Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Media informatics

Self-Supervised Learning for Cardiac MRI Segmentation

Scientific Students' Association Report

Author:

Benedek Juhász

Advisor:

dr. Bálint Gyires-Tóth
András Kalapos

2022

Contents

Kivonat

Az emberi szív testünk egyik legfontosabb szerve, így a legkisebb hiba/gond is halálos következményekkel járhat. Szív MRI felvételek nagyon hasznosnak bizonyultak a rendellenességek monitorozásában és diagnosztizálásában. Az elmúlt években több gépi látás algoritmust is javasoltak a radiológusok munkájának támogatására. A leggyakoribb kihívás ezeknél a feladatoknál az elérhető adatok alacsony száma, ugyanis egy MRI felvételt csakis egy tapasztalt szakértő tud megfelelően címkézni. Erre a transzfer tanulás az egyik legjobban kivitelezhető megoldás a transzfer tanulás, ami jelentős mértékben támaszkodik a két adathalmaz hasonlóságára. Azonban, az elérhető nagyméretű adathalmazok általában nem tartalmaznak az orvosi képdiagnosztikához releváns képeket.Továbbá, az egészségügyben komoly követelmény a modellek interpretibilitása/megmagyarázhatósága, ami nem a neurális halók erőssége, ez vezetett az eXplainable AI (XAI) kutatási téma megszületéséhez.

A munkám során olyan technikákat és módszereket kerestem amikkel az ImageNet élőtanításnál jobb eredmények elérhetők. Ehhez két state-of-the-art önfelügyelt tanuló algoritmust választottam, amik eredményeit összehasonlítom az ImageNet súly inicializációval, a szív MRI szegmentálás cél taszkon. Betanítottam és összehasonlítottam ezeket a modelleket több adathalmazon is, utána pedig elemeztem a modellek predikcióinak magyarázatait egy legújabb XAI technikával, hogy pontosabban megállapíthassam a orvosi képdiagnosztikában való használhatóságukat. Utóbbi folyamat során nem csak a modellek legjobb predikcióit figyeltem meg, hanem a rosszabbjaikat is.

Mindegyik tanítás ugyanazt az eredményt adta: az egyik önfelügyelt tanulási technika szignifikánsan túlteljesítette az ImageNet előtanítást mindegyik adathalmazon. Továbbá megfigyeltem, hogy minél több tanító adat áll rendelkezésre, annál jobban közelít a kettő teljesítménye.

E félév munkája megmutatta, hogy milyen nagy hatása is van az előtanításnak és, hogy a nem orvosi képfeldolgozás specifikus önfelügyelt tanulási technikák is milyen jól használhatóak e területen.

Abstract

The human heart is one of our most vital organs and a very minor and short dysfunction can be fatal. Cardiac MRI images proved to be very useful in monitoring and diagnosing several heart related disorders. In recent years, several computer vision algorithms have been proposed to support the work of radiology experts. The most common challenge these algorithms have to face during development is the data shortage, since an MRI image of the heart can only be correctly labeled by an experienced radiologist. Therefore the best viable option is transfer learning, which relies heavily on the relevance of the two datasets. Unfortunately large, public datasets aren't likely to contain images from the medical field. Another challenge is the medical sector's major major requirement about model interpretability and explainability, which simply put, aren't the strong suits of neural networks given their black-box nature, leading to the birth of the topic of eXplainable AI (XAI).

In my work I look for new methods and techniques for more effective pretraining for medical imaging tasks. I have chosen two state of the art self-supervised learning algorithms, which I compare with the ImageNet weight initialization on the downstream task of cardiac MRI semantic segmentation. I trained and compared these models on multiple datasets, then analyzed their predictions using a state of the art explainability technique to fully get their capabilities for medical imaging projects. During the latter process, not only the models' most accurate predictions were used, but their less accurate ones too.

Each one of the segmentation trainings produced the same results: the model pretrained using one of the two self-supervised learning algorithms, significantly outperformed the ImageNet pretrained model on all dataset sizes.

My contributions show that pretraining indeed greatly effects the performance of the target task and that today's state of the art self-supervised learning techniques, despite not being medical imaging specific algorithms, can make a real difference in the end.

Chapter 1

Introduction

Computer science has been playing a vital part in the development of medical imaging since its early days, so that it could become the essential tool of modern medicine's diagnosis process. However, with the latest breakthroughs in computer vision and artifical intelligence, the new goal is to have new softwares and algorithms supporting the diagnosing process. The reason for this is that although medical experts are highly skilled and need a lot of experience, they are still prone to human error and due to the aforementioned experience requirement, there's a general shortage of such experts in the field. Recommending systems and various softwares supporting them can potentially decrease the chances of mistakes and maybe computers can even find features and signs on medical images that haven't been registered yet. The heart is one of the most vital organs in the human body, even a minor malfunction can be fatal, so it can't be overstated how beneficial a smart diagnosing algorithm could be.

Self-supervised learning has gained a lot of attention in the recent years and many researchers and research labs have put additional focus on this topic nowadays, although there were for example even in 2016[22] publications on the subject (and "Attention Is All You Need [31] gave another boost too). One of the main reasons why it became so popular is the massive amount of unlabeled data available, which this learning method uses for pretraining. A less known source of large amounts of unlabeled is medical imaging, clinics generate surprisingly large amounts of data everyday . Radiologists and cardiologists already have their hands full on the daily basis, so they have limited time for labelling the afore mentioned data and unfortunately they are the only ones who can. Reaching this point, I believe it's getting clear how self-supervised learning comes into the picture, in short this abundance of unlabeled and shortage of labeled data is the perfect fit for self-supervised learning.

Data shortage isn't the only possible challenge to overcome during projects. After training and testing the models, shareholders in specific fields often require some level of assurance of the algorithms reliability. The medical field is no different (even stricter) in this case. Explainable AI (XAI) gets its own share of fame ever since the first intelligent learning algorithms and many useful and promising techniques have been proposed over the years. Explainable computer vision is a difficult nut to crack however, most explainability methods aren't applicable to, due to the problem's inherent computational complexity. On the flipside, the information encoded in the convolutional neural networks layers offers some opportunities. Luckily, a limited number of techniques can be somewhat universally used, for example saliency maps proved quite useful in practice. During my work I used an extension of GradCAM TODO.

In this paper I compared 2 state of the art self-supervised learning techniques with the very popular ImageNet pretraining. The downstream task, on which the 3 methods will be evaluated, is cardiac MRI segmentation. After looking at the numerical results of multiple trainings, using the previously mentioned XAI technique the models' prediction are analyzed from a changed point of view. This paper's chapters are organized as follows: in Chapter 2 we discuss briefly the background of said techniques and then give a general overview of the related works in this field. Chapter 4 comprises the concept of the used self-supervised learning techniques in detail, the architectures used for training and the datasets. After this, I give a brief and clear overview of the aims and goals of my work in Chapter 3. In Chapter 5 I present the results of the trainings and evaluations and draw conclusions in Chapter 6.

Chapter 2

Background and related works

In this chapter I would like to go over the related work and the background of each technology to give the full picture about the motivation of this paper and its significance. There are two sections: the first, SSL or self-supervised learning, a bit longer and more detailed section and the sceond, XAI or Explainable AI. The latter compared to the first section is a bit shorter, but all the relevant materials are included.

2.1 Self-Supervised Learning (SSL)

2.1.1 Background

Neural networks as we all know are black-box models, still researchers were able to get glimpse into some of the workings of these algorithms. Now it's known that the deeper we go into the network's layers, the more task specific features the layers will look for (for instance human eyes) and the earlier the layer, the more low-level features are detected by the layers (like horizontal and diagonal lines). Since the these layers detect the low-level features, they can be ideally extracted and used for another task with transfer learning, freezing their weights and only training the top layers, which are responsible for the detection task specific features.

All of the above sound well and simple in theory, however in practice a number of problems can arise. Just to name one, how can the engineer or researcher accurately determine which of the layers need no more training and can be frozen? As we can see this is yet another hyperparameter that needs optimization during the project. Also, their might be some exceptional cases when the fundamental features extracted don't contain the necessary pieces of information for the target task.

The goals are the same during self-supervised learning, we supply the large number of unlabeled data to the network in order to make its layers learn useful features for the downstream task. So far, neural networks with weights trained on the ImageNet dataset with supervied learning were the popular choice for transfer learning, even in medical imaging projects.

2.1.2 SSL in computer vision

The self-supervised algorithms and techniques published so far can be put into the following categories:

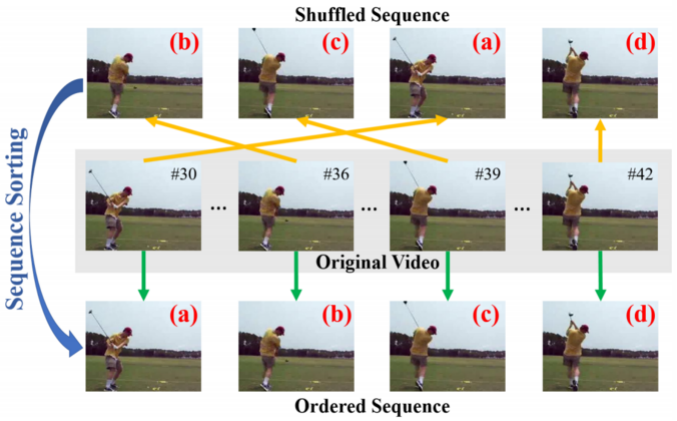


Figure 2.1: Pretext task: putting the video frames into correct order, source: `https://www.fast.ai/images/self_supervised/image3.png`

Figure 2.2: Pretext task: colorizing the image using a generative network, source: `https://www.fast.ai/images/self_supervised/image1-2.png`

- Generative tasks: visual features are learned through generating images, simplest example can be the already mentioned masked autoencoder [16]. Other tasks can be listed here too: image inpainting[24], image colorization[33] (a great example for this task can be seen on Figure 2.2), image super resolution[18] or image generation with Generative Adverserial Networks (GAN)[11]

- Context-based pretext tasks: unlike the previous tasks, the output or label in this case is not an image, rather a class label, a continuous value or it can be a sequence (for frame orders for example [19]). The design of these tasks mainly employ the context features of images or videos, like context similarity and spatial structure. Examples for tasks like this: jigsaw puzzle[22], video frame sequence prediction[19] (see Figure 2.1 for an example), geometric transformation prediction [17], image clustering tasks [3],[4].

