. Mixing or finetuning on the other datasets definitely helps the performance on the $\mathcal{D}_{\text{GSPS}}$ set compared to training only on $\mathcal{D}_{\text{GSPS}}$ data. Notably, mixing both datasets has not only an advantage on the DeepLesion dataset over finetuning but also over the model trained solely on the DeepLesion data. Looking at the performance at 2 false positives per image (see Table 4.4) shows a similar picture. This is surprising, especially as we cannot see the same pattern when repeating the experiments using ResNet101 (see Table 4.3).

A possible explanation could be that the distribution of lesion sizes of

	GIoU based			Distance based		
	DeepLesion	$\mathcal{D}_{\text{GSPS}}$	$\mathcal{D}_{\text{report}}$ (without text)	DeepLesion	$\mathcal{D}_{\text{GSPS}}$	$\mathcal{D}_{\text{report}}$ (without text)
DeepLesion only	0.059	0.006	0	0.225	0.028	0.03
DeepLesion + 20% $\mathcal{D}_{\text{GSPS}}$ mixed	**0.245**	0.029	0	**0.302**	0.095	0
DeepLesion only finetuned on 20% $\mathcal{D}_{\text{GSPS}}$	0.016	**0.072**	0	0.096	**0.165**	0.041
$\mathcal{D}_{\text{GSPS}}$ only	0.014	0.004	**0.011**	0.075	0.052	**0.045**

Table 4.4: Sensitivity at 2 FPs per image.

FPs per image	0	1	2
DL only	1	0	0.059
DL + 20% $\mathcal{D}_{\text{GSPS}}$ mixed	0	0.129	0.245
DL only finetuned on 20% $\mathcal{D}_{\text{GSPS}}$	0	0	0.016
$\mathcal{D}_{\text{GSPS}}$ only	0	0	0.014

Table 4.5: Sensitivity at various FPs per image on the DeepLesion test set.

the $\mathcal{D}_{\text{GSPS}}$ dataset includes more larger lesions than the DeepLesion dataset. Even if the DeepLesion dataset contains more instances overall, these tend to be smaller. Combining both datasets together results in a more varied dataset that prevents the model from overfitting. However, the DeepLesion samples are still the largest portion of the data, which explains the larger gain for this dataset. The sensitivity of the mixed model is roughly four times higher than trained solely on DeepLesion. Finetuning on the other hand overwrites part of the features learned with the specific features for the $\mathcal{D}_{\text{GSPS}}$ data. The basic features of the pretrained model help to boost the sensitivity, explaining the higher sensitivity.

Taking a closer look at the sensitivity on the DeepLesion dataset at various FPs per image in Table 4.5 shows that the mixed dataset model and the $\mathcal{D}_{\text{GSPS}}$ only model both are stronger with lower FPs per image compared to the other models. A possible reason is that the DeepLesion dataset has more samples with multiple lesions in one slice.

Nonetheless, all approaches do not give us decent results on the $\mathcal{D}_{\text{report}}$ data. In fact it is zero in all cases except the model trained only on $\mathcal{D}_{\text{GSPS}}$ data. As the images from both datasets originate from the same hospital it is surprising that the performance in this case is so low.

In general the DETR and MDETR models are prone to overfitting, which results in outputting the same bounding box, roughly in the center of the image, independent of the input image. This is similar to the issue we described previously that forced us to reduce the transformer size (see Section 4.1.2.1). The difference is that the bounding boxes during the training of a larger transformers barely change, while in this case they do vary in the beginning of the training process but then soon converge to a single box. To resolve this issue using the $\mathcal{D}_{\text{GSPS}}$ dataset required lowering the learning rate (compared to the learning rate used for DeepLesion). This overfitting issue does not occur when using ResNet101.

4.2 Finetuning

In this section we present the results on the influence of textual data on the detection model. We compare three BERT variants as text backbones and

Text backbone	DETR pretrained		no pretraining	
	GIoU based	Distance based	GIoU based	Distance based
BioBERT	0.0378	0.1045	0.0072	**0.108**
Bertje	**0.054**	**0.1153**	**0.0126**	0.1009
RobBERT	0.0162	0.097	0.0072	0.1027
None	0.0000	0.0467	0.0072	0.054

Table 4.6: The sensitivity of MDETR on the $\mathcal{D}_{\text{report}}$ test set.

compare them to a model without the text backbone. Each of the MDETR models took ~ 10 hour of total training time.

