3-8-2023

Efficient Scopeformer: Towards Scalable and Rich Feature Extraction for Intracranial Hemorrhage Detection using Hybrid Convolution and Vision Transformer Networks

Yassine Barhoumi
Rowan University

EFFICIENT SCOPEFORMER: TOWARDS SCALABLE AND RICH FEATURE EXTRACTION FOR INTRACRANIAL HEMORRHAGE DETECTION USING HYBRID CONVOLUTION AND VISION TRANSFORMER NETWORKS

By

Barhoumi Yassine

A Thesis

Submitted to the
Department of Electrical & Computer Engineering
College of Engineering
In partial fulfillment of the requirement
For the degree of
Master of Science in Electrical and Computer Engineering
At
Rowan University
January 27, 2023

Thesis Chair: Nidhal C. Bouaynaya, Ph.D., Professor and Associate Dean for Research, Department of Electrical & Computer Engineering

Committee Members:

Ghulam Rasool, Ph.D., Assistant Professor, Department of Electrical & Computer Engineering

Ravi Ramachandran, Ph.D., Professor, Department of Electrical & Computer Engineering

Acknowledgment

This work was partly supported by the National Science Foundation (NSF) Awards ECCS-1903466 and OAC-2008690.

Abstract

Yassine Barhoumi
EFFICIENT SCOPEFORMER: TOWARDS SCALABLE AND RICH FEATURE EXTRACTION FOR INTRACRANIAL HEMORRHAGE DETECTION USING HYBRID CONVOLUTION AND VISION TRANSFORMER NETWORKS.
2022-2023
Nidhal C. Bouaynaya, Ph.D.
Ghulam Rasool, Ph.D.
Master of Science in Electrical and Computer Engineering

The field of medical imaging has seen significant advancements through the use of artificial intelligence (AI) techniques. The success of deep learning models in this area has led to the need for further research. This study aims to explore the use of various deep learning algorithms and emerging modeling techniques to improve training paradigms in medical imaging. Convolutional neural networks (CNNs) are the go-to architecture for computer vision problems, but they have limitations in mapping long-term dependencies within images. To address these limitations, the study explores the use of techniques such as global average pooling and self-attention mechanisms. Additionally, the study investigates the performance of vision transformers (ViTs), which have shown potential for outperforming CNNs in image classification tasks. The Scopeformer, a new end-to-end architecture that combines the unique strengths of both CNNs and ViTs, is proposed to improve upon their individual performance. The study contributes to the conversation about effective approaches for tackling challenging computer vision tasks in medical imaging.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

List of Figures

Figure Page

List of Figures (Continued)

Figure Page

List of Tables

Figure Page

Chapter 1

Introduction

1.1 Problem Statements

Stroke is a serious medical condition that occurs when the blood supply to the brain is disrupted, resulting in brain cell death and permanent brain damage [1]. It is a leading cause of death and disability worldwide, with over 15 million people experiencing a stroke each year [2]. Intracranial hemorrhaging, or bleeding within the skull, is a common type of stroke that can result in serious medical complications, including increased pressure within the skull, brain swelling, and extensive brain damage if left untreated [3]. Timely detection and treatment of brain hemorrhages, or bleeding within the skull, is critical for improving patient prognosis and treatment outcomes [4], as it can significantly reduce the risk of serious medical complications and prevent long-term disability or death [5].

One effective way to detect and classify brain hemorrhages at an early stage is through the use of head computed tomography (CT) scans [6]. These scans produce detailed images of the brain and are capable of accurately identifying the location and extent of brain hemorrhages [7]. In addition to being a valuable diagnostic tool [8], head CT scans are also quick and easy to perform [9], making them a practical option for the management of brain hemorrhages [10].

The importance of prompt detection and treatment of brain hemorrhages cannot be overstated [11]. Early detection and treatment can significantly reduce the risk of serious medical complications and improve patient prognosis and treatment outcomes [12,13]. Therefore, it is crucial for healthcare professionals to have access to the necessary diagnostic tools, such as head CT scans [14], to accurately diagnose and manage brain hemorrhages [15].

Currently, the detection and classification of brain hemorrhages is often reliant on the expertise of qualified physicians, who must manually evaluate CT scans to identify indications of bleeding or abnormalities within the brain tissues [16]. This process can be time-consuming and may not always yield accurate results, particularly in cases where the hemorrhages are small or subtle [17]. It is also prone to human error, as it relies on the subjective interpretation of the scans by the physician [18, 19].

Automated detection methods using deep learning algorithms have been developed to address these issues and improve the accuracy of hemorrhage detection [20]. Machine learning algorithms have the potential to significantly improve the accuracy and speed of brain hemorrhage detection, particularly within the first 24 hours after the onset of symptoms, when the risk of serious complications is highest [2, 21-23]. These algorithms can be trained to autonomously identify and classify brain hemorrhages by analyzing CT scans, without the need for human intervention [24]. This can greatly reduce the detection time, enabling faster and more effective treatment of brain hemorrhages, and potentially preventing serious medical complications and extensive brain damage [25].

The use of machine learning algorithms for the detection and classification of brain hemorrhages is a rapidly evolving field [26], with numerous studies and clinical trials exploring the potential benefits of this approach [27, 28]. Some machine learning algorithms have demonstrated high levels of accuracy in identifying brain hemorrhages, rivaling the performance of experienced physicians [29, 30]. Emerging computer vision techniques, such as deep learning [31], offer the possibility of creating faster and more robust models for the detection and classification of brain hemorrhages [32,123-126]. These algorithms, combined with advanced feature extraction methods [33], can analyze CT scans more quickly and accurately than a human analyst, potentially saving valuable time and resources [34]. By leveraging the power of deep learning [35], these algorithms can identify patterns and features within the CT scans that may not be immediately visible to the human eye, improving the accuracy and speed of the detection process [36].

In addition to improving the accuracy and speed of brain hemorrhage detection [37], the use of machine learning algorithms has the potential to reduce the workload of healthcare-qualified physicians and improve access to diagnostic services in underserved or remote areas [38]. It can also help to reduce the cost of healthcare by automating the detection process and freeing up medical professionals to focus on more complex and nuanced cases [39]. This can help to improve the overall efficiency of the healthcare system and reduce the burden on qualified medical professionals [40]. The use of machine learning algorithms for the detection and classification of brain hemorrhages can also assist expert physicians and radiologists in triaging patients, prioritizing those with more severe or acute conditions [41]. This can help to ensure that patients receive the appropriate level of care and attention [42].

Overall, as a promising breakthrough in healthcare, the use of machine learning for the early detection and classification of brain hemorrhages has the potential to greatly improve patient outcomes and reduce the burden of stroke on healthcare systems worldwide. Accurate and efficient detection and classification can lead to timely and effective treatment, potentially preventing serious complications and extensive brain damage. By leveraging the power of deep learning and computer vision techniques, these algorithms have the potential to significantly improve patient outcomes and streamline the diagnostic process. It is an exciting area of research with much potential for future development and application in clinical practice.

