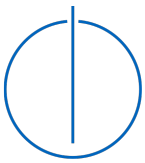
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Attention Mechanisms for End-to-End Therapy Response Prediction on PDAC CTs

Alp Güvenir



DEPARTMENT OF INFORMATICS

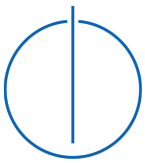
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Attention Mechanisms for End-to-End Therapy Response Prediction on PDAC CTs

Aufmerksamkeitsmechanismen für die Vorhersage des Ende-zu-Ende Therapieansprechens auf PDAC CTs

Author:	Alp Güvenir
Supervisor:	Prof. Dr. Daniel Rückert
Advisor:	Alexander Ziller, Ayhan Can Erdur
Submission Date:	June 15, 2023



I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, June 15, 2023 Alp Güvenir

Acknowledgments

I would like to express my sincere gratitude and appreciation to my supervisor Prof. Dr. Daniel Rückert for giving me the opportunity to pursue my thesis studies in his lab.

I would like to further express my gratitude to my advisors Alexander Ziller and Ayhan Can Erdur for their invaluable guidance and continuous support. For solving such a challenging task, their crucial advice along with constructive comments were really valuable. I would like to thank you both for this great opportunity and for the inspiring discussions we had.

I would also like to thank my family for their genuine support during my thesis. Moreover, I would like to thank my friends, for their continuous encouragement and support.

Abstract

Pancreatic ductal adenocarcinoma (PDAC) is the most common disease among the malignant neoplasms of the pancreas. It is reported as one of the most common causes of cancer-related deaths. The symptoms of PDAC are often unrecognized and lead the tumor to be discovered in the later stages, causing the treatment process for the disease to be challenging. In the medical workflow for the diagnosis of the PDAC, computed tomography (CT) images are used. In this thesis, we investigate end-to-end deep learning approaches, that directly learn from input CT data to produce therapy response outputs without the need for multiple consecutive steps. To predict the therapy response for PDAC patients by classifying their diagnosis/pre-treatment CTs, we use Response Evaluation Criteria in Solid Tumors labels, a standardized method for cancer response evaluation by clinicians, for supervision. As the pancreas consists of a small volume within the 3D CT structure, we investigate to what extent attention mechanisms can extract information regarding the task from the whole CT volume and improve the classification, as attention mechanisms capture global information and learn where to focus on. For this task, we introduce 2.5D and 3D networks which utilize attention mechanisms in spatial, channel and temporal domains of a 3D CT volume. Furthermore, we demonstrate the performance of these networks on different tasks and datasets in the medical domain and show the interpretability power of attention mechanisms by showing which slices obtain a higher attention weight depending on the task. Moreover, we compare performance change using the transfer learning scheme from related domains to predict therapy response for PDAC patients. Finally, we provide end-to-end networks from each architecture type of 2.5D and 3D that outperform other networks in their own cohort based on their Matthews Correlation Coefficient scores.

Contents

1. Introduction

1.1. Motivation and Problem Statement

With the increasing interest and research done on computer vision tasks, the field of Deep Learning is showing continuous progress. Furthermore, Deep Learning techniques have shown their effectiveness and performance in medical image analysis. Convolutional Neural Networks (CNN) achieved groundbreaking progress on computer vision tasks in both 2D and 3D domains. In recent years, with the introduction of attention mechanisms in the Natural Language Processing (NLP) domain, the Transformer architecture has been found to be outperforming other approaches as well as providing interpretability of the algorithm by presenting where the attention is being focused. Moreover, the Transformer approach has been introduced to the field of computer vision and become the state-of-the-art architecture for many tasks by outperforming CNN approaches.

Among the many fields that benefit from the Deep Learning era, the field of medicine has also highly profited from the recent developments. One of the areas that need further investigation in the medical domain is Pancreatic ductal adenocarcinoma (PDAC) disease. PDAC is the most common disease among the malignant neoplasms of the pancreas [33]. Furthermore, it was reported that PDAC is the fourth most common cause of cancer-related deaths. The outcome of PDAC treatments highly depends on which stage the disease is being diagnosed. It is argued that surgical resection followed by chemotherapy is the only treatment methodology to cure patients with PDAC. However, only a few of the patients are often suitable for surgical resection. The reason behind this is that the symptoms of PDAC are often unrecognized, which leads the tumor to be discovered in the later stages [41]. Thus, surgical resection becomes impractical for many patients, as the tumors become large and may infiltrate other organs or vessels. For such patients, chemotherapy is applied as the first-line treatment [33].

Response Evaluation Criteria in Solid Tumors (RECIST) is a widely acknowledged standard for assessment of response in solid tumors [38]. Tumor progression can be categorized within the defined criteria. However, based on the patient's diagnosis data, it is a challenging task for doctors to estimate the progression criteria for patients after receiving treatment. Previous studies examined the use of pre-operative computed

tomography (CT) for tumor cellularity characterization in PDAC patients [19]. Furthermore, another research was conducted for therapy response prediction on PDAC diagnosis/pre-operative CTs which proposed a 3-staged pipeline using deep learning techniques [56]. As the task is challenging, authors have benefited from intermediate representations, instead of an end-to-end fashion of deep learning which is performed by using input data to produce the desired output in a single network. As an advantage over the pipeline architectures, the end-to-end networks aim to train a single model directly mapping inputs to outputs, without designing and optimizing intermediate stages. Therefore, these approaches can simplify the overall architecture and reduce training and classification time complexity.

In this thesis, our aim is to predict therapy response to chemotherapy treatment for PDAC patients by classifying diagnosis/pre-treatment CTs using RECIST progression labels for supervision in an end-to-end way. Prediction of the therapy response would provide information to doctors for creating individualized treatment. For this task, we introduce 2.5D networks by applying attention pooling to each CT slice for the classification of a 3D input, as well as 3D networks. Furthermore, by utilizing attention mechanisms, our goal is to investigate to what extent using attention mechanisms can improve the classification performance using features extracted from spatial, channel and temporal domains of a 3D CT. Given that the pancreas consists of a small volume within the 3D CT structure, we investigate if attention mechanisms are able to extract information regarding our task from the whole CT volume as they learn where to focus by capturing global dependencies and relationships between different parts of the input. Also, by using the attention mechanisms, we aim to present the interpretability of the models as they learn where to extract information from. Moreover, we investigate the effects of transfer learning from related and unrelated domains and compare its effect on our final task.

1.2. Contributions

The contributions of this thesis are:

- Investigate the end-to-end therapy response prediction networks operating on diagnosis or pre-treatment CTs using RECIST progression labels for supervision.
- Provide an approach and a comprehensive investigation on using 2.5D network approaches, applying attention pooling to features extracted per CT slices using 2D encoders, for classifying 3D images.
- Compare the integration of various attention mechanisms for the 2.5D network approaches that we define.

- For the wide scope of network architectures we define, we conduct a comprehensive analysis of the performance of these networks on similar tasks such as sex prediction, PDAC tumor classification and Radiological ImageNet classification.
- Investigate the interpretation ability of the attention mechanisms.
- Compare the approach of 2.5D networks and 3D networks and the influence of transfer learning from a related domain for PDAC therapy response prediction.

1.3. Outline

This thesis is structured as the following:

In **Chapter 2**, we provide an overview of the related work conducted on attention mechanisms in various domains. Furthermore, we introduce convolutional neural networks for 2D and 3D computer vision tasks. Furthermore, we introduce works done on transfer learning. Lastly, we provide approaches conducted for therapy response prediction.

In **Chapter 3**, we introduce the medical background for PDAC disease. Furthermore, we introduce the RECIST categorization that we use for the supervision of our research.

In **Chapter 4**, we introduce the various datasets that we use including datasets we utilize for transfer learning as well as the dataset for therapy response prediction on PDAC CTs.

In **Chapter 5**, we first provide an overview of the existing architectures. Furthermore, we present networks that we use for our task.

In **Chapter 6**, we demonstrate the experiments that we conduct on datasets that we plan to use as proof of concept and transfer learning purposes. Later, we compare the benefits of using transfer learning from a related domain and an unrelated domain. Lastly, we present our final findings for therapy response prediction.

In **Chapter 7**, we briefly summarize our findings. Furthermore, we provide the limitations that we encounter and future research direction.

2. Related Work

2.1. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a class of artificial neural networks, such that their capacity can be controlled by varying their depth and range [22]. The first application of CNNs was presented in an article for handwritten digit recognition [23, 24]. Moreover, with the success of AlexNet on ImageNet classification, the impact of convolutions was once again highlighted [22]. Compared to standard feed-forward neural networks, CNNs have fewer connections and parameters, thus they are easier to train. Despite the early formation of the CNNs, they still show promising results for image classification, object detection, action recognition, human pose estimation, image segmentation and other computer vision tasks. The overall architecture of CNNs contains, convolutions, pooling layers, normalization layers, activation functions for non-linearity and fully connected layers [22]. It is also known that CNNs also produce outstanding results in the domain of medical imaging, due to their efficient yet powerful representation power [37, 40].

2.1.1. ResNet

Among the many well-known CNN architectures, one of the famous architectures that have shown promising results was published in the paper *Deep Residual Learning for Image Recognition*, also known as ResNet [14]. Deep CNNs have shown groundbreaking results for image classification tasks. Furthermore, the authors have pointed out that research done in the field provides evidence of the importance of the depth of the networks being crucial for attaining successful results. However, authors have also pointed out that as the networks get deeper the issue of degradation is being observed (deeper networks having higher training and testing error), which indicates the challenge of optimization. From this motivation, they propose a *deep residual learning* framework that addresses the degradation problem.

The authors formulate their idea by claiming, if the added layers can be formed as identity mappings, then the degradation should not be greater than the shallower part of the network. Although they acknowledge that the identity mappings are unlikely to the optimal, it may help to precondition the issue. Thus, a residual learning building

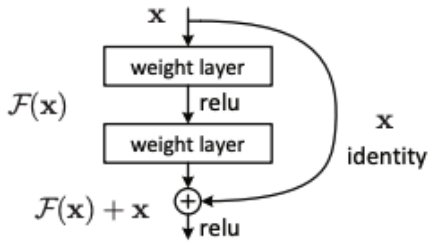


Figure 2.1.: Residual learning building block [14]

block as in Figure 2.1, is formally defined as $y = F(x, W_i) + x$, where x being the input, y being the output. Furthermore, $F = W_2\sigma(W_1x)$ where σ represents the ReLU operation. With this approach, ResNet outperformed other networks on image classification task on ImageNet. Moreover, the authors show its performance also on other tasks such as object detection.

2.1.2. ResNet 3D

As the success of 2D convolutions and networks built on top of them showed their performance on 2D image tasks, researchers investigated on 3D convolutions for another computer vision task known as action recognition [18]. The idea of 3D convolutions was argued to be able to extract features from both spatial and temporal domains, thus capturing the motion information embedded in between adjacent frames [18]. The authors showed that 3D CNN models were outperforming frame-based 2D CNN models in action recognition. On top of this, Convolutional 3D (C3D) network was proposed which has improved the benchmark on video analysis tasks [44]. Moreover, another research was conducted on benefiting from 2D convolutional networks, by inflating all the filters and pooling kernels adapting them to have a temporal dimension, namely Two-Stream Inflated 3D convolutional networks (I3D) [6]. This approach was shown to be more successful than the 3D CNN networks at the time.

Furthermore, in another research, the authors have challenged the view of inflated networks and wanted to revisit the concept of temporal reasoning in action recognition task benefiting from 3D CNNs in their article *A Closer Look at Spatiotemporal Convolutions for Action Recognition* [45]. Although 3D CNNs were investigated previously for the task of action recognition, authors employed 3D convolutions in the *residual network* (ResNet [14]) concept. Along with the concept of 3D convolutions in a *residual network (R3D)*, authors also introduce two new forms of spatiotemporal convolutions that can be viewed as a form in between 2D and 3D convolutions, which can be found in Figure 2.2 [45].

The first concept is named mixed convolution where the authors propose using 3D convolutions only in the early layers of the network and then employing 2D convolutions. Here, they argue that temporal modeling can be caught in the early stages by 3D convolutions, whereas for action recognition spatial features need to be captured using 2D convolutions. Employing this idea in ResNets, they named the architecture *MCx*.

The second concept they introduce is the (2+1)D convolutional block which factorizes 3D convolution into two separate operations of 2D convolution for spatial and 1D convolution for the temporal domain. With this modification, an additional nonlinear operation is added between these two convolutions. They also point out that this decomposition also facilitates optimization, thus lowering the train and test losses. Employing this idea in ResNets, they named the architecture *R(2+1)D*.

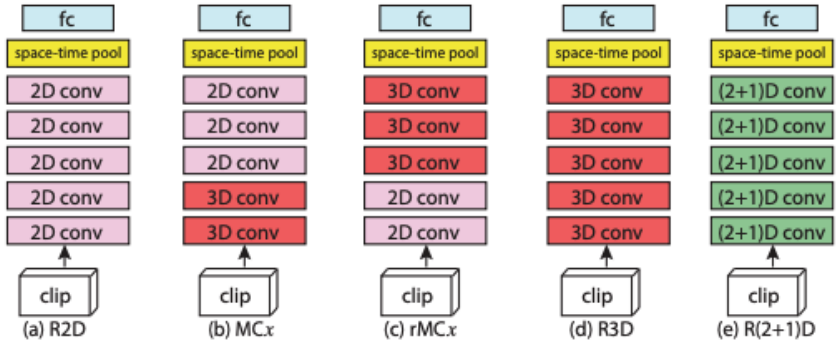


Figure 2.2.: Residual network architectures for video classification. R2D represents 2D ResNets; MCx represents mixed convolutions; rMCx represents reversed mixed convolutions; R3D represents 3D ResNets; R(2+1)D represents ResNets with (2+1)D convolutions. [45]

2.2. Attention Mechanisms

Studying human visual attention in cognitive science, researchers are motivated by the observation that, rather than compressing an entire image into a static representation, attention allows for salient features to dynamically come to the forefront as needed [52]. Inspired by the human visual attention mechanism, researchers invest in various models for simulating the similar effect of attention distribution for various types of inputs such as images and videos [53]. Thus, attention mechanisms can be defined as dynamic weight adjustment of a mask based on features of the input where this mask is used for enhancing salient information [13, 42, 53].

Although the CNNs are groundbreaking for many visual tasks such as image classification, image segmentation, object detection, image reconstruction and medical image processing, these networks hold an inductive bias [11]. In CNNs, the locality of information extraction as of a two-dimensional neighborhood structure generates an image-specific inductive bias [11]. In other words, using convolutions the influence of global information is not being captured, as they are processing local information [42]. However, using attention mechanisms, researchers aim to gather global information in the layers by applying weight distribution on the masks [11].

The idea of attention mechanisms furthermore evolved into mainly two different types; soft attention and hard attention. The main distinction between these two attention types can be referred as soft attention is a deterministic mechanism trainable by standard back-propagation, whereas hard attention is a stochastic mechanism trainable by maximizing approximate variational lower bound or reinforcement learning [52]. In other words, the main difference between hard and soft attention can be pointed out as soft attention mechanisms are masked using values from 0 to 1, whereas hard attention mechanisms are masked using values of 0 and 1 [42].

Figure 2.3 shows the difference between soft attention and hard attention techniques masking the relevant parts of the input image for generating the same neural image caption [52].

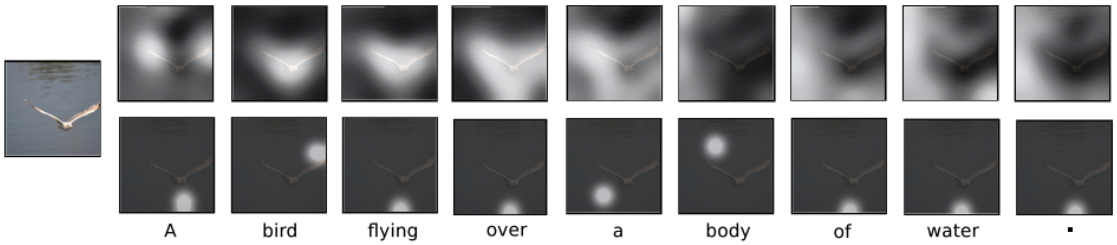


Figure 2.3.: Soft attention (top row) vs hard attention (bottom row) [52]

In Equation 2.1, the location variable s_t denotes the attention when generating the t^{th} word [52]. Furthermore, $s_{t,i}$ denotes a one-hot encoder whether the i^{th} location is set to 1 for extracting information. Thus, as attention locations being defined as latent variables, a multinoulli distribution can be assigned. For selecting the location to be used, a simple $argmax$ function can be used, although this approach is not differentiable and often achieved by reinforced learning.

$$\hat{z_t} = \sum_i s_{t,i} a_i \tag{2.1}$$

$$p(s_{t,i} = 1 | s_{j<t}, a) = \alpha_{t,i} \tag{2.2}$$

$$s_t^{\tilde{n}} \sim Multinoulli_L(\{\alpha_i^n\}) \tag{2.3}$$

In Equation 2.4, learning stochastic attention requires sampling the attention location s_t each time, instead, authors take the expectation of the context vector z^t [52]. Thus, the soft attention is realized by gradient descent and is differentiable and continuous [53].

$$\mathbb{E}_{p(s_t|\alpha)}[\hat{z}_t] = \sum_{i=1}^{L} \alpha_{t,i} a_i \tag{2.4}$$

In this thesis, we focus on soft attention mechanisms, in other words, techniques that are differentiable and continuous, and thus can be learned via forward and backward propagation. In the rest of this section, we introduce various types of attention mechanisms and categorize them based on their data domains.

2.2.1. Spatial Attention

Spatial attention can be defined as an adaptive spatial region selection mechanism, directing the attention to a specific location in space, also known as *where to pay attention* [13].

Gather-Excite

Inspired by the idea of pooling information in local descriptors and forming a global representation, Gather-Excite (GENet) defines two operators namely gather and excite [15]. As displayed in Figure 2.4, the gather operator aggregates responses in a given spatial representation with a reduction ratio and the excite operator applies the aggregation to the original input tensor, resulting in a new tensor with the same dimensions as the input tensor. This operation is performed by interpolating the aggregated feature representation from the gather operation and performing an element-wise product (Hadamard product) between the two matrices.

The GENet is a lightweight module that can be applied after a convolution operation. Within this operation, important features are emphasized, while the noise is being suppressed [13].

Self-Attention

Self-attention is defined as an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence [46]. Transformer

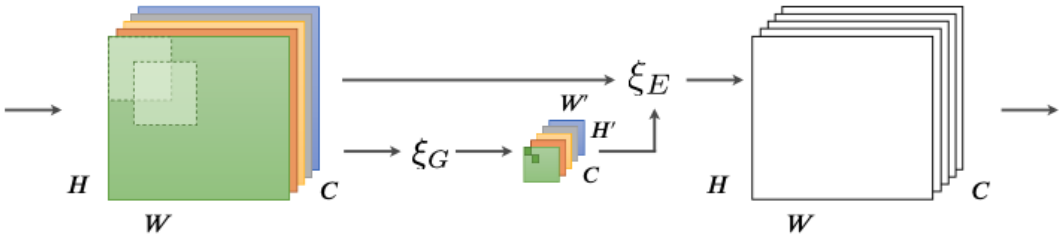


Figure 2.4.: Interaction of a Gather-Excite operator pair. Gather operator ξ_G aggregates features extracted across spatial neighborhoods. Resulting aggregations are passed together with input tensor to excite operator ξ_E, producing an output matching the dimensions of the input [15]

being the first transduction model relying entirely on self-attention to compute representations of its input and output was introduced by the pioneering research, *Attention Is All You Need* [46]. In this article, authors present a simple network architecture, eliminating the need for recurrence and convolutions [46]. The aim of the research done is to rely entirely on self-attention mechanisms to draw global dependencies between the input and output.

For the task of neural sequence transduction models, the authors observed that the competitive models use an encoder-decoder architecture. Therefore, the Transformer follows a similar architecture using stacked self-attention and fully connected layers for both the encoder and decoder as in Figure 2.5 [46].

The attention function in the Transformer can be described by mapping a query and a key-value pair to an output [46]. The output is computed being a weighted sum of the values, such that the weights assigned to each value are computed using the query with the respective key vector. The defined attention is referred as Scaled Dot-Product Attention consisting of query, key and values. The dot products are calculated as in Equation 2.5 and visualized as in Figure 2.7.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.5}$$

Furthermore, the authors point out that instead of performing a single attention function, linearly projecting the queries, keys and values h times with different, learned linear projections results in a better performance [46]. Each of these projections is performed in parallel, then concatenated and projected once again for final values as shown in Figure 2.6 describing the Multi-Head Attention.

In this research, authors experiment with their approach to machine translation tasks.

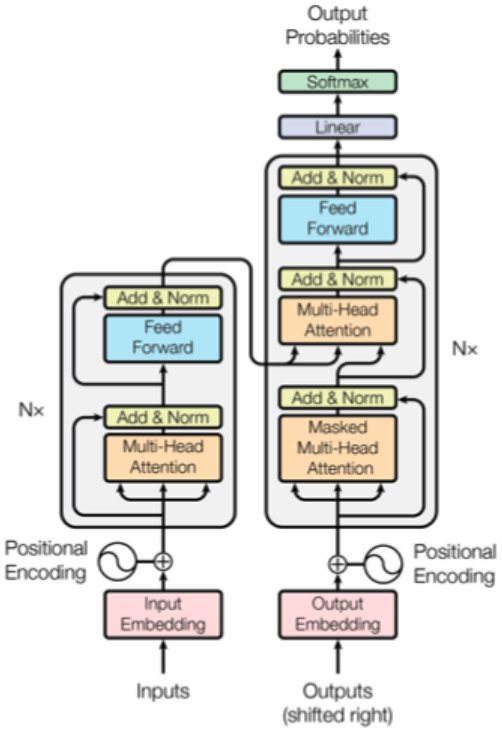


Figure 2.5.: The Transformer model architecture [46]

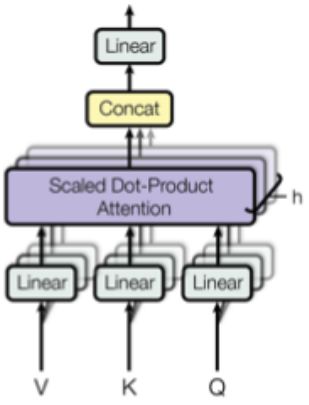


Figure 2.6.: Multi-Head Attention [46]

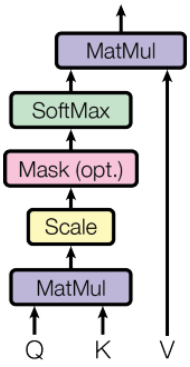


Figure 2.7.: Scaled Dot-Product Attention [46]

They report that the approach was successful for Natural Language Processing (NLP) tasks. However, self-attention is also applied to computer vision.

Non-Local Neural Networks

One of the leading research done relating to self-attention in the field of computer vision is the *Non-local Neural Networks*, presenting a novel non-local network in video understanding and object detection [48]. For capturing the long-range dependencies for sequential data, models use recurrent operations, whereas, for image data, stacks of convolutional operations are being used. These operations process a local neighborhood, therefore they cover long-range dependencies as these operations are applied repeatedly. However, repeating local operations have limitations such as computationally being inefficient, optimization difficulties and challenges in information flow at distant positions [48]. Therefore, in this research authors present a *non-local* operation for capturing long-range dependencies with an efficient and generic component in deep neural networks.

The authors point out the advantages of using non-local operations as they capture long-range dependencies by computing interactions between any two positions, regardless of their distance. Moreover, non-local operations can maintain variable input sizes and can be combined with other operations such as convolutions [48]. The authors showcase the effectiveness of the non-local operations in the video classification task, where long-range interactions occur between pixels in the space and time domain. This work relates to the self-attention method as it can be viewed as a form of the non-local mean (computing a weighted mean of all pixels in an input) that connects the idea to computer vision tasks and apply spatial attention despite the fact that its ability to also apply attention in the time domain [48].

Authors define a non-local operation as in Equation 2.6, where i is the index in output position and j is the index that enumerates all possible positions, as to clarify x being the input and y being the output signal [48]. Here, the pairwise function f is for computing a scalar and the function g is for computing a representation of the input signal at position j.

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{2.6}$$

$$z_i = W_z y_i + x_i \tag{2.7}$$

The non-local block is defined as in Equation 2.7, where y_i defined in Equation 2.6, x_i denoting a residual connection and W being the weight matrix to be learned [48]. An example non-local block is shown in Figure 2.8.

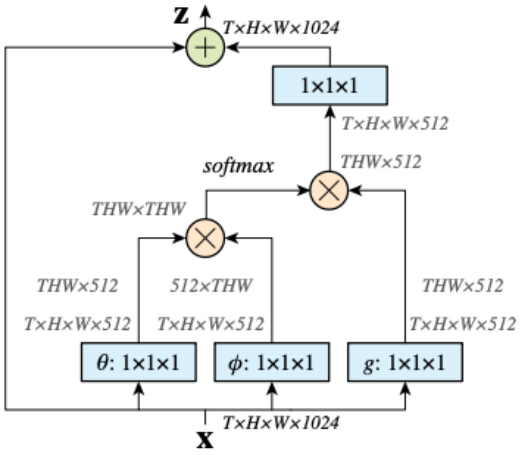


Figure 2.8.: Non-local block [48]

The research done concludes that non-local blocks combined with an existing architecture can show a significant improvement in the tasks such as video classification, as well as in static image recognition tasks to improve object detection/segmentation and pose estimation. The non-local block can use the set of positions in the space domain as well as in the spacetime domain and model pairwise relations. Although, it is important to note that the non-local operator does not process any temporal ordering information, while this is explicitly modeled in other spatial-temporal networks [49].

Vision Transformer

As the Transformer architecture has shown its success in the NLP tasks [13], recently its application to computer vision tasks is also being researched. However, the research done were investigating approaches where attention is either applied in conjunction with convolutional networks or to replace some components of the convolutional networks while keeping the main structure the same [11]. Thus, in computer vision, architectures currently rely on convolutional operations, and in large-scale image recognition tasks, classic ResNet-like architectures are still state-of-the-art [11].

Motivated by the success of Transformers in the NLP tasks, the authors experimented by applying the Transformer architecture to images and following the original model design as closely as possible. The standard Transformer receives input as a sequence of token embeddings in a 1D form. Therefore, to handle images, authors propose to reshape images into flattened patches. Thus, a 2D image of shape $x \in \mathbb{R}^{H \times W \times C}$ can be represented in a form of $x_p \in \mathbb{R}^{N \times (P^2 \times C)}$ such that (P, P) denoting the resolution of each path and $N = (H \times W)/P^2$ denoting the number of patches [11]. The Transformer uses a constant-sized latent vector through all of its layers, thus in this work similarly authors mapped the flattened patches through a linear projection and named the output as patch embeddings.

As depicted in Figure 2.9, authors prepend a learnable embedding to the patch embeddings, to serve as a classification token [11]. Using a Multilayer Perceptrone (MLP) at the end of the network, the network classifies the images using the classification embeddings. In addition, 1D position embeddings are added to the patch embeddings in order to retain positional information [11].

The authors point out that, compared to CNNs, the Vision Transformer has much less image-specific inductive bias as only the MLP layers are local and translationally equivariant, whereas the self-attention layers are global [11]. However, in CNNs the locality issue is persistent throughout the whole model. The Vision Transformer attain groundbreaking results on some public datasets for image classification and become the state-of-the-art approach.

2.2.2. Channel Attention

Channel attention can be defined as an adaptive recalibration of the weight of each channel, directing the attention to specific channels in a feature map, also known as *what to pay attention* [13]. The pioneering work for channel attention was the research known as *Squeeze and Excitation Networks* [16]. On top of this article, many improvements were suggested on the operations that are applied known as *squeeze* and *excitation* [12, 47, 54].