- Contrastive tasks: contrastive methods actually share a lot of similarities with the context-based tasks, but the separate place in our list is needed, due to their popularity and success in the last few years. The technique's concept is that two images are fed to the network, but with different transformations(random cropping, rotating, color jitter) and the network's job is to learn representations by contrasting the positive and negative examples. The positive image pair in this case means that the two input images were the same prior to the transformations. Intuitively, a negative input pair originally were different images. See figure Figure 2.3 for better understanding and in the next section I will give further details on these techniques while discussing the Sim Siam algorithm. Examples: the afore mentioned SimSiam[9], SimClr[8], Momentum Contrast (Moco)[15]

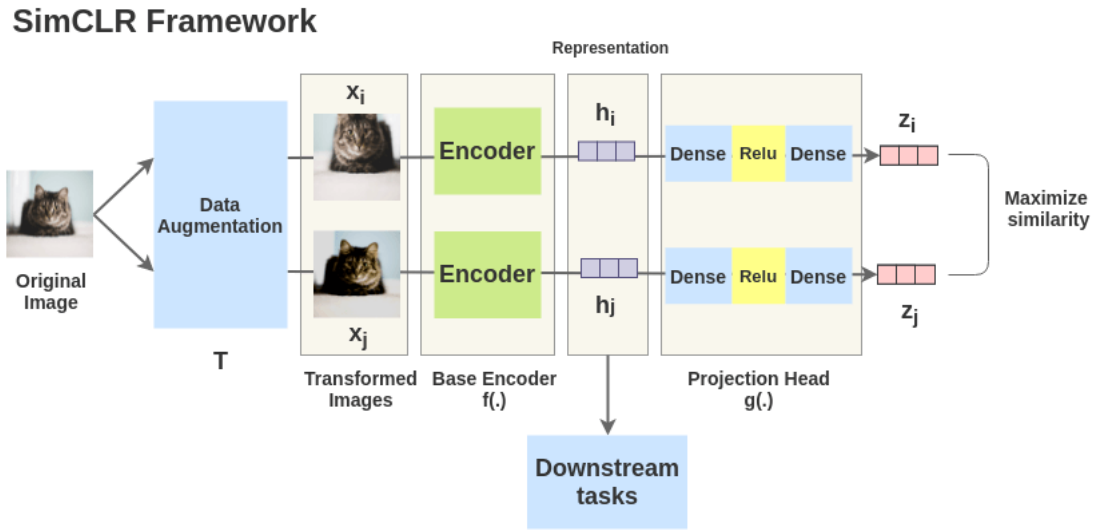


Figure 2.3: Showing one of the contrastive techniques: SimCLR, `https://miro.medium.com/max/1400/0*UNHJt_PyIEfOmQQx`

2.1.2.1 Sim Siam

The name of the algorithm and the title of this subsection stems from the paper's original title "Exploring Simple Siamese Representation learning"[9].

Siamese networks are used quite frequently for tasks such as face/signature verification, object tracking and zero/few-shot learning. The name stems from its architecture, as you can see on figure Figure 2.4, the model is visualized as if it had two identical branches. In reality these are shared weights, there's only one branch, but the model takes two inputs, which are then transformed into feature space and then a fully connected layer creates the final output (either a label of 0 or 1 or maybe a representation in Euclidean vector space), taking the two feature vectors as input (usually prior to that they are simply concatenated).

In siamese networks the contrastive nature is inherently present and their architecture is used in nearly all contrastive self-supervised learning techniques [12], [8],[15]. Mentioning the previous algorithms is quite on cue, because the Sim Siam algorithm is related to all and was developed using the knowledge and other experiences of the earlier works, modifying them slightly. Please see the previous works, unfortunately the length of this paper would explode if it was all included. The main changes SimSiam introduced:

- No negative pairs. Contrary to SimCLR[8] and the previous definition of contrastive learning tasks, SimSiam uses only positive pairs
- SimSiam is basically SwAV[4] without online clustering and BYOL[12] without momentum encoding
- The algorithm works with smaller batch sizes contrary to SimCLR

Interestingly, the main reason for the model's convergence is the stop-grad operation (only the Encoder-Predictor branch is updated during backpropagation). Also, this is the only added structure or operation, which prevents the model from identity mapping.

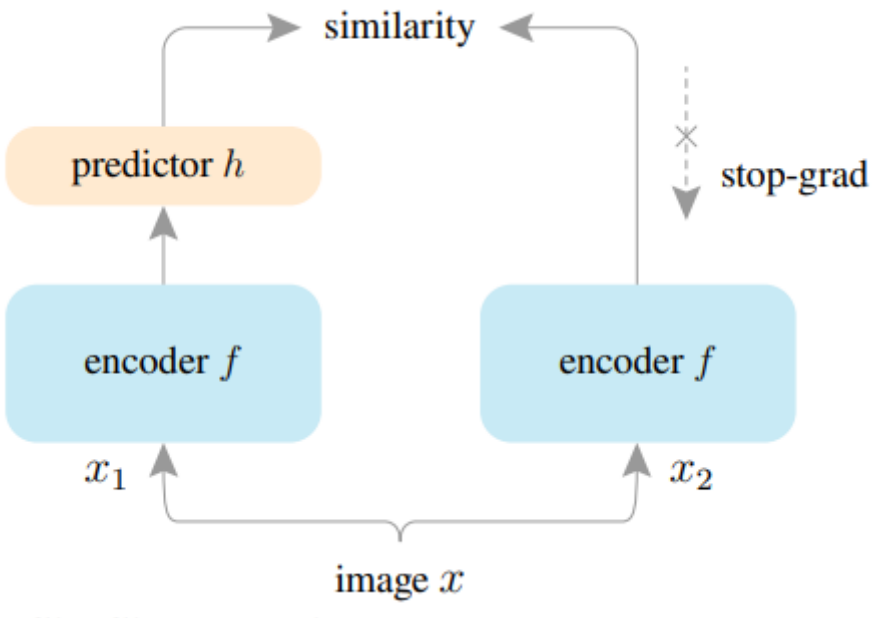


Figure 2.4: Demonstrating the Sim Siam[9] algorithm, the image is from the actual paper

2.1.2.2 Swapping Assignments Between Views (SwAV)

The short name stands for Swapping Assignments Between Views. The task of clustering is actually a well-known and quite popular task in machine learning, which is usually introduced very early during educational courses. The main goal of clustering is to create clusters (groups) of data points which belong together, according to some distance metric. The latter can be the usual and simple Euclidean distance or something more specific like the Manhattan distance. One of the key feature of clustering is that it's usually an unsupervised learning task, when we cluster we usually have no knowledge of the desired output, we only determine which distance metric to use and how many data points to take into account.

The SwAV algorithm differs from the typical contrastive learning techniques in comparing feature vectors, because the former doesn't directly compare feature vectors, rather it uses intermediate codes.These codes are created online from the feature vectors, which the paper interchangeably calls clusters and prototypes. The algorithm consists of the following steps:

- multiple views are created from the input image using a set of transformations and calculating the representations
- calculating the softmax normalized similarities and then iteratively creating the codes
- cross-entropy loss calculation, which is then averaged over the multiple views

2.1.3 SSL in medical imaging

After working on a number of project, on soon begins to experienc that techniques trained and evaulated on the ImageNet dataset don't necessarily mean good results on the special

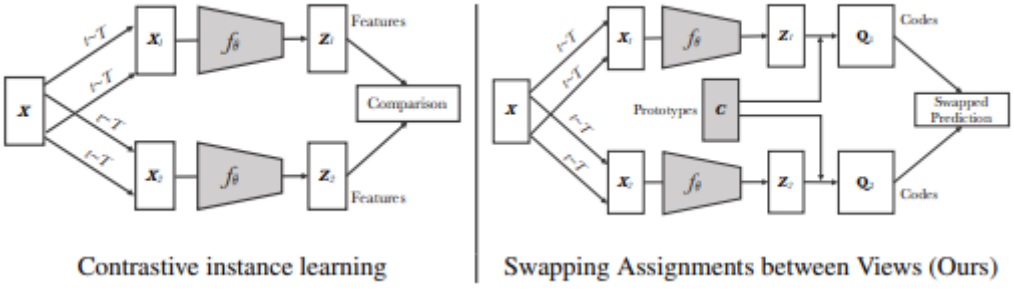


Figure 2.5: Demonstrating the SwAV [4] algorithm, the image is from the actual paperimage source: [4]

image datasets(such is the situation with medical image dataset). Of course this was bit of an overstatement, many in the previous section mentioned techniques achieved significant results in medical imaging. However, much work has been put into finding more task specific methods, more fit for the domain of medical imaging.

A very interesting new technique was the result of taking the originally 2D jigsaw puzzle and with slight modifications, creating the Rubik cube[34] task, which is essentially the original puzzle, but for 3D data. Extending the dimensionality of the original task allowed the authors to pretrain their network for their 3D medical data.

Other very unique solutions and approaches have come to light, for example in the case of [21], the authors used the problem of skull reconstruction on CT images as a pretext task for bone flab volume estimation.