1.2 Thesis Objectives

The aim of our project is to make significant strides in the use of cutting-edge tools from deep learning algorithms in medical image classification, particularly in the detection of hemorrhages from CT scans [43]. We hope to significantly enhance the capabilities of these algorithms in this field, enabling more accurate and efficient identification of brain hemorrhages [44]. By focusing on intracranial hemorrhages specifically, we hope to address a critical need in the field of stroke care [45] and reduce the burden of this condition on healthcare systems worldwide [46]. We intend to develop a reliable and efficient method for detecting multi-type brain hemorrhaging in CT scans [47], enabling medical professionals to promptly initiate the necessary treatment and potentially prevent further complications [48].

The goal of our research is to explore the potential benefits of integrating two state-of-the-art deep learning models, the convolutional neural network (CNN) [49] and the vision transformer (ViT) [50], into a single end-to-end model called Scopeformer. By combining the strengths of both architectures, we aim to create a hybrid model that outperforms either model alone on the task of detecting hemorrhages from medical images [51]. This is a critical problem in the field of healthcare, as early and accurate identification of intracranial hemorrhages can facilitate prompt treatment and potentially prevent further complications [52].

To validate the effectiveness of our proposed hybrid model, we will apply it to the RSNA hemorrhage detection challenge [53], a widely recognized benchmark in the field of medical image analysis [54]. Our aim is to demonstrate that the Scopeformer model is able to achieve superior performance compared to either the CNN or ViT model alone, as well as other state-of-the-art approaches [55]. In addition to its potential practical value, our research will also contribute to the growing body of knowledge on the application of deep learning models in medical image analysis [56] and the potential benefits of hybrid models [57]. Overall, our work has the potential to make a meaningful impact on the field of healthcare and improve patient outcomes [58].

This thesis aims to achieve the following objectives:

1. Design a hybrid CNN-ViT architecture to address the RSNA hemorrhage detection challenge.
2. Demonstrate the trainability of the model in various configurations.

3. Investigate different pretrained CNN architectures as feature map extractors for use with various ViT variants.
4. Evaluate the quality of the proposed backbone and suggest training approaches to improve it.
5. Optimize trainable parameters through the use of dimensionality reduction techniques and hyperparameter exploration.

1.3 Hypotheses Statements

We propose that convolutional neural networks (CNNs) can be used effectively as feature extractor modules to generate high-quality feature maps that can serve as input patches for vision transformer (ViT) models. We believe that the incorporation of correlation contexts among features across designated axes within the ViT model can lead to improved classification results. In this study, we will test our hypothesis by using CNNs to extract features from images and feeding these features as input to various ViT models. We will then evaluate the classification performance of these models and explore different techniques for optimizing their trainable parameters.

Our proposed model is predicated on the premise that a combination of pretrained CNNs can generate strong and comprehensive features for the Vision transformer (ViT) block. Various research supports the idea that using a combination of multiple CNNs in a single architecture can generate strong and comprehensive features for use in a subsequent model [59]. An example is where authors propose a hybrid CNN model that combines multiple CNNs in a single architecture to generate a single feature map for the purpose of a segmentation task [59]. The vision transformer (ViT) block is designed to extract functional correlations within the feature map and transmit them to the final classification block while maintaining a constant output dimension [50]. This capability is described in detail in the paper on ViT, which outlines the design and functionality of the model [50]. The ability to extract functional correlations within the feature map and transmit them to the final classification block is a key characteristic of the ViT model and is an important factor in its effectiveness for various image recognition tasks. We believe that incorporating self-attention layers within the model will improve its performance by strengthening the correlation between input patch features [60]. In our end-to-end Scopeformer architecture, we expect that the more ViT encoder blocks we stack, the stronger the extracted feature correlations will be for classification. However, we theorize that there is a critical number of ViT encoders that can be stacked before classification performance reaches a plateau and starts to decrease for a particular dataset. To test these hypotheses, we plan to study the impact of the size and number of ViT encoders on the model's performance, as well as the size of the input feature map determined by the number of CNN architectures used in the Scopeformer model. We will use experimental methods and statistical analysis to evaluate the performance of the model under different configurations and identify the optimal settings for achieving the best classification results.

Convolutional neural networks (CNNs) [61] are powerful machine learning models that are capable of learning a wide range of low-level and high-level features through training on large datasets [62]. These features serve as hard inductive biases [63], guiding the model's decision-making process and helping it to generalize to new data [64]. However, we hypothesize that pretraining the CNNs in our model on diverse types of datasets [65] and using various data augmentation techniques [66] can further bias the learned features towards the RSNA classification application. This is because the core of our model consists of multiple CNNs, each of which is responsible for learning different aspects of the data [67]. By pretraining each CNN independently and significantly altering the training paradigm, we believe that we can improve the final decision-making process of the model [68,69]. To test this hypothesis, we will investigate different approaches to pretraining and training the CNNs in our model, including variations in the datasets and data augmentation techniques used. We will then evaluate the impact of these changes on the classification performance of the model, using experimental methods and statistical analysis to assess the effectiveness of each approach.

The use of multiple CNNs in a single model can be computationally expensive, so we have carefully selected our CNN backbones in an effort to obtain a more consistent and comprehensive feature map. We believe that this will help us to build a more efficient and effective model for the RSNA classification application. To further optimize the model, we will conduct an interpretability analysis [70] using saliency maps [71] to identify CNN architectures that may be redundant or not contribute significantly to the final decision. This will allow us to eliminate unnecessary CNNs and reduce the computational cost of the model without sacrificing performance. Additionally, we will use self-attention visualizations spanning various layers in the ViT pipeline to understand the richness of CNN features and their role in the decision-making process. By using these techniques, we hope to minimize the number of CNN architectures in the model while maintaining or improving its performance.

The objectives of this project are to address the following significant problems in our formulation of the problem hypothesis:

1. We expect that the feature map produced by the various CNN architectures using multiple CNNs will contain redundancies that need to be addressed.

2. The inclusion of various CNN architectures in the backbone of the Scopeformer for end-to-end training may impose computational and memory constraints. We suggest that enriching lower dimensional feature maps produced by the CNNs through pretraining and transfer learning may improve the feature richness and diversity presented to the ViT as a potential solution to this issue.

3. We propose that CNNs learn hard inductive biases, and we hypothesize that soft inductive biases within patterns among these features can be extracted using the self-attention layers of the ViT blocks.

1.4 Thesis Focus and Organization

The primary focus of this thesis is to develop a machine learning model that can accurately detect and classify various types of hemorrhaging in CT scan images from the RSNA hemorrhage detection dataset. This is a challenging problem that requires advanced image analysis techniques to identify and classify the different types of hemorrhaging in the images. To address this problem, we will conduct multiple simulations of the proposed model to optimize its performance. We will use a range of techniques, including data augmentation, dimensionality reduction, and hyperparameter optimization, to fine-tune the model and improve its accuracy in detecting and classifying different types of hemorrhaging. We will also explore different CNN architectures and ViT configurations to identify the most effective combination for the task. Through these simulations, we hope to develop a model that can accurately and efficiently detect and classify different types of hemorrhaging from processed CT scan images.