4.2.1 Text Backbone

The results suggest that adding textual information to our model enhances the model sensitivity compared to finetuning on the images only. The pretraining described in the last section proves to be useful. Interestingly, without pretraining the performance of all models is really similar. The only exceptions are the model without text backbone, where the predictions are farther away from the ground truth, and the BERTje text backbone, which is a little better than the rest in terms of GIoU.

It is surprising that the RobBERT backbone sensitivity is lower than that of the BERTje backbone considering that BERTje is trained on a third of the data RobBERT is trained on. Also, RobBERT is based on RoBERTa which is designed to train a BERT model more efficiently. BERTje on the other hand has been trained on a curated dataset that is collected with a focus on qualitative aspects of the texts. As the difference in performance is small, this comparison is informal and no conclusion should be drawn from the difference observed. However, it could be the case that more text-image pair data would increase the difference. It could be an indication that more data is not always favourable over qualitative selection criteria. We see a similar tendency with our image datasets. The $\mathcal{D}_{\text{GSPS}}$ dataset was assembled using a pipeline with as little human interaction as possible. Therefore there was no manual reviewing of all samples. We observed ample noisy points in the form of, for example, multiple slightly different bounding boxes for the same lesion. The $\mathcal{D}_{\text{report}}$ dataset is a subset of these images with a reference in the report. This leads to a substantially smaller dataset while still improving the qualitative aspect.

Despite these uncertainties we can conclude that both the biomedical aspects as well as the Dutch aspects of the text backbones are important. Consequentially, having a BERT model that is trained on Dutch radiology reports could be capable of combining the positive aspects of both text

backbones. See Figure 4.3 for some good predictions of the MDETR models.

Building on the DETR DeepLesion-only model from Section 4.1.2.2 improves the sensitivity of the models with the RobBERT text backbone only slightly, but using the two other text backbones in the models increases sensitivity even more. It is interesting that the DETR DeepLesion-only weights are counterproductive when leaving out the text information. The performance is even lower than that of the DETR $\mathcal{D}_{\text{GSPS}}$ only model (compare Table 4.2) in terms of GIoU. This supports our theory that the accompanying sentences include information that enhances the lesion detection sensitivity of our model.