Here's a list of other medical imaging specific pretext in medical imaging, according to the very helpful survey of [30] tasks used for segmentation downstream tasks:

- Slice ordering (ordering the 2D slices of 3D data)
- Anatomical position prediction
- Geodesic distance prediction
- Spatial awareness: contest restoration[7] technique inspired task for medical imaging

Coming back to more "conventional" SSL techniques, the following not medical imaging specific techniques were used for image segmentation:

- Jigsaw puzzle
- Image inpainting
- Image denoising
- BYOL
- SimCLR

After going through the survey and different papers, one can't help but notice that, in very few cases are the constrastive pretext tasks used when image segmentation is the downstream task, classification seems to be a much more preferred choice. It's likely that

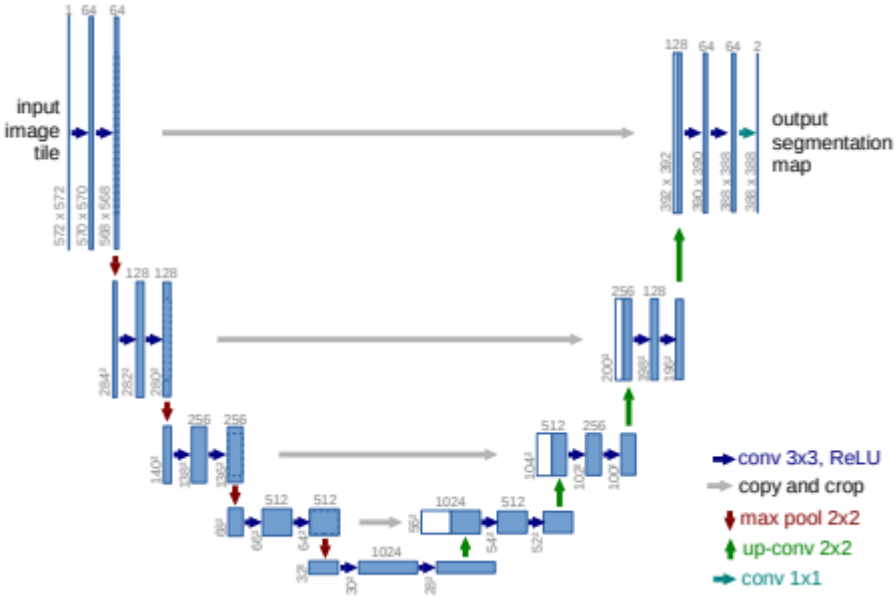


Figure 2.6: The architecture of U-net from the paper[27]

this is only caused by the fact that these are the newest techniques, which is all the more reason for researching and comparing these techniques with segmentation downstream tasks.

Other than the previously mentioned techniques, I would like to mention an inherent problem in finding the optimal pretext tasks in medical imaging: pretext tasks designed for regular images often focus on larger parts of the image, meanwhile in medical imaging the most important parts are the infected or injured ones, which aren't that dominant in size.

2.1.4 Semantic segmentation

In computer vision the semantic segmentation refers to the task of classifying all the pixels in the input image, the simplest example is when we want to segment an object from the background, meaning that basically it's a pixel level binary classification.

Since the model has to classify each pixel, the output format is a one channel image with the same height and width as the input. Generally in deep learning these model are called encoder-decoder networks. The name comes from the fact that just like a simple classifier, the input image is transformed into a feature space vector and then using upsampling(transposed convolutional layers, often incorrectly coined deconvolutional layers) the segmented image is the output.

On Figure 2.6 the standard and widespread U-net[27] architecture can be seen. A few lines back, the more experienced readers could have immediately spotted the problem with general description of the segmenting networks up- and downsampling process. The problem is that during the encoding phase, the fine details are lost due to the nature of the downsampling. Therefore, the creators of the U-net architecture adding skip connections from the encoding to decoding or downsampling to upsampling phase, so the original(less abstract) inputs are available for the latter phase too, resulting in more accurate predictions.

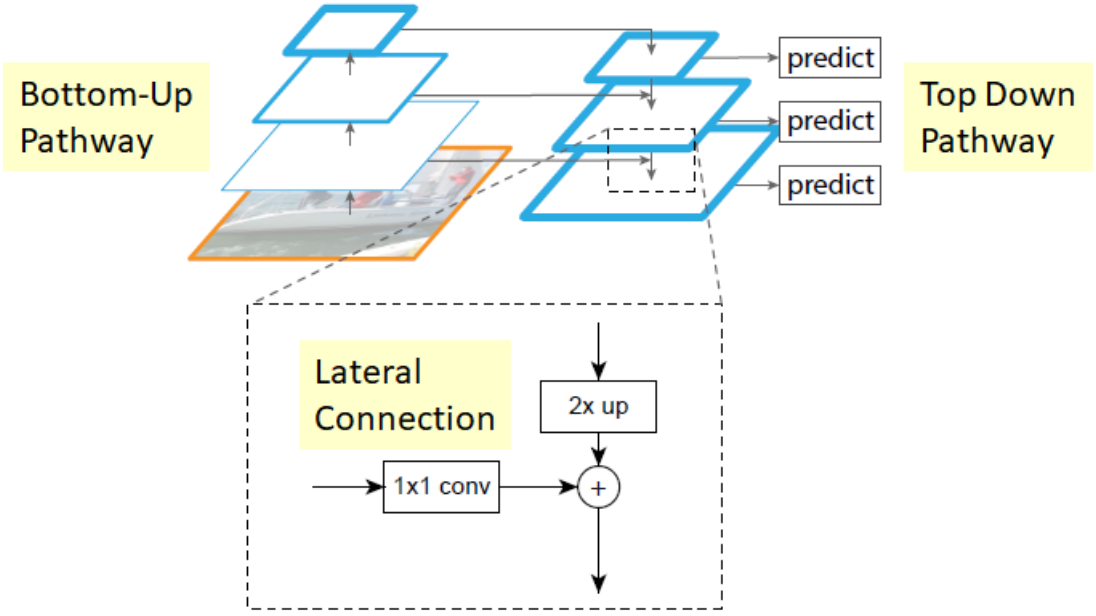


Figure 2.7: Feature Pyramid Network, source: https://miro.medium.com/max/1380/1*D_EAjMnlR9v4LqHhEYZJLg.png

Nowadays the original Unet pipeline have been already surpassed lately by the Feature Pyramid Network (FPN) design. The latter design's properties:

- Combines semantically important, but low resolution features with high resolution ones
- Rich semantics at each level, providing speed and accuracy for inference
- Lateral connections, so at each stage extra information is available

Semantic segmentation models aren't the primary topic of this theis work, so I will conclude this section with a brief mention of the other few notable techniques in the field.

Other popular works in semantic segmentation include: Mask-RCNN[14], a segmentation model with close ties to object detection and a non convolutional arhcitecture [6].

2.2 XAI

Deep learning and its machine learning counterparts have achieved incredible levels of accuracy in multiple problems and fields, yet their reliability is often questioned due to their lack of transparency and black-box nature. The news for example of accidents related to self-driving cars generally makes people doubtful of these algorithms, therefore it is every developer's interest to put heir minds at ease by providing more transparency to their products. Oftentimes it's a specific requirement from the client or other stakeholder in the project to know what the model learns and how the prediction was made. Honestly they have every right for that, especially in the medicine where human lives are at stake and maltreatment can lead to very serious consequences.

I would like to take the time right here to clarify a bit confusing and subtle difference between the interchangebly used interpretability and explainability words. They are very

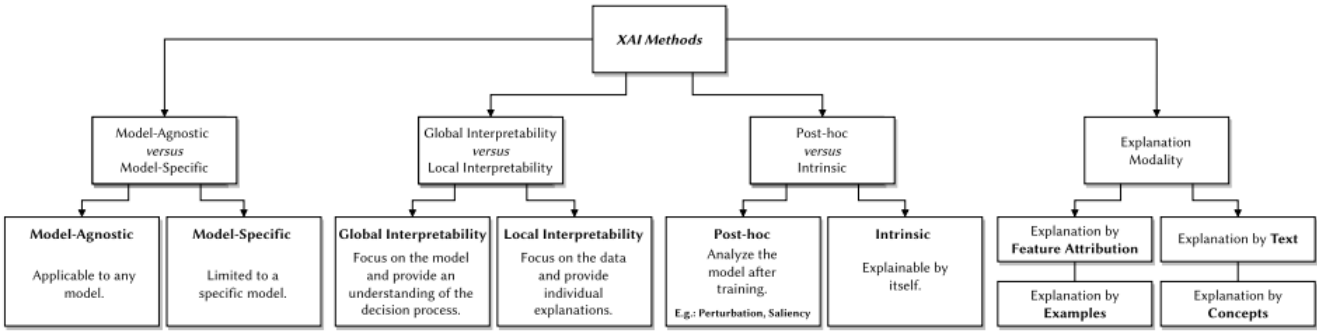


Figure 2.8: XAI categories from [25]

similar in every day speaking, but hide from our point of view very significant differences. Explainability as in the word means explaining, providing/showing reasons why something happened. A relevant to our topic example is highlighting pixels on an input image which affected the model's prediction the most. Interpretability meanwhile refers to knowing how the model works in a general sense, not only for a specific input. [1] refers to interpretability as static characteristic of the model, conversely explainability is a dynamic characteristic of the model.

2.2.1 Categories of XAI

OnFigure 2.8 we can see the proposed categorization of the authors of [25], which in my opinion perfectly shows the different types of XAI methods. Most of the types have quite self-explanatory names, all other non evident details will be covered later.

On that note, a short digression is relevant to address some critical opinions regarding model explainability. In [28] the authors views regarding intrinsic vs post-hoc methods can be said controversial, but from a different point of view are completely understandable. In short it is their firm belief that instead of post-hoc methods (post-training model explaining) universally interpretable machine learning algorithms should be the main goal of this research branch. From my point of view these are very reasonable thoughts and probably many researchers agree too, however in present circumstances there are no accurate white-box semantic segmentation algorithms, so post-hoc methods are left only currently in computer vision. (Interestingly, white-box neural networks have been proposed for image classification: [5].)