The first chapter of this thesis provides an in-depth introduction to the problem of detecting intracranial hemorrhaging in medical images, including a discussion of the challenges and importance of accurately identifying and classifying different types of hemorrhaging. We also provide an overview of the RSNA intracranial hemorrhage detection challenge and the motivation behind our work in this area.

The second chapter of the thesis covers the technical background knowledge necessary to understand the computer vision and deep learning techniques used in this study. We discuss the basics of convolutional neural networks (CNNs) and vision transformers (ViTs), as well as various feature extraction methods and their role in improving model performance.

In the third chapter, we describe the approach and methodology behind the simulations conducted on the Scopeformer model. This includes details on the datasets and data augmentation techniques used, the design of the model architecture, and the evaluation criteria used to assess the model's performance.

The fourth chapter presents the results of the proposed hybrid model, including comparisons to other state-of-the-art methods and an analysis of the model's performance on different types of hemorrhaging. We also discuss the strengths and limitations of the model and provide insights into how it can be further improved in the future.

Finally, in the fifth and final chapter, we provide a summary of the overall results and accomplishments of this thesis. We discuss the contributions made by this work and suggest potential directions for future research in this area.

Chapter 2

Literature Review

This chapter provides a thorough review of the technical and theoretical aspects relevant to our proposed work on the RSNA hemorrhage detection problem. We present a literature review of various approaches that have been used to tackle this problem, including common architectural designs and hemorrhage identification methods, as well as data augmentation and dimensionality reduction techniques. We also discuss the adoption of convolutional neural networks (CNNs) and transformer architectures in this context, and the advantages and disadvantages of using these approaches for image classification tasks. In this chapter, we delve into the process of feature extraction executed by different CNN architectures, including the concept of convolutional layers and their role in extracting features from images. We discuss the benefits of using CNNs for image classification tasks, such as their ability to learn hierarchical representations of image data and their robustness to translation and rotation invariance. However, we also highlight some of the limitations of CNNs, including their sensitivity to the choice of hyperparameters and their reliance on large amounts of labeled data for training. In addition to CNNs, we also explore the use of transformer architectures for image classification tasks. We introduce the concept of self-attention and discuss how it allows transformers to capture long-range dependencies and global context in images. We compare the benefits of using transformers over traditional

CNNs for image classification, including their ability to handle variable-length sequences and their greater efficiency in terms of memory and computation. Finally, we present some of the most popular transformer architectures and their applications in the literature, including ViTs and Scopeformers. Overall, this chapter provides a comprehensive overview of the technical and theoretical foundations of our proposed work and sets the stage for the subsequent chapters where we describe the details of our approach and present the results of our simulations.

2.1 Introduction

The application of machine learning in the field of radiology has the potential to significantly enhance the accuracy and speed of intracranial hemorrhage detection, which is a critical task in medical practice [72]. Delayed diagnoses of this type of bleeding can lead to serious complications and even death [5], making it crucial to identify and classify different types of hemorrhaging as quickly and accurately as possible [4]. In this study, we conducted a systematic comparison of various transformer network architectures to assess their ability to extract features that can improve the classification performance of computed tomography (CT) scans for detecting different types of hemorrhaging. By identifying the most effective transformer architecture for this task, we aim to improve the accuracy and reliability of automated hemorrhage detection systems in radiology, which could ultimately lead to better patient outcomes and a reduction in the risk of medical complications caused by delayed diagnoses.

2.2 Vision Transformer

Vision Transformers (ViTs) have gained popularity in a variety of computer vision identification applications [73, 74] and have demonstrated success in a range of vision tasks, including the ImageNet classification challenge [75]. The key component of ViT-based models is the transformer block [76], which was originally introduced by Vaswani et al. [60] in the field of natural language processing (NLP). The successful implementation of the Transformer model [60] applied to images, known as vision Transformer or ViT, was a milestone in the computer vision field [50] with comparable performance to SOTA convolutional neural networks such as Residual neural networks [77] and EfficientNet neural networks [78].

ViTs have been shown to be particularly effective in the medical field, with various successful implementations being proposed that outperform standard convolution-based models by a significant margin [79]. One of the key advantages of ViTs [50] is their ability to extract high-level features from images, which can be used for tasks such as diagnosis and treatment planning [80]. In order to do this, the ViT model divides a natural image into equal, 3-channel square patches, which are then flattened and represented as uni-dimensional tokens. Each patch represents local semantic information from the raw image, and the model learns to extract patterns from their correlations [50]. Using smaller patches allows for the extraction of higher local correlations and improved semantics, as the model is able to analyze the relationships between the different patches in greater detail [81]. However, this increased complexity also results in more expensive computations and a greater need for large amounts of data [82]. It has been shown that ViT models only outperform standard CNNs in high data regimes during pre-training or training [82-83]. In other words, in order to achieve the best results, these models require a large amount of data to learn from in order to extract the most relevant features [82- 83].

Despite the challenges posed by their complexity and data requirements, ViTs have the potential to revolutionize the way that medical diagnoses are made [84]. By providing more accurate and reliable results, they can help doctors to make more informed treatment decisions, ultimately leading to improved patient outcomes. As such, the development of ViT-based models in the medical field is an area of active research and development, with many researchers and engineers working to optimize their design and performance [84].

Vision transformers are strong feature correlation modules [85]. Truong et al. [86] showed this attribute by incorporating feature layers to estimate a large number of confident matches between image pairs. These layers compute each confidence value in the correspondence volume by taking the dot product of two feature vectors extracted from specific locations in the source and target images. [87].

The key success behind ViT-based models are the multi-headed self-attention (MHSA) blocks [60]. MHSA operation within each transformer block enables each inputted tokenized vector to interact with all input vectors which allows the model to construct global correlations crucial in learning semantics. These semantics are further improved with successive computations of the MHSA operations in an end-to-end architecture composed of multitude of ViT encoders [88]. The hierarchical stacking comes with a high-rate increase of computational costs [89]. In every transformer block, the complexity degree of self-attention is quadratic to the number of input tokens. As such, the computation of the attention matrix in MHSA, which requires computationally and volatile memory demanding procedures to compute batch-wise matrix multiplication, can overdraw the memory and computation resources when scaling up to high data regimes [90]. It is especially difficult to simulate larger vision transformer-based models on resource-constrained systems with restricted processing capabilities, stringent memory limits, or a limited power budget.

A comprehensive overview of different vision transformer attempts of implementation in computer vision was presented by Salman H. Khan et. al. [91] in their survey article. The authors provide a detailed analysis of the use of transformers in various computer vision tasks. These tasks include object detection, which involves identifying and localizing objects in images or videos; segmentation, which involves partitioning an image into different regions or classes; and action recognition, which involves identifying and classifying actions performed in videos.