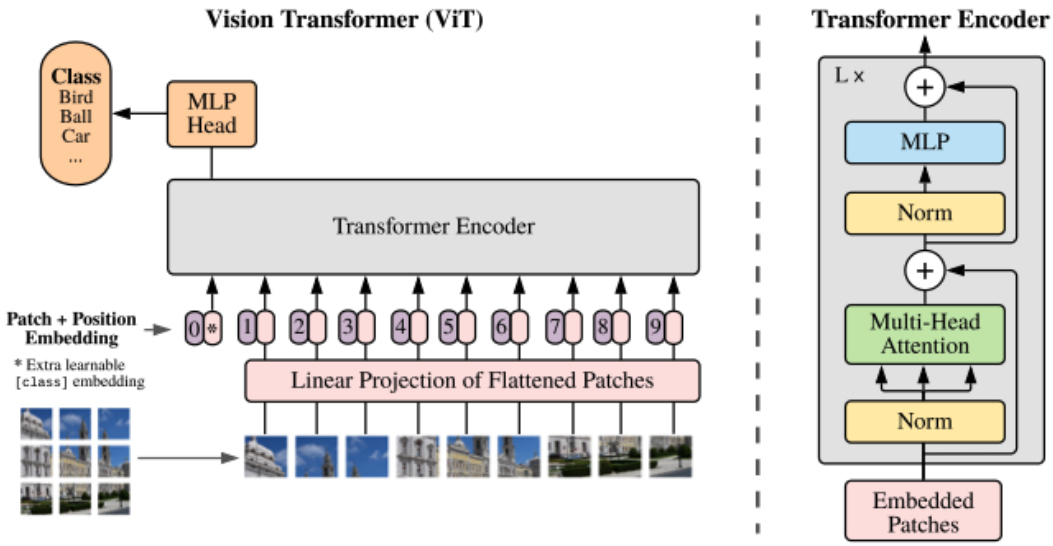


Figure 2.9.: Vision Transformer model overview: Split an image into fixed-size patches, linearly embed each patch, add position embedding, feed the resulting sequence to a Transformer encoder. Add an extra learnable "classification token" to the sequence [11]

Squeeze-and-Excitation networks

CNNs have shown that they are useful models for many computer vision tasks. At each convolutional layer, a collection of filters gather spatial connectivity patterns along the input channels and fuse them together, creating spatial and channel-wise information [16]. In this research, the authors investigate a different approach for applying attention to the relationship between the channels of a feature representation. Thus, the authors introduce the Squeeze-and-Excitation (SE) block with the goal of improving the representation quality by modeling the interdependencies between the channels of features by collecting global information and performing a feature recalibration by selectively emphasizing informative features and suppressing less useful ones [16].

The SE block consists of a two-step process as depicted in Figure 2.10. For a feature map represented as $U \in \mathbb{R}^{H \times W \times C}$ the features of U are first passed a *squeeze* operation, producing a channel descriptor by aggregating feature maps in terms of their spatial dimension [16]. Thus, an embedding of the global distribution of channel-wise feature responses is produced. Next, this aggregation is provided to an *excitation* operation, taking the embedding as an input and outputting a weight distribution per channel.

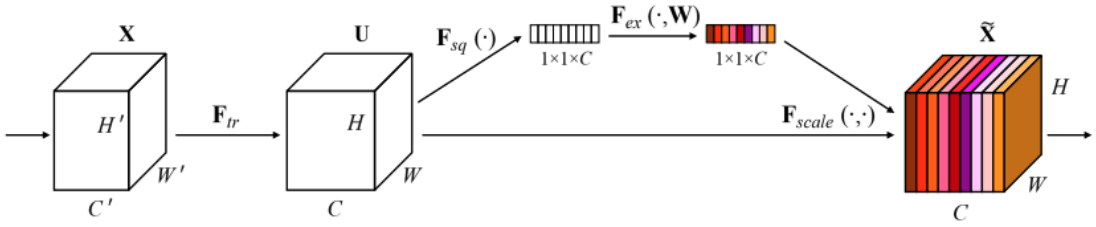


Figure 2.10.: Squeeze and Excite block [16]

Formally, the *squeeze* operation can be defined as in Equation 2.8, where z can be created by shrinking the feature map U through its spatial dimensions such that z_c being the c-th element of z. The *excite* operation can be defined as in Equation 2.9, where s being the scaled embeddings, such that the z from the first operation is trained with a weight vector of $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ and a ReLU function is applied to the output. Next, it is trained with a weight vector of $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$ where the output is passed to a sigmoid activation function. It is important to point out that C is the notation for channels and r is the reduction ratio of the embedding. Furthermore, the output of the weight distributed embedding is multiplied with the initial inputting feature embedding represented as u [16].

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{2.8}$$

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{2.9}$$

With this research, authors introduce a dynamics conditioned on the input, which maps the input-specific embedding to a set of channel weights, which can be regarded as a self-attention function on channels [16].

2.2.3. Temporal Attention

Temporal attention can be defined as a dynamic time selection mechanism, also known as *when to pay attention* [13]. This attention mechanism is usually used for video tasks. For temporal relation modeling, Recurrent Neural Networks (RNN) and temporal pooling operations were widely suggested to be used for video tasks. However, these operations are limited in terms of modeling success and efficiency [13].

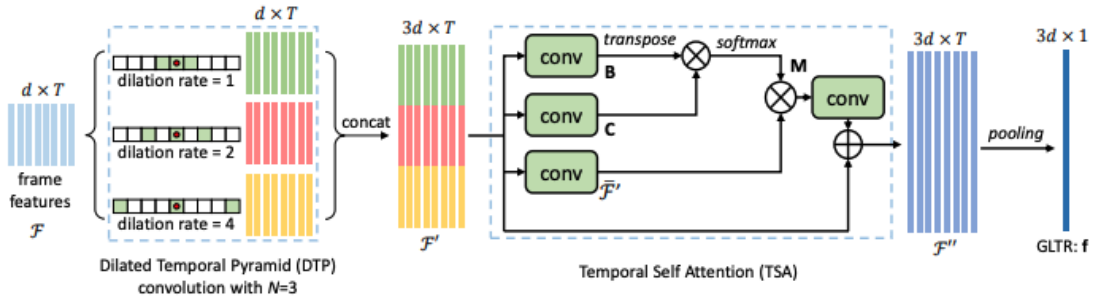


Figure 2.11.: Feature aggregation subnetwork for GLTR extraction consisting of Dilated Temporal Pyramid (DTP) convolution for local temporal and Temporal Self-Attention (TSA) model for global temporal cues [25]

One of the works that have focused on temporal attention is the *Global-Local Temporal Representation* (GLTR) research [25]. The task that was investigated by the authors is person Re-Identification from videos. As depicted in Figure 2.11, initially, Dilated Temporal Convolutions are used for local temporal feature learning [25]. Moreover, for global temporal features, a Temporal Self-Attention module is used for reducing the impact of occlusions and noises in the video data [25]. With this approach, authors were able to reach state-of-the-art results in person Re-Identification.

2.2.4. Branch Attention

Branch attention can be defined as in structures with multiple branches a dynamic branch selection mechanism, also known as *which to pay attention* [13].

One of the works that focus on branch attention is the *Selective Kernel Networks* [26]. In their work, authors propose a dynamic selection mechanism in CNNs, which allows each neuron to dynamically adjust the size of the receptive field based on multiple scales of input feature [26]. For this purpose, they define three operators, namely *split*, *fuse* and *select*. As provided in Figure 2.12, two branches are given, although the number of branches can be scaled to more.

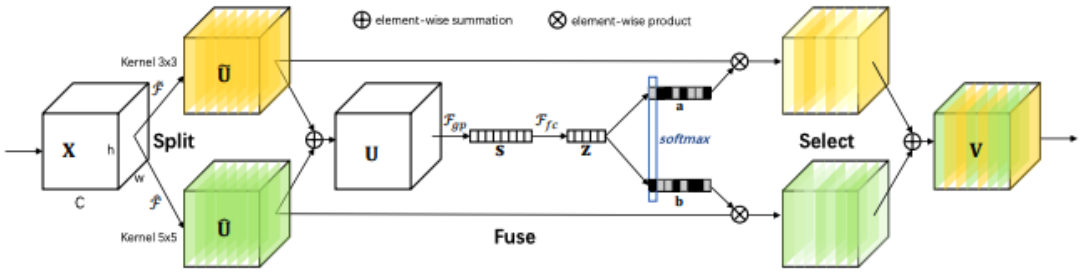


Figure 2.12.: Selective Kernel Networks [26]

In the *split* phase, transformations with different kernel sizes are applied and separate feature maps are generated [26, 13]. Next, in the *fuse* phase, with the basic idea of gates to control, information from all branches is fused to compute a gate vector, in order to control information flow from multiple branches. Finally, in *select* operation, soft attention across channels is used for aggregating feature maps from the branches. As the *Selective Kernel Networks* hold a lightweight design, this block can be applied to CNNs efficiently.

2.2.5. Mixed Attention

Spatial and Temporal Attention

The pioneering research covered self-attention application on computer vision, namely *Non-local Neural Networks* is designed in a way that can apply attention over space, time or spacetime [48]. Although, it is important to emphasize that the non-local operator does not process any temporal ordering information, while this is explicitly modeled in other spatial-temporal networks [49].

As the recent research done on the computer vision domain shows the success of Transformer architectures on image classification tasks compared to CNN approaches,

Transformer-based architectures for video recognition are also widely investigated [3, 4]. Both ViViT [3] and TimeSformer [4] employ factorized self-attention mechanisms, where in the transformer blocks defined the spatial and temporal attentions are applied separately and in sequentially, however, defined in opposite orders in both researches.

Video Swin Transformer

In the research done, known as *Video Swin Transformer*, authors present a pure-transformer architecture that reaches state-of-the-art accuracy on video recognition benchmarks [28]. The architecture benefits from the spatiotemporal locality of videos, such that pixels that are closer to each other in distance are more correlated to each other. Authors propose a spatiotemporal adaptation of the Swin Transformer [27] that was a prior work done utilizing self-attention mechanisms for image recognition tasks. This prior work builds hierarchical feature maps merging image patches in deeper layers [27], different from Vision Transformer [11].

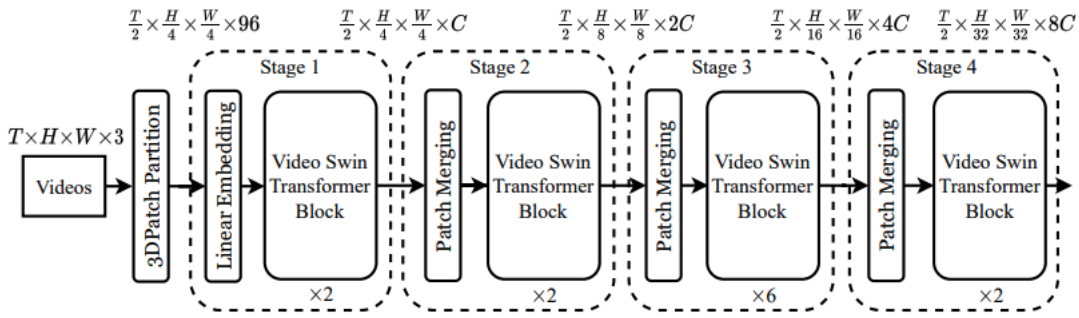


Figure 2.13.: The overview of Video Swin Transformer [28]

The architecture of the Video Swin Transformer, as provided in Figure 2.13, treats the input within smaller patches, as in Vision Transformer [11]. Without down sampling in the temporal dimension, the architecture consists of four stages, where the spatial domain is down sampled within the power of 2 in each stage. Furthermore, the patch merging layers after each stage concatenate the features of spatially neighboring patches into a group of 2×2, and apply a linear layer.

Within the *Video Swin Transformer Block*, a 3D Shifted Window based Multi-Head self-attention module exists, where a video represented in 3D tokens of $T^{'} \times H^{'} \times W^{'}$ and a window of size $P \times M \times M$, the tokens are partitioned by the windows in a non-overlapping way. As introduced in Figure 2.14, layer l adopts a regular window partitioning, whereas in layer l+1 the windows are being shifted.

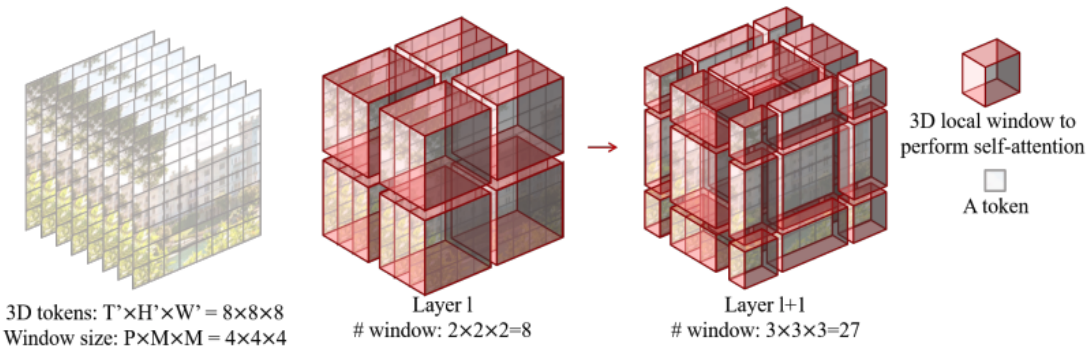


Figure 2.14.: 3D Shifted Windows. Layer l adopts a regular window partitioning, where layer l+1 has the windows shifted and the number is increased [28]

With this approach, the authors demonstrate that this approach was achieving state-of-the-art results among all other spatiotemporal networks for action recognition tasks on Kinetics 600 dataset.

Spatial and Channel Attention

Furthermore, there are many research done on combining spatial and channel attention in the same network. In other words, *where to pay attention* and *what to pay attention*. An important research done for combining two different domains of attention mechanisms were investigated in the article, *Convolutional Block Attention Module* (CBAM).

Convolutional Block Attention Module

CNNs demonstrated their success in computer vision tasks. Recent research mainly investigated *depth*, *width* and *cardinality* of networks [51]. In this research, authors investigate a different aspect of the architecture design, the attention mechanisms. The goal of the authors is to increase representation power by using attention mechanisms, thus focusing on important features and suppressing the unnecessary features [51]. As the convolution operations extract information by using cross-channel and spatial information together, authors design their module for emphasizing meaningful features along the channel and spatial domains [51]. As in Figure 2.15 where the main idea is depicted, authors propose to apply channel and spatial attention modules sequentially.

Given that a feature map $F \in \mathbb{R}^{C\times H\times W}$, CBAM infers a 1D channel attention map $M_c \in \mathbb{R}^{C\times 1\times 1}$ and a 2D spatial attention map $M_s \in \mathbb{R}^{1\times H\times W}$ [51]. Thus, the overall process depicted in Figure 2.15 can be mathematically formulated as $F^{'} = M_c(F) \otimes F$

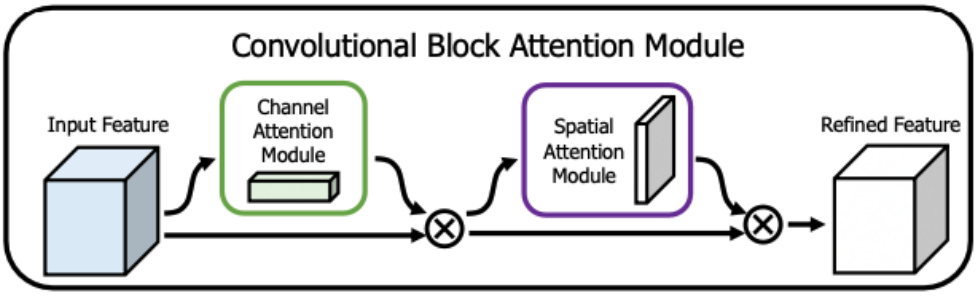


Figure 2.15.: The overview of CBAM. The module consists of two sequential sub-modules, channel and spatial. The feature map is refined through the CBAM module [51]

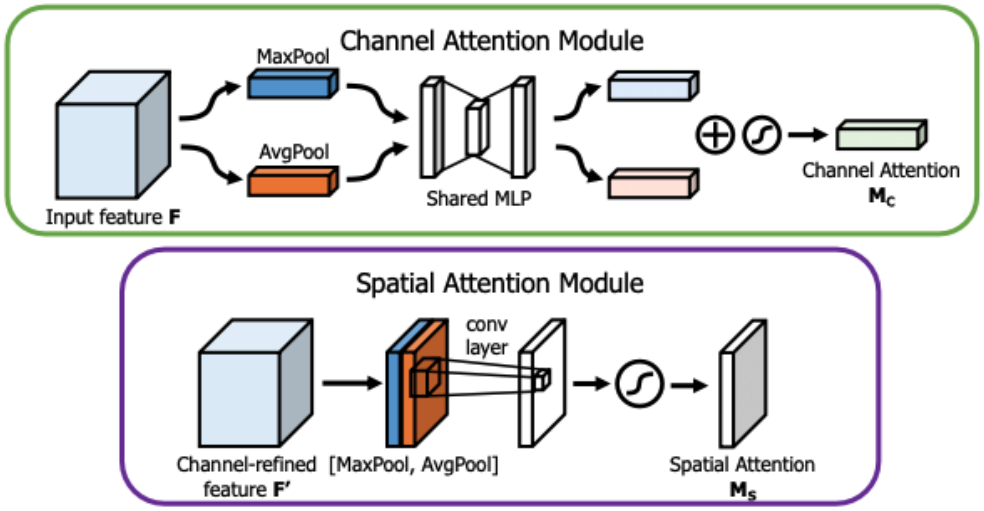


Figure 2.16.: The overview of each attention sub-module in CBAM. The channel module utilizes both max pooling and average pooling outputs with a shared network. The spatial module similarly utilizes two outputs pooled along the channel axis [51]

and $F'' = M_s(F') \otimes F'$ where $\otimes$ denotes element-wise multiplication [51].

Channel attention map makes use of inter-channel relationship of features. To compute the channel attention, the feature map F is squeezed from the spatial information, generating a 1D channel attention map. For aggregating the spacial dimension, average pooling is widely used [51]. However, authors also argue that max pooling gathers important feature representation also for inferring channel-wise attention. Thus, they propose to extract 1D representations by using both average pooling and max pooling. For generating the channel attention maps, both descriptors are forwarded to a shared MLP, and furthermore, the outputs of both descriptors are element-wise summed, as shown in Figure 2.16. To formulate mathematically, the channel attention is computed as $M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F)))$ [51].

The spatial attention map makes use of inter-spatial relationship of features. In order to compute the spatial attention, 2D average pooling and max pooling operations are applied to the intermediate feature map F' that is generated after the channel attention is applied to the inputting feature map [51]. The outputting 2D feature representations after the pooling operation are then concatenated, and a convolution layer is applied to generate the $M_s(F)$, as shown in Figure 2.16. To formulate mathematically, the spatial attention is computed as $M_s(F) = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)]))$, where f denotes a 2D convolution kernel with size 7 by 7 [51].

The CBAM block is also argued by the authors that it can be rearranged in different ways. For instance, the two sub-modules can be placed in parallel instead of sequentially [51]. Furthermore, the sub-modules can be placed as first the spatial attention instead of the channel attention. However, based on the experiments that they performed on various datasets, they found out that sequential arrangement and channel-first order was producing better results compared to other combinations [51].

2.3. Transfer Learning

Transfer learning can be defined as, transferring information from a related domain to improve a learner from another domain [50]. Among the various public datasets, the practice of training a CNN to perform an image classification task on ImageNet [10], and then adapting these features learned for a new task has been a widely accepted practice in computer vision tasks [17]. This procedure is also known as pre-training on another dataset and then fine-tuning the network for the target task and dataset. Using ImageNet pre-trained CNN weights shown promising results on various 2D image tasks such as image classification, object detection, action recognition, human pose estimation, image segmentation and other computer vision tasks [17].

Although the images in ImageNet [10] are widely different from the images in

medical datasets, recent works demonstrate that fine-tuning networks on ImageNet weights produced better results rather than training from scratch. Transfer learning is a common practice in medical image tasks, especially in cases where the training data size is limited [35]. A research was conducted on investigating the impact of transfer learning for melanoma screening [32]. The authors investigated classifying the melanoma screening dataset Interactive Atlas of Dermoscopy (Atlas) [2] using Deep Neural Networks (DNN). They have conducted their research by training the network from scratch, fine-tuning on another medical dataset namely Retinopathy [20] and fine-tuning on ImageNet [10]. Authors have obtained results such that fine-tuning on ImageNet was producing higher area under the ROC curve (AUC) values compared to other techniques. Although, extensive research conducted that argues on fine-tuning medical imaging tasks on ImageNet models offers limited performance gains against random initialization [36]. Furthermore, smaller architectures can perform comparably to the standard ImageNet models [36]. Another research compares the performance of the networks that use pre-trained weights on ImageNet against Radiological ImageNet (RadImageNet) [31]. In this work, authors show that for medical imaging tasks, using pre-trained network weights of RadImageNet for transfer learning provides better results.

Similar to 2D image tasks, 3D image tasks are also shown to be benefiting from transfer learning. In medical image datasets such as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), data is represented in the 3D domain as slices of multiple 2D images. In other terms, each frame of the data represents a slice of a 2D image, where concatenating these 2D images results in a 3D volume. A research conducted on investigating the pulmonary embolism classification from 3D CT images shown that, 3D medical image classification tasks can benefit from 3D CNNs that are pre-trained on Kinetics 600 [5] dataset [35]. It is important to note that the Kinetics [5] dataset is a human action dataset with the data type of short video clips for 600 different classes. However, there is also research conducted on how effective is transfer learning from Kinetics data. An article argues that although fine-tuning on top of Kinetics dataset utilizes 3D spatial information, the temporal video data and medical volume are different, causing a bias [8]. Nevertheless, authors show that transfer learning from Kinetics dataset is still beneficiary compared to training from scratch, although using a medical dataset produces better results [8].

2.4. Therapy Response Prediction

As mentioned previously, our goal in this thesis is to predict the therapy response of a patient by classifying diagnosis/pre-treatment CTs, using tumor progression labels for

supervision, which are assigned by doctors to patients after receiving therapy.

An approach for predicting therapy response using diagnosis/pre-treatment CTs was recently addressed in a research that uses the same in-house dataset that we are using in this thesis. The authors address the same goal using a three-stage pipeline as in Figure 2.17. The pipeline they propose is a hybrid deep neural network for predicting tumor response based on the Response Evaluation Criteria in Solid Tumors (RECIST) score.

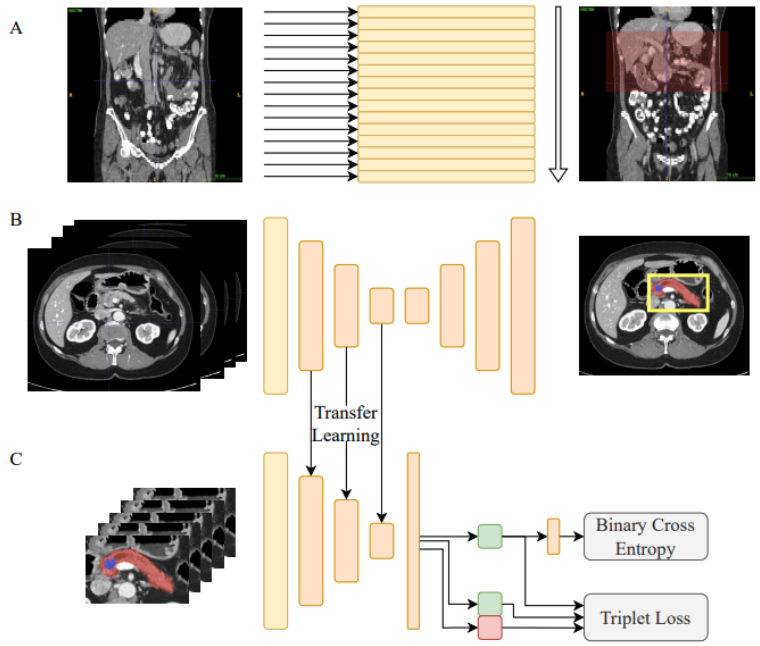


Figure 2.17.: The overview of three stage pipeline. Stage A: Binary classifier for each slice containing pancreatic tissue or tumor in a 2.5D approach. Stage B: Segmentation network for cropping to a bounding box. Stage C: Tumor progression classifier. [56]

The patients with pancreatic ductal adenocarcinoma (PDAC) have a 3D CT image scan with a varying number of CT slices either covering the whole body or a regional part. Therefore, in the first stage of the pipeline represented as *A* in Figure 2.17, authors aim to limit the search space in the z-Dimension of the 3D scan [56]. Therefore, the authors trained a binary classifier that predicts for each slice if it contains at least one pixel of the pancreas or tumor. Comparing to a segmentation network, such an approach is claimed to be more robust and can be trained on large datasets. For modeling a network applicable to 3D data, they applied a Long Short-Term Memory

(LSTM) cell after a 2D encoder, thus creating a 2.5D classifier.

In the second stage of the pipeline, authors trained a 3D segmentation network, a voxel-based architecture known as DynU-Net represented as *B* in Figure 2.17. Although training such a network is more expensive in terms of time and prune to inaccuracies, it allows them to reduce the search space in all x, y and z directions, converting segmentation to a bounding box around the pancreas [56].

Lastly, the authors reuse the network weights of the encoder of the 3D segmentation network benefiting from the transfer learning and building a RECIST classifier network. As the final component of their pipeline, authors benefited from representation learning and triplet loss by using an anchor image, a positive image from the same class and a negative image from the negative class [56].

For the first two stages, authors were dependent on segmentation labels. Since these are not present in our in-house dataset, they used the public Medical Segmentation Decathlon dataset [1]. With this approach, they were able to attain AUC of 63.7%, Matthew's Correlation Coefficient (MCC) score of 33.1% and accuracy of 67.2%.

Another work done on predicting therapy response was investigated on non-small cell lung cancer (NSCLC) [7]. The authors developed a deep multiple-instance learning (DMIL) based model for predicting chemotherapy response for treating NSCLC patients using their pre-treatment CT images. Multiple instance learning (MIL) is defined as a weakly supervised method where the model is trained using a label for a series of images (bag) instead of a single slice of image. This approach is argued to be beneficiary for extracting global features and also avoids the effect of a single false positive instance [7].

For preprocessing the data, CT voxels were re-sampled to have a resolution of $1 \times 1 \times 1mm^3$ where a cuboid was cropped from the CT images containing the whole tumor in the dimensions of $64 \times 64 \times 32$. The cuboid was treated as a bag, where each slice in the cuboid was treated as an instance. Furthermore, for the feature extraction module, five different models (AlexNet, VGG16, ResNet34, DenseNet, and MobileNetV2) were used which were pre-trained on ImageNet [10]. On top of the features extracted, three pooling mechanisms were applied; namely max pooling, convolutional pooling and attention mechanism pooling as in Figure 2.18.

With this approach, as the pre-trained feature extractor, the model with VGG16 obtained the best results with an AUC of 95.8% and an accuracy of 85.0%.

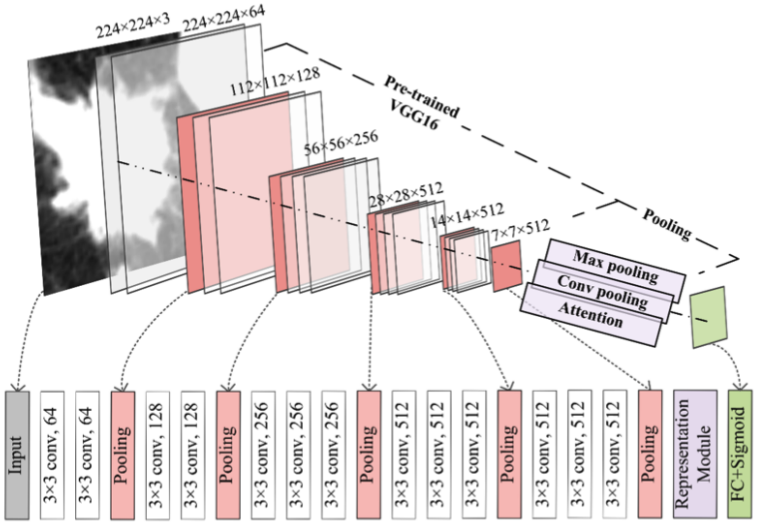


Figure 2.18.: Structure of the pre-trained VGG16 for feature extraction [7]

3. Background

3.1. Pancreatic Ductal Adenocarcinoma

Among the malignant neoplasms of the pancreas, pancreatic ductal adenocarcinoma (PDAC) is the most common disease accounting for more than 90% of all [33]. Moreover, PDAC is the fourth most frequent cause of cancer-related deaths [33]. To point out how lethal this disease is, it is important to note that the survival rate for a five-year period of time is less than 8% [39]. Furthermore, it is expected for diagnosed instances and deaths to increase by a factor of two from 2020 to 2030 within a 10-year frame [33].