The popular post-hoc XAI techniques for computer vision are very hard to categorize effectively, since there are many characteristic, but many methods posess more than one of these and there are also many different names for every proposed category. Nonetheless, it all boils down to two main classes. **pertubation based** techniques are the more intuitive way. The basic idead behind these techniques is to perturbe, change the input data and observe the models's outputs and draw conclusions from that. These are a bit more computationally expensive, but very efficient and often model-agnostic. Another kind of methods are the **saliency maps based** methods, which essentially perform a very similar algorithm to backpropagation and in case of images they calculate the importance of input's pixels using the gradients calculated during the prediction. Reading blogspots and papers other categories are often mentioned, like reference-based or not, or if only gradients are used during computation. Let's see some popular choices from the afore mentioned categories:

Figure 2.9: With green color those areas are highlighted which supported the "Cat" prediction, conversely the red ones didnt. Source:`https://github.com/marcotcr/lime`

- **LIME** its name stands for Local Interpretable Model-agnostic Explanations [26] and it is the perfect example for perturbation based XAI methods. In addition to its advantages, which are all listed in its name, the methodes requires task specific references. Ideally these should be chosen as to minimize the number of activated neurons. For image classification they often use a completely black image. Obviously this choosing is quite difficult for more complex datasets. On Figure 2.9 we can see an example of its output.

- **SHAP:** this is another perturbation based technique, whic evolved from the game theory's Shapely values[20]. It calculates each features average importance by going over their possible values in ther domain. Safe to say, this is a very computationally complex method, works incredibly well for tabular data, however very slow due to the increasing sizes of regular input images.

- **GradCAM:** as the name suggests, this is a gradient-based method originally published in 2017 [29] (since then many modifications and extensions have been proposed, just like the one that we will be used eventually for segmentation explanation [32]). The main disadvantage of this method is in its original description of its paper, where it is mentioned that this is method only works for CNNs, so unlike LIME[26] it isn't model agnostic. In the next paragraphs the algorithm and how it can be used for semantic segmentation will be discussed in detail.

The problem is that the afore mentioned methods share a mutual shortcoming: none of them were originally intended for explaining semantic segmentation. In [32] the authors propose and extension of the Grad-CAM algorithm, by modifying the original formulas.

$$L^c = ReLU(\sum_k \alpha_c^k A^k)$$

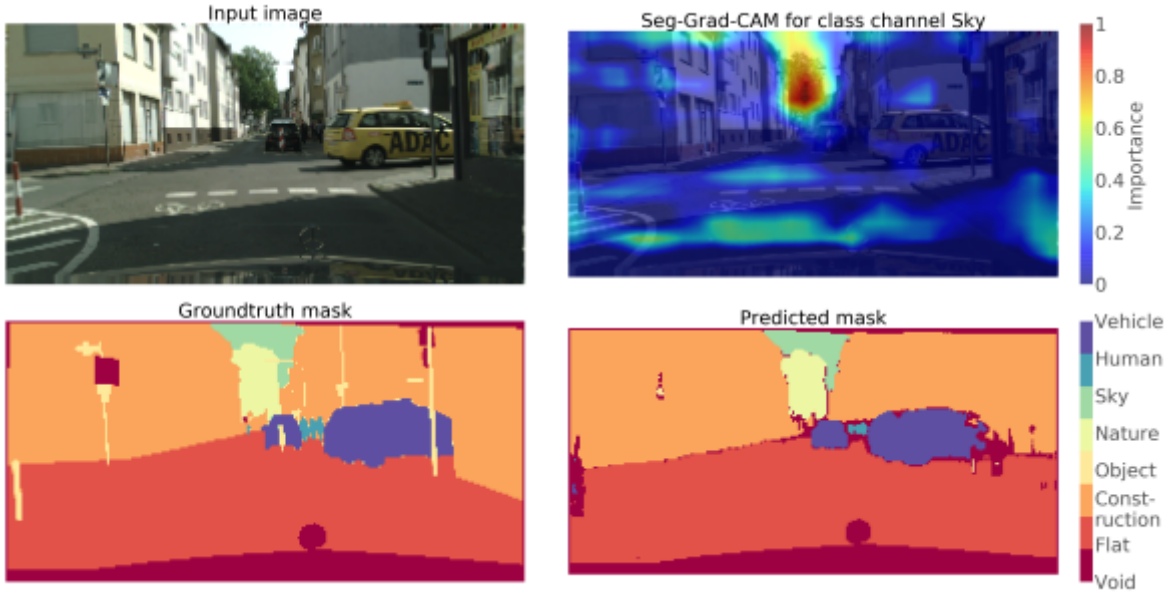


Figure 2.10: Example outputs of a U-net from the Cityscape dataset

with

$$\alpha_c^k = 1/N \sum_{u,v} \frac{\partial y^c}{\partial A_{uv}^k} \tag{2.1}$$

Where A^k is the selected feature map,y^c is the logit for chosen class c. u and v are the indexes of the N pixels, which are used to average gradients of y^c for each feature map, yielding α_c^k as the weight denoting its importance. This is the formula for explaining classification predictions, which is essentially one value at the output, meanwhile a segmentation network produces outputs with N*N sizes. In SEG-Grad-CAM the authors replaced y^c with $sum_{i,j}y^c$, where i and j are the pixel indices of the interested area. On Figure 2.10 we can see an example from the paper.

Authors also checked the heatmaps of each layer in their U-net and corresponding to the common knowledge that first layer's look for low level features. Therefore they proposed to use the last layer of the encoder(often coined bottleneck), since it provided the most humanly understandable information about the prediction.

I used the implementation of this algorithm at `https://github.com/jacobgil/pytorch-grad-cam`.

Chapter 3

Aims and Goals

The overall goals and task of my research is to use and review the current state of the art self-supervised learning techniques in the medical imaging tasks. As mentioned before, one of the greatest problems in medical imaging is the lack of sufficiently labelled datasets due to a number of different reasons. Self supervised learning has the potential of solving the problem by using unlabeled data, which is widely available.

In this paper, my task was to compare the performance of two self-supervised learning techniques with each other, and with a model initialized with the ImageNet weights on semantic segmentation of cardiac MRI scans. My two chosen SSL techniques are SimSiam[9] and SwAV[4], (the ones discussed in the previous chapter).

Models used for semantic segmentation have two parts as we have seen in the last chapter, an encoder and a decoder. One of the questions asked currently in this thesis work is how much of an effect pretraining an encoder has on the performance of the segmentation model. Many doubts can arise since, the encoder typically contains only (at most)half of the parameters of the network(in the cases it doesn't there's always a significant amount of parameters in the decoder), so as we can expect it's not exactly a level playing field.

Chapter 4

Proposed methods

In this short chapter the more training implementation related details are discussed, for example which datasets, models and metric were used. No time like the present, I would like to take this opportunity and clarify that in this work I use the word "image" for 2D data, not multidimensional data.

4.1 Dataset

Just before delving into the details important from the deep learning perspective, I would like to take the time and give a brief overview of the creation of our inputs. Cardiac MRI scans are taken over longer periods of time, in order to counter the effect of the constantly pumping heart, the movement of which largely complicates the 3D image synthesis. Unfortunately, the effects of the latter movement impairs the output's quality(bringing a significant amount of noise), therefore medical experts use and label multiple (at least 2) scans, so the decisions can be made more confidently(either by a medical expert or software). In essence this means that datasets nearly always have more than 1 labeled scan per patient. Also, minor, but not trivial detail: as I mentioned before, the MRI machines and softwares aren't the same everywhere, therefore the 3D scans from two different datasets can have very different dimensions and resolutions, most of the time according to my experience, differing in the third dimensions.

For the self-supervised pretraining I used the ACDC(Automatic Cardiac Diagnosis Challenge)[2] and MSD datasets, in total comprising about 60 MRI labelled scans. The ACDC dataset alone contains scans from 100 patients, 2 labelled scans per patient (with a significant time difference between them) and a 4D data array comprising all 3D scans taken at different times. Taking 2D slices from these scans yields a little bit more than 20000 images, ready for the self-supervised pretraining. The other dataset was the MSD, Medical Segmentation Decahtlon challenge (https://decathlon-10.grand-challenge.org/home/), which consists of 3D MRI scans from 10 different body parts, one of them being the heart. Slicing these 3D scans into 2D arrays yielded another, about 2200 images. In the end, for the self-supervised pretraining over 23 thousand images were ready.

For the downstream segmentation task, I chose the ACDC dataset, owing to its variety (more patient scans are available) and its simplicity. Meaning, the downstream tasks dataset consisted of 2251 images.

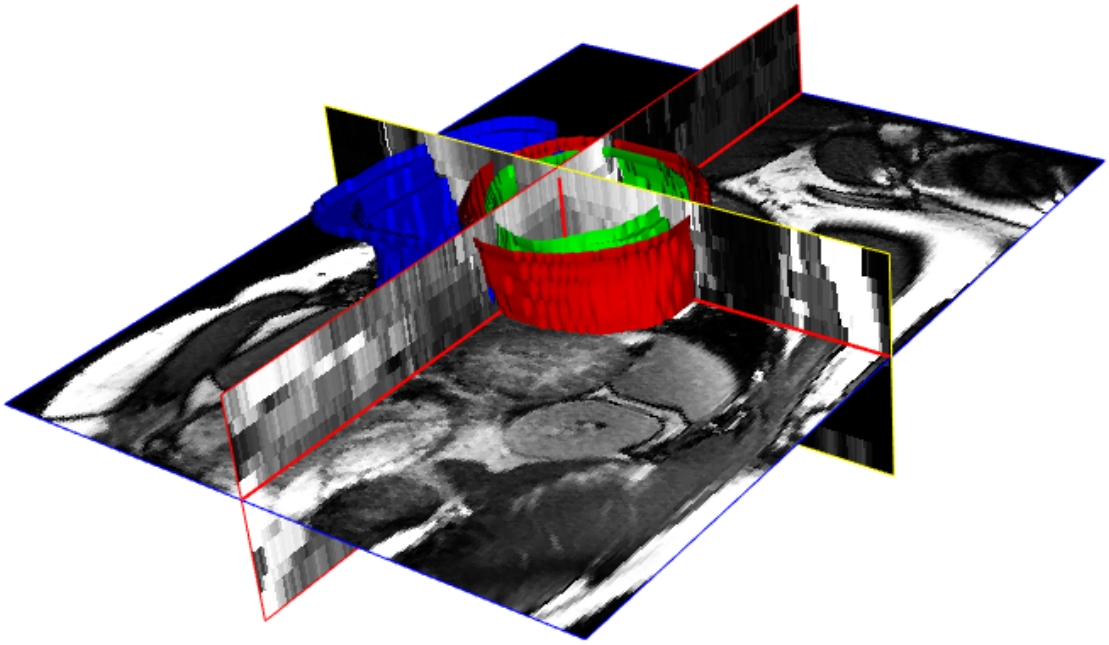


Figure 4.1: A sample image from the ACDC dataset, source: `https://acdc.creatis.insa-lyon.fr/description/files/illustrationChallengeACDC.png`

During the creation of the 2D slices, each image's height and width was converted to 224×224. Despite all of them being black and white images, I chose to use all 3, RGB channels, simply to later avoid any unnecessary error with preprocessing libraries and (so the input of the neural network was 3×224×224) so these images could be fed to the ImageNet pretrained models too. (In my code, at various preprocessing pipelines, there might be further resizes (for eg. to 256×256, this is done so the RandomCrop transformations can be used).