The authors discuss the strengths and limitations of using transformers in these tasks, highlighting the advantages of transformers in terms of their ability to process large amounts of data and capture long-range dependencies. However, they also note that transformers can be computationally expensive and require a large number of parameters, which can be a challenge when scaling to large datasets. In addition, the authors provide insights on future research directions in the use of transformers in computer vision. They suggest that there is potential for further development and improvement of transformer-based approaches in tasks such as image generation and unsupervised learning and suggest that future research should focus on developing more efficient transformer architectures and methods for training and fine-tuning transformers on specific tasks. They also discuss the strengths and limitations of using transformers in these tasks and provide insights on future research directions in this area.

2.2.1 Vision Transformer Architecture Overview

The vision transformer architecture proposed by Dosovitskiy et. al [50], involves a unique approach to processing RGB images. The images are hard split into 16 by 16 by 3 patches without overlap, and each patch is flattened and considered as an input token specific to that patch of the image. These flattened patches are then mapped to a constant latent vector z through a trainable projection to the embedding matrix E, forming a sequence that is taken as the input to the first ViT encoder. This sequence is then packed and used to form the input matrix X, which is projected onto a trainable embedding matrix. This projection allows the model to learn powerful and task-specific features from the input data, enabling it to perform various vision tasks with high accuracy.

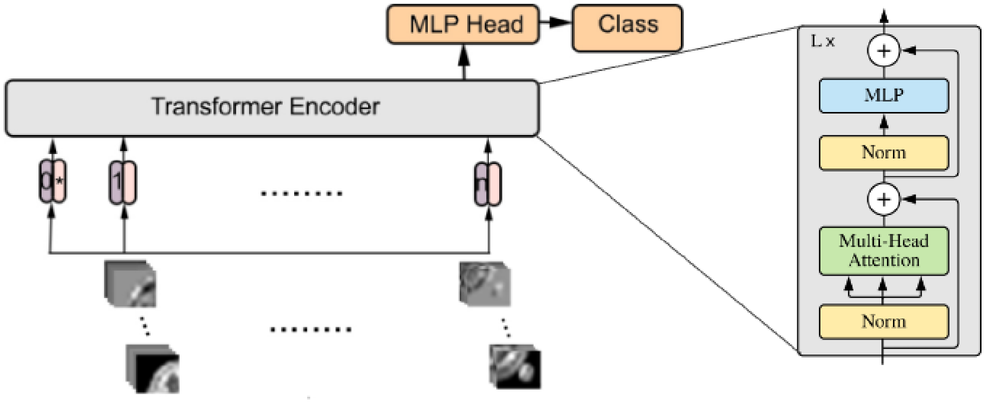
- **Resolution of the input image:** 224 x 224 x3
- **Resolution of the patch:** 16 x 16 x 3
- **Number of patches:** N=196

The authors added a class embedding that can be learned and included it in the input matrix X for the vision transformer model. The input matrix X has a first dimension of N+1, where N represents the number of rows in the matrix. The size of each flattened patch vector is determined by the number of pixels in the patch across the three channels, which is calculated by multiplying the number of pixels in each channel (16 x 16 x 3 = 768). This value of 768 represents the second dimension of matrix X. After being inputted, matrix X is projected onto the embedding matrix E. The first dimension of E is the same as the second dimension of X (768), while the second dimension is determined by the chosen variant of the vision transformer. If the variant selected is "base," the projection will be onto a matrix of 768 x 768. If the variant chosen is "large," the matrix dimension will be 768 x 1024. If the variant selected is "huge," the matrix dimension will be 768 x 1280. The variant of the ViT encoder that is chosen plays a significant role in determining the complexity of the overall vision transformer model. The additional transformation that occurs after the flattening layer allows for control over the desired dimensionality reduction based on the variant that has been selected. This added flexibility allows the user to tailor the model to their specific needs and desired level of complexity.

- **Input:** $\underline{x}\,\underline{\underline{E}} = \underline{z_0}$
- **Dimension:** $(1 * 768)(768 * D) => (1 * D)$
- **Matrix form:** $\underline{\underline{E}}_{pos} + \underline{\underline{X}}\,\underline{\underline{E}} = \underline{\underline{Z}}$
- **Dimension:** $((N+1) * D) + ((N+1) * 768)(768 * D) => ((N+1) * D)$

Figure 1

Overview of the Vision Transformer Model



The positional embedding matrix should have the same dimensions as the resulting inner product multiplication between the matrices X and E. The matrix Z has the same dimensions as the positional embedding matrix, which is also identical to the result of the projection of both matrices X and E. The positional embedding matrix is a trainable matrix, with each row representing the position of the relative vector being added. This allows the model to learn and incorporate the positional information of each vector into the overall representation. The dimensions of the positional embedding matrix are determined by the projection of matrices X and E, and this resulting value is used to set the dimensions for both the positional embedding matrix and the matrix Z.

In this implementation, the same input matrix Z is used at the input level and is passed through a normalization layer three times to create the Key, Query, and Value matrices. These matrices, denoted as K, Q, and V, respectively, have the same dimensions as the input to the encoder, which is the dimension of Z. The dimensions of K, Q, and V are therefore equal to N+1 by D, where N is the number of patches (+1 for the class embedding) and D is the latent vector's dimension. In contrast, the original Transformer work [60] multiplies these matrices by trainable square weight matrices, but this step is not included in this implementation. It is important to note that the dimensions of K, Q, and V are determined by the input to the encoder and are therefore fixed.

- **Key, Query, and value matrices dimensions:** $\dim(K) = \dim(Q) = \dim(V) = ((N + 1) * D)$

The dimension of the latent vector, denoted as D, is determined by the chosen model and remains constant throughout all of the encoder layers. Similarly, the value of N+1 is also conserved. The number of patches is equal to the resolution of the input image divided by the dimension of the patch, and the dimension of the latent vector is equal to the number of pixels in the patch multiplied by the number of channels projected onto the embedding matrix. In other words, the dimensions of the matrix can be simplified to either N by D or (N+1) by D. It is important to note that these values are fixed and determined by the chosen model and input data.

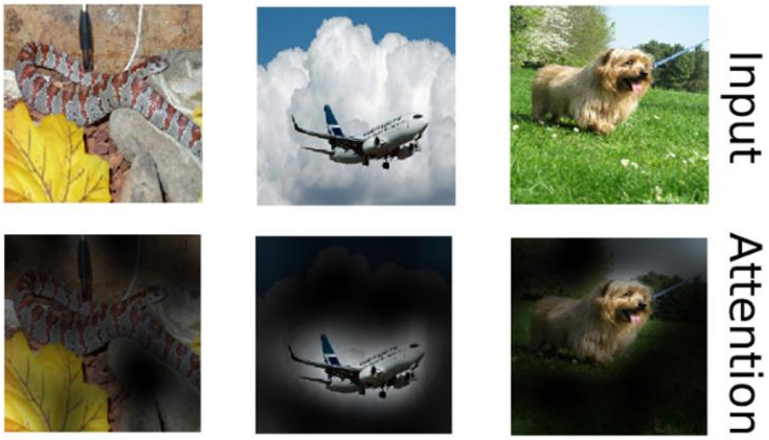
The attention weights, which are obtained through the SoftMax function applied to the normalized product of Q and K, are based on the pairwise similarity between elements of the sequence and their corresponding query and key representations (q and k, respectively). These attention weights indicate how much emphasis should be placed on each element of the sequence when generating the output. The SoftMax function is used to ensure that the attention weights sum to 1, allowing them to be interpreted as probabilities. The attention weights are calculated using the query and key representations of each element, allowing the model to determine the relevance of each element to the current task at hand.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{Q \cdot K^{\top}}{\sqrt{d_k}}) \cdot V.$$

The attention weights are used to determine the amount of emphasis that should be placed on each element of the value matrix, which is identical to the original input matrix Z. This value matrix represents a mapping for the input image itself, and the attention weights can be visualized to show how much the model is focusing on specific pixels that are relevant to the classification or learned class. By applying the SoftMax function to the attention weights, they can be interpreted as probabilities, indicating the importance of each element in the value matrix. This allows us to see which pixels the model is paying the most attention to and how this attention is distributed across the image. Overall, the attention weights SoftMax play a crucial role in determining the relevance of each element in the input image for the current task at hand.