The increase in PDAC instances is explained by the researchers to be related due to obesity and type 2 diabetes along with the aging of society. On top of that, alcohol and tobacco abuse are also listed as factors that have an unfavorable impact on survival rate [55, 9].

The outcome of the PDAC treatments largely depends on which stage the disease is being diagnosed [33]. Surgical resection followed by chemotherapy is the only possible therapy to cure PDAC [33]. However, only 10%-20% of the patients are suitable for surgical resection. This is mainly due to the disease being diagnosed as of symptoms that are often unspecific and unrecognized, leading to diagnostic delay [41]. The residual 80%-90% of the patients presented locally advanced, non-respectable stages including metastases [33]. Therefore, for patients that present non-resectable or borderline-resectable tumors, chemotherapy is the first-line treatment [33]. Predicting tumor response to a particular treatment, such as the choice of chemotherapy, is a challenging task. However, if tumor response can be predicted, personalized treatment could generate a higher survival rate for PDAC patients.

Previous studies have shown that pre-operative computed tomography (CT) imaging can be used for tumor cellularity characterization in PDAC patients [19]. Therefore, using state-of-the-art computer vision techniques, in this thesis our aim is to predict the therapy response of a patient by classifying diagnosis/pre-treatment CTs, using tumor progression labels for supervision. These labels are assigned by doctors to patients after receiving therapy and monitoring their progress, which is known as *Response Evaluation Criteria in Solid Tumors* (RECIST).

3.2. Response Evaluation Criteria in Solid Tumors

For managing patients with cancer, radiological imaging plays a crucial role in diagnosis and treatment procedures. Therefore, a standard assessment of response in solid tumors was introduced in 2000 and modified in 2009, namely, *Response Evaluation Criteria in Solid Tumors* [38].

The evaluation of time to disease progression was categorized into four classes with RECIST criteria [43]:

Response Label	**Definition**
Complete Response (CR)	Disappearance; confirmed at 4 weeks
Partial Response (PR)	30% decrease; confirmed at 4 weeks
Stable Disease (SD)	Neither PR not PD
Progressive Disease (PD)	20% increase; other labels not documented before increase

Table 3.1.: Definition of best response according to RECIST criteria [43]

Furthermore, these RECIST labels can also be used in a way that consists of three response labels as *Regressive Disease* (RD), *Stable Disease* (SD) and *Progressive Disease* (PD). Such labeling would refer to tumor size reduction, no significant change in tumor size and tumor size growth or metastasis development where the labels would be RD, SD and PD, respectively.

Moreover, in this thesis, we narrow down these labels into two classes for modeling tumor response prediction as a binary task. We group patients with *Regressive Disease*, *Partial Response* (PR) and *Stable Disease* as one class where the therapy response can be claimed to have a positive response and patients with *Progressive Disease* as another class for negative response to the therapy applied. The argument behind this grouping is that, although it is most favored to observe tumor size reduction, it can be also considered for the tumor to not grow or not develop metastasis as a favorable response.

4. Dataset

4.1. Pancreatic Ductal Adenocarcinoma Dataset

In this thesis our aim is to predict the therapy response of a patient by classifying diagnosis/pre-treatment computed tomography (CT) image, using tumor progression labels for supervision. These progression labels are assigned by doctors to patients with pancreatic ductal adenocarcinoma (PDAC) disease after receiving therapy and monitoring the progress. Such labels are known as Response Evaluation Criteria in Solid Tumors (RECIST). For this purpose, we use an in-house dataset that was collected over several years by Klinikum Rechts der Isar der Technische Universität München in Munich, Germany.

The dataset contains a total of 878 patients. Furthermore, there are 38 features for each patient which consist of therapy response, sex, age of diagnosis and other medical measurements. More importantly, the dataset also contains a CT scan that depicts either a whole or a part of the patient's body. Therefore, the number of slices in CTs varies between 29 to 1116. The spatial size of the CTs is 512×512.

Among these 878 patients, only 538 of them have the therapy response label. The response labels include *Regressive Disease* (RD), *Partial Response* (PR), *Stable Disease* (SD) and *Progressive Disease* (PD). As mentioned before in this thesis we narrow down these labels into two classes for modeling tumor response prediction as a binary task. Therefore, we group patients with *Regressive Disease*, *Partial Response* and *Stable Disease* as one class where the therapy response can be claimed to have a positive response and patients with *Progressive Disease* as another class for negative response to the therapy applied. From the 538 patients, 186 of them had *Progressive Disease* label whereas 352 were within the set of positive responses. It is important to note that the labels are sparse and applicable to certain patients. Moreover, there is a significant class imbalance as the ratio of PD to other labels is 34.6% to 65.4%. The number of slices in CTs varies between 38 to 1116. The mean number of slices in this dataset is 284.1 and the standard deviation is 250.1 rounded to one decimal point.

Among the patients with a therapy response label assigned, the therapy procedure applied can be classified into two classes. There are 240 patients that first had a resection operation and then chemotherapy. Furthermore, there are 291 patients that received first chemotherapy then a resection operation, or only chemotherapy.

Among these 878 patients, 865 have their sex labels assigned either as male or female. Of these 878 patients, 412 of them are female and 453 of them are male. Thus, contrary to the therapy response label, the female-to-male ratio is 47.6% to 52.4%, indicating class balance. The number of slices in CTs varies between 29 to 1116. The mean number of slices in this dataset is 298.7 and the standard deviation is 263.3 rounded to one decimal point.

TUM Ethics vote number: 180/17S.

4.2. Radiological ImageNet Dataset

Another dataset that we benefit from in this thesis is the Radiological ImageNet (RadImageNet) [31]. As in their work, authors argue that image classes in the ImageNet dataset [10] consist of object classes, therefore the models trained on ImageNet consist of images that are not related to the medical domain. Therefore, along with their research, the authors provide a dataset known as RadImageNet which consists of data collected from various sources such as computed tomography (CT), magnetic resonance imaging (MRI) and Ultrasound. Within the part of the dataset that consists of CTs, authors have split their dataset into two parts based on the location of the human body, namely *lung* and *abdomen/pelvis*. In line with our area of interest, the CT dataset of *abdomen/pelvis* is more related due to the pancreas being located there. In this dataset, there are 139,825 CT instances and 28 classes where each CT is a scan of a lesion, tissue or organ. The spatial size of the CTs is 224×224. Furthermore, the instances in this dataset are in the form of 2D, and thus have a depth of 1 slice.

4.3. Normal Pancreas Dataset

Another dataset that we utilize in our thesis is a separate in-house dataset that consists of 758 patient CTs. This dataset is also gathered over several years by Klinikum Rechts der Isar der Technische Universität München in Munich, Germany. The patients in this dataset have a CT scan taken for them and were controlled by radiologists such that none of the patients in this dataset were diagnosed with PDAC but may possess another disease. Similar to the Pancreatic Ductal Adenocarcinoma Dataset, this dataset also contains a CT scan that depicts either a whole or a part of the patient's body. Therefore, the number of slices in CTs varies between 1 to 326. Note that there is only one instance with a layer count of 1. The spatial size of the CTs is 512×512. The mean number of slices in this dataset is 129.7 and the standard deviation is 27.3 rounded to one decimal point.

4.4. External Pancreatic Ductal Adenocarcinoma Dataset

For evaluating our models, we have an external PDAC dataset that consists of 112 patient CTs. This dataset is for final evaluation purposes. Similar to the PDAC dataset, this dataset consists of diagnosis/pre-treatment CTs and their respective therapy response label. Furthermore, once again, we group patients with *Regressive Disease*, *Partial Response* and *Stable Disease* as positive responses and patients with *Progressive Disease* as negative responses to the therapy applied. Among the 112 patients, 79 of them have a positive response and 33 of them have a negative response. Thus, there is again a significant class imbalance as the ratio of PD to other labels is 29.5% to 70.5%. The number of slices in CTs varies between 44 to 284. The mean number of slices in this dataset is 104.7 and the standard deviation is 33.4 rounded to one decimal point.

5. Methodology

In this thesis we aim to predict the therapy response of pancreatic ductal adenocarcinoma (PDAC) patients by classifying their diagnosis/pre-treatment computed tomography (CT) using response evaluation criteria in solid tumors (RECIST) labels for supervision, which are assigned by doctors for tumor progression after a therapy being applied. As the nature of the task, our input data consists of 3D images, therefore we aim to tackle this classification problem by using deep neural networks that are capable to perform supervised learning on 3D images. Furthermore, we investigate the state-of-the-art attention approaches, where attention mechanisms can be defined as dynamic weight adjustment of a mask based on features of the input where this mask is used for enhancing salient information [13, 42, 53]. These attention mechanisms can be employed in spatial, channel, branch, temporal and mixed domains. In this thesis, we focuse on spatial, channel, temporal and mixed attention mechanisms.

The input data that we are conducting our experiments known as pancreatic ductal adenocarcinoma (PDAC) dataset consists of 3D CT images that although the spatial size is the same, CTs have a varying number of slices. The reason for this variation in the number of slices is due to some CTs rather displaying a smaller region of the patient's body, whereas other CTs are in greater detail. Even though, 3D networks can handle such inputs which a differentiating slice size by using a batch size of 1, we also investigate in novel 2.5D networks where a 2D encoder network is being used with slice attention mechanisms for targeting the temporal domain.

5.1. 2.5D Networks

In this section, we investigate the 2.5D network approaches that we mentioned where a 2D classifier is used as an encoder of the input with a slice attention mechanism attached to the output of the 2D encoder network aggregating the information extracted per slice and enhancing the slices that hold salient information for the classification task.

5.1.1. ResNet

For our 2D encoders, we benefit from the well-known residual networks, also known as ResNets [14]. With the residual networks, the authors aim to solve the degradation issue where as the network depth increases the training and testing errors are reported to be higher. Thus, the degradation of training accuracy indicates the challenge of optimizing the networks [14]. Thus, authors propose to formulate layers in deep neural networks by introducing *deep residual learning* framework as demonstrated in Figure 5.1. Authors hypothesized that optimizing residual mappings would be easier than an unreferenced mapping. The formulation can be realized as shortcut connections performing an identity mapping where neither extra parameters nor computational complexities are introduced.

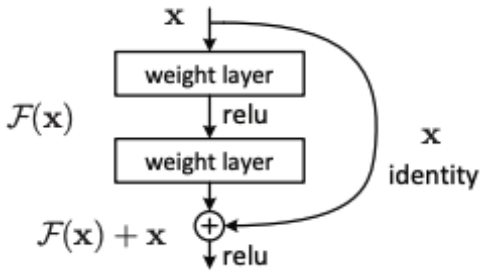


Figure 5.1.: Residual learning building block [14]

Authors provide various versions of the residual networks that they defined which differ in the number of layers, in other terms the depth. For networks such as ResNet18 and ResNet34, they benefit from the residual blocks provided on the left-hand side of the Figure 5.2. In this thesis, this block definition is referred to as a basic block.

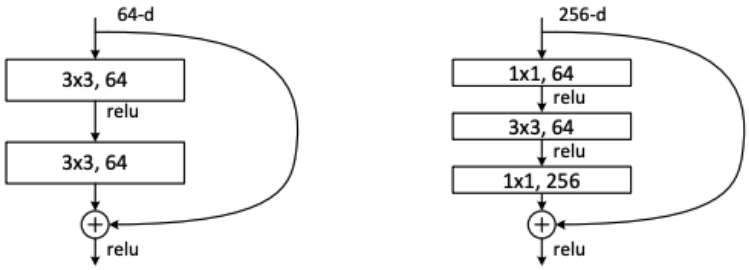


Figure 5.2.: Left: A "basic" building block. Right: A "bottleneck" building block [14]

In the article, the authors present multiple versions of ResNets with a different number of layers and parameters. They present shallower networks that are more robust and deeper networks that attain lower train error. For deeper networks such as ResNet50, ResNet101, and ResNet152 authors introduce the bottleneck building blocks as provided in the right-hand side of the Figure 5.2. Due to concerns about the training time, authors modified the basic block and created a new version named bottleneck block which consists of a 1×1 layer for reducing and then restoring the dimensions, leaving a 3×3 layer as a bottleneck. Thus, the basic block represented on the left-hand side of the Figure 5.2 has two convolutional layers whereas the bottleneck block represented on the right-hand side has three convolutional layers. In between these convolutional layers, there are also batch normalization and rectified linear unit operations.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 5.3.: ResNet architectures for ImageNet, building blocks are shown in brackets with the number of blocks stacked. Downsampling performed at $conv3_1$, $conv4_1$ and $conv5_1$ with stride 2 [14]

Various architecture definitions are provided in the research done. Each of the ResNets consists of four main layers. However, they differentiate based on the number of blocks in each layer and the type of block being used. In the paper, authors provide five different setups namely, ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152 as in Figure 5.3. The first two architectures consist of basic building blocks whereas the latter three consist of bottleneck building blocks. For comparing both approaches with observing the accuracy and loss metrics, in this thesis, we utilize ResNet18 and ResNet152 as the two extreme networks defined.

The ResNet18 network defined has four main layers. Based on the definition, in each

main layer, there are two basic blocks, thus making a total of 16 basic blocks distributed as 2, 2, 2, 2 from the first to the last main layer. The ResNet18 network consists of 11,689,512 parameters. Similarly, ResNet152 has four main layers. According to the definition, authors utilize bottleneck blocks total of 50 which are distributed as 3, 8, 36, 3 from the first to the last main layer. The ResNet152 network consists of 60,192,808 parameters.

Multi-Head Attention

ResNet18 + Att

This network is an extension of the ResNet18 that we employ in this thesis for benefiting from 2D networks to be applicable to handle 3D inputs for classification tasks. As provided in Figure 5.4, the ResNet18 architecture designed for ImageNet classification task consists of the following modules. ImageNet data consists of input images with three channels, therefore, the first convolution operator is defined as receiving three input channels and producing 64 output channels. However, the CT images are in grayscale, therefore consist of only one channel. Furthermore, the ImageNet dataset consists of 1000 image classes, however, for classification tasks that we handle in this thesis are binary classification tasks where the input data is 3D instead of 2D.

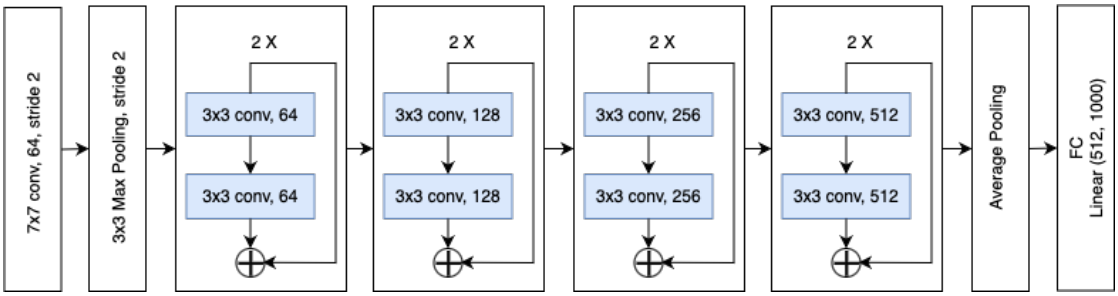


Figure 5.4.: ResNet18 architecture for ImageNet

In order to address the issues mentioned, we prepend an additional convolution operation of 1×1 where the input channel is one and the output channel is three. The reason that we do not directly modify the initial original convolution operator is for the sake of being able to hold the flexibility to use pre-training weights on ImageNet. Furthermore, we modify the fully connected layer at the end to output a vector size of 256 instead of 1000.

In order to use this architecture as 2.5D classifier we can forward a CT where the batch size is equivalent to the number of slices each CT has. Moreover, using features extracted from each slice, we have the novel intuition to apply Multi-Head Attention

[46] to these features for attention pooling and applying a fully connected layer to the attention output.

For this purpose as shown in Figure 5.5, we define the *query* of the Multi-Head Attention as taking the mean of the features with respect to the number of slices available per CT being forwarded. As each CT is represented within a dimension of [# of slices, 256], the result of the mean operation is equivalent to a feature dimension of [1, 256]. For the *key* and *value* parameters of the Multi-Head Attention, the whole CT volume representation is used. Thus, with regards to the Equation 5.1, matrix multiplication of *query* and the transpose of *key* produces an intermediate representation with dimensions [1, # of slices], which is then normalized by the square root of the embedding dimension being 256, producing the attention output weights. Furthermore, matrix multiplication of attention output weights and *value* generates the attention output with a dimension of [1, 256]. Finally, using a fully connected layer, we are able to produce a single output weight for a whole 3D CT volume.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{5.1}$$

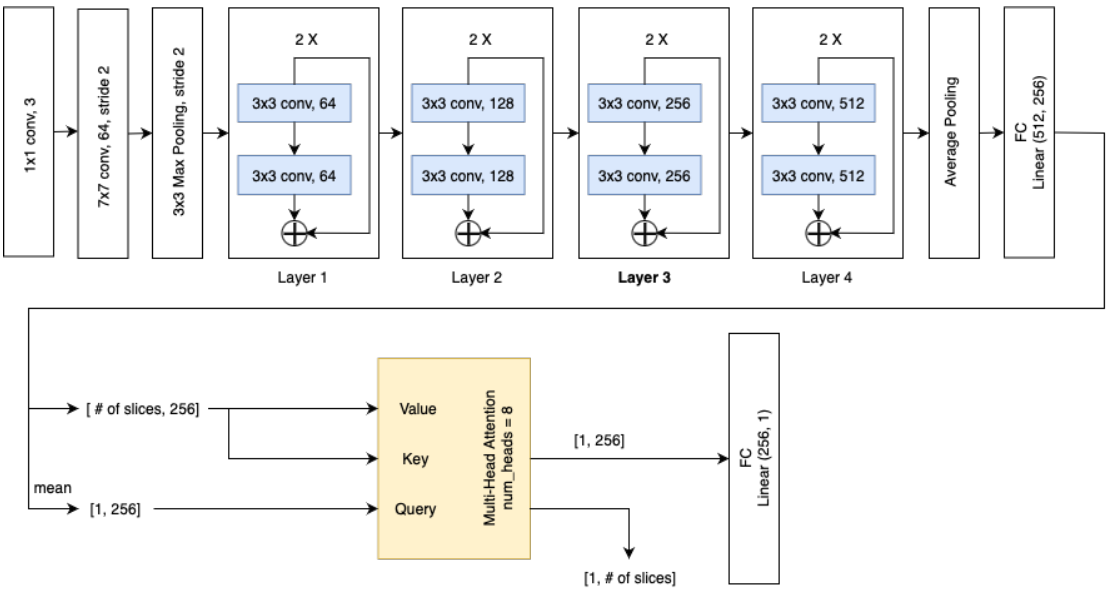


Figure 5.5.: *ResNet18 + Att* architecture

Using the modified ResNet18 architecture and applying Multi-Head Attention to the features extracted for a 3D volume as the number of slices being equivalent to the batch size, we define the 2.5D image classification architecture *ResNet18 + Att* as shown in

Figure 5.5. With the changes in the ResNet18 as the encoder part of this network, the adapted part holds 11,307,846 parameters. Furthermore, the Multi-Head Attention part consists of 263,168 and the classifier linear layer has 257 parameters. Thus creating a total of 11,571,271 parameters, which is less than the original ResNet18 proposed by the authors.

ResNet152 + Att

This network is an extension of the ResNet152 that we have employed in this thesis for benefiting from 2D networks to be applicable to handle 3D inputs for classification tasks, similar to *ResNet18 + Att*. Similar to ResNet18, as provided in Figure 5.6 the ResNet152 architecture designed for the ImageNet classification task consists of the following modules. It is important to recall ImageNet data instances consist of three channels thus the first convolution of ResNet152 accepts three input channels. Furthermore, the ImageNet dataset consists of 1000 image classes.

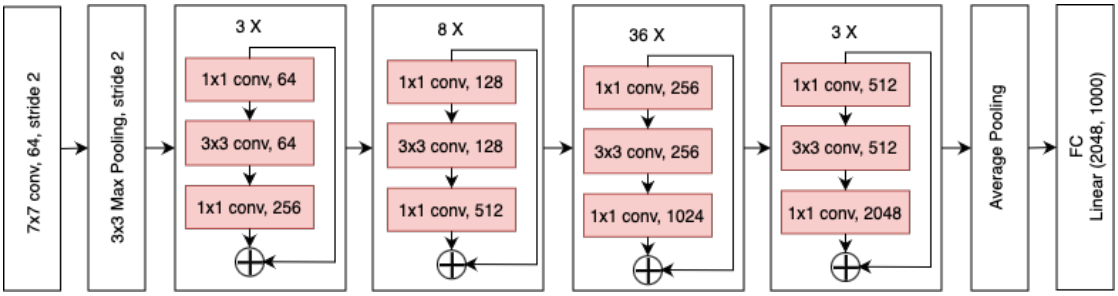


Figure 5.6.: ResNet152 architecture for ImageNet

Similar to the modifications applied for *ResNet18 + Att* an additional convolution operation of 1×1 where the input channel is one and the output channel is three. Furthermore, we modify the fully connected layer at the end to output a vector size of 2048 instead of 1000 as the representation power is greater for this network and we want to benefit from this. In order to use this architecture as 2.5D classifier we can compute each CT where the batch size is equivalent to the number of slices each CT has.

Likewise to our approach at *ResNet18 + Att*, for *ResNet152 + Att* using features extracted from each slice, we have the novel intuition to apply Multi-Head Attention [46] applying attention pooling and a fully connected layer to the attention output.

As depicted in Figure 5.7, we define the *query* to be the mean of the features extracted from all slices per CT. The features extracted from all slices have a dimension of [# of slices, 2048], and the mean operation results in a feature size of [1, 2048]. Similarly,

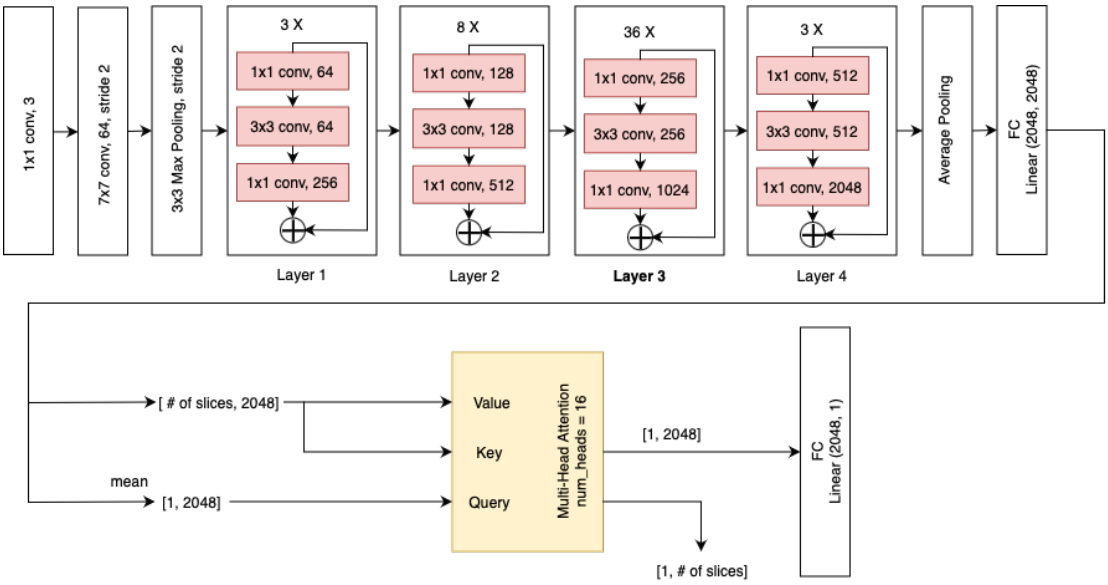


Figure 5.7.: *ResNet152 + Att* architecture

for *key* and *value* inputs of Multi-Head Attention, the whole CT representation is used. With regards to the Equation 5.1, the attention output weights hold a feature size of [1, # of slices] and the attention output is in a dimension of [1, 2048]. Furthermore, by applying a fully connected layer, a single output for a whole CT volume is obtainable.

With the changes in the ResNet152 as the encoder part of this network, the adapted part holds 62,340,166 parameters. Furthermore, the Multi-Head Attention part consists of 16,785,408 and the classifier linear layer has 2,049 parameters. Thus creating a total of 79,127,623 parameters, which is more than the original ResNet152 proposed by the authors.

Gather-Excite Networks

As mentioned previously, one of the leading research done in the field of spatial attention is the Gather Excite (GE) block [15]. In the article that they published, the authors show the benefit of GE blocks by embedding them into ResNet networks. As the GE block is a lightweight flexible model, it can be appended to a basic or bottleneck block in any main layer in the ResNet architectures as introduced above. Furthermore, the GE blocks can be selectively added to some or all of the blocks belonging to any of the main layers. In their findings, authors mentioned that, while GE blocks introduce improvements at every main layer, the greatest improvement comes from to mid and

late layers, arguing that there are more channels [15]. Thus, based on this intuition, throughout the thesis, we decide to add any form of attention mechanisms that can be individually introduced to ResNets to be added to every block located in the third main layer for 2D architectures and likewise for 3D architectures.

In the research done regarding GE blocks, authors define the attention mechanisms where a gather operator aggregates feature responses from spatial neighborhoods and an excite operator to produce an output that matches the initial input [15]. Thus, the intuition behind GE blocks is aggregating contextual information across large neighborhoods and modulating feature maps based on the information extracted. Furthermore, they define various versions of the GE blocks based on this definition, and many more can be defined as well. The authors demonstrate that parameter-free versions can be defined as well as parameterized versions. For the sake of simplicity and also observing the performance of parameter-free applications of attention mechanisms, we select to use the parameter-free version that is shown as in Figure 5.8.

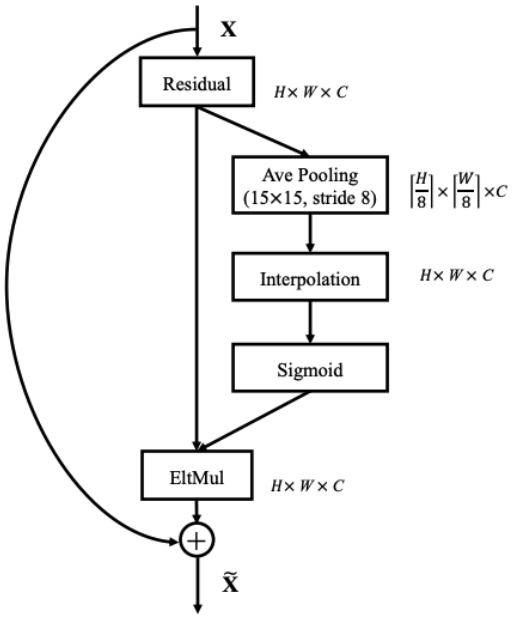


Figure 5.8.: Gather and Excite ResNet module parameter-free version ($GE-\theta^{-}$) [15]

The GE block defined in Figure 5.8 is designed to be added inside the basic or bottleneck blocks in ResNet between the identity addition to the output of the latest convolutional block. The gather operator is defined as applying an average pooling operation with sampling the spatial area to a reduced size by a factor of 8. Furthermore, the excite operator is defined as applying an interpolation operation to resize to the

original spatial size and applying a sigmoid operation to the output. Lastly, the output of the sigmoid operation is element-wise multiplied with the input.