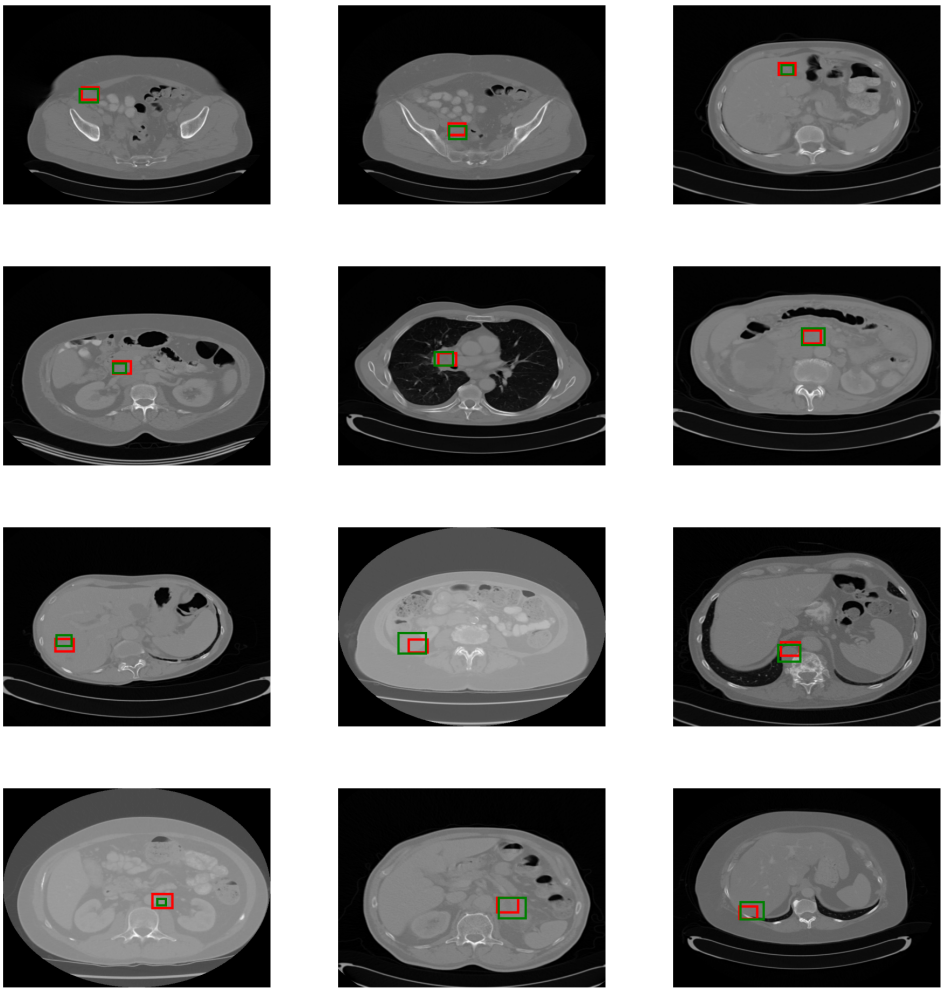


Figure 4.3: Examples of (nearly) good predictions made by the three MDETR models. Green boxes are the ground truth annotations, red boxes the predictions.

Chapter 5

Conclusions and Future Work

In this thesis we have studied how data from different sources and modalities impact the performance of the DETR and MDETR models. The goal was to gain insight in the working of the model to translate the results from the original object detection model to the medical domain. Compared to the previous work using DeepLesion (see section 2.4) the performance is substandard. Despite the overall weak performance of all models we tested, our experiments show that adding textual information indeed boosts the performance compared to an image only model.

We encountered problems with unstable training resulting in a model that makes the same prediction for every image. Due to resource limitations we were only able to use a batch size of 4, while the original model has a batch size of 64, which could not only have influenced the overall performance but explain the stability issue as well.

Our results give various insights in the data, model loss and model details that can be used in future work to improve medical lesion detection models. There are multiple points that could be explored in future research to improve the performance of our models:

GIoU loss: Using the MDETR and DETR model, it would be interesting to explore other loss functions to replace the GIoU, like the DistIoU presented by Zheng et al. [44] who saw a faster convergence and better performance adding the loss to various state-of-the-art models such as YOLO v3 and faster R-CNN for bounding box regression.

Bounding boxes: We reduced the output from 100 bounding boxes to 3 in our experiments. Another interesting approach would be to investigate bounding box fusion of overlapping boxes.

DETR training: The DeepLesion dataset provides lesion type information that could be transformed into class labels for training the DETR model, determining the importance for detection as well as the difference to training on full sentences.

$\mathcal{D}_{\text{GSPS}}$ dataset: Even with careful compilation of the datasets we noticed some noise especially in the $\mathcal{D}_{\text{GSPS}}$ dataset. Currently the GSPS objects coordinates are considered to be center points of a possible lesion. However, GSPS objects can be other forms of annotation such as circles and lines as well. Differentiating between GSPS objects would improve the data quality of the $\mathcal{D}_{\text{GSPS}}$ dataset and the annotation pipeline. The annotation pipeline could not only be valuable for this research but be used as a helpful tool for all research needing bounding box annotations on medical images.

NER tagger: The annotation pipeline for the $\mathcal{D}_{\text{report}}$ dataset could be improved by adding more handlabeled sentences to train the NER tagger [42]. The current training sentences were pre-selected using a regular expression. Thus favouring a certain sentence structure. Labeling more data without the regular expression pre-filtering would result in a more varied dataset that improves the NER tagger reliability, thus make it possible to process whole reports.

ImageNet initialisation: As mentioned previously, within the medical domain small details are often crucial and medical images differ greatly from images from the general object detection domain. Therefore avoiding ImageNet initialisation for a model in the mediacl domain could improve results. However, to achieve that, a dataset of comparable size is needed to train the image backbone. As the ImageNet images are vastly different to the medical images, a network only trained on medical images can learn better domain specific features in the lower layers.

Summarizing, the current model is not fully comparable with the related work. However, if the problems mentioned above are addressed, we expect a better overall performance. The biggest challenge is how to obtain a large and representative dataset. Within the medical domain, a long tail of rare cases makes it hard to collect enough samples for every phenomenon to observe. This limitation of the domain is an overall challenge in medical research.

In a more general direction, further research aiming for a deeper understanding of the information of the text and image modalities and their contribution to detection models would help building more efficient methods combining modalities. As within the medical domain small details are often crucial, translating general object detection research is but a start. As

the data shortage is a continuous struggle for medical research the generation of images is an interesting route of research as well. Chambon et al. [45] for example uses text prompts to generate images of chest X-Rays. In their experiments those prompts are artificially generated. Having real live examples could provide details for the image generator resulting in more accurate depictions of the text description. Understanding the dependencies of the two modalities would help encoding the small details that are often more important in the medical domain that in a general image domain. The generation of images could be used as an augmentation method.