4.2 Semantic Segmentation

The first step in designing the downstream task was choosing the encoder network. Given the complexity of the downstream task and taking into account which architectures perform well in self-supervised pretraining, I chose the resnet18[13] model. With its 11 million parameters it was fast to train and so most likely a deployment process would go easily too.

The other designing step was choosing a right architecture for the complete segmentation network. I chose the Feature Pyramid Network as the segmentation networks architecture, see Chapter 2 for further details.

In semantic segmentation the question of metrics is a tough one and not really not evident at first. Let's start with the loss function, the basic requirement of a neural network. I used the following metrics:

- **Loss function:** as a loss function I chose dice loss. Its name comes from the dice coefficient(F1 score), which is a value used to measure the overlap between two samples. The coefficient's range is from 0 to 1, 1 means perfect overlap.
- **Evaulating models:** intersection over union, aka IoU or Jaccard Index. In short it's the ratio of the area of overlap and are of union, basically we get a score for the overlap if the predicted area and the ground truth.

For semantic segmentation I used the Qubvel library(`https://github.com/qubvel/segmentation_models.pytorch`).

4.3 Implementation

The purpose of this section is to sum up and tie all of the above together in a concise and easy to understand manner, also to touch upon the more code centric implementation details.

The following submodules were created during my work:

- **Data processing submodule:** this module handled the integration, conversion and sorting to different directories of all the gathered data. By conversion here I mean from 3D data to 2D slices (or simply put from NIfTI(`https://radiopaedia.org/articles/nifti-file-format`) format to RGB) and both applied for the inputs and corresponding labels.
- **SimSiam submodule:**every code related to the SimSiam algorithm was gathered here. Following the example of the official Pytorch Lightly documentation (`https://docs.lightly.ai/getting_started/lightly_at_a_glance.html`), I too implemented the creating of positive pairs using the Pytorch library's [23] dataloader functions.
- **SwAV module:**getting the Pytorch Lightning SwAV implementation to work involved less coding, but more understanding compared to the previous algorithm. In the end however using the library, the entire module really became the opposite of boiler plate code
- **Segmentation module:**This module required the most oversight in order to make it able to support all kinds of experimental trainings in the future, regardless of the amount of training data and architecture. Another very important requirement was to ensure that all segmentation trainings (each with a different encoder) use the same validation dataset. I implemented this feature using python's very handy Pickle package, so all indexes of the validation and training datasets are saved.

Fortunately, the SSL libraries all came with bulit-in augmentation pipelines, the default configuration of which seemed perfect for the task. Unlike these libraries, the Qubvel(the library used for semantic segmentation) has room for improvement in terms of image augmentation support. Therefore I implemented my own input image and segmentation label flipping(random horizontal and vertical) algorithm in the Segmentation module.

Chapter 5

Experiments and results

In this chapter I will present the results of both the self-supervised pretraining, training of the downstream tasks and the experiments made using the SEG-Grad-CAM[32]. In the downstream training phase I trained and compared the performance of the Resnet18 with the following three weight initializations: ImageNet, SwAV pretraining and SimSiam pretraining, then selecting the two best model configurations procedeed with additional experiments (few-shot learning and XAI) on them.

In the previous chapters I already mentioned(probably more than once) the causes of the general lack of data in the medical field. Due to this data shortage, I decided to conduct multiple experiments using smaller training datasets, so the networks' perfomances can be compared in this new aspect too.

5.1 Self-supervised pretraining

For the SimSiam algorithm the Pytorch Lightly(`https://github.com/lightly-ai/lightly` library was used. This is a rather new and easy-to-use repository, many times building on the services and utilizing Pytorch Lightning's (`https://www.pytorchlightning.ai/`) API. Unfortunately Pytorch Lightly doesn't have an available implementation for the SwAV paper, so I had to use the Pytorch Lightning. Working with two very different libraries caused some complications, but it didn't effect the accuracy of the training with one bit.

Originally I wanted both of the the SSL trainings to run for at least 35 epochs each, however the networks' losses converged way quicker and both of the trainings, due to the parameters of Pytorch's EarlyStopping stopped around 15 epochs. I actually of course ran these trainings multiple times to make sure and in all the cases the training and validation losses stopped improving around the 15 epoch mark. (Of course, the 15 epoch mark is just an average of all the trainings, naturally there were some discrepancies, but nothing major changes, only 1 or 2 epochs in both directions.) On Figure 5.1 the SimSiam self-supervised training's training and validation losses are plotted until the 15th epoch.

The fast convergence of the networks is actually quite surprising compared to the performance of the two algorithms' pretraining on the ImageNet dataset. My opinion is that probably this is caused by the difference in the variety of the datasets. In a sense, between two of the ImageNet's images, there can be much bigger differences in visual features (for instance between a boat and a cat). Meanwhile, between two MRI images there are

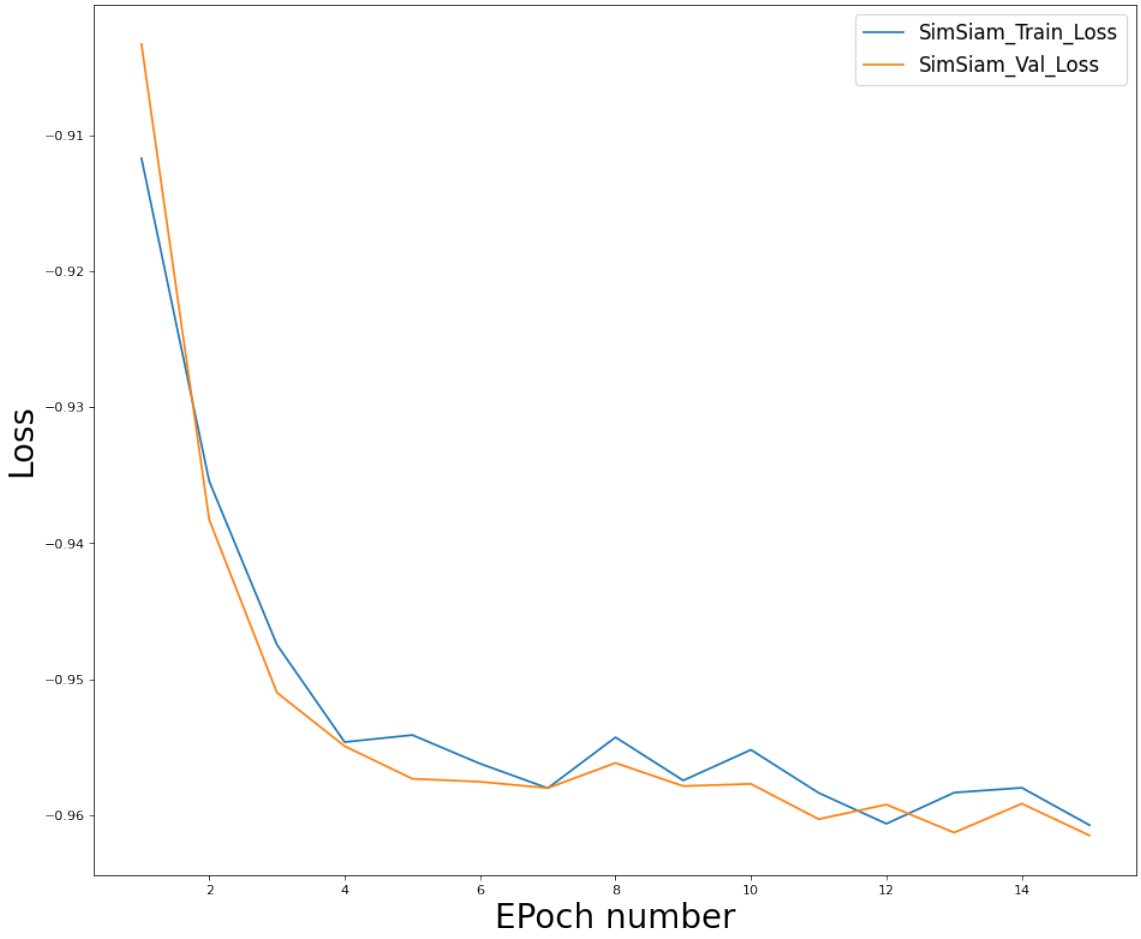


Figure 5.1: SimSiam pretraining validation and training loss

only very subtle differences, since if we were talking about classification, both of them are instances of the same class.

5.2 Segmentation training

For implementating the segmentation networks with the modified encoders, I used the Qubvel library, in my opinion it's a fairly simple library used in many competitions.

For comparing and measuring the performances of the different networks I used pixel accuracy and IOU score, and for training the loss function was dice loss. In our case, where the background (everything but the heart) is quite large compared to the other regions, the pixel accuracy is unfortunately not a useful metric.

During segmentation, initially I set the number of epochs to 40, which later proved to be a good choice, because each of the models reached their peaks before that, usually around epoch number 36-38. Of course as an experiment I once or twice let the networks train for another 20 epochs (so 60 epochs in total), still their performance didn't budge, only showed signs of mild overfitting. Other hyperparameters: when training the network on the entire dataset, I used batches of size 128 and the Adam optimizer, with an initial learning rate of 0.0001.