ResNet18 + GE + Att

The GE block is a flexible module that can be placed after a convolutional operation that outputs a 3D feature representation with height, width and depth. In this thesis, we benefit from the GE block by placing the module into the two basic blocks in the third layer of the *ResNet18 + Att* and defining an architecture namely *ResNet18 + GE + Att*. As shown in Figure 5.9 where the third main layer is modified, defining a separate branch an average pooling operation is applied to the output of the latest convolution operation within the block. The average pooling operation is defined to have a kernel size of 8×8 and a stride of 8. Thus an input with the spatial size of [16, 16] is sampled to a size of [2, 2]. Furthermore, applying interpolation we resize the features extracted by the pooling operation back to [16, 16]. Next, applying a sigmoid function, the values are sampled between 0 and 1. Lastly, by multiplying the output from the sigmoid function with the initial input to the branch we are able to insert the GE block into a ResNet basic block.

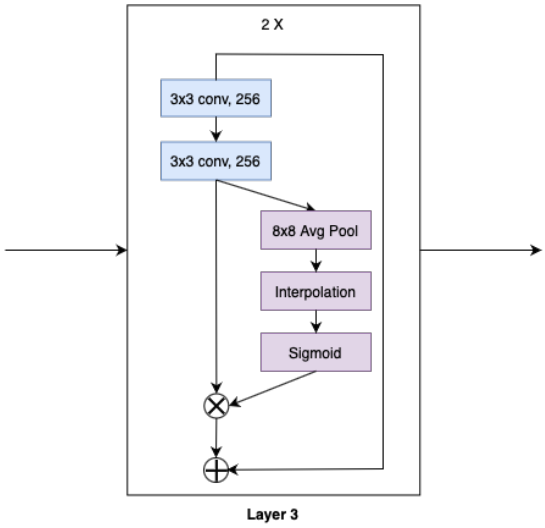


Figure 5.9.: *ResNet18 + GE + Att* architecture

As the GE block that we utilize in *ResNet18 + GE + Att* is the parameter-free version, the number of parameters in this network is the same as *ResNet18 + Att* consisting of 11,307,846 parameters in the encoder, 263,168 parameters in the Multi-Head Attention and 257 parameters in the fully connected classifier.

ResNet152 + GE + Att

As introduced in *ResNet18 + GE + Att*, this network is the version that utilizes the GE block in the third main layer of *ResNet152 + Att*. In this network, the third main layer of ResNet152 consists of 36 bottleneck blocks. Furthermore, the bottleneck blocks consist of three convolution operations. Thus, the bottleneck block in the third layer of this network is demonstrated as in Figure 5.10. Since, the rest of the network is the same as in *ResNet152 + Att*, only the changes in the third layer are shown. Similarly, as introduced above, the output of the last convolution operation in the bottleneck block is the input to a separate branch defined for gather and excite operations. An average pooling operation with kernel size 8×8 is applied with the stride of 8, causing the feature representation size to get reduced. Later on, applying interpolation, the feature size is rearranged and lastly, a sigmoid operation is applied. Furthermore, element-wise multiplying the output of the sigmoid operation with the output of the last convolution operation in the bottleneck block, the connection in the bottleneck block is achieved.

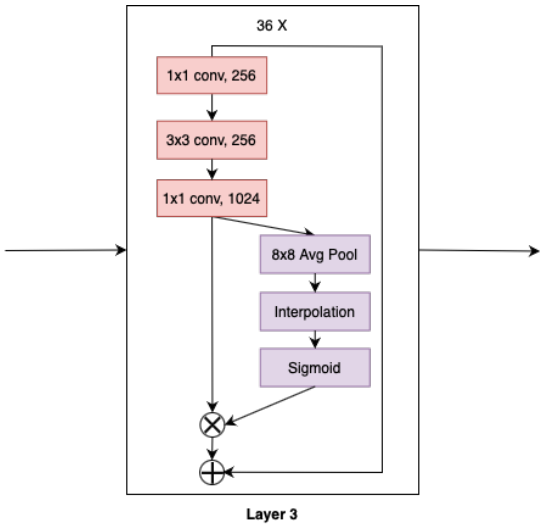


Figure 5.10.: *ResNet152 + GE + Att* architecture

As the GE block that we utilized in *ResNet152 + GE + Att* is the parameter-free version, the number of parameters in this network is the same as *ResNet152 + Att* consisting of 62,340,166 parameters in the encoder, 16,785,408 parameters in the Multi-Head Attention and 2,049 parameters in the fully connected classifier.

Squeeze-and-Excitation Networks

One of the early approaches for employing attention mechanisms and the pioneering work on channel attention is the *Squeeze-and-Excitation Networks* [16]. The aim of the Squeeze-and-Excitation (SE) block is to improve the representation quality produced by a network, by modeling the interdependencies between channels of the convolutional features. Thus, using SE blocks, the aim is to allow networks to perform feature recalibration using global information to selectively emphasize informative features while suppressing less useful ones.

For embedding SE blocks into ResNet architectures, similar to GE blocks we decide to include them in all basic and bottleneck blocks residing in the main third layer of the ResNet18 and ResNet152 architectures. Thus, including these blocks in the mid and later stages of the networks where more channels are present, we had the intuition that channel-wise feature recalibration would be much more crucial and effective.

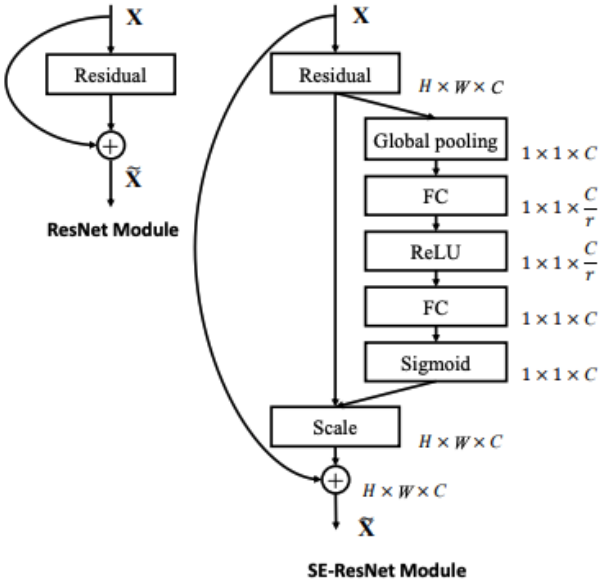


Figure 5.11.: Original ResNet module (left) and Squeeze and Excitation ResNet module (right) [16]

The SE block introduced by the authors, as shown in Figure 5.11, creating a new branch from a feature map extracted from the last convolution operation in a basic or bottleneck block with dimensions $H \times W \times C$, using pooling techniques a more refined feature representation with dimensions $1 \times 1 \times C$ can be extracted. From the refined features, a fully connected linear layer is used where the input channel is equal to the

number of channels C and the output channel is equal to C/r where r is the reduction ratio. After applying a rectified linear unit function another fully connected layer is used once again to this time increase the number of channels back to C. Next, applying a sigmoid function to these refined features we eventually retrieve a map of dimensions $1 \times 1 \times C$ which holds a weight multiplier for each channel. Lastly, multiplying the input that has dimensions of $H \times W \times C$ with the map of $1 \times 1 \times C$ we recalibrate the feature map with respect to channels. In their research, the authors presented multiple reduction ratios. In this thesis, we set the reduction ratio to be 16.

ResNet18 + SE + Att

This network is an extension of *ResNet18 + Att* where SE blocks are utilized in the third main layer of the ResNet18 encoder. The SE block is a flexible module that can be included after a convolutional block that produces a feature extraction representation with height, width and channels. In this thesis, as we also mentioned above, we decide to add attention mechanisms to the third main layer of the networks. Thus, in this network, we proceed with the same approach.

As shown in Figure 5.12, the SE block is implemented within the basic blocks in the third main layer. The SE block is formed by creating a new branch from the latest convolution operation in the basic block. Within the branch, first, an adaptive average pooling is applied such that the feature map obtains the dimensions $1 \times 1 \times 256$ from a feature map of size $H \times W \times 256$. Furthermore, as the reduction ratio size is set to 16, thus we obtain a new feature map of size $1 \times 1 \times 16$. After applying an activation function, using a fully connected layer we again increase the number of channels to 256. Next applying a sigmoid function to the feature map resized, we hold a feature map with dimensions $1 \times 1 \times 256$, which is equivalent to the initial number of channels. Thus, with the end product of the branch, multiplying each channel with the recalibrated channel values, we are able to apply channel-wise attention.

With the changes in the ResNet18 as the encoder part of this network with the applicable changes third main layer for SE block, the adapted part holds 11,324,774 parameters. Due to additional linear layers, the encoder part now consists of more parameters. Furthermore, again the Multi-Head Attention part consists of 263,168 and the classifier linear layer has 257 parameters. Thus, creating a total of 11,588,199 parameters.

ResNet152 + SE + Att

This network is an extension of *ResNet152 + Att* where the third main layer of ResNet152 is modified such that the bottleneck blocks consists of SE blocks. Similar to the approach

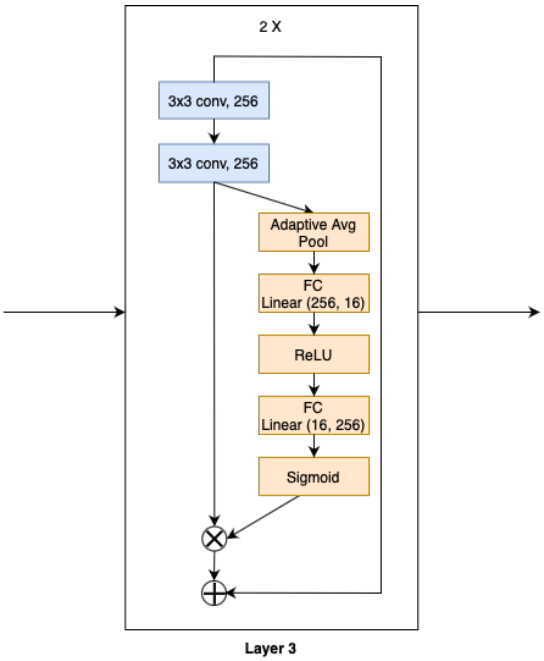


Figure 5.12.: *ResNet18 + SE + Att* architecture

described for *ResNet18 + SE + Att*, from the output of the last convolution operation in the bottleneck blocks a new branch is created. As shown in Figure 5.13, the latest convolution operation in the bottleneck block produces a feature map with dimensions $H \times W \times 1024$. As stated previously, the reduction ratio was decided to be 16, thus after applying a global pooling, using a fully connected layer we are reducing the number of channels from 1024 to 64. Furthermore, using another linear layer we are resizing to the original dimensions. After applying the sigmoid function, we eventually hold a recalibrated feature map of dimension $1 \times 1 \times 1024$, which we then use to redistribute weights per channel.

With the changes in the ResNet152 as the encoder part of this network with the applicable changes third main layer for SE block, the adapted part holds 67,097,926 parameters. Furthermore, again the Multi-Head Attention part consists of 16,785,408 and the classifier linear layer has 2,049 parameters. Thus, creating a total of 83,885,383 parameters.

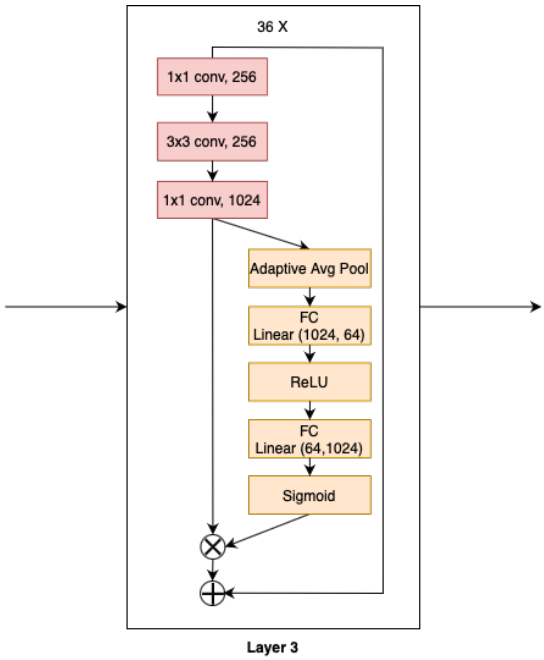


Figure 5.13.: *ResNet152 + SE + Att* architecture

Convolutional Block Attention Module

An approach that combines both spatial and channel attention is introduced as the Convolutional Block Attention Module (CBAM). Similar to GE and SE blocks, CBAM is also a lightweight module that can be integrated into CNN architectures. As to recall, CBAM consists of two sub-modules which are sequentially connected in an order of first channel attention and then second spatial attention. Authors state that they also experimented with other combinations such as having a spatial first channel second and also connecting two modules in a parallel way. However, they observed that the best performance was attained with the order introduced as in Figure 5.14.

Despite the fact that the CBAM module can be included in any CNNs, authors have also experimented with the performance of their mixed attention module by using ResNets as provided in Figure 5.14. Furthermore, similarly, as we decide to keep GE and SE blocks included in the third main layer of the ResNet architectures that we are using, once again we have included CBAM modules in all of the basic and bottleneck blocks in the third main layer. With this purpose, we are able to observe changes in performance by keeping the overall structure of the main architecture fixed and modifying the relevant part of the network.

As mentioned before the CBAM block consists of two sub-modules, namely channel

attention and spatial attention. In the channel attention part, the principle is similar to the SE block. Using the output of the last convolution operation in a basic or bottleneck block, two pooling operations are applied to the feature map independently. From a feature map with dimensions $H \times W \times C$, a branch is created where an average pooling operation is applied to obtain a representation with dimensions $1 \times 1 \times C$, and a max pooling operation is applied to obtain another representation with dimensions $1 \times 1 \times C$. Thus, two we obtain an average pooling per channel and a max pooling per channel. Furthermore, these two representations are forwarded from a shared fully connected layer with a reduction ratio of 16 and from another fully connected layer for rescaling back to the original size of channels. These two fully connected layers contain a rectified linear unit operation in between. Furthermore, the output for the average and max pooling operations are summed and a sigmoid is applied to the output of the summation. Furthermore, this recalibrated vector is multiplied with the initial output from the last convolutional layer, thus reweighting the channels of the feature map.

From this recalibrated feature map another branch is created for the spatial attention. Similar to what we did for the channel attention, from the feature map of $H \times W \times C$ shape, a mean operation is performed in the channel dimension producing a plane of $H \times W \times 1$ and a max operation is performed in the channel dimension also producing a plane of $H \times W \times 1$. Concatenating these two planes, we obtain a $H \times W \times 2$ and next we apply a convolution operation which inputs two channels and outputs one channel, where the kernel size is 7×7. Forwarding the result from the convolution operation to a sigmoid function, we once again obtain a spatial map of weight distribution. Using this spatial map, we multiply the feature map that was recalibrated by the channel attention and this time reweighting the spatial area within each channel.

With this sequential operation being applied after the last convolution of each basic or bottleneck block, we are able to apply a mixed convolution within the blocks of ResNet networks.

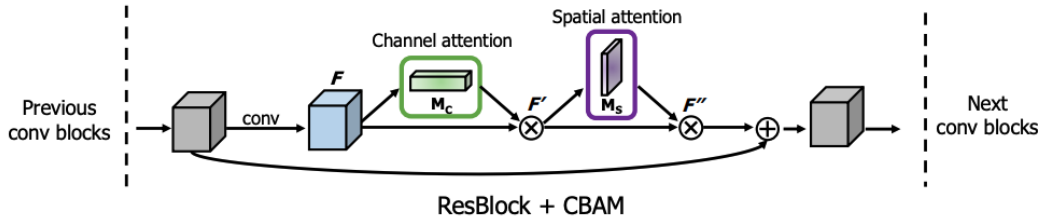


Figure 5.14.: CBAM integrated with a block in ResNet [51]

ResNet18 + CBAM + Att

This network is an extension of *ResNet18 + Att* where we introduce the CBAM module to every basic block in the third layer of the encoder part of this architecture. As shown in Figure 5.15, the CBAM module is implemented by sequentially applying channel and then spatial attention mechanisms.

Firstly, for the channel attention module a new branch is formed by using the output of the latest convolution operation in the basic block which produces a feature representation with dimensions $H \times W \times 256$. From this representation, two separate pooling operations are performed as average pooling and max pooling with respect to the channel dimension producing pooling outputs of $1 \times 1 \times 256$. Using a reduction ratio of 16, forwarding to a fully connected layer the pooling outputs are reduced to $1 \times 1 \times 16$ and then with another fully connected layer reformed back to $1 \times 1 \times 256$. A rectified linear unit operation is performed in between. Summing the outputs from the shared multi-layer perceptron and using a sigmoid function, the channel-wise attention is applied by multiplying it with the produced recalibration vector.

Secondly, for spatial attention, two planes are extracted from the by-product by taking the mean and max in terms of the spatial dimension. Concatenating these two planes and applying a convolution operation followed by a sigmoid function we hold a spatial attention map which is once again multiplied with the by-product. Lastly, the identity summation operation is done for the residual networks principle.

With the changes in the ResNet18 as the encoder part of this network with the applicable changes third main layer for CBAM block, the adapted part holds 11,324,426 parameters. Due to additional linear layers, the encoder part now consists of more parameters. Furthermore, again the Multi-Head Attention part consists of 263,168 and the classifier linear layer has 257 parameters. Thus, creating a total of 11,587,851 parameters.

ResNet152 + CBAM + Att

Similar to *ResNet18 + CBAM + Att*, this network shown in Figure 5.16 is an extension to the ResNet152 being used as an encoder network component for *ResNet152 + Att*. Different than in *ResNet18 + CBAM + Att*, the number of channels produced by the last operation in the bottleneck block in the third main layer possess 1024 channels. Therefore the pooling operations output channel attention vectors of $1 \times 1 \times 1024$. Furthermore, as the reduction ratio is set to 16, the fully connected layer in the channel attention first downsizes to a vector of $1 \times 1 \times 64$ and then the second fully connected layer increases the number of channels back to $1 \times 1 \times 1024$. The rest of the operations are the same as the ones described in the *ResNet18 + CBAM + Att*.

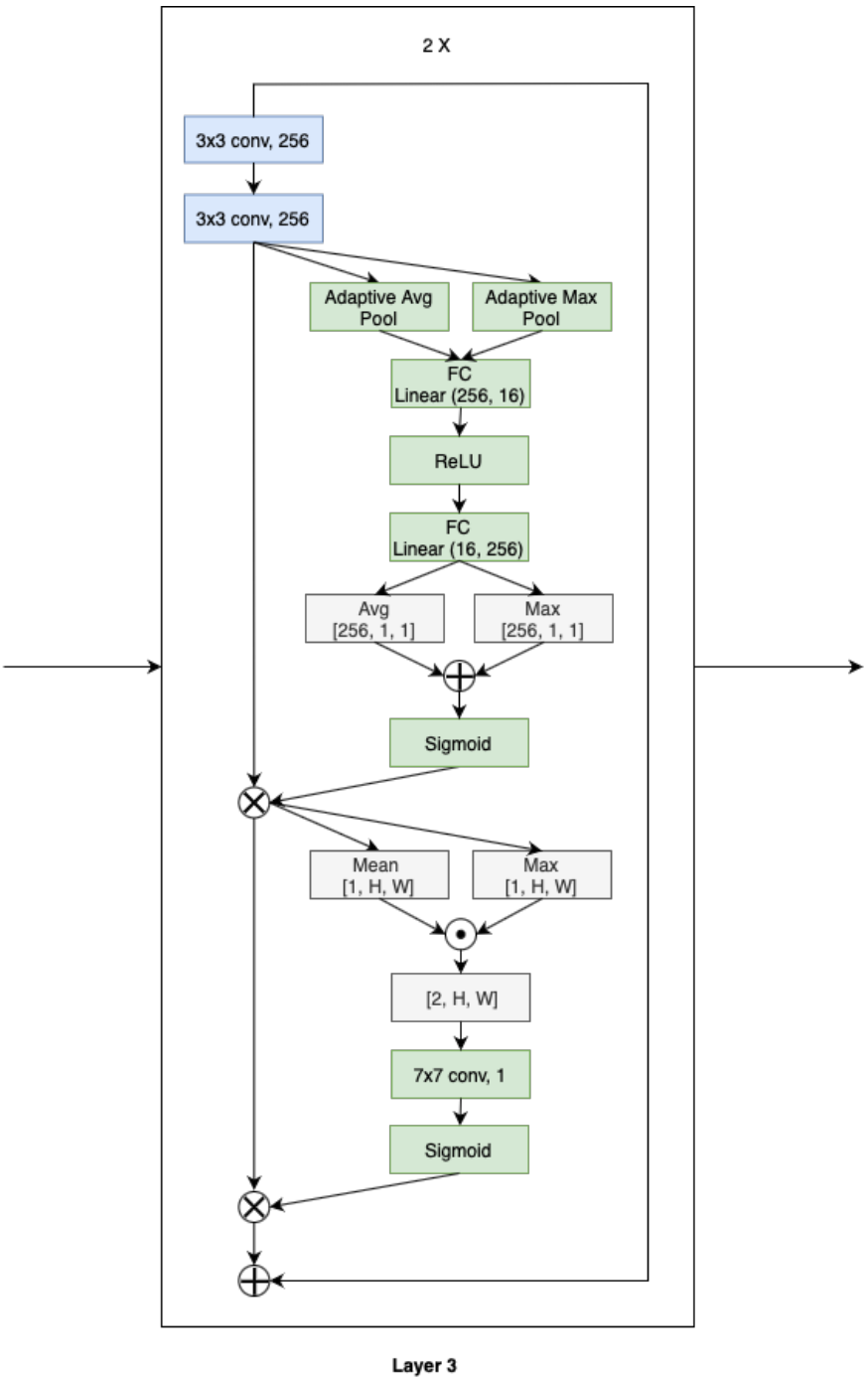


Figure 5.15.: *ResNet18 + CBAM + Att* architecture

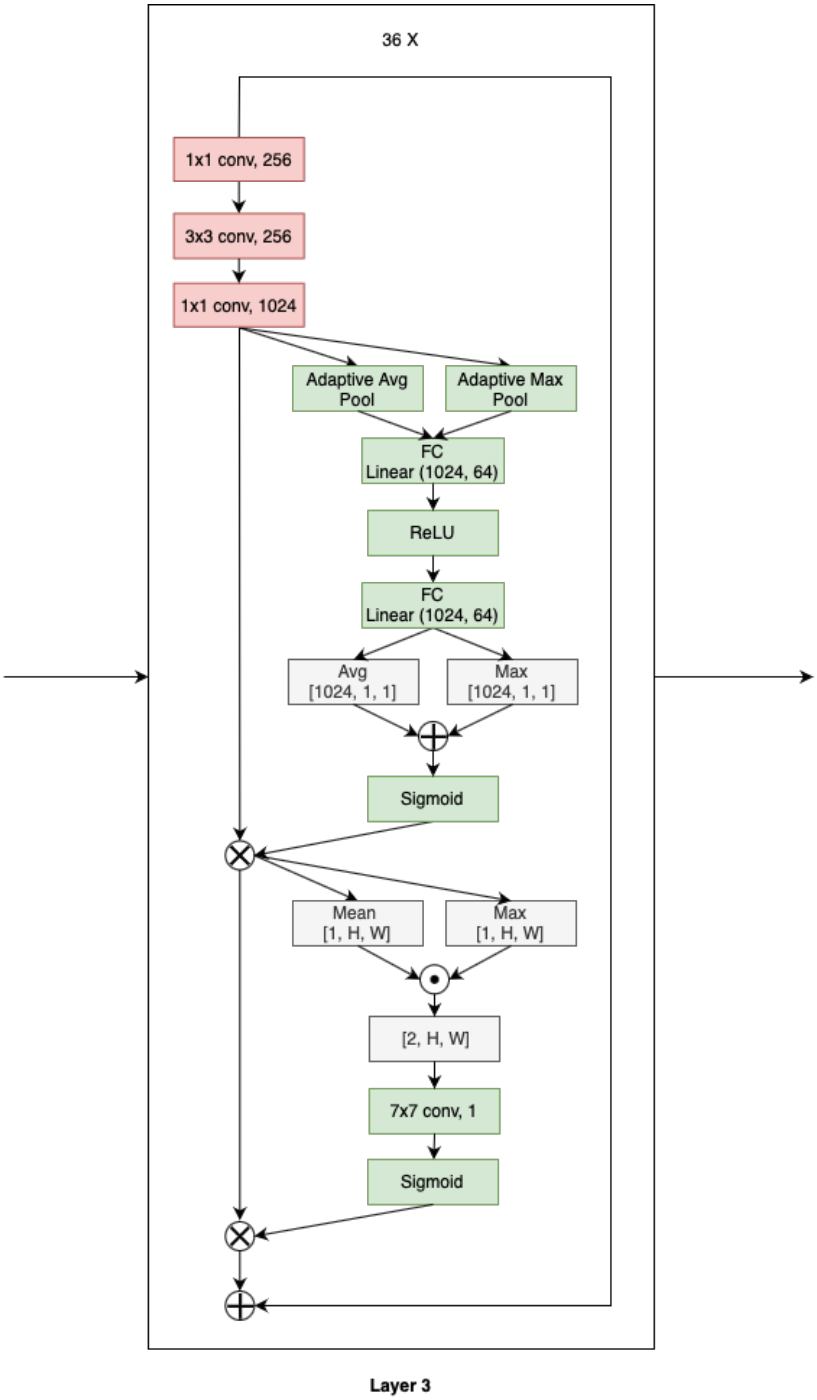


Figure 5.16.: *ResNet152 + CBAM + Att* architecture

With the changes in the ResNet152 as the encoder part of this network with the applicable changes third main layer for CBAM block, the adapted part holds 67,062,286 parameters. Furthermore, again the Multi-Head Attention part consists of 16,785,408 and the classifier linear layer has 2,049 parameters. Thus, creating a total of 83,849,743 parameters.

5.1.2. Vision Transformer

Another 2D encoder that we have used in our networks is the Vision Transformer [11]. In this network, contrary to the ones introduced as residual networks, the architecture does not rely on the convolutional operations. Instead, inspired by the Multi-Head Attention [46] and Transformer architecture authors proposed to applying the same approach to computer vision tasks.

The standard Transformer receives input as a 1D sequence of token embeddings. Therefore, to handle images, authors propose to reshape images into flattened patches. Thus, a 2D image of shape $x \in \mathbb{R}^{H \times W \times C}$ can be represented in a form of $x_p \in \mathbb{R}^{N \times (P^2 \times C)}$ such that (P, P) denoting the resolution of each path and $N = (H \times W)/P^2$ denoting the number of patches [11].

Flattened patches are forwarded through a linear projection producing patch embeddings. Furthermore, a classification token is prepended to these patch embeddings. Moreover, position embeddings are added to the patch embeddings for retaining positional information. For the positional information, 1D embeddings are used. Within the Transformer Encoder as in Figure 5.18, multiple layers of encoder blocks are located which are connected sequentially. Using the output of the last encoder block and taking the classification tokens, located as the top vector, which is a vector with the size of the embedding dimension, a fully connected layer is used for producing weights that are equal to the number of classes.

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Figure 5.17.: Details of Vision Transformer model variants [11]

The architecture of Vision Transformer proposed by authors with a patch size of 16 and 12 encoder layers consists of 86,567,656 parameters.

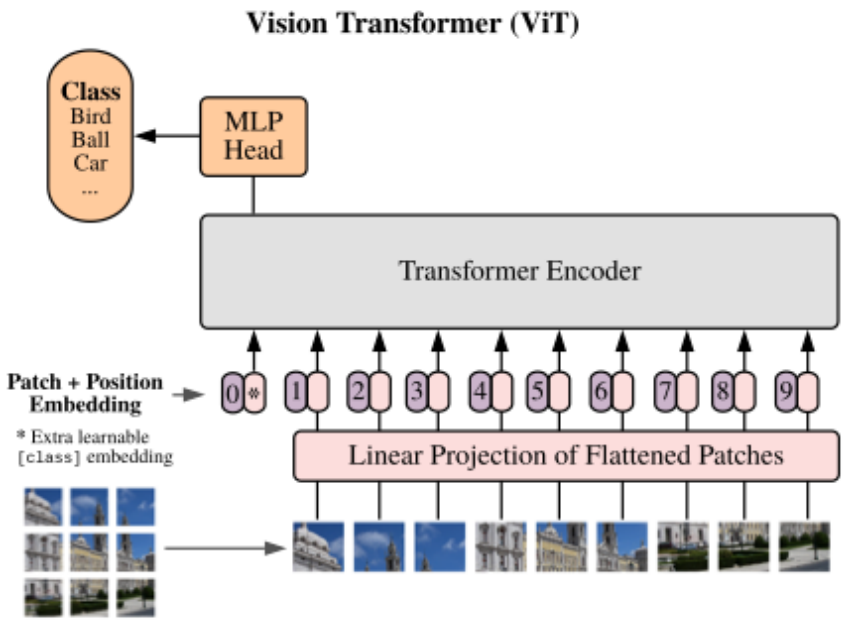


Figure 5.18.: Vision Transformer model overview [11]

Multi-Head Attention

ViT b 16 + Att

This network is adapted from the Vision Transformer architecture and converted to a 2.5D classifier using the attention pooling approach described in the previous classifiers introduced as *ResNet18 + Att* and *ResNet152 + Att*.