Figure 2

Representative Examples of Attention from the Output Token to the Input Space by the ViT Model



Note. The model attends to image regions that are semantically relevant for classification.

The authors utilized Attention Rollout in their work to calculate attention mappings from the outputted tokens to the input space [92]. This technique involves averaging the attention weights across all 16 heads of the ViT (large variant) before recursively multiplying the weight matrices of all layers. This process allows for the blending of attention among tokens in different ViT encoders, as the attention weights are calculated and averaged across all heads before being used to determine the attention mappings. Attention Rollout allows for a more comprehensive understanding of how the model is attending to different tokens and how this attention is distributed across the input space. It is an important aspect of the ViT model and contributes to its effectiveness in various tasks.

At this point in the process, we have arrived at the level of the key K, query Q, and value V matrices, which have been normalized from the original input matrix Z. Although the ViT model typically includes multiple heads in the encoder, we will assume that there is only one head in this case. When examining the attention operation, we can see that the resulting matrix has the same dimensions as the input matrix Z, meaning that the input dimensions are preserved after the self-attention layer is applied.

The process of preserving the input dimensions begins with the product of Q and the transpose of K, which results in a square matrix with dimensions of $(\mathrm{N}+1)^2$. The SoftMax function is then applied to this matrix, producing a set of probabilities for each element in the matrix. When this matrix is multiplied with the matrix V, the result is a matrix with dimensions of N+1 by D. The square matrix generated by the attention operation puts more emphasis on the elements of the value matrix, which is identical to the original input matrix Z. This ensures that the input dimensions are preserved after the self-attention layer is applied.

In the next step, there is a skip connection and the matrices from the self-attention layer and the skip connection are added together and normalized. We are then at the level of the multi-layer perceptron, which is applied to each position separately and identically. In order to preserve the dimensions, the input and output of the multi-layer perceptron have the same dimensions, with the inner layer having a larger dimension that is dependent on the chosen model variant. The skip connection serves to reinforce the flow of information and, in order to be able to add the two matrices, they must have the same dimensions. This allows the model to effectively incorporate both the self-attention layer and the skip connection into its overall representation.

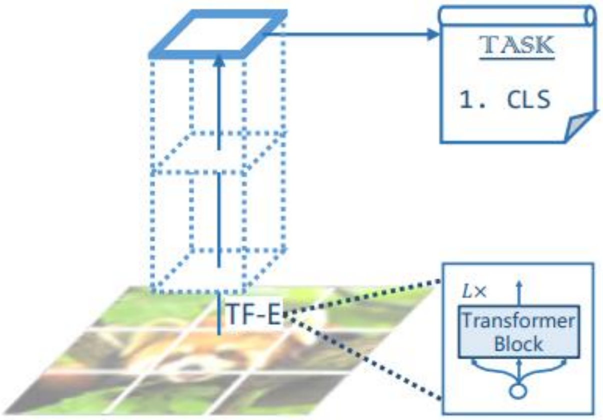
To summarize, we began with an input matrix Z with dimensions of N+1 by D, split it into identical K, Q, and V matrices, applied the multi-head attention layer and passed it through a multi-layer perceptron (MLP) layer, and ended up with a matrix with dimensions of N+1 by D. This process demonstrates that the input to the Transformer layer is identical to its output, regardless of the number of encoder layers used. This means that the dimensions of the input are preserved throughout the entire process. The vision transformer pipeline in this case is columnar rather than pyramidal [93], and the complexity of the problem is determined by the values of N (the number of patches) and D (the dimension of the latent vector). It is important to properly set these values in order to accurately address the problem at hand.

In the case of multi-head attention, or parallel attention layers, we are able to maintain the dimensions of the model by concatenating the heads and reducing the dimensions of the key, query, and value matrices by the number of heads. This allows us to effectively reduce the complexity of the model while still preserving the relevant information. When we concatenate the heads, we are able to retrieve the original dimensions of the model. It is important to properly balance the number of heads with the dimensions of the key, query, and value matrices in order to ensure that the model is able to effectively process the input data while still being computationally tractable.

In summary, we can see that the skeleton of the vision transformer has a columnar structure. Within the transformer block, the input matrix X is attended to itself at the transformer encoder block level through the multi-head self-attention (MHSA) layer. After each block, we are left with an output matrix that has improved semantics and higher correlations among the projected pixel counterparts, leading to better classification results. It is important to note that the dimensions of the input matrix are preserved throughout the transformer blocks, which is a powerful feature that allows the model to effectively process the data while still being computationally tractable. The complexity of the problem is controlled through only two parameters: D and N. However, this particular configuration has some limitations, which motivated the authors of the "Pyramid vision transformer" paper [93] to focus on improving this structure in their work.

Figure 3

Overview of the Vision Transformer Model



Note. The Vision Transformer model has a columnar shape specifically engineered for image classification problems.

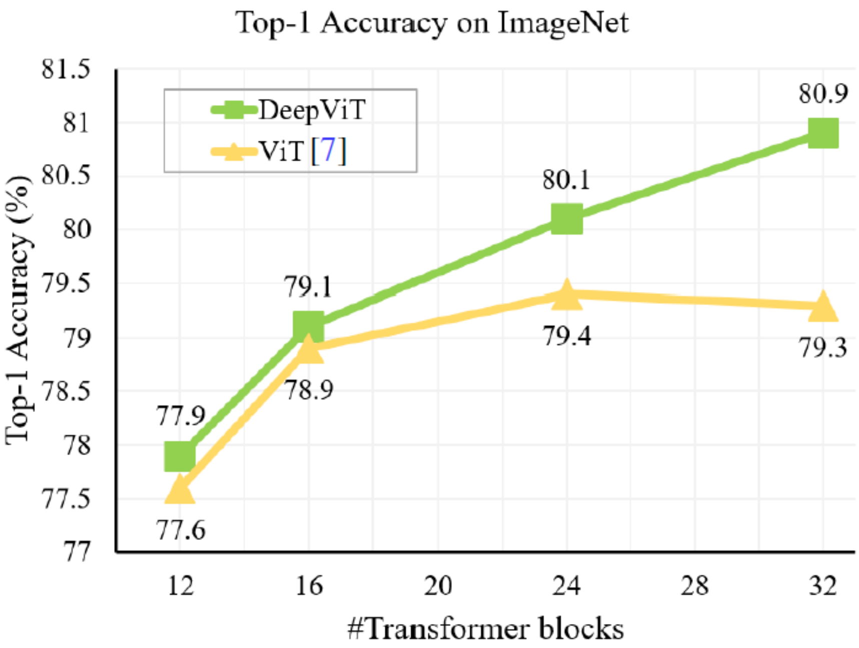
However, vision transformers require high data regimes and huge architectural variants to reach such performances Specifically, if ViT is trained on datasets with more than 14M images it can approach or beat state-of-the-art CNNs.