If one has already taken a glimpse at the upcoming figures, than he could see that there are 4 curves on both figures. In order to see the power of transfer learning, I threw in the data of a training where, the encoder's weights were randomly initialized.

To summarize: in all the different training cases the relative order of their validation IOU scores were the same:

1. **SimSiam**
2. **ImageNet**
3. **SwAV**
4. **Random initialization**

The relative differences between their IOU scores more or less stayed constant. As we can see on Figure 5.2, the networks with SimSiam and ImageNet weights are usually about 3-4% apart in IOU scores, while the SwAV network is bringing up the rear with about another steady 5% below the scores of the runner up, but still well above the result of random initilaization.

Below is a table to give a more concise summary to all the results of the different models. The "Test dataset" column refers to the models' achieved scores on the dataset's [2] original test images.

Highest IoU scores per model and dataset			
Weight initialization	Train dataset	Validation dataset	Test dataset
Random	0.4996	0.4398	0.4405
SwAV	0.7173	0.6953	0.6783
ImageNet	0.7685	0.7538	0.7471
Sim Siam	**0.7806**	**0.7761**	**0.7674**

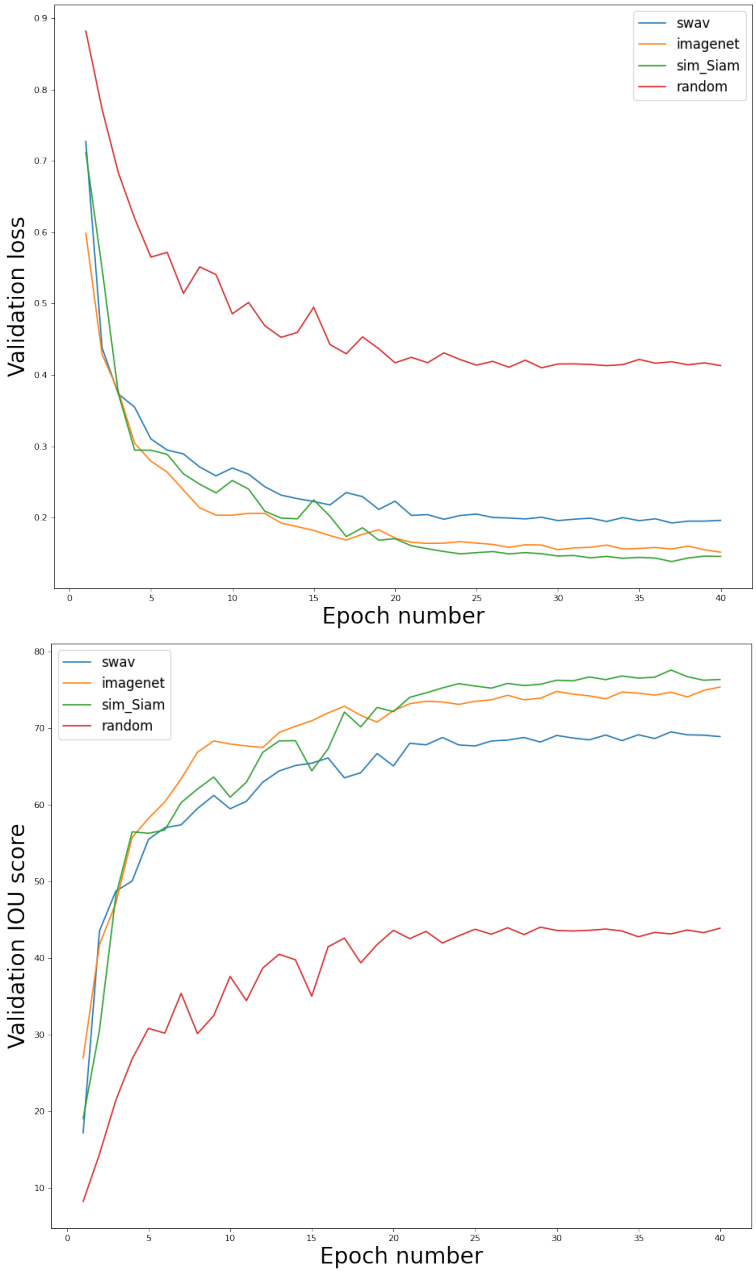


Figure 5.2: Validation loss and IOU score (times 100 for better understanding) during training

I conducted the extra experiments with fewer training images on the best and the runner-up models (SimSiam and Imagenet initialization). The 4 different experiments were conducted with 1200, 700, 500 and 150 training dataset sizes, leaving the validation dataset intact. The motivation for these experiments is to simulate a real life project, during which there's usually a data shortage(due to the reasons mentioned in the introduction). In layman's terms: let's see which initialization is better if the dataset is small. For this experiment, if the dataset size was less than 1000 (150, 500, 700), I froze the weights of the encoder layers. In order to account for any non-deterministic effects interfering with the results, for each dataset size the training of the two models were redone multiple times, but initialized with different random seeds.

The results can be seen on **??**, the dataset size is on the horizontal axis and on the vertical one, the maximum achieved IOU score is presented (again for better understanding, the IOU score is scaled by 100). Clearly, the SimSiam weight initialization beats its opponent in all of the cases. To be specific for the dataset sizes 1200, 700, 500, 150 respectively the following maximum validation IOU score pairs were the output: 41.04 - 31.08, 55.25 - 48.49, 60.94 - 54.12, 66.00 - 59.45.

The results can be seen on Figure 5.3, the dataset sizes is on the horizontal axis and on the vertical one the maximum IOU score can be seen of each training, which are represented by the colored points. The curves are the connected mean IOU scores for each dataset size, for example the first point on the curve is the result of the 4 SimSiam trainings with dataset size of 150. The vertical red lines are the deviations of the training's maximum IOU scores. learly, the SimSiam weight initialization beats its opponent in all of the cases.

As we can see the difference the two pretrainings slowly decreases as the number of training images increases. The only "bump" in the curve is at the 1200 training dataset size. Most likely, unfreezing the models' weights is behind this slight change.

All of the models were trained for 40 epochs as before, the only additional difference being the the batch size, lowering it to 32 resulted in much better performance than leaving it at 128. The previously mentioned phenomeon happened again: the networks usually reached their maximum validation IOU scores around the 30th epoch (give or take 2 epochs) and than typically only showed signs of overfitting again.

A sample output can be seen on Figure 5.4. The model was trained using 700 images with a SimSiam weight initialization. In this case it actually the prediction is relatively good, given the low amount of training data the network has seen.

I would like to make another interesting observation about regarding the above mentioned trainings, which caught my attention during supervising the training cycles and analyzing the logs. In most cases ImageNet pretrained models tended to take the lead in the first half of the epochs. My theory is that this is caused by the fact that these models weights begun training from a more "stable" state, since these weights were trained using a supervised learning task. Rather insignificant difference in the end, but it's still interesting to see the differences between the two types of learning tasks' outputs.

5.3 XAI experiments

In the previous section, I confidently stated based on the results of the few-shot trainings' maximal valdiation IOU scores that the SimSiam weight initialization is the superior choicee. If this was part of a real life project the question would arise before long, how can we surely know and put the clients mind at ease that we chose the better model? As much as we would all like to, not all of these questions can be answered with maximum confidence, but we can try at least. Admittedly without the knowledge of a radiologist or cardiologist, very accurate and groundbreaking medical observations aren't likely, however apperent patterns canbe found in the explanations.

During these more "client perpespective" experiments I first calculated the IOU scores inputting all test images to both models and selected the following images from their predictions:

- 5 predictions with the highest IOU scores
- 5 predictions with the lowest IOU scores

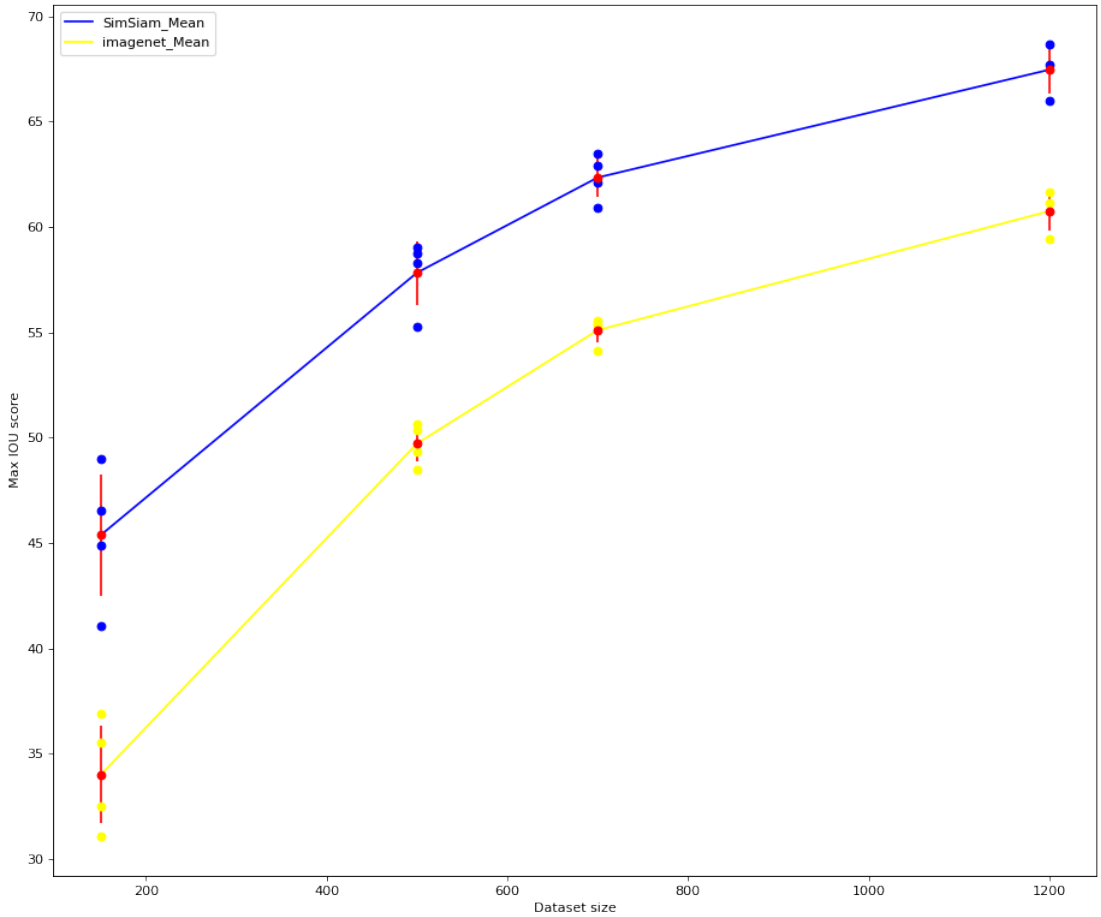


Figure 5.3: Peformance of SimSiam and Imagenet initializations on different dataset sizes