As provided in Figure 5.19, the ViT b 16 architecture is designed for the ImageNet classification task which consists of the following modules. The ImageNet dataset consists of images with three channels, thus the first convolution operation inputs three channels and outputs 768 channels as being the latent vector size as the patch size is equal to 16. Furthermore, it is also important to note that the ImageNet dataset consists of images with 1000 different classes thus the final fully connected layer produces 1000 weights by inputting a vector size of 768 taking the classification tokens.

In order to adapt the ViT b 16 architecture for our task which consists of CT images that are in grayscale we address the issue by prepending an additional convolution operation of 1×1 where the input channel is one and the output channel is three. We do not modify the initial convolution layer due to the fact that we want to keep the structure for holding the option of being able to use pre-trained weights. Moreover, we

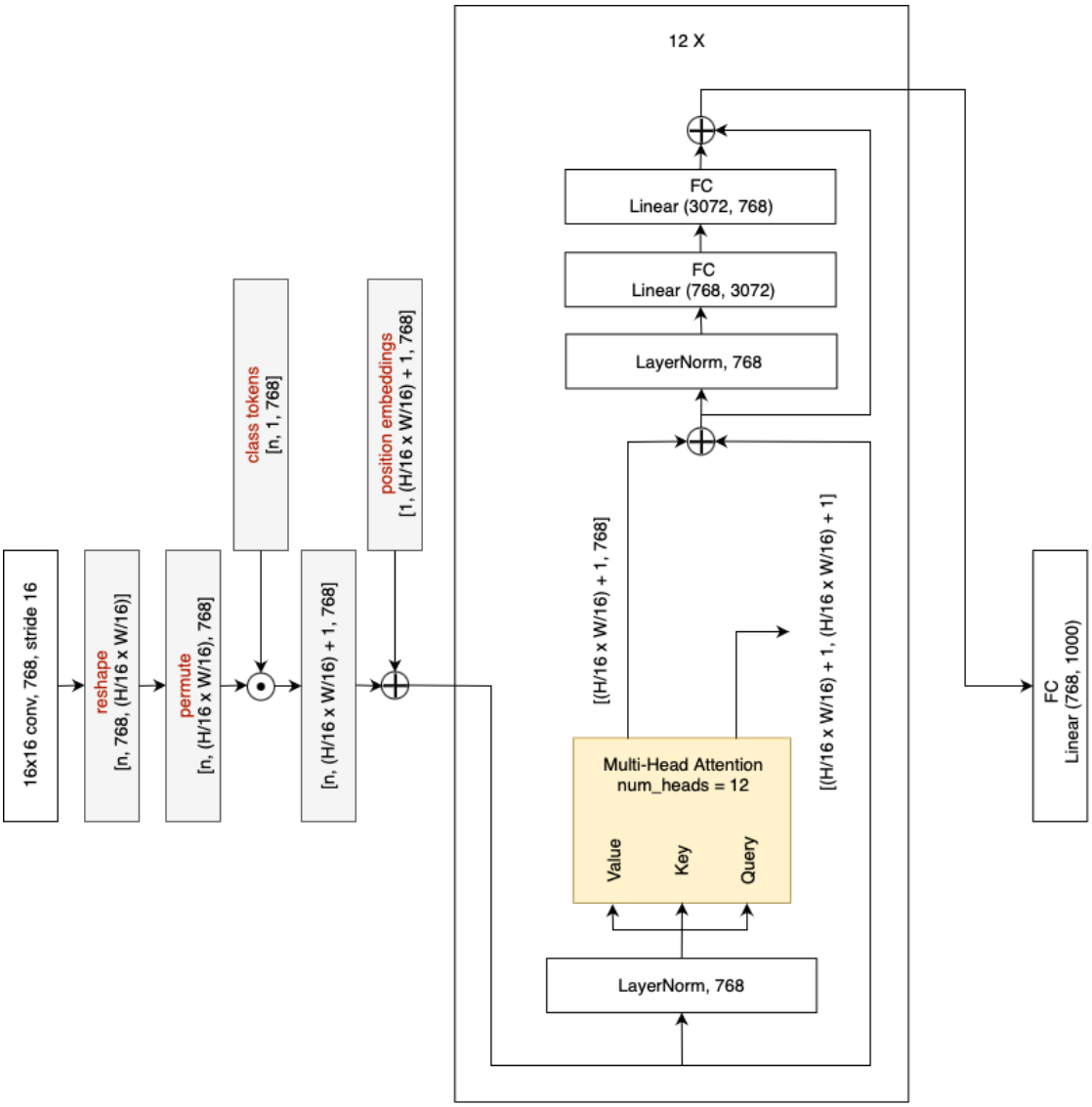


Figure 5.19.: ViT b 16 architecture for ImageNet

modify the last fully connected layer that outputs 1000 weights to 256.

For the purpose of using the adapted ViT b 16 architecture as a 2.5D classifier, similar to the approach introduced at *ResNet18 + Att* and *ResNet152 + Att*, we can forward a CT instance where the batch size is equivalent to the number of slices each CT posses. Using the features extracted per slice, we benefit from applying an additional Multi-Head Attention [46] and perform attention pooling among the slices and lastly applying a fully connected layer to the attention output to produce a single weight for performing a binary classification.

As the architecture for *ViT b 16 + Att* is shown in Figure 5.20, we define the input *query* as the mean of the features retrieved from classification tokens with respect to the slices per CT. As we define the output of the fully connected layer to be 256 for the Vision Transformer, being the feature extractor of our 2.5D classifier network, each layer of a CT is represented by a vector of size 256. Thus, once again, the mean of the features in terms of a CT is equivalent to a feature representation of [1, 256]. Furthermore, as introduced in the earlier architectures mentioned, for the *key* and *value* inputs of the Multi-Head Attention, the feature representations from the whole CT volume extracted by passing all of the layers as a batch are used.

It is important to note that, using this architecture, Multi-Head Attention is not only employed in the temporal domain but as it is also applied spatially as the Vision Transformer performs a self-attention. In previous networks introduced, we were also including attention mechanisms spatially, however in this network the spatial attention is also computed using Multi-Head Attention instead of using other techniques for applying a dynamic weight adjustment benefiting from a mask-based feature representation.

With the changes in the ViT b 16 as the encoder part of this network, the adapted part holds 85,995,526 parameters. Furthermore, the Multi-Head Attention part consists of 263,168 and the classifier linear layer has 257 parameters. Thus creating a model total of 86,258,951 parameters.

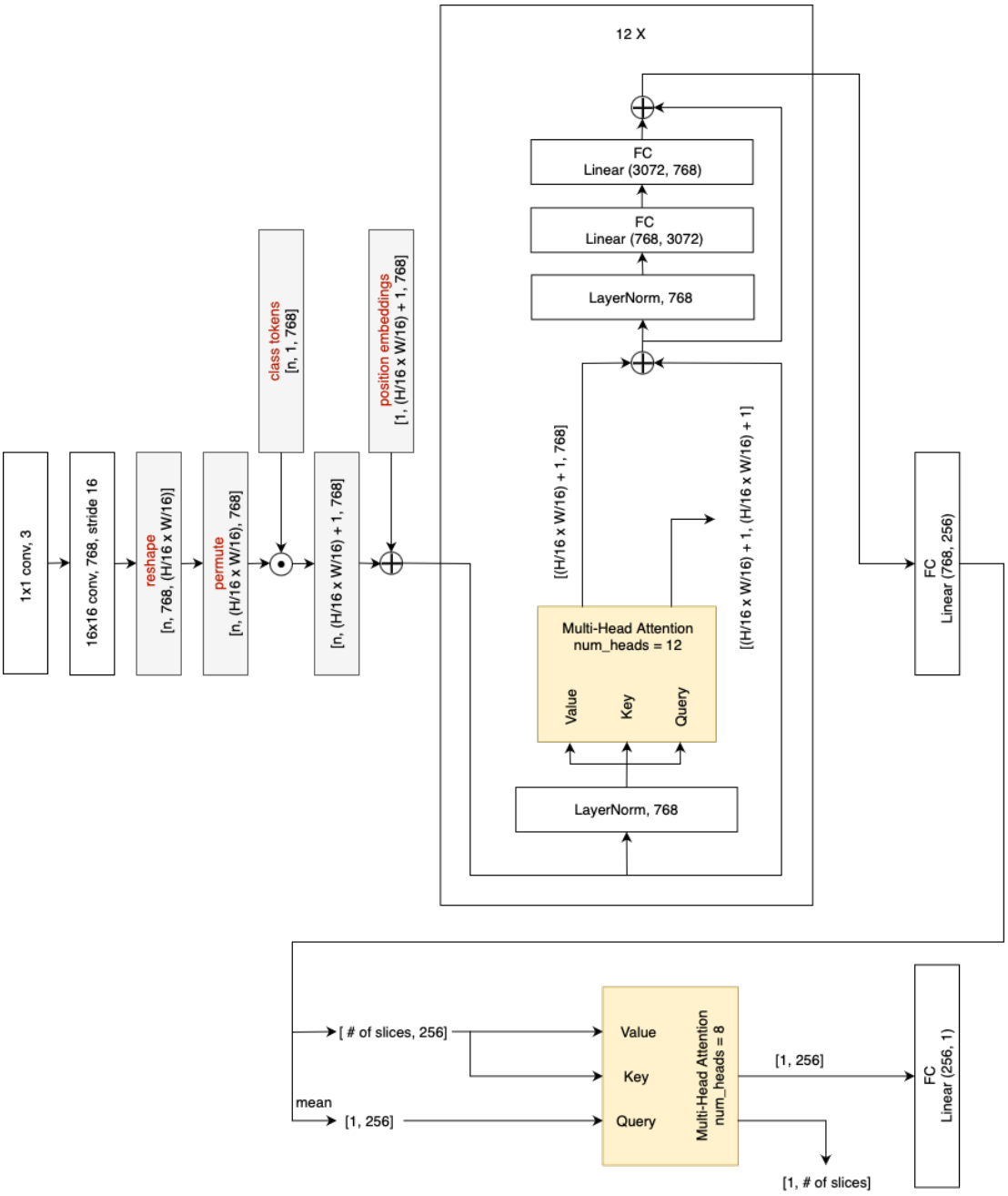


Figure 5.20.: *ViT b 16 + Att* architecture

5.2. 3D Networks

In this section, we investigate networks that are designed to be used for 3D inputs. As these networks are already capable to handle 3D inputs, we perform relevant modifications to adapt the network for handling input images with one channel and also modify the output of the network to be used for a binary classification task.

5.2.1. 3D ResNet

As convolutions have proven their success in image tasks, in a study conducted, authors investigated several forms of spatiotemporal convolutions for video tasks [45]. Authors demonstrate that using 3D CNNs instead of 2D CNNs for video tasks is much more advantageous, as the 3D convolutions operate within both spatial and temporal domains. Furthermore, in this work, authors also benefit from the residual learning as it was demonstrated to be successful in 2D CNNs.

layer name	output size	R3D-18	R3D-34
conv1	L×56×56	3×7×7, 64, stride 1 × 2× 2	
conv2_x	L×56×56	$\begin{bmatrix} 3\times3\times3, 64 \\ 3\times3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3\times3, 64 \\ 3\times3\times3, 64 \end{bmatrix}\times3$
conv3_x	$\frac{L}{2}\times28\times28$	$\begin{bmatrix} 3\times3\times3, 128 \\ 3\times3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3\times3, 128 \\ 3\times3\times3, 128 \end{bmatrix}\times4$
conv4_x	$\frac{L}{4}\times14\times14$	$\begin{bmatrix} 3\times3\times3, 256 \\ 3\times3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3\times3, 256 \\ 3\times3\times3, 256 \end{bmatrix}\times6$
conv5_x	$\frac{L}{8}\times7\times7$	$\begin{bmatrix} 3\times3\times3, 512 \\ 3\times3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3\times3, 512 \\ 3\times3\times3, 512 \end{bmatrix}\times3$
	1×1×1	spatiotemporal pooling, fc layer with softmax	

Figure 5.21.: R3D architectures. Downsampling performed at $conv1$, $conv3_1$, $conv4_1$ and $conv5_1$ [45]

In the article, as shown in Figure 5.21, for 3D ResNet authors introduce two variations namely R3D-18 and R3D-34 which have 18 and 34 layers respectively. Furthermore, the modules in the R3D architectures are denoted as in Figure 5.22. As argued for 3D convolutions, 3D CNNs preserve temporal information as it propagates it through the

layers [45]. In our networks, we consider using only the R3D-18 with 18 layers. Likewise, to 2D networks of ResNet, 3D ResNet is also defined with four main layers. Based on the definition each layer has two basic blocks, thus making 16 basic blocks from the first to last the main layer. Along with the convolution operations, the network also includes batch normalization and rectified linear unit operations. With this configuration, the R3D-18 contains 33,371,472 parameters.

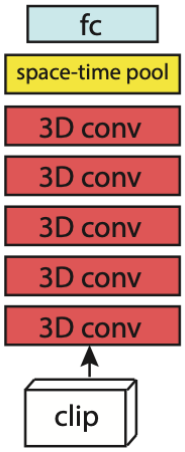


Figure 5.22.: R3D architecture modules [45]

3D ResNet (R3D)

This network is adapted from the R3D-18 network mentioned above, and since we only utilize the version with 18 layers the network is named *3D ResNet (R3D)*. As provided in Figure 5.23, the R3D-18 is designed for Kinetics 400 action recognition dataset. As the frames in the Kinetics dataset consist of three channels, the first convolution operation receives three input channels and produces 64. However, for our input modality, since CT images are in grayscale, they consist of a single channel. Furthermore, the version of the Kinetics dataset investigated by the authors consists of 400 classes whereas we are interested in a binary classification task.

In order to make this network applicable for grayscale 3D inputs and a binary classification task, first we prepend an additional convolution operation with kernel size $1 \times 1 \times 1$ which has input channel size one and output channel size three. By using an additional convolution operation, we manage to keep the option of using pre-training weights on the Kinetics dataset. Furthermore, we modify the fully connected layer at the end which holds an output weight size of 400. Since we do not want the classification output dimension to be preserved as 400 and class outputs from

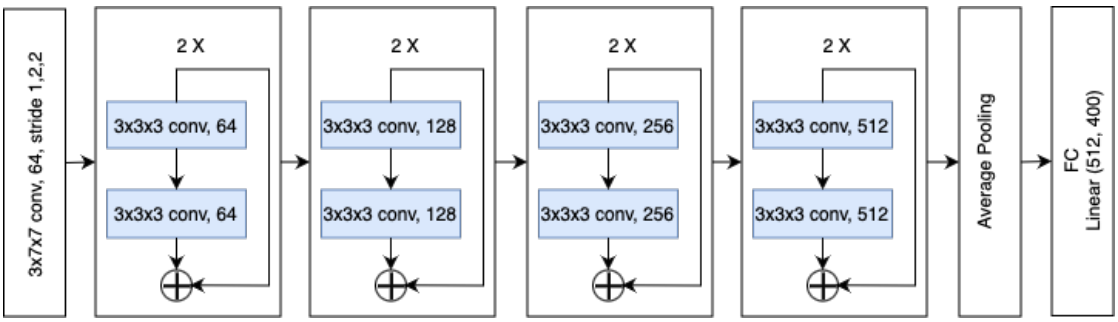


Figure 5.23.: R3D-18 architecture for Kinetics

pre-trainings are not relevant to our tasks, we modify the fully connected layer to a vector size of 256. Lastly, we add a fully connected layer that takes an input size of 256 and outputs a single weight, thus being able to be used for a binary classification task.

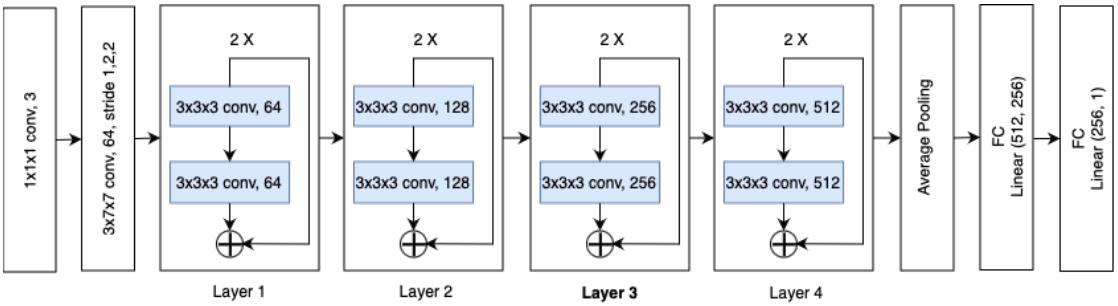


Figure 5.24.: *3D ResNet (R3D)* architecture

With the changes in the R3D-18 and adding an additional classifier layer, *3D ResNet (R3D)* holds a total of 33,297,863 parameters.

Non-Local Neural Networks

Convolutions and recurrent operations process one local neighborhood at a time. Therefore, in order to capture long-range dependencies these operations are repeated. However, these repetitions are computationally inefficient and cause optimization difficulties. Therefore, authors propose the *non-local* operations for capturing long-range dependencies [48]. Using the non-local operation, the authors show that response at a position can be calculated as a weighted sum of the features at all positions.

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{5.2}$$

$$z_i = W_z y_i + x_i \tag{5.3}$$

The non-local mean operation is defined as in Equation 5.2 where i is the index in output position and j is the index that enumerates all possible positions where x is the input and y being the output signal [48]. Here, the pairwise function f is for computing a representation relationship between i and j. The function g is for computing a representation of the input signal at position j. Furthermore, using the non-local operation in Equation 5.2, a non-local block can be defined as in Equation 5.3.

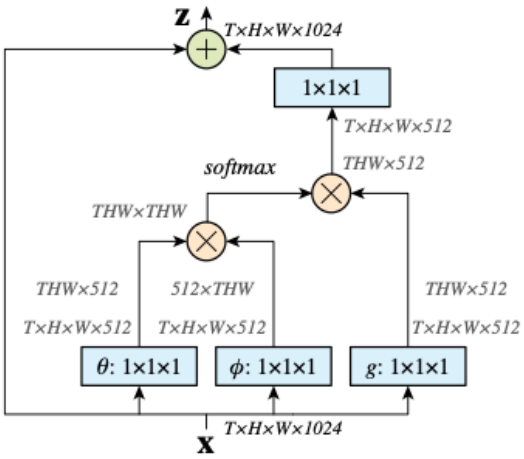


Figure 5.25.: A spacetime Non-Local block [48]

As functions f and g are positioned in a generic way, authors defined various versions where the function f can relate to different positions using a Gaussian function, embedded Gaussian, dot-product similarity and concatenation. The function f for embedded Gaussian is defined as in Equation 5.4 and the normalization factor $C(x)$ is defined as in Equation 5.5.

$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)} \tag{5.4}$$

$$C(x) = \sum_{\forall j} f(x_i, x_j) \tag{5.5}$$

Authors point out that, the embedded Gaussian produces the form in the self-attention module introduced in Multi-Head Attention [46] as shown in Equation 5.6. Thus, the non-local block relates to the Multi-Head Attention and operates in spacetime, making the block applicable to be used with 3D networks.

$$y = softmax(x^T W_\theta^T W_\phi x) \cdot g(x) \tag{5.6}$$

The formulation expressed above is demonstrated in Figure 5.25, which consists of four different 3D convolutions where the input to the block is related to the input itself within the space and time domains.

3D ResNet (R3D) + NL

This network is an extension of *3D ResNet (R3D)* where we utilize the Non-Local Neural Network (NL) block. As we did for GE, SE, and CBAM blocks in the 2.5D networks, for this network we also decide to implement our changes in the third main layer of the 3D ResNet network. Therefore, the NL block is added at the end of the last block in the third layer of *3D ResNet (R3D)*. For the aforementioned attention modules that were added to 2.5D networks, we included them at the end of every basic or bottleneck block. However, as the NL block contains more parameters, we decide to include it only after the last 3D basic block located in the third main layer.

As depicted in Figure 5.26, the NL block takes as an input of the feature representation created by the last convolution of the last basic block located in the third main layer of *3D ResNet (R3D)*. Three different convolution operations are defined namely, θ, ϕ and g. The convolution operations have a kernel size of one and produce half of the inputting channels, thus decreasing the number of channels from 256 to 128. Furthermore, the output of the convolution operations is reshaped such that a spacetime feature map is obtained. For the products of θ and g, the tensor representation is permuted to be in the form of [$T \times H \times W$, 128] and ϕ in the form of [128, $T \times H \times W$], where H and W represents the spatial domain and T stands for the temporal domain. The rearranged tensors θ and ϕ are multiplied, creating a new tensor [$T \times H \times W$, $T \times H \times W$]. Applying softmax to this by-product where each row of the matrix is summed to one, creating a self-attention form. Furthermore, multiplying the output of the Softmax and g, we obtain a feature map once again in the form of [$T \times H \times W$, 128], which is reshaped to the form of [128, T, H, W]. This reshaped form is then used as an input to a final convolution operation where the number of channels is increased from 128 to 256. Lastly, the output of the last convolution of the last basic block is added to the output of the final convolution, thus forming the NL block output.

With the changes in the *3D ResNet (R3D)* accumulating for the NL block consists of additional 132,224 parameters, thus *3D ResNet (R3D) + NL* holds a total of 33,429,830 parameters.

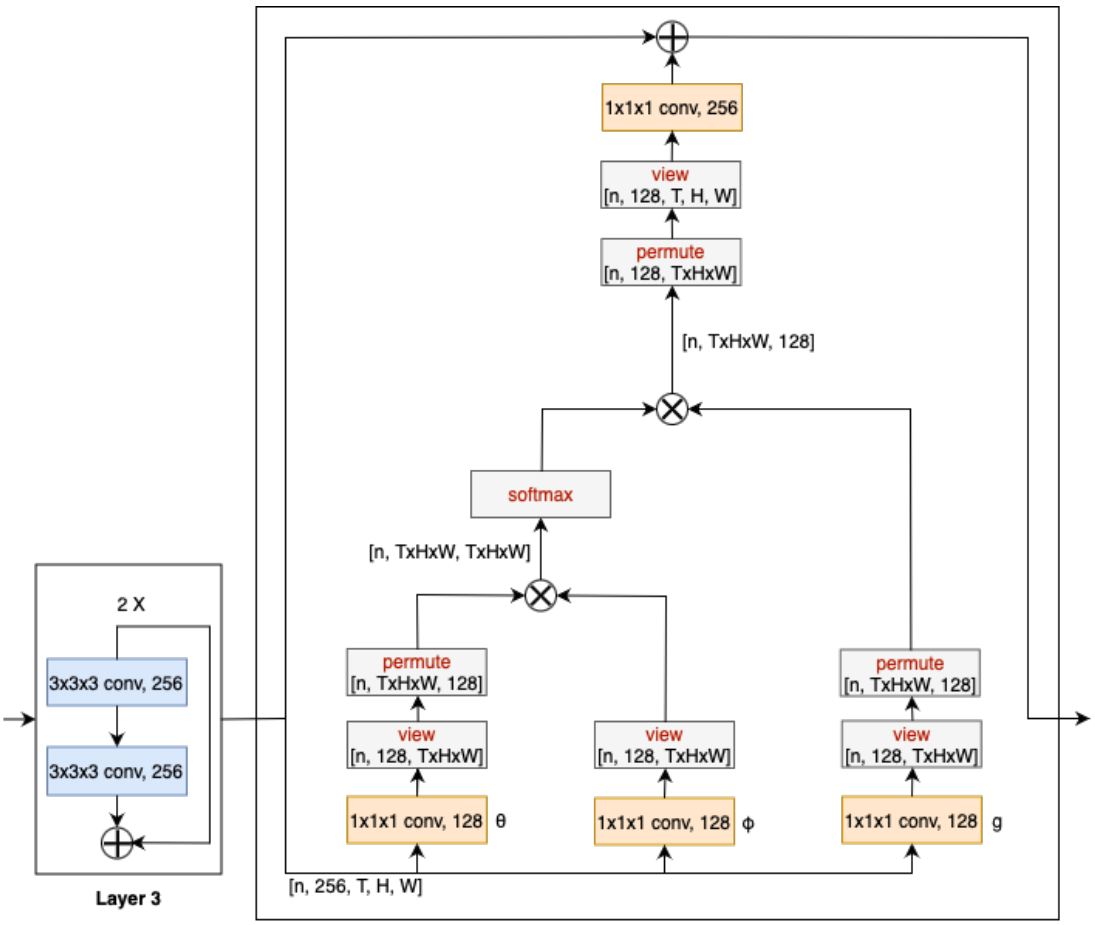


Figure 5.26.: *3D ResNet (R3D) + NL* architecture

5.2.2. Video Swin Transformer

As the Transformer based architectures started to show their success in image tasks, they are also applied to video tasks that perform attention across spatial and temporal dimensions. A further study was conducted, known as Video Swin Transformer, using an adapted version of the Shifted Window Transformer design that was introduced for image tasks which applies attention in the spatial domain only [27]. Thus authors adapted this design to apply attention to spatial and also temporal domains. The architecture surpasses other transformer-based video recognition architectures by taking advantage of the spatiotemporal locality of videos, where claiming the idea that pixels closer to each other in the spatiotemporal distance are more likely to be correlated than the ones that are distant [28].

The 3D input holds dimensions of $T \times H \times W \times 3$. There are T frames/slices with a spatial representation of $H \times W$ and a channel size of 3. In this architecture, authors define a 3D patch size contrary to the patch definition in Vision Transformer for 2D inputs. For this network, authors define the patch size as $2 \times 4 \times 4$ for temporal, height and width respectively.

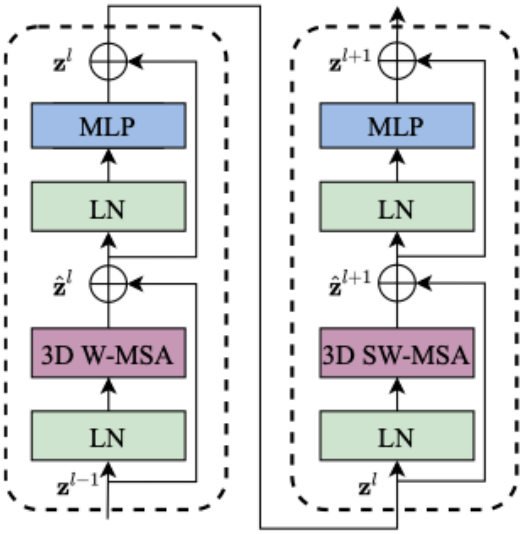


Figure 5.27.: Two successive Video Swin Transformer blocks [28]

There are multiple variations introduced by the authors such as tiny, small, base and large. For the base version, the Video Swin Transformer consists of four main stages. Within the stages, there are Video Swin Transformer blocks with counts of 2, 2, 18, 2 from the first to the last stage. An illustration of the Video Swin Transformer is shown

in Figure 5.27. The major component change is the 3D W-MSA where the Multi-Head self-attention module in the standard Transformer architecture is adapted to a form with a 3D shifted window-based attention module.

Given that a video composed of tokens $T^{'} \times H^{'} \times W^{'}$ and a window size defined as $P \times M \times M$, the input tokens would be partitioned into non-overlapping windows of $\lceil \frac{T^{'}}{P} \rceil \times \lceil \frac{H^{'}}{M} \rceil \times \lceil \frac{W^{'}}{M} \rceil$ as shown in Figure 5.28.