The original MDETR model is not only capable of bounding box detection, but can be trained for other downstream tasks such as segmentation and question answering. Those tasks are of interest in the medical domain as well. Segmentation for example is used to provide measurements of lesions automatically. Fusion of different predictions improve the quality of the measurements [46]. Medical visual question answering could speed up the lookup process of certain structures within an image for medical experts. Combining all possible predictions of the model could enable us to integrate the model within a hospital setting to speed up processes around the diagnosis (looking up lesions, measuring and possible comparing of measurements over time).

All in all, we give insights that can be used to build systems supporting medical staff in their daily work and providing annotated data for machine learning projects.

Bibliography

[1] A. Kamath, M. Singh, Y. LeCun, I. Misra, G. Synnaeve, and N. Carion, "MDETR–modulated detection for end-to-end multi-modal understanding," *arXiv preprint arXiv:2104.12763*, 2021.

[2] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object detection," *Computer Vision: A Reference Guide*, pp. 1–9, 2020.

[3] K. Yan, M. Bagheri, and R. M. Summers, "3D context enhanced region-based convolutional neural network for end-to-end lesion detection," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds. Cham: Springer International Publishing, 2018, pp. 511–519. [Online]. Available: https://arxiv.org/abs/1806.09648

[4] G. Cai, J. Chen, Z. Wu, H. Tang, Y. Liu, S. Wang, and S. Su, "One stage lesion detection based on 3D context convolutional neural networks," *Computers & Electrical Engineering*, vol. 79, p. 106449, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790619301107

[5] K.-Y. Lung, C.-R. Chang, S.-E. Weng, H.-S. Lin, H.-H. Shuai, and W.-H. Cheng, "ROSNet: robust one-stage network for CT lesion detection," *Pattern Recognition Letters*, vol. 144, pp. 82–88, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865521000246

[6] K. Yan, Y. Tang, Y. Peng, V. Sandfort, M. Bagheri, Z. Lu, and R. M. Summers, "MULAN: multitask universal lesion analysis network for joint lesion detection, tagging, and segmentation," *MICCAI*, 2019. [Online]. Available: http://arxiv.org/abs/1908.04373

[7] K. Chen, K. Long, Y. Ren, J. Sun, and X. Pu, *Lesion-Inspired Denoising Network: Connecting Medical Image Denoising and Lesion Detection.* New York, NY, USA: Association for Computing Machinery, 2021, p. 3283–3292. [Online]. Available: https://doi.org/10.1145/3474085.3475480

[8] M. A. Khan, K. Muhammad, M. Sharif, T. Akram, and V. H. C. d. Albuquerque, "Multi-class skin lesion detection and classification via teledermatology," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 12, pp. 4267–4275, 2021.

[9] S. Mehralivand, D. Yang, S. A. Harmon, D. Xu, Z. Xu, H. Roth, S. Masoudi, T. H. Sanford, D. Kesani, N. S. Lay, M. J. Merino, B. J. Wood, P. A. Pinto, P. L. Choyke, and B. Turkbey, "A cascaded deep learning–based artificial intelligence algorithm for automated lesion detection and classification on biparametric prostate magnetic resonance imaging," *Academic Radiology*, vol. 29, no. 8, pp. 1159–1168, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1076633221003779

[10] K. Shankar, A. R. W. Sait, D. Gupta, S. Lakshmanaprabu, A. Khanna, and H. M. Pandey, "Automated detection and classification of fundus diabetic retinopathy images using synergic deep learning model," *Pattern Recognition Letters*, vol. 133, pp. 210–216, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865520300714

[11] M. Usman Akram, S. Khalid, A. Tariq, S. A. Khan, and F. Azam, "Detection and classification of retinal lesions for grading of diabetic retinopathy," *Computers in Biology and Medicine*, vol. 45, pp. 161–171, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482513003430

[12] A. Baccouche, B. Garcia-Zapirain, C. C. Olea, and A. S. Elmaghraby, "Breast lesions detection and classification via YOLO-based fusion models," *Computers, Materials & Continua*, vol. 69, no. 1, pp. 1407–1425, 2021. [Online]. Available: http://www.techscience.com/cmc/v69n1/42797

[13] H. T. Nguyen, H. H. Pham, N. T. Nguyen, H. Q. Nguyen, T. Q. Huynh, M. Dao, and V. Vu, "VinDr-SpineXR: a deep learning framework for spinal lesions detection and classification from radiographs," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, Eds. Cham: Springer International Publishing, 2021, pp. 291–301.