2.2.2 DeepViT Vision Transformer Architecture Overview

Convolutional neural networks [77, 94] integrate global information by combining multiple convolutional subsequent operations, whereas vision transformers (ViTs) [50] establish patterns from spatial information and non-local dependencies across the encoder block's multi-head self-attention (MHSA) function [60]. This allows vision transformer-based models to acquire richer global context without manually building layer-wise local features extracted by convolution-based filters. Attending to all pixels of the image yields more meanings from global feature correlations. As proven in [73], on ImageNet classification problem, a model composed of 12-blocks of ViT encoders topped a ResNet model composed of more than 30 bottleneck convolutional blocks.

Figure 4

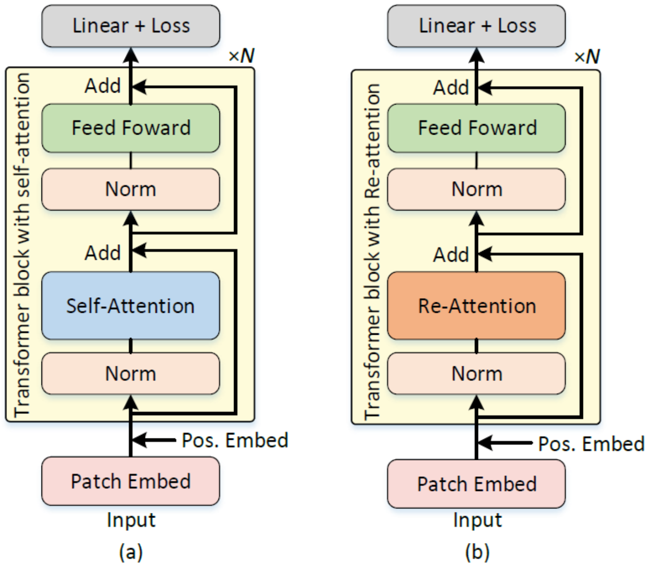
Overview of Top-1 Classification Performance of Vision Transformer Models as Function of the Stacked ViT Encoder Blocks



As shown in Figure 4, when the depth of the ViT model is increased by adding more transformer blocks on top of one another, the performance does not consistently improve. In fact, the model's performance tends to plateau and eventually decline. In contrast, the DeepViT model, which incorporates a re-attention mechanism, is able to improve performance by going deeper into the data. This model does not exhibit the same tendency to plateau, at least under the conditions tested. However, it is worth noting that this plateauing effect may still occur for larger models. Overall, these findings suggest that the re-attention mechanism implemented in the DeepViT model allows for more effective and efficient processing of the input data.

Figure 5

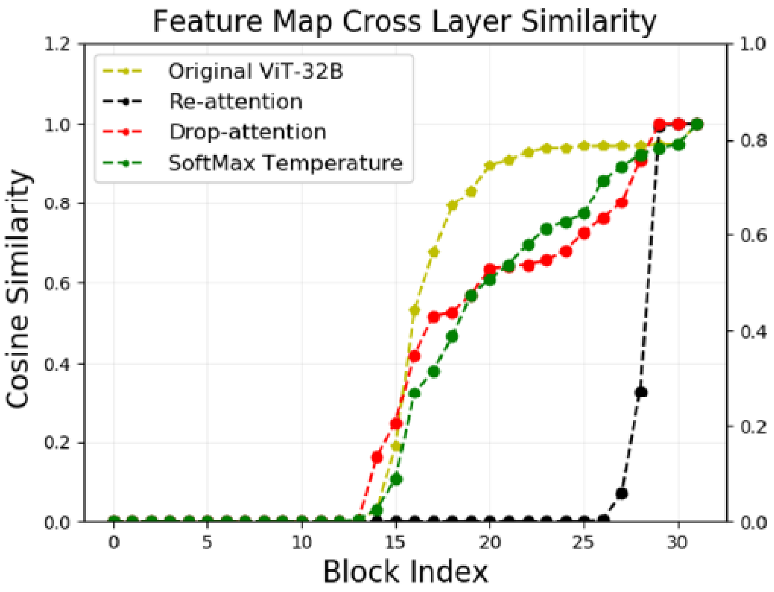
Comparison of the (a) Basic ViT with N Transformer Blocks and (b) the DeepViT Model Suggested in DeepViT Model



One of the key differences between DeepViT and ViT is the inclusion of a re-attention layer within the transformer block. In ViT, this layer is typically replaced with a self-attention layer, which can lead to an "attention collapse" issue that limits the ability to train deeper models. By replacing the self-attention layer with a re-attention layer, DeepViT is able to address this issue and allow for the training of deeper ViT models. This modification allows DeepViT to process the input data and improve performance on various tasks more effectively. Overall, the inclusion of the re-attention layer is a key feature that sets DeepViT apart from ViT and allows it to achieve better results.

Figure 6

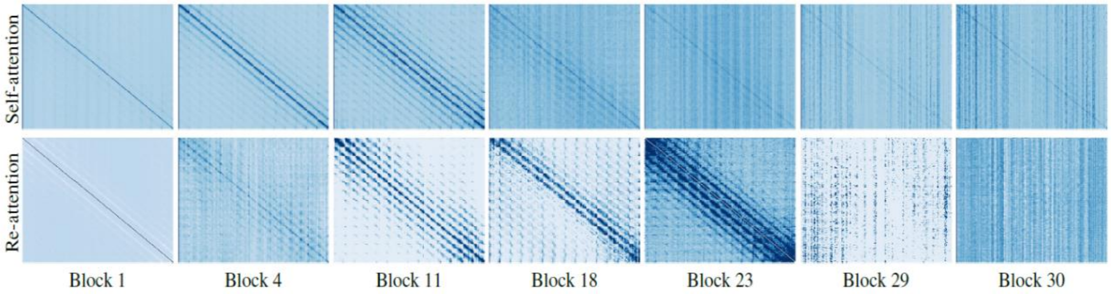
Cosine Similarity Between the Feature Map of the Last Block and Each of the Previous Block



In order to understand the behavior of ViT and DeepViT, the authors analyzed the cosine similarity between two successive feature maps before and after each transformer encoder block. They found that the ViT model initially has no similarities across feature maps in the initial encoder layers. However, as more encoder layers are added, the similarity increases, and the feature maps start to resemble one another. This is due to the fact that, without actual transformations occurring on the transitionary set of tokens, we are simply repeating the same transformer encoders, which is redundant. A similar effect can be seen in the DeepViT model, but it is prolonged, allowing for continuous progress until the point where repeated feature maps are produced that do not include any added information. This analysis helps to shed light on the limitations of both ViT and DeepViT and suggests potential areas for improvement.