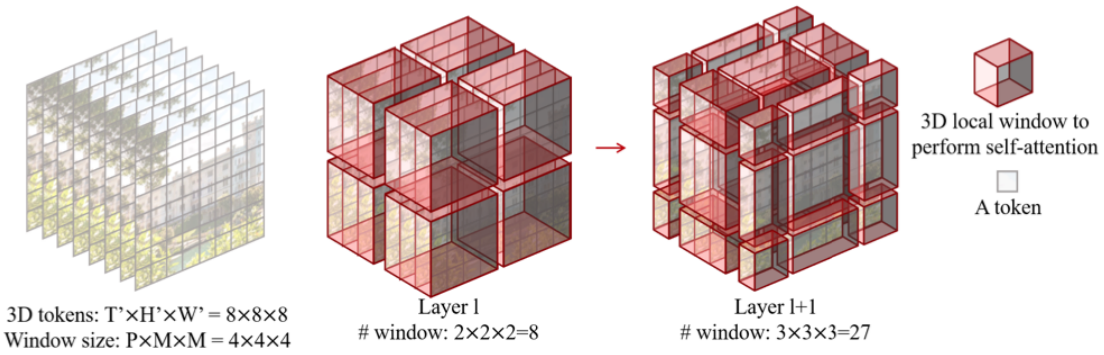


Figure 5.28.: 3D shifted windows [28]

3D Swin Transformer

This network is an extension of the Video Swin Transformer where we select the base variation to be used, referred to as *3D Swin Transformer* in this thesis. The base version of Video Swin Transformer uses patch size as $2 \times 4 \times 4$ and window size $8 \times 7 \times 7$ for temporal, height and width respectively. The architecture consists of four main stages with block counts of 2, 2, 18, 2. At the end of first, second and third stages there is a so-called patch merging layer for projecting the features to half of their dimensions.

The embedding dimension is set to 128. For each stage, the dimension is calculated as the multiplication of the embedding dimension by two to the power of the stage index starting from zero. Thus dimensions are 128, 256, 512, 1024 from the first to the last stage respectively. An additional parameter known as stochastic depth probability which is calculated by dividing the block index, starting from zero, to one subtracted from total number of blocks which is 24. Thus, we are able to calculate stochastic depth probability by multiplying the output of the division operation with a constant factor of 0.1. Thus, beginning from 0, the latest block has a stochastic depth probability equal to 0.1. At the end of the last stage, there is a fully connected layer inputting 1024 weights and outputting 400 weights which is equal to the number of classes in the action recognition dataset Kinetics.

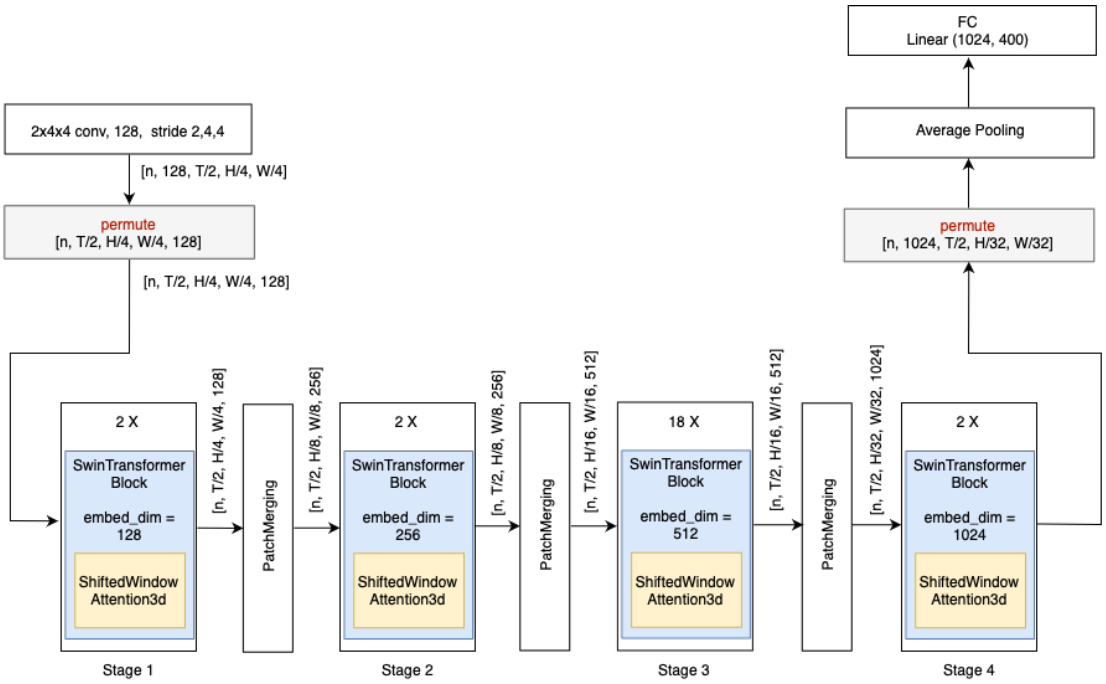


Figure 5.29.: Video Swin Transformer architecture for Kinetics

The main structure of the Video Swin Transformer is shown in Figure 5.29. In order to adapt the Video Swin Transformer for our task on multi-layer CT images that are grayscale, we address the input issue by prepending an additional convolution operation of $1 \times 1 \times 1$ which increases the input channels from one to three. Thus, using an additional convolution operation we manage to keep the initial form to be applicable for using pre-trained weights on the Kinetics dataset. Moreover, we modify the last fully connected layer of the Video Swin Transformer to output 256 weights instead of 400. Lastly, we append an additional fully connected layer that outputs a single weight, so that the architecture would be applicable to our binary classification task. With the mentioned changes we obtain the *3D Swin Transformer* architecture as illustrated in Figure 5.30.

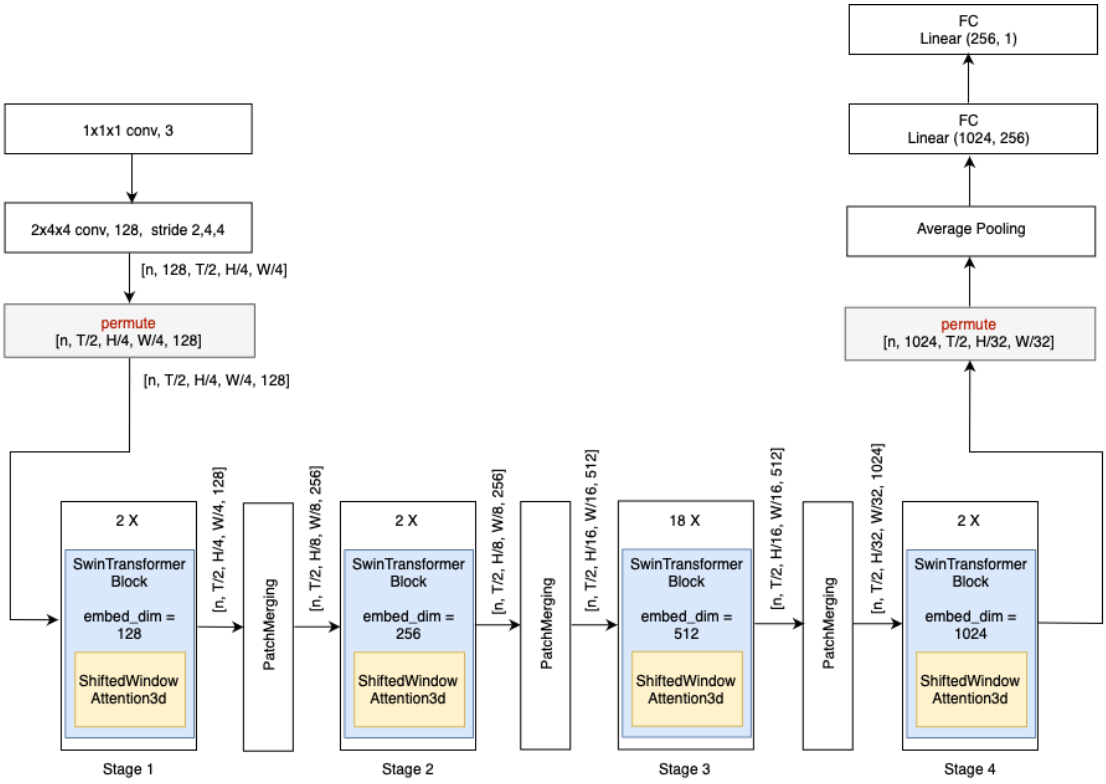


Figure 5.30.: *3D Swin Transformer* architecture

6. Experiments

6.1. Pre-trainings

Transfer learning is defined as transferring information from a related domain to improve a learner from another domain [50]. For the networks that we aim to employ as 2.5D and 3D architectures, pre-trained network weights are available for ImageNet [10] and Kinetics 400 [21] in the PyTorch library [34]. Although using ImageNet and Kinetics pre-trainings show promising results in terms of networks converging faster and producing higher metrics such as accuracy, Matthews Correlation Coefficient (MCC), and Area Under The Receiver Operating Characteristic (ROC) Curve (AUC), some argue that transfer learning from a network trained on a medical dataset is more beneficiary for a medical task [31]. Therefore, in this thesis, we experiment with our architectures in other medical datasets or tasks and use them as pre-trainings to compare the effect of transfer learning from a medical domain either showing an improvement or not.

6.1.1. Sex Prediction

As a proof of concept, we experiment with sex prediction by classifying the computer tomography (CT) images using sex labels for supervision. For this experiment, we use our in-house pancreatic ductal adenocarcinoma (PDAC) dataset. With this experiment, we demonstrate the performance of our networks on a relatively simpler task and conduct a sanity check with our in-house dataset.

Among the 878 patients, 865 have their sex labels assigned either as male or female. Furthermore, with class stratification, we split the set into train, validation and test with ratios of 80%, 10% and 10% respectively. With splitting the data, in the train set, we have 330 female 362 male patients, in validation we have 41 female 45 male patients and in test we have 41 female 46 male patients. For all of the experiments conducted, we select the model with the highest validation MCC score and run the test set on that model for final evaluation. For calculating the metrics and forming the confusion matrix, the threshold is set to be 0.5 default value as the task is a binary classification.

As our input types have a differentiating number of slices, for 2.5D networks, the batch size for 2D encoder networks is equal to the number of layers of the inputting

CT, whereas for 3D networks the batch size is set to 1. For ResNet18 and its attention module variations, with regards to the model parameters and GPU memory limits, the input shape of the CTs are selected as $200_{\leq} \times 256 \times 256$ being the depth, height and width of the input CT with respectively. It is important to note that $200_{\leq}$ represents the notation such that since the number of slices for each CT is not equal, we set a maximum threshold of 200 slices, therefore for CTs with more slices than the threshold, the slices are selected with uniform intervals from the total data representation and fitted to 200. For ResNet152 with attention pooling the input shape is $90_{\leq} \times 256 \times 256$, whereas for attention module variations of ResNet152 the input shape is $80_{\leq} \times 224 \times 224$. Lastly, for the Vision Transformer with attention pooling, the input shape is $100_{\leq} \times 224 \times 224$. As some of the networks consume more memory due to a high number of parameters and operations, we need to use a smaller threshold for some networks.

For sex prediction, accuracy is defined as the total number of female and male instances being correctly predicted divided by the whole evaluation set such as $\frac{TP+TN}{TP+TN+FP+FN}$. Precision is defined as among the instances that are classified as male, how many of them are correct such as $\frac{TP}{TP+FP}$. Recall is defined as among the male instances, how many of them are correct such as $\frac{TP}{TP+FN}$. F1 is defined as the multiplication of precision with recall by 2 and divided by the summation of precision and recall. Lastly, MCC is defined as the $\frac{TP*TN-FN*FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$. MCC takes a value between -1 to 1, where 1 indicates perfect prediction and -1 indicates total disagreement with prediction and class labels.

For this experiment, female instances are defined to be negative and male are defined to be positive instances. Thus, female instances are labeled as 0 and males are labeled as 1.

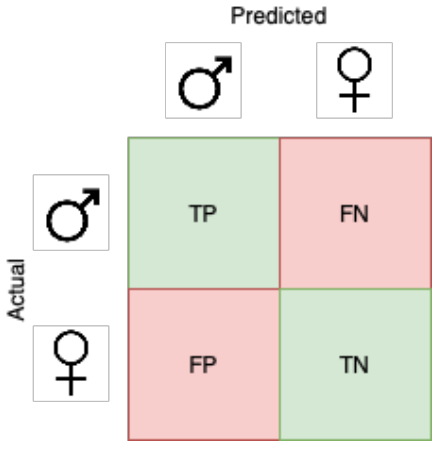


Figure 6.1.: Sex Prediction Confusion Matrix

It is also important to note that, before starting to the final experiments, in order to justify the claim that we mentioned in the methodology section on where to include the forms of attention mechanisms in the ResNet architectures, we experimented with the Gather-Excite blocks on ResNet18 and observed that for sex prediction the best metrics where attained at the third main layer. Thus with this intuition, we execute attention mechanisms that can be individually introduced to ResNets to be added to every block located in the third main layer for 2D architectures and likewise for 3D architectures.

	Accuracy	Precision	MCC	F1	AUC
ResNet18 + Att	0.920	**1.000**	0.851	0.918	0.960
ResNet18 + GE + Att	**0.943**	0.956	0.885	**0.945**	0.978
ResNet18 + SE + Att	0.920	**1.000**	0.851	0.918	**0.986**
ResNet18 + CBAM + Att	0.920	0.953	0.841	0.921	0.951
ResNet152 + Att	0.874	0.872	0.746	0.882	0.964
ResNet152 + GE + Att	0.874	0.889	0.747	0.879	0.947
ResNet152 + SE + Att	0.920	0.933	0.839	0.923	0.971
ResNet152 + CBAM + Att	0.862	0.886	0.725	0.867	0.943
ViT b 16 + Att	**0.943**	0.977	**0.887**	0.944	0.980

Table 6.1.: Sex Prediction using 2.5D architectures pre-trained on ImageNet. Attention pooling after a network that is used as an encoder is represented as Att. GE denotes Gather-Excite operator added into the blocks of ResNet layer 3 [15]. SE denotes Squeeze-and-Excite networks added into the blocks of ResNet layer 3 [16]. CBAM denotes Convolutional Block Attention Module added into the blocks of ResNet layer 3 [51].

As mentioned before, we are interested in the model parameters where the highest validation MCC value was achieved. Likewise, for selecting the best architecture, we are also interested in the model configuration that generated the highest MCC value for the test set. Furthermore, we also calculate the accuracy, precision, F1 and AUC values for the relevant test evaluations.

When we compare the metrics in Table 6.1, we are able to observe that in two out of five metrics that we consider, *ViT b 16 + Att* outperforms the other approaches for sex prediction. It is important to note that all of the encoder networks used pre-trainings on ImageNet that were presented by the PyTorch library. Furthermore, we can observe that for all of the ResNet152 configurations with attention mechanisms, its respective configuration with ResNet18 outperforms, despite the fact that they were not performing above *ViT b 16 + Att* when accuracy and MCC metrics were considered.

Furthermore, in Figure 6.2 we present the temporal attention output weights for

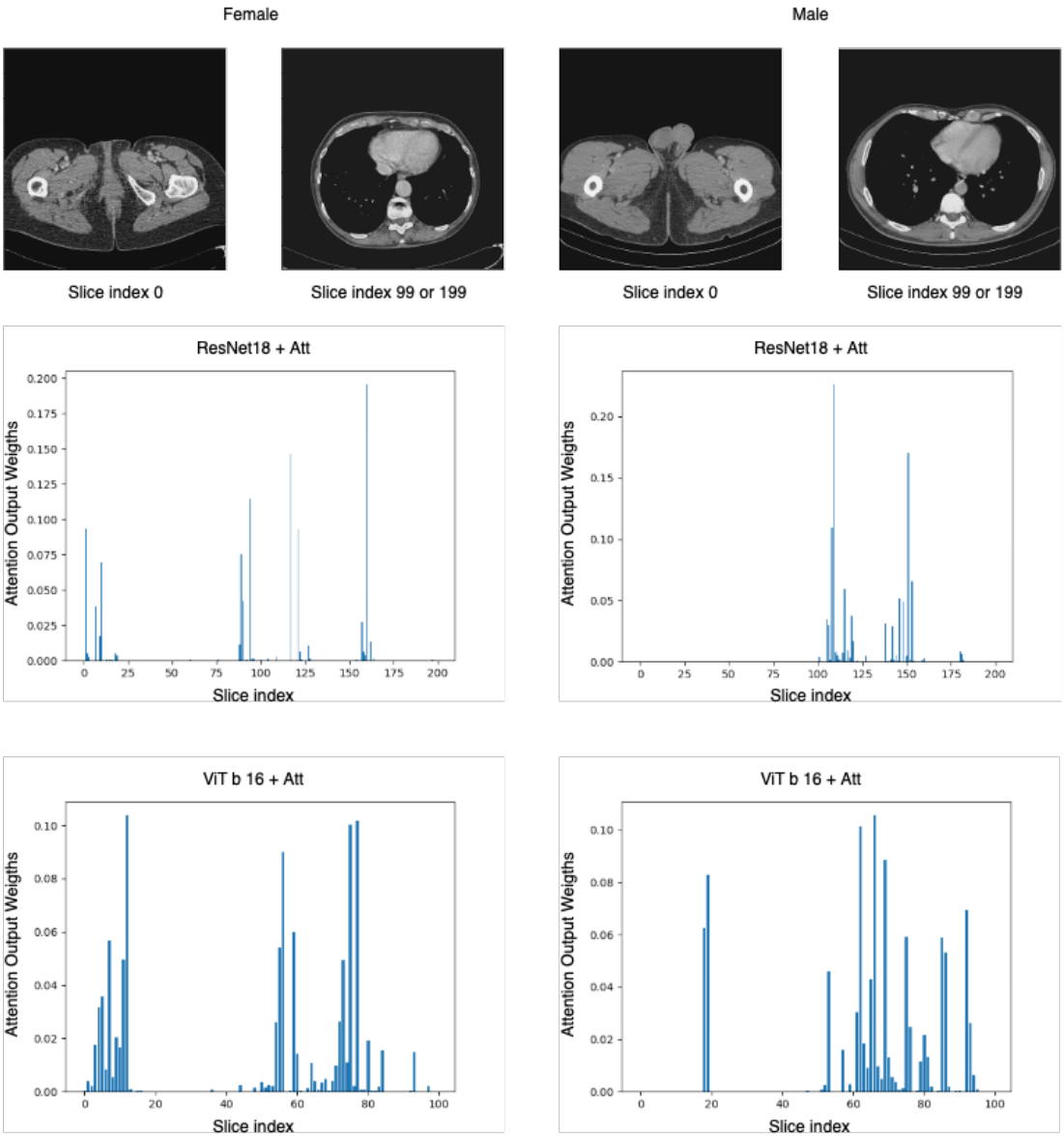


Figure 6.2.: Sex Prediction: Attention output weights for slice attention comparison using *ResNet18 + Att* and *ViT b 16 + Att* for same female and male patient

slices when using our 2.5D networks *ResNet18 + Att* and *ViT b 16 + Att*. For the same female and male patients that were classified correctly by both of the networks, at the top we have presented the first and the last CT slices for the patients. The CT slices are in an incremental index from pelvic to thoracic body parts. As mentioned before, for *ResNet 18 + Att* the input shape is $200_{\leq} \times 256 \times 256$, whereas for *ViT b 16 + Att* the input shape is $100_{\leq} \times 224 \times 224$ corresponding to depth, height and width respectively. Using the attention output weights produced in the slice attention part of the architectures, we are able to see that for both female and male samples, *ViT b 16 + Att* network produced higher values towards chest and abdominopelvic areas. Whereas, although *ResNet18 + Att* was producing similar results for the female sample, it was only producing weights towards chest parts for the male patient.

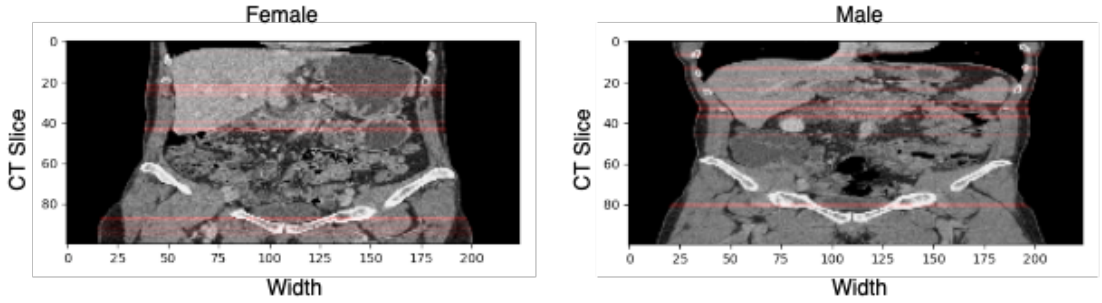


Figure 6.3.: Sex Prediction: Attention output weights for slice attention using *ViT b 16 + Att* for male and female patient visualized on CT scan

In Figure 6.3 we visualize the attention output weights for the CT slices when using *ViT b 16 + Att* network for the same male and female patients provided in Figure 6.2. It is important to note that, the frontal view of the CTs seems to be compressed in the y direction. The reason for this is that since *ViT b 16 + Att* needs a greater memory size to operate, the maximum number of slices that can be processed for each instance was 100, whereas for these patients selected, the CT scan for female had 494 and male had 323 slices. Despite the threshold applied to the CTs, here we can observe and emphasize that the attention pooling applied to slices can focus on the relevant slices according to the task. Thus the attention-pooling technique was able to capture dependencies globally even under restrictions. Here we are able to observe which slices of the CT scans had a higher attention output weight being colored by red. For both of the patients, we are able to see that the attention mechanism focused on the chest and pelvic areas in the body.

Similar to our 2.5D networks, for our 3D networks we also need to change the spatial and temporal dimensions of the inputs according to the network that we are training.

For 3D ResNet, the input shape is $200_{\leq} \times 256 \times 256$, whereas for Non-Local block attention version of 3D ResNet the input shape is $100_{\leq} \times 256 \times 256$. Furthermore, for the Video Swin Transformer, the input shape is $100_{\leq} \times 224 \times 224$.

	Accuracy	Precision	MCC	F1	AUC
3D ResNet (R3D)	0.931	0.917	0.862	0.936	0.977
3D ResNet (R3D) + NL	0.851	0.867	0.701	0.857	0.906
3D Swin Transformer	**0.977**	**0.978**	**0.954**	**0.978**	**0.985**

Table 6.2.: Sex Prediction using 3D architectures pre-trained on Kinetics 400. NL represents a Non-Local block in the end of ResNet layer 3

When we compare the performance of 3D networks for sex prediction, *3D Swin Transformer* outperforms the other approaches in all of the metrics. Moreover, it is also recognized that *3D Swin Transformer* also outperforms the *ViT b 16 + Att* in all of the metrics. As the nature of the task is a comprehensive 3D task, using self-attention in spatial and temporal domains integrated into the same network is more successful rather than applying them separately as spatial attention in the encoder part and temporal attention in the pooling part of *ViT b 16 + Att*.

The experiments for sex prediction are trained for 50 epochs. We employ the Adam optimizer using Multi-Step Learning Rate decay starting from 0.001 and decaying by a gamma of 0.01 with milestones 3 and 4. For the loss function, we benefit from Binary Cross Entropy with Logits Loss. As we use ImageNet and Kinetics 400 pre-trained weights, we gradually included first the latter and newly added parts of the networks in the training and then the former regions. Furthermore, since we are using pre-trained weights, we trained by freezing the batch normalization layers.

6.1.2. Radiological ImageNet

The Radiological ImageNet (RadImageNet) consists of 2D CT images per instance, therefore for this dataset, architectures modeling 3D data structures were not suitable. Therefore, we experiment with networks that we use as the encoders for our 2.5D approaches, which are 2D networks.

Among the 139,825 instances from 28 different classes, with class stratification, we split the set into train, validation and test with ratios of 80%, 10% and 10% respectively. With splitting the data, in the train set we have 111,859 instances, in validation we have 13,982 instances and in test we have 13,983 instances. Here different from sex prediction, since the input images are in the data form of 2D, we keep our batch size as 32 for all trainings in the RadImageNet section.

In Table 6.3 we find the results for the trainings where the 2D networks were trained from scratch, in other words, the network was randomly initialized without using any pre-trained weights.

	Accuracy	Precision	MCC	F1	AUC
ResNet18	**0.763**	**0.738**	**0.671**	**0.739**	**0.952**
ResNet152	0.725	0.685	0.614	0.688	0.926
ViT b 16	0.498	0.257	0.057	0.335	0.630

Table 6.3.: Radiological ImageNet Classification using 2D architectures trained from scratch

Although the *ViT b 16* performed relatively poor in all metrics compared to *ResNet18* and *ResNet152* when the model was trained from scratch, the model was displaying an increasing accuracy and MCC curves and a decreasing loss curve. However, since we trained all models up to 25 epochs for RadImageNet, the training and validation loss metrics are implying that the Vision Transformer would perform much better results if it was trained for longer. Furthermore, it is also known that Transformer architectures tend to produce better results compared to convolutional neural networks when there is excess data [29]. However, with the data size and train duration of 25 epochs, we are able to see that *ResNet18* outperformed the other two approaches when the model was trained from scratch.

In Table 6.4 we find the results for the trainings where the 2D networks were trained using ImageNet pre-trained weights. Here we can observe that all three networks that are trained for the radiological image classification task benefited from transfer learning from ImageNet weights.

	Accuracy	Precision	MCC	F1	AUC
ResNet18	**0.800**	**0.784**	**0.724**	**0.785**	**0.962**
ResNet152	0.777	0.763	0.690	0.761	0.950
ViT b 16	0.734	0.713	0.629	0.717	0.941

Table 6.4.: Radiological ImageNet Classification using 2D architectures pre-trained on ImageNet

Similarly, the *ViT b 16* performed relatively poorly in all metrics compared to *ResNet18* and *ResNet152* when the model was trained using ImageNet weights. Although, the *ViT b 16* model was producing an increasing accuracy and MCC values and a decreasing loss value. However, since we trained all models up to 25 epochs, the metric are implying that the Vision Transformer would perform much better results if it was

trained for longer. However, it is important to note that, the performance of the Vision Transformer was much closer to the residual networks when ImageNet weights were used for network initialization compared to not using pre-trained weights. Furthermore, it is also known that Transformer architectures tend to produce better results in excess data size compared to convolutional neural networks. However, with the data size and train duration of 25 epochs, we are able to see that *ResNet18* outperformed the other two approaches when the model was trained using ImageNet weights as pre-training.

The experiments for RadImageNet were trained for 25 epochs as mentioned before. We employ the Adam optimizer using Multi-Step Learning Rate decay starting from 0.001 and decaying by a gamma of 0.01 with milestones 3 and 4. For loss function, we benefit from Cross Entropy. For trainings that we use ImageNet weights as network initialization, we gradually include first the latter parts of the networks in the trainings and then the former regions. Furthermore, for trainings that we are using pre-trained weights, we trained by freezing the batch normalization layers.

6.1.3. PDAC Tumor Classification

In comparison to tumor response prediction, the classification of instances with PDAC positive or negative is relatively a less challenging task. However, our motivation is that using weights of a model that is trained to classify pancreatic tumors would be more related to our aim and be more successful to extract features from the pancreas for therapy response prediction.

Among the 878 patient records in the PDAC dataset, when removing the patients that are listed with erroneous records 867 of them are left. Furthermore, in the Normal Pancreas Dataset, we have 758 patients who had a CT scan, but PDAC disease was not present. In the PDAC dataset, for this task, we have not taken into consideration whether or not the patients had their therapy response label present since these were patients known to have a PDAC disease but their therapy response labels can be empty due to any reason such as the patient not receiving therapy at all or any other medical issue.