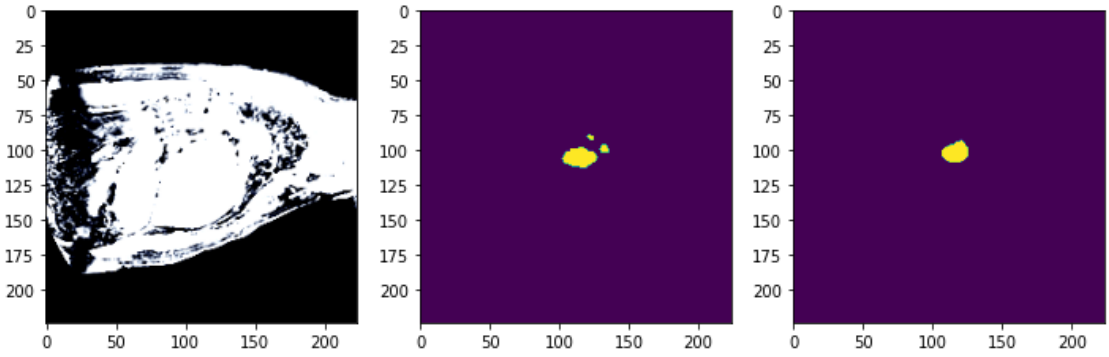
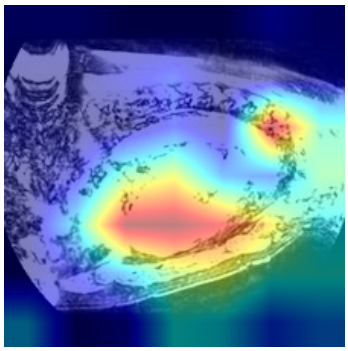
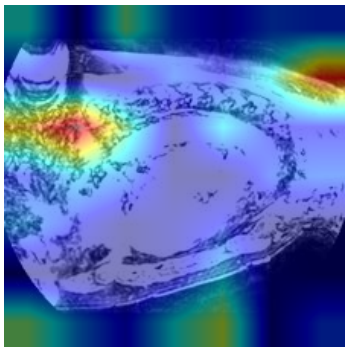


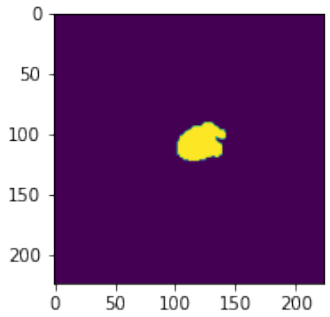
Figure 5.4: From left to right: input image, the output label and the predicted output



(a) SimSiam weight initialization **best** prediction



(b) Imagenet weight initialization **best** prediction



(c) The common mask

Figure 5.5: The best predictions

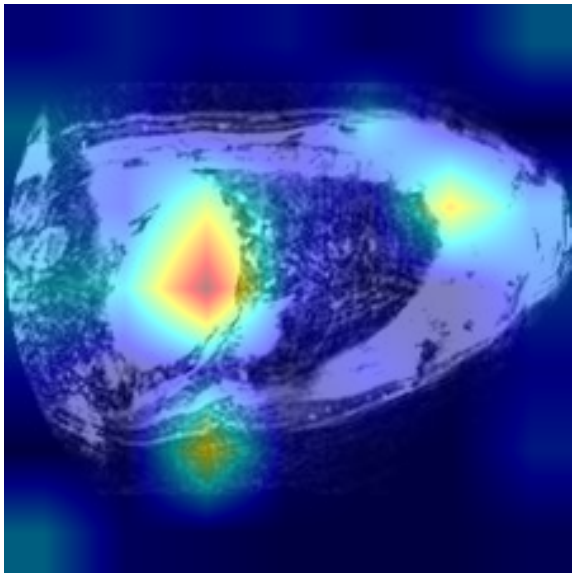
- 3 predictions with IOU scores closest to the prediction's median

Using the SEG-Grad-CAM implementation even the passing of multiple layers was possible, still after checking the explained outputs of multiple layers, the rule of thumb proved to be right again: the encoder's last layer yielded the most informative and understandable explanations. No point in procrastinating, the best explanation outputs can be seen on Figure 5.5, the medians are on Figure 5.6 and Figure 5.7, and the worst predictons' explanations on Figure 5.9 and Figure 5.8.

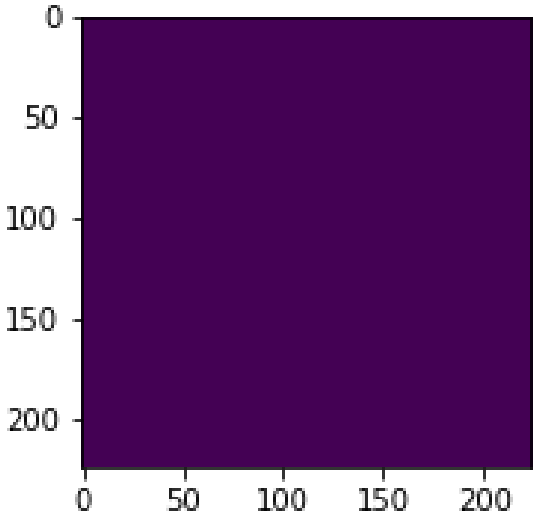
On Figure 5.5 in comparison to the other figures contains 3 images, because both models had very high IOU score for the same input image, giving a great opportunity to compare the two predictions. On this figure it seems that the model with ImageNet weight initialization focuses less on the relevant area. Of course I hadn't the time to check and compare each explanation of the test set, but for the chosen top inputs with thr top IOU scores the former observation has more or less true. Somehow, the ImageNet weight initializatied model puts its attention to the side of the images, not to the centre. Interestingly, on their worst predictions Figure 5.8, Figure 5.9 the ImageNet model seems to be focusing on the more relevant features, meanwhile the SimSiam model was presumably biased and focused on the wrong feature.

Looking at separately the ImageNet model's and SimSiam model's saliency maps, we can see a couple of reoccuring patterns. First of all the former model puts more attention to the background and many such areas cause high activations. Meanwhile the SimSiam model's focus is way more centered, the background pixels have very small contributions overall.

In the end, it is very hard to come to an impartial decision, since not all of the images were analyzed, and there must be other test sets for which the exact opposites would be true. Nevertheless, these examples were able to clearly show the difference of pretraining. As we could see overall the encoder pretraining using the SimSiam algorithm focused the network's more on the task specific features.

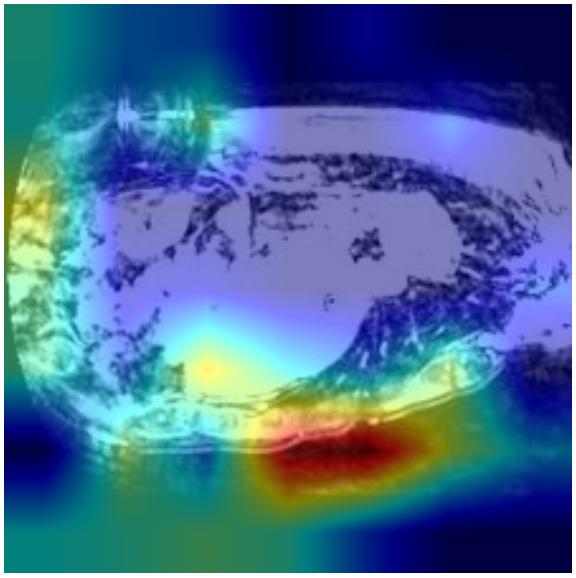


(a)

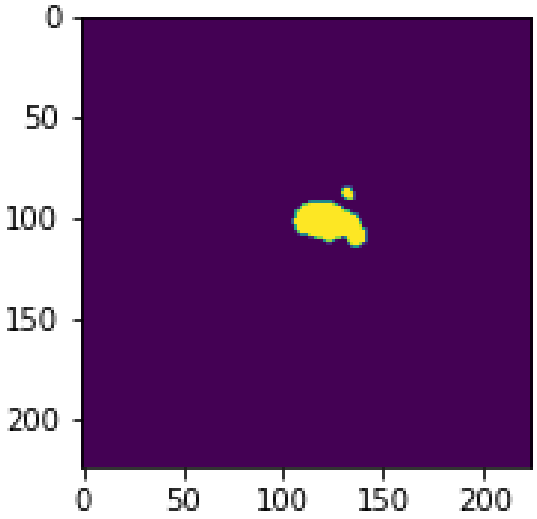


(b)