Figure 7

Attention Map Visualization of a Baseline ViT Model with Re-Attention - Comparison of Shallow and Deep Blocks with and without Re-Attention



The attention map visualization shown here demonstrates the behavior of both the ViT and DeepViT models. It can be seen that these models primarily learn local patch relationships at the shallow blocks, with most of the attention values near zero. As the block becomes deeper, the scope of the attention maps increases gradually, but they tend to become nearly uniform and lose diversity. This suggests that the models struggle to effectively incorporate information from a wider range of patches as they process the data. Overall, the attention maps provide insight into the patterns of information that the models are able to learn and how they incorporate this information into their overall representation of the input data.

2.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) have been widely used in computer vision tasks, such as image classification, due to their ability to extract high-resolution features from data [95, 96]. These models are particularly effective at analyzing images because they are able to learn and recognize patterns and features within the data, allowing them to effectively classify and identify objects or other relevant features in the images. However, CNNs have certain limitations, such as a reliance on pre-defined kernel sizes and the inability to efficiently process data with long-range dependencies. In recent years, alternative models such as the vision transformer (ViT) have been proposed as a potential alternative to CNNs for certain applications. These models are able to process data in a more flexible and efficient manner, allowing them to potentially outperform CNNs on certain tasks. In the RSNA challenge, the top-ranking solutions for classifying cerebral bleeding in CT scans employed multi-stage classification models that incorporated a convolution-based feature extraction stage [97]. These models were able to accurately identify the presence of bleeding in the scans, highlighting the potential for machine learning to make a significant impact in the medical field. The use of convolution-based models for feature extraction allowed these solutions to effectively analyze the images and extract relevant features for classification. This demonstrates the utility of machine learning in medical applications and the potential for these techniques to improve patient care and diagnosis.

The architecture of a machine learning model, including the stacking and arrangement of its convolutional layers, can significantly impact the features that the model is able to extract from the data. These features are influenced by a variety of factors, such as the architecture structure, the parameters controlling the flow of visual information, and the depth of the model [98]. To improve the performance of the model, it is crucial to carefully consider these factors and optimize the design of the model. This may involve adjusting the architecture, selecting appropriate parameters, or increasing the depth of the model, among other approaches. By carefully designing the model's architecture, it is possible to improve its ability to extract relevant features and improve its performance on various tasks.

One approach to improving the performance of a machine learning model is to increase the depth of its architecture. This can lead to improved feature representations due to the higher non-linearity and increased receptive field of the model. In addition to increasing the depth, there are various other strategies that can be employed to optimize the design of the model, including the use of different convolution layers, activation functions, loss functions, regularization methods, and optimization processes [99]. For example, certain off-the-shelf architectures have been proposed that aim to increase the perceptual field, improve feature extraction efficacy, and reduce the trainable parameter space, resulting in faster and more efficient computation. By carefully considering the design of the model and applying appropriate strategies, it is possible to significantly improve its performance and effectiveness [23,22,39,6,40].

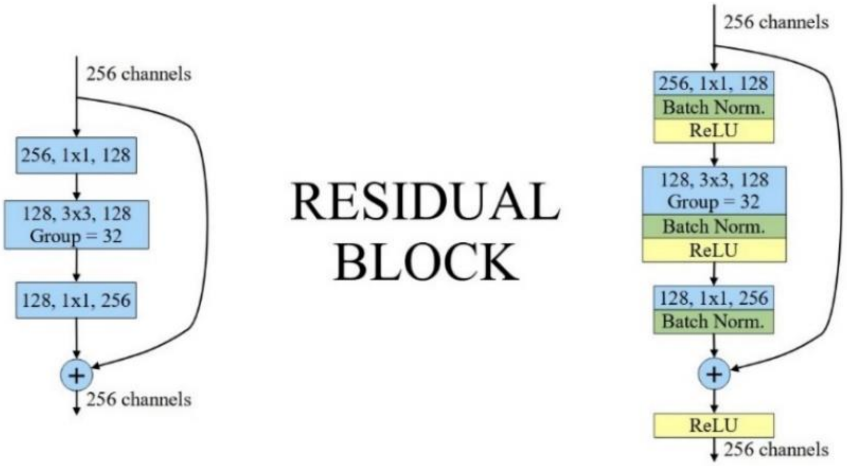
Overall , the design of a machine learning model's architecture plays a critical role in its ability to effectively extract and utilize relevant features from the data. By carefully considering and optimizing various configurations, such as the architecture structure, the parameters controlling the flow of visual information, and the depth of the model, it is possible to significantly improve the model's performance and accuracy. These considerations are particularly important for tasks such as image classification, where the ability to extract and analyze relevant features is critical for accurate and reliable results. By properly designing the model's architecture, it is possible to improve its effectiveness and maximize its potential for success on various tasks.

2.3.1 Residual Neural Networks Overview

The concept of residual networks was first introduced in 2015 in order to address the issue of vanishing gradients that can occur as a network grows in depth. When a network has a large number of layers, the gradients can become very small during the backward propagation process, which can negatively impact the gradient descent algorithm used to update the weights and biases. Residual networks, also known as ResNets, address this issue by incorporating a shortcut link, or skip connection, into the network design. These skip connections allow data from a previous layer to be injected directly into a deeper layer, helping to alleviate the problem of vanishing gradients and improving the performance of the network. Overall, the use of skip connections in residual networks has proven to be an effective method for addressing the challenges of training deep neural networks and improving their performance.

Figure 8

Overview of Two Residual Block Types Used in ResNet Architectures



The skip connection in a residual network, as shown in Figure 12, allows the activation x from an earlier layer to be added to the activation of F(x) a few layers deeper in the network. This sum is then passed through a ReLU non-linearity. By allowing the activation x to bypass several layers and be directly injected into a deeper layer, the skip connection helps to alleviate the problem of vanishing gradients. Even if a significant amount of information is lost in the function F(x), the presence of the activation x from the earlier layer means that some of this information is still present in the deeper layer. This can help to stabilize the gradients and improve the performance of the network. Overall, the skip connection is an important aspect of residual networks and plays a key role in helping to address the challenges of training deep neural networks.

Before the introduction of skip connections, it was difficult to train deep neural networks with more than 25 convolution blocks due to the problem of vanishing gradients. However, the use of residual blocks with skip connections has allowed for the creation of networks with hundreds or even thousands of layers, while still maintaining good performance. These residual blocks enable the network's architecture to grow very deep, while still being able to effectively learn and extract useful features from the data. As a result, the use of residual blocks has greatly expanded the capabilities of deep learning models and has led to significant advances in a wide range of applications.