For this task, we combine the patients in both datasets, creating a set of 1625 patients. Furthermore, with class stratification, we split the set into train, validation and test with ratios of 75%, 10% and 15% respectively. With splitting the data, in the train set we have 568 PDAC false 650 PDAC true patients, in validation we have 76 PDAC false 86 PDAC true patients and in test we have 114 PDAC false 131 PDAC true patients. For all of the experiments, we select the model with the highest validation MCC score and run the test set on that model for final evaluation. For calculating the metrics and forming the confusion matrix, the threshold is set to be 0.5 default value as the task is a binary classification. For notation purposes, we denote this new dataset as, PDAC 0/1.

For PDAC tumor classification, similar to sex prediction, the task is a binary classification task. Therefore, for the metrics that we use for this task, the similar description in Figure 6.1 holds, whereas the positive instances are PDAC true and the negative instances are PDAC false. Thus, PDAC true patients are assigned to have class label 1 and PDAC false patients are assigned to have class label 0.

As our input types have a differentiating number of slices, for 2.5D networks, the batch size for 2D encoder networks is equal to the number of layers of the inputting CT, whereas for 3D networks the batch size is set to be 1, similar to the approach conducted for sex prediction.

	Accuracy	Precision	MCC	F1	AUC
ResNet18 + Att	**0.931**	**0.938**	**0.861**	**0.935**	**0.982**
ResNet18 + GE + Att	0.922	0.931	0.844	0.927	0.965
ResNet18 + SE + Att	0.918	0.937	0.837	0.922	0.971
ResNet18 + CBAM + Att	0.906	0.909	0.811	0.913	0.972
ResNet152 + Att	0.890	0.900	0.779	0.897	0.961
ResNet152 + GE + Att	0.910	0.916	0.820	0.916	0.959
ResNet152 + SE + Att	0.886	0.912	0.772	0.891	0.928
ResNet152 + CBAM + Att	0.898	0.914	0.796	0.903	0.948
ViT b 16 + Att	0.894	0.889	0.787	0.902	0.967

Table 6.5.: PDAC Tumor Classification using 2.5D architectures pre-trained on ImageNet. Attention pooling after a network used as an encoder is represented as Att. GE denotes Gather-Excite operator added into the blocks of ResNet layer 3 [15]. SE denotes Squeeze-and-Excite networks added into the blocks of ResNet layer 3 [16]. CBAM denotes Convolutional Block Attention Module added into the blocks of ResNet layer 3 [51].

When we compare the metrics in Table 6.5, we are able to observe that *ResNet18 + Att* outperforms all other networks for this tasks in all of the five metrics defined. Furthermore, similar observation holds for this task as well, where all of the ResNet18 network configurations has outperformed all the respective ResNet152 configurations. Furthermore, in this task, *ViT b 16 + Att* performed better than *ResNet152 + Att* and *ResNet152 + SE + Att* in terms of MCC score, however worse than all of the other networks.

In Figure 6.4, once again we investigate the attention output weights for slices when using our 2.5D networks *ResNet18 + Att* and *ViT b 16 + Att*. For the same PDAC false and PDAC true patients that were classified correctly by both of the networks, in the topmost we have present the first and the last CT slices for both patients. Using the

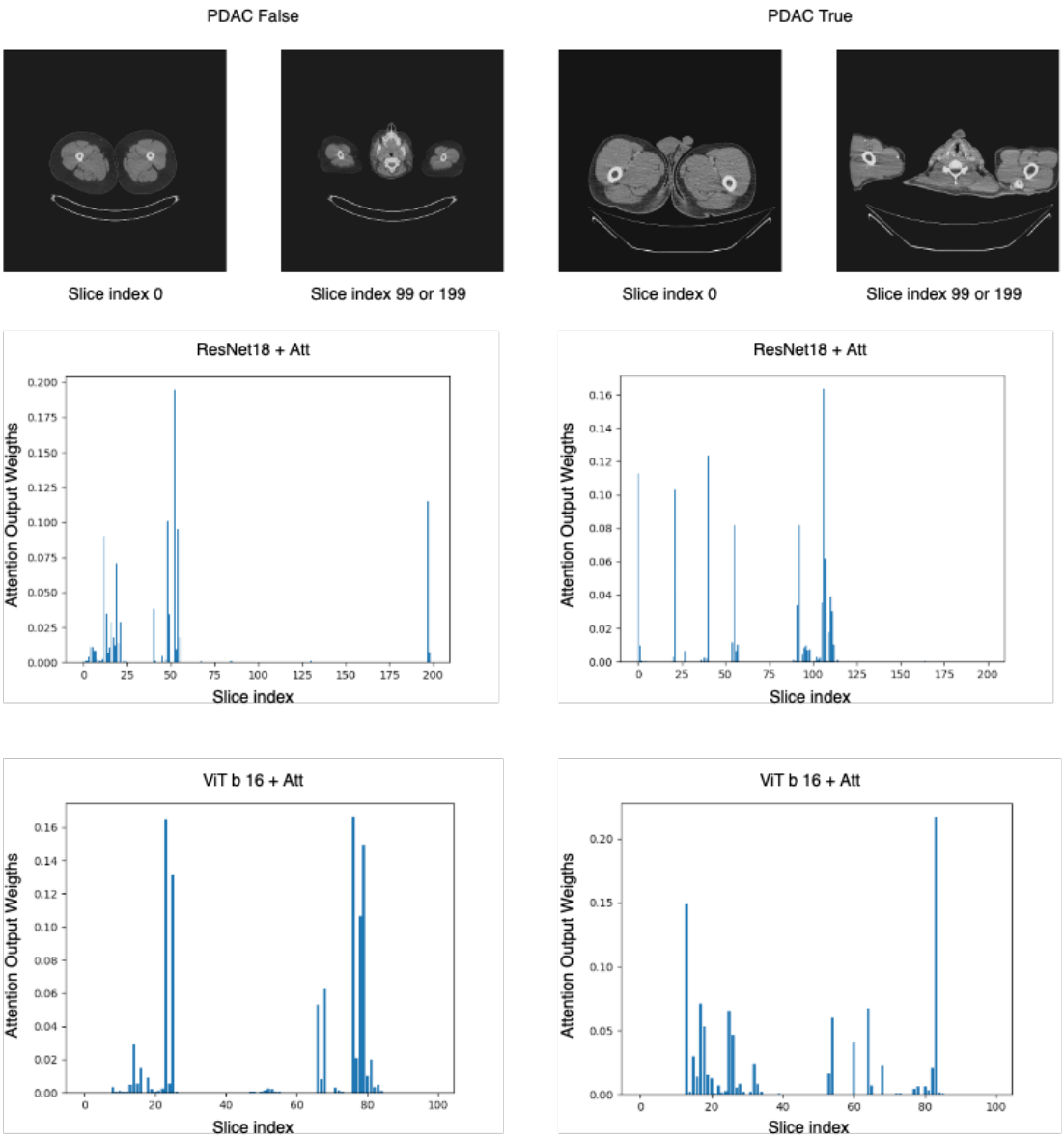


Figure 6.4.: PDAC Tumor Classification: Attention output weights for slice attention comparison using *ResNet18 + Att* and *ViT b 16 + Att* for same PDAC false and PDAC true patient

attention output weights produced in the slice attention part of both of the networks, we are able to see that for both of the samples from each class, *ResNet18 + Att* networks produced higher values abdominopelvic area. However, the *ViT b 16 + Att* produced higher values to both ends of the patient where the pancreas is not existing. Thus, the evaluation metrics provided in Table 6.5 are correlated with the findings of slice attention as *ResNet18 + Att* was resulting in better metrics. Thus, it can be argued that the interpretability of the attention mechanisms is correlated with the evaluation metrics.

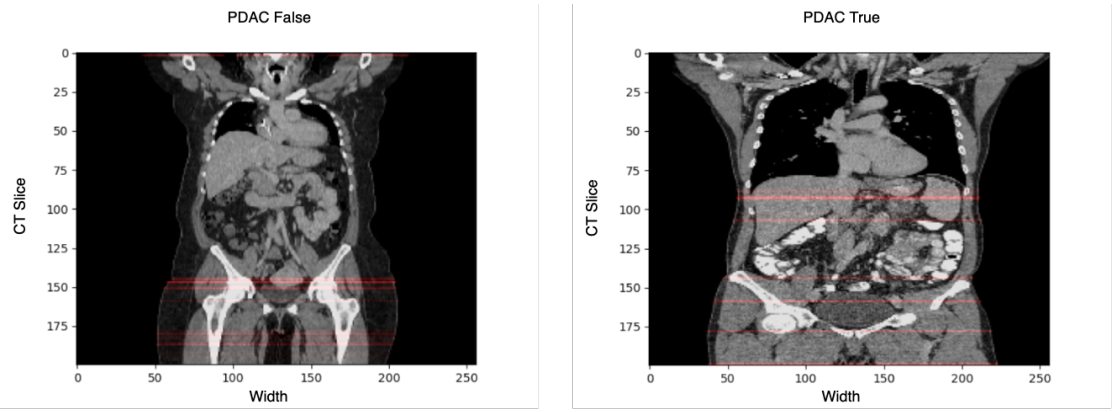


Figure 6.5.: PDAC Tumor Classification: Attention output weights for slice attention using *ResNet18 + Att* for PDAC false and PDAC true patient visualized on CT scan

Moreover, in Figure 6.5, we visualize the attention output weights for the CT slices when using *ResNet18 + Att* network for the same patients PDAC false and PDAC true provided in Figure 6.4. Using a frontal body CT, we are able to observe which slices of the CT scans had a higher attention output weight being colored by red. As the intensity of the color increases, it signifies that the impact of weight was greater. Here it is important to note that, for PDAC true sample, the attention weights for the slices are greater towards the mid regions of the body scan. Indeed, both the intensity of the attention weights located in the mid regions are greater and at the same time also the number of slices that have a significant attention weight are also clustered towards the mid regions of the body as can be seen in the plot located in the right-hand side in Figure 6.5.

Furthermore, we experiment with 3D networks for the PDAC tumor classification dataset, where *3D ResNet (R3D)* outperforms other networks by obtaining the highest MCC score as well as other metrics also. Here we observe that two other networks that utilize attention mechanisms are producing similar whereas *3D ResNet (R3D) + NL* was

	Accuracy	Precision	MCC	F1	AUC
3D ResNet (R3D)	**0.935**	**0.939**	**0.869**	**0.939**	**0.981**
3D ResNet (R3D) + NL	0.918	0.930	0.836	0.923	0.975
3D Swin Transformer	0.910	0.892	0.820	0.919	**0.981**

Table 6.6.: PDAC Tumor Classification using 3D architectures pre-trained on Kinetics 400. NL represents a Non-Local block in the end of ResNet layer 3

outperforming *3D Swin Transformer* in terms of MCC. Nevertheless, when we compare the *3D ResNet (R3D)* with the *ResNet18 + Att*, the 3D architecture was outperforming the 2.5D architecture considering the MCC score. Thus, once again as in sex prediction, we can claim that as the nature of the task 3D networks are more suitable.

The experiments for tumor classification were trained for 50 epochs. We employ the Adam optimizer using Multi-Step Learning Rate decay starting from 0.001 and decaying by a gamma of 0.01 with milestones 3 and 4. For the loss function, we benefit from Binary Cross Entropy with Logits Loss. As we use ImageNet and Kinetics 400 pre-trained weights, we gradually include first the latter and newly added parts of the networks in the trainings and then the former regions. Furthermore, since we are using pre-trained weights, we train by freezing the batch normalization layers.

6.2. PDAC Therapy Response Prediction

6.2.1. Comparing Pre-trainings

In this section, we compare the performance gain on PDAC therapy response prediction by the pre-trainings that we conducted on Sex Prediction, RadImageNet classification and PDAC tumor classification. For this purpose, we select two network architectures that we conduct our pre-trainings. For the 2.5D classification network, we select the *ResNet 18 + Att* and for the 3D classification network, we select the *3D ResNet (R3D)* as being the base architectures without any attention mechanisms employed other than the slice attention. In addition, these two networks were the best performing for the PDAC0/1 classification task, which is the most related pre-training that we conducted.

	Accuracy	Precision	MCC	F1	AUC
ResNet18 + Att					
No pre-training	0.652	0.000	0.000	0.000	**0.548**
ImageNet	**0.667**	0.571	0.159	**0.262**	0.439
ImageNet + Sex Prediction	**0.667**	**1.000**	**0.168**	0.082	0.473
ImageNet + RadImageNet	0.652	0.000	0.000	0.000	0.540
ImageNet + PDAC 0/1	**0.667**	**1.000**	**0.168**	0.082	0.481
3D ResNet (R3D)					
No pre-training	0.652	0.000	0.000	0.000	0.451
Kinetics	0.674	**1.000**	0.206	0.120	0.536
Kinetics + Sex Prediction	0.667	0.750	0.147	0.118	0.526
Kinetics + PDAC 0/1	**0.689**	0.778	**0.241**	**0.250**	**0.542**

Table 6.7.: PDAC Therapy Response Prediction using ResNet18 + Att architecture comparing no pre-training, pre-trained on ImageNet, pre-trained on ImageNet + Sex Prediction, pre-trained on ImageNet + RadImageNet and pre-trained on ImageNet + PDAC 0/1. PDAC Therapy Response Prediction using 3D ResNet (R3D) architecture comparing no pre-training, pre-trained on Kinetics 400, pre-trained on Kinetics 400 + Sex Prediction and pre-trained on Kinetics 400 + PDAC 0/1.

Among these 878 patients, only 538 of them have the therapy response label assigned. As mentioned before in this thesis we narrow down these labels into two classes for modeling therapy response prediction as a binary task. From the 538 patients, 186 of them had *Progressive Disease* label whereas 352 were within the set of positive responses also referred to as *Others*. So instances in the class of *Others* are negative and instances in the class of *Progressive Disease* are positive. Thus, instances in the class of *Others* are

assigned to have a label 0, and instances in the class of *Progressive Disease* are assigned to have a label 1.

As the data size is comparably less than the sex prediction, RadImageNet and PDAC Tumor classification, for this classification task, with class stratification, we split the set into train and validation with ratios of 75% and 25% respectively. Since the dataset is relatively small and labels for therapy response are sparse, extracting a subset from the validation set as a test set would cause both validation and test set to be more prone to errors and not be generalized. Furthermore, using a smaller train set would also cause the model to be less generalized. With splitting the data, in the train set, we have 264 *Others* and 139 *Progressive Disease* patients and in the validation set, we have 88 *Others* and 47 *Progressive Disease* patients. Moreover, we have our external PDAC dataset, which we reserve for our final evaluations.

Among the trainings that we conducted on sex prediction, RadImageNet classification and PDAC Tumor classification, with our findings from Table 6.7 we select the PDAC Tumor classification weights to be the most relevant pre-training as these weights are already containing information that is related to the pancreas organ itself. Therefore, for therapy response prediction of PDAC patients, we have the intuition to use these weights for the initialization of our models. Furthermore, when taking into account both networks, using PDAC0/1 generates an improvement in the metrics compared to other setups.

Furthermore, we compare the performance change when using weights of the models trained for a medical task, indeed for the pancreas organ itself, and weights trained on ImageNet [10]/Kinetics[21] classification task. When we are conducting transfer learning, we reinitialize the last fully connected layer responsible for the final output generation for the binary classification task for PDAC therapy response. As it can also be observed in Table 6.7 for *ResNet18 + Att* in terms of MCC using pre-trainings from sex prediction and PDAC tumor classification produced the same output. However, when we experiment with *3D ResNet (R3D)* using PDAC tumor classification, it produces better results in terms of most of the metrics including MCC.

As our input type has a differentiating number of slices, for 2.5D networks, the batch size for 2D encoder networks is equal to the number of layers of the inputting CT, whereas for 3D networks the batch size is 1.

6.2.2. 2.5D Networks

Furthermore, as shown in Table 6.8 we compare all of the 2.5D networks that we introduced earlier, using ImageNet and PDAC tumor classification pre-trainings. When we utilize the PDAC tumor classification pre-trainings, we use slice attention weights and weights in other modules as initialization, but re-initialize the last fully connected

layer responsible for the final weight generation for the binary classification. As the classification task is not PDAC tumor classification but PDAC therapy response prediction, reusing the weights from another classification task might mislead the training process. Thus, we create a training environment that would not be biased toward previous classification tasks.

In this experiment, we observe that for most of the networks, using PDAC tumor classification pre-trainings results in a higher MCC value compared to using ImageNet pre-trainings. However, using *ViT b 16 + Att* with ImageNet pre-training outperform all other 2.5D network setups including the version where PDAC tumor classification pre-training is used with *ViT b 16 + Att* itself.

Furthermore, it is important to note that *ResNet152 + Att* and most of its variants tends to have a higher F1 score. The reason for this is that all of the other networks are showing a high tendency to predict the class label for *Others*, whereas this tendency was much lower for *ResNet152 + Att* and its versions that employ attention mechanisms. Since there is a tendency towards the class *Others* in the remaining networks the precision values are recorded to be high but the recall values are low. The ResNet18 variants and Vision Transformers are predicting *Progressive Disease* labels for actual *Progressive Disease* instances with low false positives, however, the false negatives were relatively high compared to ResNet152 versions. Thus, the F1 scores are higher for ResNet152 variants, however, the MCC scores are lower.

The reason behind this bias of networks highly predicting the class label for *Others* can be explained due to the reason because of the class imbalance favoring *Others*. As mentioned previously, the ratio of *Progressive Disease* to *Others* is 34.6% to 65.4%. This was not an issue in sex prediction or PDAC0/1, since these datasets are much more balanced in terms of their class distributions.

Moreover, in Figure 6.6, we demonstrate the slice attention weights on a frontal CT view to signify the areas where the weight is greater for our outperforming network *ViT b 16 + Att* with ImageNet pre-training. We demonstrate the weight distribution for samples from both classes that were correctly predicted. Different from the sex prediction and PDAC0/1, the weights are much more distributed among different layers showing that the Multi-Head Attention was challenged to converge to specific regions. To recall, for sex prediction *ViT b 16 + Att* was the outperforming network. In Figure 6.4, we are able to see that the attention output weights were focused on specific regions. Nevertheless, for therapy response prediction task, attention output weights also emphasize weights to specific layers that are relevant to the pancreas. The top two slices with the highest attention weights are provided on the left-hand side of the plots.

	Accuracy	Precision	MCC	F1	AUC
ResNet18 + Att					
ImageNet	0.667	0.571	0.159	0.262	0.439
ImageNet + PDAC 0/1	0.667	**1.000**	0.168	0.082	0.481
ResNet18 + GE + Att					
ImageNet	0.667	**1.000**	0.168	0.082	0.492
ImageNet + PDAC 0/1	0.667	0.556	0.171	0.308	0.496
ResNet18 + SE + Att					
ImageNet	0.674	**1.000**	0.206	0.120	0.460
ImageNet + PDAC 0/1	0.674	0.800	0.186	0.154	0.491
ResNet18 + CBAM + Att					
ImageNet	0.652	0.500	0.039	0.041	0.434
ImageNet + PDAC 0/1	0.667	**1.000**	0.168	0.082	0.524
ResNet152 + Att					
ImageNet	0.422	0.368	0.104	0.524	0.587
ImageNet + PDAC 0/1	0.548	0.408	0.142	0.504	0.570
ResNet152 + GE + Att					
ImageNet	0.659	**1.000**	0.118	0.042	0.529
ImageNet + PDAC 0/1	0.659	0.516	0.193	0.410	0.564
ResNet152 + SE + Att					
ImageNet	0.667	0.750	0.147	0.118	0.509
ImageNet + PDAC 0/1	0.607	0.450	0.191	0.505	0.586
ResNet152 + CBAM + Att					
ImageNet	0.659	0.529	0.158	0.324	0.519
ImageNet + PDAC 0/1	0.533	0.411	0.187	**0.540**	**0.602**
ViT b 16 + Att					
ImageNet	**0.689**	0.857	**0.250**	0.222	0.534
ImageNet + PDAC 0/1	0.667	0.556	0.171	0.308	0.507

Table 6.8.: PDAC Therapy Response Prediction using 2.5D architectures pre-trained on ImageNet and ImageNet + PDAC 0/1. Attention pooling after a network used as encoder is represented as Att. GE denotes Gather-Excite operator [15]. SE denotes Squeeze-and-Excite networks [16]. CBAM denotes Convolutional Block Attention Module [51].

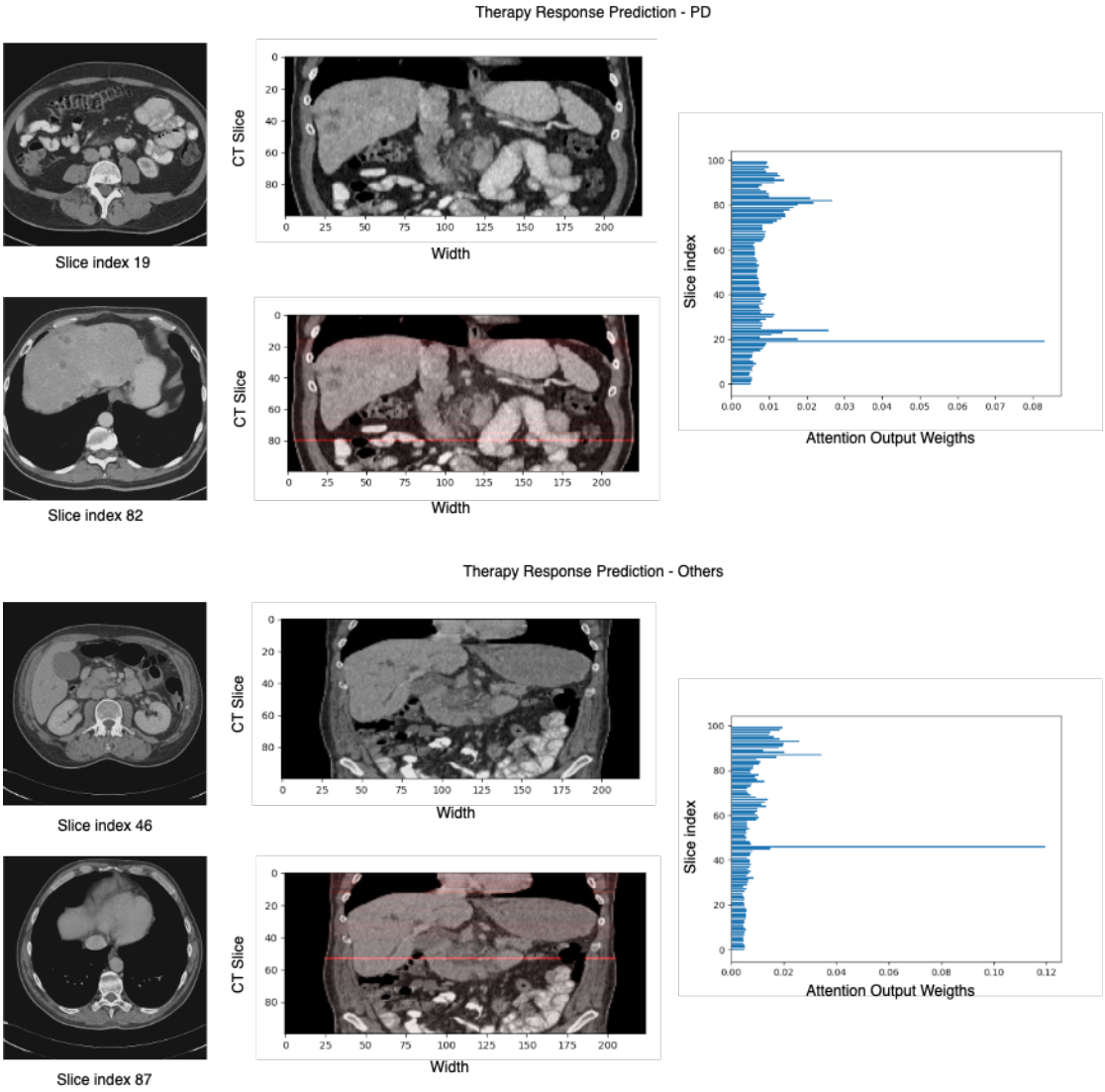


Figure 6.6.: PDAC Therapy Response Prediction: Attention output weights for slice attention using *ViT b 16 + Att* for therapy response *Progressive Disease* and *Others*

6.2.3. 3D Networks

	Accuracy	Precision	MCC	F1	AUC
3D ResNet (R3D)					
Kinetics	0.674	**1.000**	0.206	0.120	0.536
Kinetics + PDAC 0/1	**0.689**	0.778	**0.241**	0.250	0.542
3D ResNet (R3D) + NL					
Kinetics	0.659	**1.000**	0.118	0.042	0.461
Kinetics + PDAC 0/1	0.667	**1.000**	0.168	0.082	0.441
3D Swin Transformer					
Kinetics	0.622	0.438	0.105	0.354	0.521
Kinetics + PDAC 0/1	0.674	0.565	0.206	**0.371**	**0.550**

Table 6.9.: PDAC Therapy Response Prediction using 3D architectures pre-trained on Kinetics 400 and Kinetics 400 + PDAC 0/1. NL represents a Non-Local block in ResNet layer 3

Furthermore, we also experiment with our 3D networks on the effect of using Kinetics pre-trainings and PDAC tumor classification trainings. Similarly, as we did for our 2.5D architectures when using PDAC0/1 pre-trainings we re-initialize the last fully connected layer for final output generation.

Contrary to our finding for 2.5D networks, as provided in Table 6.9, we observe that *3D ResNet (R3D)* with using PDAC tumor classification as pre-training outperforms all other setups in terms of MCC. However, it is important to point out that, 2.5D network *ViT b 16 + Att* using ImageNet pre-trainings produce higher performance in terms of MCC compared to *3D ResNet (R3D)* with using PDAC tumor classification as pre-training. Although, we observe that for both sex prediction and PDAC tumor classification tasks, 3D networks resulted in a higher MCC value.

The experiments for therapy response prediction are trained for 150 epochs. We employ the Adam optimizer using Multi-Step Learning Rate decay starting from 0.001 decaying with a gamma of 0.01 with milestones 3 and 4. For the loss function, we benefit from Binary Cross Entropy with Logits Loss. As we use ImageNet and Kinetics 400 pre-trained weights, we gradually include first the latter parts of the networks in the trainings and then the former regions. Furthermore, since we are using pre-trained weights, we train by freezing the batch normalization layers.

6.2.4. Training Outperforming Networks

Moreover, from the 2.5D and 3D network setups that outperform other networks in their cohort namely, *ViT b 16 + Att* and *3D ResNet (R3D)*, we conduct additional four more trainings where we change the random seed within each run. As we have not used a test split for the preceding experiments conducted for PDAC therapy response prediction, we proceed with including randomization in order to assess the performance of our networks.

	Accuracy	Precision	MCC	F1	AUC
ViT b 16 + Att					
(*run 1*) ImageNet	**0.689**	0.857	**0.250**	0.222	0.534
(*run 2*) ImageNet	0.674	0.588	0.191	**0.313**	**0.561**
(*run 3*) ImageNet	0.667	0.556	0.171	0.308	0.450
(*run 4*) ImageNet	0.667	**1.000**	0.168	0.082	0.511
(*run 5*) ImageNet	0.674	**1.000**	0.206	0.120	0.504
Mean	0.674	0.800	0.197	0.209	0.512
Standard deviation	0.008	0.194	0.030	0.095	0.037
3D ResNet (R3D)					
(*run 1*) Kinetics + PDAC 0/1	0.689	0.778	0.241	0.250	**0.542**
(*run 2*) Kinetics + PDAC 0/1	**0.696**	0.875	0.278	**0.255**	0.533
(*run 3*) Kinetics + PDAC 0/1	0.681	**1.000**	0.239	0.157	0.529
(*run 4*) Kinetics + PDAC 0/1	**0.696**	**1.000**	**0.295**	0.226	0.533
(*run 5*) Kinetics + PDAC 0/1	0.681	0.833	0.220	0.189	0.535
Mean	0.686	0.897	0.254	0.215	0.534
Standard deviation	0.007	0.089	0.028	0.037	0.004

Table 6.10.: PDAC Therapy Response Prediction training outperforming networks in 2.5D and 3D with random seeds and evaluating

Conducting trainings with different random seed initialization, as given in Table 6.10, in a total of five runs are done both outperforming networks. Furthermore, we calculate the mean and standard deviation of the evaluation metrics for each network observing the reliability of approaches with randomness being introduced. For trainings conducted, the same setup defined for 2.5D and 3D trainings for therapy response prediction is used as provided in Table 6.10. Here, we observe that with respect to the mean and standard deviation calculations *3D ResNet (R3D)* outperforms *ViT b 16 + Att*.