Figure 5.6: SimSiam weight initialization **median** prediction

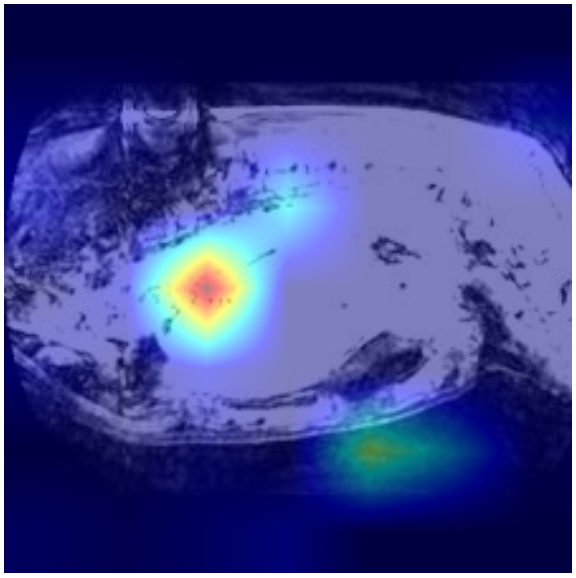


(a)

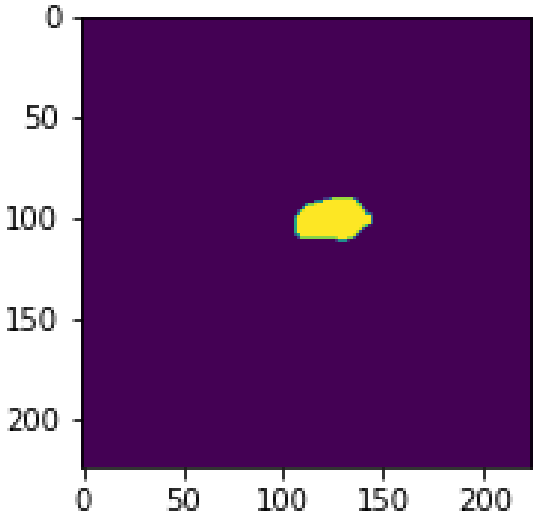


(b)

Figure 5.7: ImageNet weight initialization **median** prediction

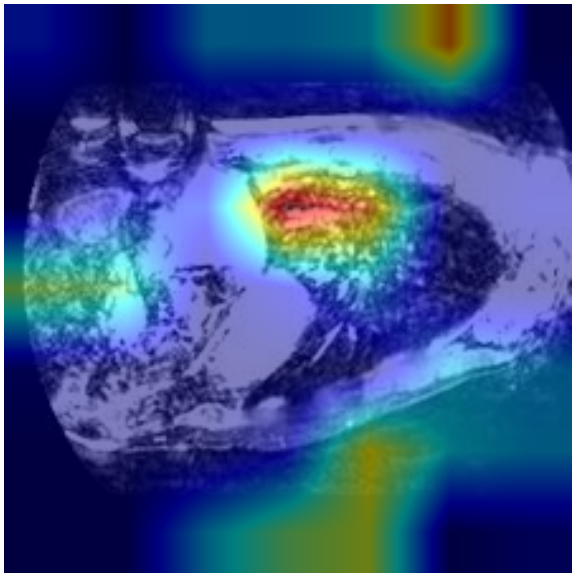


(a)

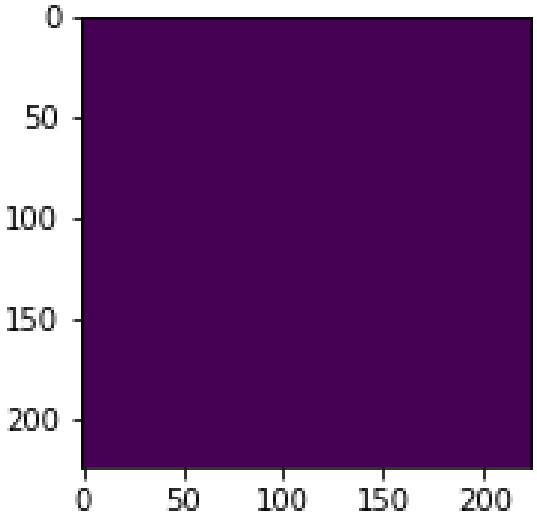


(b)

Figure 5.8: SimSiam weight initialization **worst** prediction



(a)



(b)

Figure 5.9: ImageNet weight initialization **worst** prediction

Chapter 6

Summary

In this chapter I would like to summarize this semester's work and draw some conclusions.

6.1 Conclusions

I successfully created 2D image datasets from the 3D medical datasets both for semantic segmentation and self-supervised learning, with more than 20 thousand samples in the latter. This dataset provides sufficient amount of data for self-supervised learning tasks (for instance Vision Transformer[10] models require a lot of data).

After the data processing I also overviewed and looked into the current and past works in the field of self-supervised learning and medical imaging, the intersection of the two topics being at the center of attention. Needless to say, familiarizing myself with semantic segmentation was also necessary during my work.

Using the (for the lack of a better word) created datasets, I performed self-supervised learning on a Resnet18 model with two state of the art algorithms and extracted the trained encoders. In total I trained 4 semantic segmentation models on cardiac MRI data with 4 different encoders, evaluated each one and compared them to each other.

After running the training with all the labeled training data, I performed few-shot learning in order to simulate the situation of real world medical imaging projects with various training dataset sizes: 150, 500, 700, 1200. For these experiments I used the SimSiam and ImageNet weight initializations. Speaking of real world projects, I observed and analyzed multiple predictions using a state of the art XAI technique, SEG-Grad-CAM.

All training's final metrics showed the same pattern: the model with SimSiam weights outperformed the random, ImageNet and SwAV weight initializations. After the trainings, observing the saliency maps of the different mdoels, the pretraining without a doubt had great effects for the encoders attention. Due to all this, in the end, it is my firm belief that SSL definitely has a place in medical imaging, owing to the fact that SimSiam pretrained model outperformed signifcantly the ImageNet initialized model during all of the experiments.

My contributions show that self-supervised learning techniques can make a huge difference in solving the problem of labeled data shortage and that it is a perfectly viable option for engineers in medical imaging projects, who no longer have to rely on model's pretrained on non task related datasets.

6.2 Future works

I think this semester's work can be continued in the following ways:

- Using more data for pretraining: one of the most noteworthy observations was the fast convergence of both of the SSL algorithms. I suspect that the reason for this was the small variety of the dataset, so adding to this dataset other medical recordings could result in better feature representations.
- Pretraining the decoder part too. In my opinion the performance can only be raised to an extent for the reason that nearly more than half of the model is left untouched.
- Comparing the general SSL techniques with the medical imaging specific ones mentioned in chapter 2.

Bibliography

[1] Alejandro Barredo Arrieta, Natalia Díaz Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *CoRR*, abs/1910.10045, 2019. URL http://arxiv.org/abs/1910.10045.

[2] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, Gerard Sanroma, Sandy Napel, Steffen Petersen, Georgios Tziritas, Elias Grinias, Mahendra Khened, Varghese Alex Kollerathu, Ganapathy Krishnamurthi, Marc-Michel Rohé, Xavier Pennec, Maxime Sermesant, Fabian Isensee, Paul Jäger, Klaus H. Maier-Hein, Peter M. Full, Ivo Wolf, Sandy Engelhardt, Christian F. Baumgartner, Lisa M. Koch, Jelmer M. Wolterink, Ivana Išgum, Yeonggul Jang, Yoonmi Hong, Jay Patravali, Shubham Jain, Olivier Humbert, and Pierre-Marc Jodoin. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved? *IEEE Transactions on Medical Imaging*, 37(11): 2514–2525, 2018. DOI: 10.1109/TMI.2018.2837502.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features, 2018. URL https://arxiv.org/abs/1807.05520.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2020. URL https://arxiv.org/abs/2006.09882.

[5] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf.

[6] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation, 2021. URL https://arxiv.org/abs/2102.04306.

[7] Liang Chen, Paul Bentley, Kensaku Mori, Kazunari Misawa, Michitaka Fujiwara, and Daniel Rueckert. Self-supervised learning for medical image analysis using image context restoration. *Medical Image Analysis*, 58:101539, 07 2019. DOI: 10.1016/j.media.2019.101539.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. URL https://arxiv.org/abs/2011.10566.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL https://arxiv.org/abs/2006.07733.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2017. URL https://arxiv.org/abs/1703.06870.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2019. URL https://arxiv.org/abs/1911.05722.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

[17] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction, 2018. URL https://arxiv.org/abs/1811.11387.

[18] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2016. URL https://arxiv.org/abs/1609.04802.

[19] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences, 2017. URL https://arxiv.org/abs/1708.01246.

[20] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874.

[21] Franco Matzkin, Virginia Newcombe, Susan Stevenson, Aneesh Khetani, Tom Newman, Richard Digby, Andrew Stevens, Ben Glocker, and Enzo Ferrante. Self-supervised skull reconstruction in brain ct images with decompressive craniectomy, 2020. URL https://arxiv.org/abs/2007.03817.

[22] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2016. URL https://arxiv.org/abs/1603.09246.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

[24] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. 2016. DOI: 10.48550/ARXIV.1604.07379. URL https://arxiv.org/abs/1604.07379.

[25] Cristiano Patrício, João C. Neves, and Luís F. Teixeira. Explainable deep learning methods in medical imaging diagnosis: A survey, 2022. URL https://arxiv.org/abs/2205.04766.

[26] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. URL http://arxiv.org/abs/1602.04938.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

[28] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. 2018. DOI: 10.48550/ARXIV.1811.10154. URL https://arxiv.org/abs/1811.10154.

[29] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. URL http://arxiv.org/abs/1610.02391.

[30] Saeed Shurrab and Rehab Duwairi. Self-supervised learning methods and applications in medical imaging analysis: A survey, 2021. URL https://arxiv.org/abs/2109.08685.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

[32] Kira Vinogradova, Alexandr Dibrov, and Gene Myers. Towards interpretable semantic segmentation via gradient-weighted class activation mapping. *CoRR*, abs/2002.11434, 2020. URL https://arxiv.org/abs/2002.11434.

[33] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization, 2016. URL https://arxiv.org/abs/1603.08511.

[34] Xinrui Zhuang, Yuexiang Li, Yifan Hu, Kai Ma, Yujiu Yang, and Yefeng Zheng. Self-supervised feature learning for 3d medical images by playing a rubik's cube, 2019. URL https://arxiv.org/abs/1910.02241.