[14] L. Brigato and L. Iocchi, "A close look at deep learning with small data," *2020 25th International Conference on Pattern Recognition (ICPR)*, vol. abs/2003.12843, pp. 2490–2497, 2020. [Online]. Available: https://arxiv.org/abs/2003.12843

[15] K. Yan, X. Wang, L. Lu, and R. M. Summers, "Deeplesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning," *Journal of medical imaging (Bellingham, Wash.)*, 2018. [Online]. Available: https://doi.org/10.1117/1.JMI.5.3.036501

[16] K. Yan, Y. Peng, V. Sandfort, M. Bagheri, Z. Lu, and R. M. Summers, "Holistic and comprehensive annotation of clinically significant findings on diverse ct images: Learning from radiology reports and label ontology," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8515–8524.

[17] C. P. Langlotz, "RadLex: a new method for indexing online educational materials." *Radiographics : a review publication of the Radiological Society of North America, Inc*, vol. 26,6, pp. 1595–7, 2006.

[18] Z. Li, S. Zhang, J. Zhang, K. Huang, Y. Wang, and Y. Yu, "MVP-Net: multi-view FPN with position-aware attention for deep universal lesion detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019. [Online]. Available: https://arxiv.org/abs/1909.04247

[19] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," 2022. [Online]. Available: https://arxiv.org/abs/2206.06488

[20] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[21] M. Xu, Z. Zhang, F. Wei, Y. Lin, Y. Cao, H. Hu, and X. Bai, "A simple baseline for open vocabulary semantic segmentation with pre-trained vision-language model," *Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV)*, 2022. [Online]. Available: https://arxiv.org/abs/2112.14757

[22] J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4467–4480, 2020.

[23] I. Ilievski and J. Feng, "Multimodal learning and reasoning for visual question answering," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/f61d6947467ccd3aa5af24db320235dd-Paper.pdf

[24] Y. Li, H. Wang, and Y. Luo, "A comparison of pre-trained vision-and-language models for multimodal representation learning across medical images and reports," in *Proceedings - 2020 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2020*, ser. Proceedings - 2020 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2020, T. Park, Y.-R. Cho, X. Hu, I. Yoo, H. Woo, J. Wang, J. Facelli, S. Nam, and M. Kang, Eds. United States: Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1999–2004, funding Information: This study is supported in part by NIH grant 1R01LM013337. Publisher Copyright: © 2020 IEEE.; 2020 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2020 ; Conference date: 16-12-2020 Through 19-12-2020.

[25] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz, "Contrastive learning of medical visual representations from paired images and text," *CoRR*, vol. abs/2010.00747, 2020. [Online]. Available: https://arxiv.org/abs/2010.00747

[26] P. Müller, G. Kaissis, C. Zou, and D. Rueckert, "Joint learning of localized representations from medical images and reports," *CoRR*, vol. abs/2112.02889, 2021. [Online]. Available: https://arxiv.org/abs/2112.02889

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[28] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification: An efficient way to train deep neural networks," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 196–207, 2020.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[30] S. Lee, J. S. Bae, H. Kim, J. H. Kim, and S. Yoon, "Liver lesion detection from weakly-labeled multi-phase CT volumes with a grouped single shot multibox detector," in *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018 - 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II*, ser. Lecture Notes in Computer Science, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds., vol. 11071. Springer, 2018, pp. 693–701. [Online]. Available: https://doi.org/10.1007/978-3-030-00934-2_77

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in

Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[32] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[33] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. v. Noord, and M. Nissim, "BERTje: A Dutch BERT Model," arXiv:1912.09582, Dec. 2019. [Online]. Available: http://arxiv.org/abs/1912.09582

[34] P. Delobelle, T. Winters, and B. Berendt, "RobBERT: a Dutch RoBERTa-based Language Model," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 3255–3265. [Online]. Available: https://www.aclweb.org/anthology/2020.findings-emnlp.292

[35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[36] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 09 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz682

[37] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 213–229. [Online]. Available: https://doi.org/10.1007/978-3-030-58452-8_13

[38] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2019.

[39] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union," June 2019.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[41] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," 2019.

[42] M. Bergau, "Stage report," 2021.

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[44] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *The AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[45] P. J. M. Chambon, C. Bluethgen, C. Langlotz, and A. Chaudhari, "Adapting pretrained vision-language foundational models to medical imaging domains," *arXiv preprint arXiv:2210.04133*, 2022.

[46] Y. Tang, N. Zhang, Y. Wang, S. He, M. Han, J. Xiao, and R.-S. Lin, "Accurate and robust lesion RECIST diameter prediction and segmentation with transformers," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, Eds. Cham: Springer Nature Switzerland, 2022, pp. 535–544.