6.2.5. Final Evaluation

Using the network setups with the highest MCC value in Table 6.10 for 2.5D and 3D architectures, finally, we evaluate our model performance on the External PDAC dataset.

	Accuracy	Precision	MCC	F1	AUC
ViT b 16 + Att					
(*run 1*) ImageNet	0.696	**0.333**	**0.014**	0.056	**0.503**
(*run 2*) ImageNet	0.571	0.105	-0.188	**0.077**	0.423
(*run 3*) ImageNet	0.607	0.211	-0.083	0.154	0.466
(*run 4*) ImageNet	0.696	**0.333**	**0.014**	0.056	**0.503**
(*run 5*) ImageNet	**0.705**	0.000	0.000	0.000	0.500
Mean	0.655	0.196	-0.049	0.069	0.479
Standard deviation	0.055	0.130	0.079	0.050	0.031
3D ResNet (R3D)					
(*run 1*) Kinetics + PDAC 0/1	0.688	0.000	-0.087	0.000	0.487
(*run 2*) Kinetics + PDAC 0/1	0.688	0.000	-0.087	0.000	0.487
(*run 3*) Kinetics + PDAC 0/1	**0.696**	0.000	-0.061	0.000	**0.494**
(*run 4*) Kinetics + PDAC 0/1	0.688	0.000	-0.087	0.000	0.487
(*run 5*) Kinetics + PDAC 0/1	0.688	0.000	-0.087	0.000	0.487
Mean	0.690	0.000	-0.082	0.000	0.488
Standard deviation	0.003	0.000	0.010	0.000	0.003

Table 6.11.: External PDAC Therapy Response Prediction using networks in 2.5D and 3D trained with random seeds

As provided in Table 6.11, we evaluate the networks mentioned in Table 6.10 for the external PDAC dataset for predicting therapy response. For this dataset, we obtain results that do not hold a meaningful predictive performance. Using the mean MCC metric calculated for both networks, we obtain an MCC with a value of 0. Based on the intuition that the MCC value gives, there is no agreement with the class labels and predicted labels. Therefore, we conclude that our algorithms do not transfer to the external PDAC dataset. The reasons for this can be due to a domain shift between two different datasets or no generalizability due to insufficient dataset size.

7. Conclusion

In this thesis, we considered the problem of predicting the therapy response for Pancreatic ductal adenocarcinoma (PDAC) patients using their diagnosis/pre-treatment computer tomography (CT) images. For this purpose, we introduced end-to-end architectures for classifying patient CTs using RECIST labels for supervision. By combining deep learning techniques and attention mechanisms, we introduced numerous architectures. Furthermore, we showed an approach for using 2D architectures by applying attention pooling to the number of slices of a 3D image, thus forming 2.5D architectures. Moreover, using various attention mechanisms, as the pancreas consists of a relatively small volume of a 3D CT image, we investigated whether attention mechanisms can extract and relate information regarding the task from other regions of the human body as attention mechanisms capture global information and learn where to focus. In addition, we investigated the performance gain using transfer learning using pre-trainings conducted on medical and other domains. From our experiments, we observed that *ViT b 16 + Att* from 2.5D networks and *3D ResNet (R3D)* from 3D networks outperform other networks in their cohort for therapy response prediction. In the PDAC dataset, *ViT b 16 + Att* with ImageNet pre-training achieves a maximum of 25.0% MCC and *3D ResNet (R3D)* with Kinetics + PDAC0/1 pre-training achieves a maximum of 29.5% MCC on the validation set. Furthermore, over five trainings using different random seeds, *ViT b 16 + Att* achieved an average of 19.7% MCC with a standard deviation of 3.0% and *3D ResNet (R3D)* achieved an average of 25.4% MCC with a standard deviation of 2.8%. As we focused on end-to-end networks, the testing time for a dataset with 112 patients takes 60 seconds. Whereas the 3-staged pipeline approach proposed for the same task achieves an average of 33.1% MCC with a standard deviation of 2.2%, however, testing time takes 180 seconds for the same 112 patients [56]. Thus, our approach is three times faster, yielding an MCC score of 7.7% lower. Furthermore, we evaluated our networks on an external PDAC dataset, where both approaches draw results such that therapy response prediction on diagnosis/pre-treatment CT images do not transfer. The reason for such a difference can be explained due to a large domain shift from the PDAC dataset to the external PDAC dataset or not having enough training data for this task. Thus, our models are not generalized for therapy response prediction task.

As most of the medical deep learning tasks suffer from, in this thesis, we also had the limitation of the dataset size being small relative to other publicly available datasets

such as ImageNet and Kinetics. For deep learning tasks, it is known that having an excess amount of data samples tends to make networks much more generalized and perform better in terms of many metrics [30]. Moreover, it is known that Transformer architectures need even more data samples for outperforming other deep learning methodologies [29]. Another issue that we had with our PDAC dataset was the class imbalance among *Others* and *Progressive Disease* labels, where *Progressive Disease* labels were scarce. This has caused the majority of our models to tend to predict the instances as *Others*. As a technical limitation, we have faced the issue of graphics processing unit (GPU) memory not being able to load and perform operations on the whole CT volumes that we had. Therefore, we were forced to downsample the temporal and spatial domains based on the network that we were training. For the temporal domain, we needed to take CT slices with equal intervals and form a new 3D CT. Furthermore, for some approaches, we needed to resize our CT slices to a smaller spatial size.

For both sex prediction and PDAC tumor classification, 3D networks performed better compared to our 2.5D approaches. However, it is crucial to note that the gap between best-performing approaches for PDAC0/1 in terms of MCC was minor as *ResNet18 + Att* obtained an MCC value of 86.1% and *3D ResNet (R3D)* obtained an MCC value of 86.9%.

For sex prediction task, we observed that a 3D network utilizing spatiotemporal attention mechanisms named *3D Swin Transformer* approach outperformed the model with the highest MCC among 2.5D models with a greater difference by achieving 95.4%. As the whole CT volume is more crucial for the nature of the task, we can claim that spatial and temporal attention mechanism employed over the whole body scan were able to extract and relate information successfully. From the slice attentions that we extracted using our 2.5D architectures *ResNet18 + Att* and *ViT b 16 + Att*, we observed that the *ViT b 16 + Att* was producing attention output weights towards the pelvic and chest area of the body. It is also important to note that not all of the patient CTs were including the genital or thoracic regions, therefore slice attention focusing toward the pelvis supports the generalizability of the model as it exists in most of the CTs.

However, for the PDAC0/1 task, as the pancreas consists of a relatively small portion of the 3D CT volume, networks were accessing additional information other than the relevant region for the task. As *ResNet18 + Att* being the outperforming network for PDAC0/1 classification within 2.5D networks, we have visualized the attention output weights for CT slices on top of the frontal CT view of the patient in Figure 6.5 and we have observed that the slices were focused towards the body parts that were close to the pancreas location. Thus, we have once again showed that the attention mechanisms were able to focus on relevant regions based on the task. Furthermore, *3D ResNet (R3D)* was the outperforming one among 3D networks, as well as outperforming *ResNet18 + Att*.

For the therapy response prediction task, we have compared the performance gain from transfer learning using various domains. With the experiments that we have conducted, we observed that using PDAC0/1 weights as initialization of our models tends to show a greater improvement compared to other pre-trainings. We further trained 2.5D networks with ImageNet and ImageNet + PDAC0/1 and 3D networks with Kinetics and Kinetics + PDAC0/1 weights for the therapy response prediction task. From our evaluations, we found out that from 2.5D networks *ViT b 16 + Att* and from 3D networks *3D ResNet (R3D)* were the outperforming networks. By training these networks with additional setups where the random seed was changed, we obtained a maximum of 29.5% MCC on the validation set. Next, we have evaluated the performance of the outperforming networks trained with random seeds, on the external PDAC dataset. However, with the external dataset, we observed that our models did not generalize. The reason for this difference could be due to a large domain shift between the PDAC dataset and the external PDAC dataset. As the external dataset was collected from a different hospital, the labeling of the dataset was done by different doctors. Moreover, the algorithms for the PDAC dataset may have not transferred to the external PDAC dataset due to insufficient dataset size causing no generalizability.

In the research done for Gather-Excite (GE) blocks, authors have argued that in their findings, the GE blocks introduce improvements at every main layer, however, the greatest improvements are observed from the mid and late layers [15]. With this intuition, for our 2.5D and 3D ResNet architectures, we have utilized the attention mechanisms that can be integrated into basic and bottleneck blocks within the third main layer of the architectures. Utilizing the attention mechanisms in layers with less number of channels was much more parameter and memory efficient. However, as future work, in order to find the optimal locations for all of the attention mechanisms such as Squeeze-and-Excitation, Convolutional Block Attention Module and Non-Local Neural Networks, an extensive study can be conducted on sex prediction and PDAC tumor classification tasks as a benchmark. Thus, by finding the optimal configurations for each of these attention mechanisms, different results could have been attained. Due to our GPU memory limits, we needed to apply a threshold for the number of slices we could use per CT instance. Thus, using a computational power with greater memory size, without applying any threshold the whole CT volume could be used. We have observed the benefit of transfer learning from a related domain using the PDAC tumor classification dataset. Other datasets that are more directly related to pancreas tasks can be examined. Moreover, by limiting the start and end index of the CT images and also the spatial size, we can limit the image domain with a bounding box around the pancreas and apply attention mechanisms to focus on a closer spatiotemporal area around the pancreas. Thus, reducing redundant information with respect to the task could be an approach to improve performance.

A. Appendix

A.1. Data Pre-processing

For forming the PDAC0/1 dataset, it was observed that the orientations of the CTs in two data sets were different. For instances in PDAC therapy response dataset a 90 degree anti-clockwise rotation was done, whereas for Normal CTs 270 degree anti-clockwise rotation and horizontal flip was done to have the same orientation to be obtained. For sex prediction, PDAC0/1 and therapy response prediction, the instances for these datasets where also clipped between -150 and 200.

A.2. Experiments

For the experiments that we have conducted for sex prediction, we have also extracted the attention heat maps from the attention output weights from different levels of encoder blocks in the *ViT b 16 + Att* architecture. To recall, the *ViT b 16 + Att* is a 2.5D network which performs a slice attention over number of slices per 3D CT volume has. Performing the slice attention, the network performs interpretability of which slices contribute for the final classification output. Moreover, we have investigated the interpretability of the spatial attention maps. Multi-Head attention returns attention output weights along with the attention output. The attention output weights are in the dimension of $numberof\,patches + 1 \times numberof\,patches + 1$. The additional row/column represent the class tokens added to the patch embeddings. Using the top row of the attention output weights, and using weights after first index, we can obtain a vector with size $numberof\,patches$. Reshaping the vector to a 2D spatial map, we have obtained the following attention heat maps for female patient in Figure A.1 and for male patient in Figure A.2. For both patients, we have extracted the attention maps for the first, last and a slice within the top-5 according to the layer attention. Furthermore, towards the bottom, we have plotted the average of attention output weights from each 12 encoder blocks in *ViT b 16 + Att*, and performed a bi-linear interpolation for obtaining the original CT spatial size.

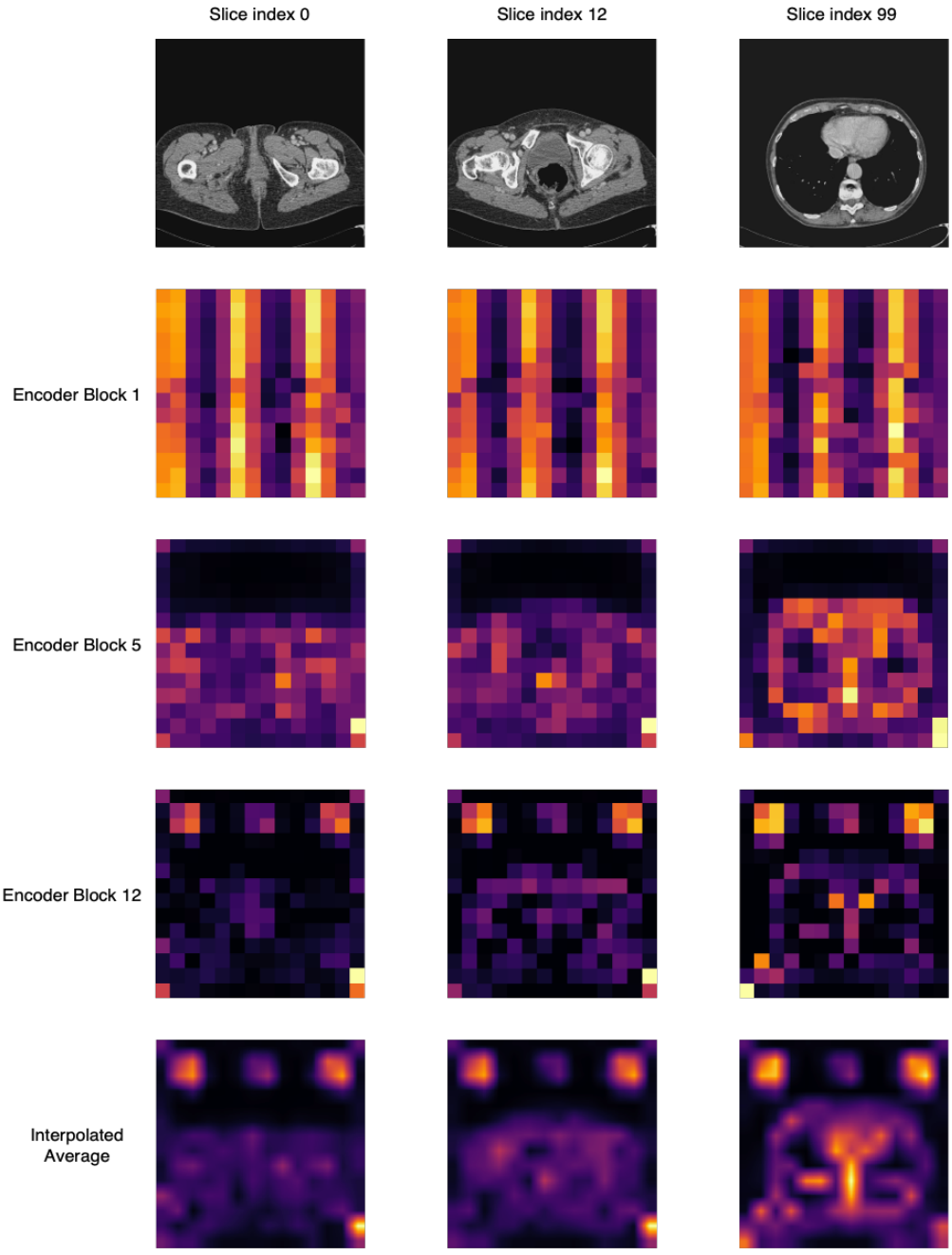


Figure A.1.: Attention output weights for first, last and a slice within the top-5 according to layer attention using *ViT b 16 + Att* for sex prediction for a female patient

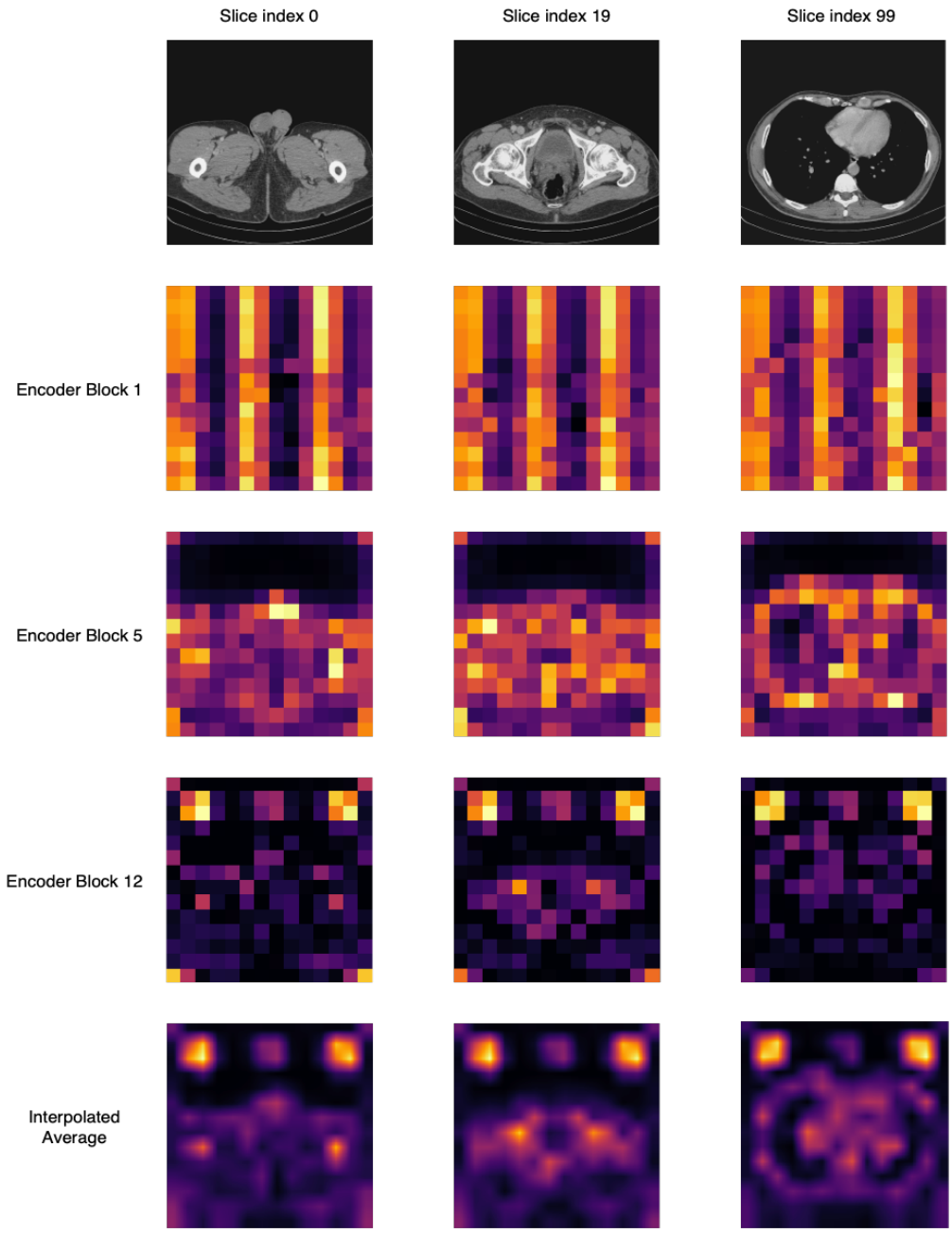


Figure A.2.: Attention output weights for first, last and a slice within the top-5 according to layer attention using *ViT b 16 + Att* for sex prediction for a male patient

List of Figures

List of Tables

Bibliography

[1] M. Antonelli, A. Reinke, S. Bakas, K. Farahani, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze, O. Ronneberger, R. M. Summers, et al. "The medical segmentation decathlon." In: *Nature communications* 13.1 (2022), p. 4128.

[2] G. Argenziano, H. P. Soyer, V. De Giorgio, D. Piccolo, P. Carli, M. Delfino, A. Ferrari, R. Hofmann-Wellenhof, D. Massi, G. Mazzocchetti, et al. "Interactive atlas of dermoscopy." In: (2000).

[3] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. "Vivit: A video vision transformer." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 6836–6846.

[4] G. Bertasius, H. Wang, and L. Torresani. "Is space-time attention all you need for video understanding?" In: *ICML*. Vol. 2. 3. 2021, p. 4.

[5] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. "A short note about kinetics-600." In: *arXiv preprint arXiv:1808.01340* (2018).

[6] J. Carreira and A. Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6299–6308.

[7] R. Chang, S. Qi, Y. Wu, Q. Song, Y. Yue, X. Zhang, Y. Guan, and W. Qian. "Deep multiple instance learning for predicting chemotherapy response in non-small cell lung cancer using pretreatment CT images." In: *Scientific Reports* 12.1 (2022), p. 19829.

[8] S. Chen, K. Ma, and Y. Zheng. "Med3d: Transfer learning for 3d medical image analysis." In: *arXiv preprint arXiv:1904.00625* (2019).

[9] D. Delitto, D. Zhang, S. Han, B. S. Black, A. E. Knowlton, A. C. Vlada, G. A. Sarosi, K. E. Behrns, R. M. Thomas, X. Lu, et al. "Nicotine Reduces Survival via Augmentation of Paracrine HGF–MET Signaling in the Pancreatic Cancer MicroenvironmentThe Effect of Nicotine on c-Met in Pancreatic Cancer." In: *Clinical Cancer Research* 22.7 (2016), pp. 1787–1799.

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

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." In: *arXiv preprint arXiv:2010.11929* (2020).

[12] Z. Gao, J. Xie, Q. Wang, and P. Li. "Global second-order pooling convolutional networks." In: *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2019, pp. 3024–3033.

[13] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu. "Attention mechanisms in computer vision: A survey." In: *Computational Visual Media* 8.3 (2022), pp. 331–368.

[14] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[15] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi. "Gather-excite: Exploiting feature context in convolutional neural networks." In: *Advances in neural information processing systems* 31 (2018).

[16] J. Hu, L. Shen, and G. Sun. "Squeeze-and-excitation networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[17] M. Huh, P. Agrawal, and A. A. Efros. "What makes ImageNet good for transfer learning?" In: *arXiv preprint arXiv:1608.08614* (2016).

[18] S. Ji, W. Xu, M. Yang, and K. Yu. "3D convolutional neural networks for human action recognition." In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2012), pp. 221–231.

[19] F. Jungmann, G. A. Kaissis, S. Ziegelmayer, F. Harder, C. Schilling, H.-Y. Yen, K. Steiger, W. Weichert, R. Schirren, I. E. Demir, et al. "Prediction of tumor cellularity in resectable PDAC from preoperative computed tomography imaging." In: *Cancers* 13.9 (2021), p. 2069.

[20] Kaggle. *Diabetic retinopathy detection*. 2015.

[21] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. "The kinetics human action video dataset." In: *arXiv preprint arXiv:1705.06950* (2017).

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[23] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. "Handwritten digit recognition with a back-propagation network." In: *Advances in neural information processing systems* 2 (1989).

[24] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation applied to handwritten zip code recognition." In: *Neural computation* 1.4 (1989), pp. 541–551.

[25] J. Li, J. Wang, Q. Tian, W. Gao, and S. Zhang. "Global-local temporal representations for video person re-identification." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 3958–3967.

[26] X. Li, W. Wang, X. Hu, and J. Yang. "Selective kernel networks." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 510–519.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. "Swin transformer: Hierarchical vision transformer using shifted windows." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[28] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. "Video swin transformer." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 3202–3211.

[29] J. Mao, H. Zhou, X. Yin, Y. C. Xu, and B. N. Rui. "Masked autoencoders is an effective solution to transformer data-hungry." In: *arXiv preprint arXiv:2212.05677* (2022).

[30] G. Marcus. "Deep learning: A critical appraisal." In: *arXiv preprint arXiv:1801.00631* (2018).

[31] X. Mei, Z. Liu, P. M. Robson, B. Marinelli, M. Huang, A. Doshi, A. Jacobi, C. Cao, K. E. Link, T. Yang, et al. "RadImageNet: An open radiologic deep learning research dataset for effective transfer learning." In: *Radiology: Artificial Intelligence* 4.5 (2022), e210315.

[32] A. Menegola, M. Fornaciali, R. Pires, F. V. Bittencourt, S. Avila, and E. Valle. "Knowledge transfer for melanoma screening with deep learning." In: *2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017)*. IEEE. 2017, pp. 297–300.

[33] M. Orth, P. Metzger, S. Gerum, J. Mayerle, G. Schneider, C. Belka, M. Schnurr, and K. Lauber. "Pancreatic ductal adenocarcinoma: biological hallmarks, current status, and future perspectives of combined modality treatment approaches." In: *Radiation Oncology* 14.1 (2019), pp. 1–20.

[34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. "Automatic differentiation in pytorch." In: (2017).

[35] L. Peng, H. Liang, G. Luo, T. Li, and J. Sun. "Rethinking Transfer Learning for Medical Image Classification." In: *medRxiv* (2022), pp. 2022–11.

[36] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. "Transfusion: Understanding transfer learning for medical imaging." In: *Advances in neural information processing systems* 32 (2019).

[37] D. Sarvamangala and R. V. Kulkarni. "Convolutional neural networks in medical image understanding: a survey." In: *Evolutionary intelligence* 15.1 (2022), pp. 1–22.

[38] L. H. Schwartz, L. Seymour, S. Litière, R. Ford, S. Gwyther, S. Mandrekar, L. Shankar, J. Bogaerts, A. Chen, J. Dancey, et al. "RECIST 1.1–Standardisation and disease-specific adaptations: Perspectives from the RECIST Working Group." In: *European journal of cancer* 62 (2016), pp. 138–145.

[39] R. L. Siegel, K. D. Miller, and A. Jemal. "Cancer statistics, 2018." In: *CA: a cancer journal for clinicians* 68.1 (2018), pp. 7–30.

[40] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás. "3D deep learning on medical images: a review." In: *Sensors* 20.18 (2020), p. 5097.

[41] C. Stornello, L. Archibugi, S. Stigliano, G. Vanella, B. Graglia, C. Capalbo, G. Nigri, and G. Capurso. "Diagnostic delay does not influence survival of pancreatic cancer patients." In: *United European Gastroenterology Journal* 8.1 (2020), pp. 81–90.

[42] J. Sun, J. Jiang, and Y. Liu. "An introductory survey on attention mechanisms in computer vision problems." In: *2020 6th International Conference on Big Data and Information Analytics (BigDIA)*. IEEE. 2020, pp. 295–300.

[43] P. Therasse, S. G. Arbuck, E. A. Eisenhauer, J. Wanders, R. S. Kaplan, L. Rubinstein, J. Verweij, M. Van Glabbeke, A. T. van Oosterom, M. C. Christian, et al. "New guidelines to evaluate the response to treatment in solid tumors." In: *Journal of the National Cancer Institute* 92.3 (2000), pp. 205–216.

[44] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. "Learning spatiotemporal features with 3d convolutional networks." In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[45] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. "A closer look at spatiotemporal convolutions for action recognition." In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 6450–6459.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need." In: *Advances in neural information processing systems* 30 (2017).

[47] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu. "ECA-Net: Efficient channel attention for deep convolutional neural networks." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11534–11542.

[48] X. Wang, R. Girshick, A. Gupta, and K. He. "Non-local neural networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7794–7803.

[49] X. Wang and A. Gupta. "Videos as space-time region graphs." In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 399–417.

[50] K. Weiss, T. M. Khoshgoftaar, and D. Wang. "A survey of transfer learning." In: *Journal of Big data* 3.1 (2016), pp. 1–40.

[51] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. "Cbam: Convolutional block attention module." In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.

[52] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. "Show, attend and tell: Neural image caption generation with visual attention." In: *International conference on machine learning*. PMLR. 2015, pp. 2048–2057.

[53] X. Yang. "An overview of the attention mechanisms in computer vision." In: *Journal of Physics: Conference Series*. Vol. 1693. 1. IOP Publishing. 2020, p. 012173.

[54] Z. Yang, L. Zhu, Y. Wu, and Y. Yang. "Gated channel transformation for visual recognition." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11794–11803.

[55] S. Zhang, C. Wang, H. Huang, Q. Jiang, D. Zhao, Y. Tian, J. Ma, W. Yuan, Y. Sun, X. Che, et al. "Effects of alcohol drinking and smoking on pancreatic ductal adenocarcinoma mortality: A retrospective cohort study consisting of 1783 patients." In: *Scientific Reports* 7.1 (2017), p. 9572.

[56] A. Ziller, A. C. Erdur, F. Jungmann, D. Rueckert, R. Braren, and G. Kaissis. "Exploiting segmentation labels and representation learning to forecast therapy response of PDAC patients." In: *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*. 2023.