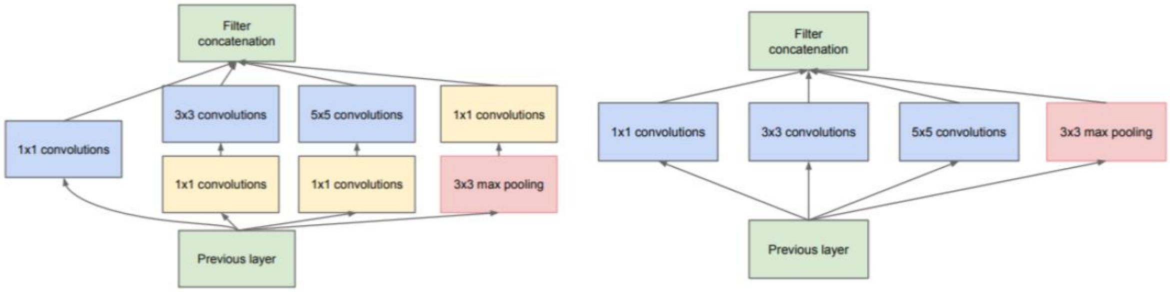
The performance of various residual network architectures was evaluated on the ImageNet dataset, which is a widely used dataset for testing the effectiveness of different network topologies. The results showed that residual networks with skip connections outperformed traditional networks without skip connections on this dataset. For example, a residual network with 34 layers achieved a training error of 7.76%, compared to a training error of 10.02% for a plain 34-layer network without skip connections. These results demonstrate the effectiveness of residual networks in improving the performance of deep learning models and highlight the importance of skip connections in overcoming the challenges of training deep neural networks.

2.3.2 Inception Architecture Overview

The inception module is a popular architecture used to improve the performance of convolutional neural networks (CNNs). It was first introduced in 2014 in the Inception paper [100], and since then it has become a widely used design in CNNs. In traditional CNNs, the size of the filters is a critical hyperparameter that must be carefully chosen by the user. The inception module was developed to address this issue by computing multiple convolutions of different sizes (1x1, 3x3, and 5x5) and a 3x3 max pooling operation, eliminating the need for the user to manually select a single filter size. To reduce the computational cost of these convolutions, the inception module also includes 1x1 convolutions that serve as a dimension reduction step before the more computationally expensive 3x3 and 5x5 convolutions. These 1x1 convolutions are activated with ReLU functions, which introduce nonlinearity to the process. Figure 13 illustrates an inception module with and without dimension reduction.

Figure 9

Overview of Two Inception Modules Used in Inception Architectures



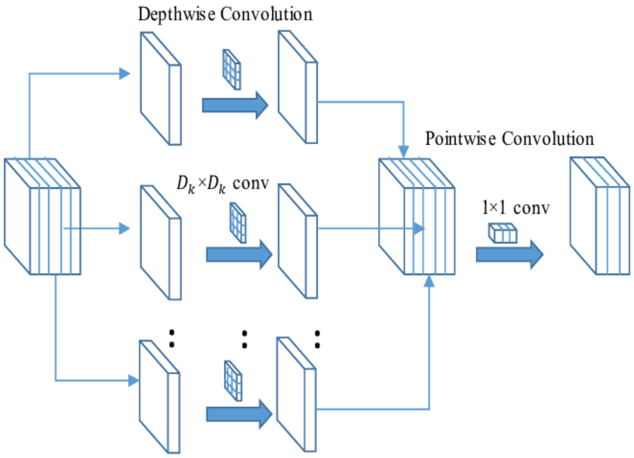
The inception module combines the outputs of convolution and pooling operations to create a single output volume, which is then used as the input for the next layer of the network. This module allows the network to determine which filter sizes and pooling techniques will be most effective at improving the model's accuracy, rather than relying on the user to manually select a filter size. The network can also determine the optimal filter size for a particular layer, eliminating the need for manual selection.

Xception Architecture Overview

The Xception architecture is a type of convolutional neural network (CNN) that is part of the Inception family of CNNs and is characterized by its use of depth-wise separable convolutional layers. Depth-wise separable convolutions are implemented by first performing a spatial convolution independently on each channel of the input, followed by a 1x1 convolution to transform the dimensions. This is in contrast to traditional convolutions, which operate over all channels of a volume at once. The Xception architecture is based on the Inception framework and has been shown to be effective in a range of image classification tasks. This reduces the number of connections and, as a result, the model's learnable parameters.

Figure 10

Overview of the Depth Wise Separable Convolution Used in Xception Architectures



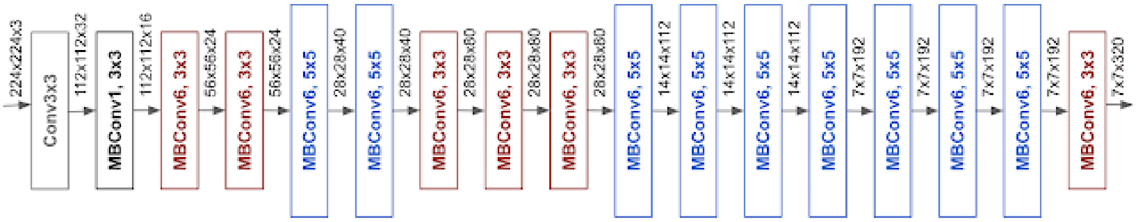
The Xception network architecture is a combination of two previous successful network designs, the ResNet and the Inception Network. It combines the use of skip connections from the ResNet design with depthwise separable convolutional layers from the Inception Network. The resulting network is made up of a linear stack of 36 depthwise separable convolution layers connected by skip connections. This combination of principles allows the Xception network to benefit from both the ResNet and Inception Network designs. When tested on the ImageNet dataset, the Xception network was able to outperform both the 152-layer ResNet and the Inception Network, demonstrating its effectiveness as a network architecture. [23][24].

2.3.3 EfficientNet Architectures Overview

The EfficientNet family of models is a collection of convolutional neural networks that are designed to be both accurate and efficient. They are created using a structural scaling technique that scales every dimension of the architecture using a set of predetermined scaling coefficients. This technique was developed by the authors of the EfficientNet paper [78] and resulted in the creation of the EfficientNet B0 to B7 models. These models have been shown to outperform the state-of-the-art in terms of accuracy on the ImageNet dataset, while also being smaller and faster than other convolution models with similar accuracy scores. The scalability of the EfficientNet models is heavily influenced by the baseline network used, and the authors used the AutoML MNAS framework [102] to conduct a neural architecture search in order to further enhance the performance of the models in terms of FLOPS (floating-point operations per second). The EfficientNet models are created using inverted mobile bottleneck convolution applied to the base model [103].

Figure 11

Overview of the Baseline Scalable and Generalizable EfficientNet-B0 Network in EfficientNet Architectures



2.3.4 Convolution Operations for Features Dimensionality Reduction

Deep convolutional neural networks often have an increasing number of feature mappings as the network depth increases, which can be a disadvantage. This problem can be exacerbated when larger filter sizes, such as 5-by-5 and 7-by-7, are used, as they can significantly increase the number of parameters and computation required to process the data. In order to address this issue, 1-by-1 convolutional layers, also known as projection layers or feature map pooling layers, can be utilized. These layers are effective at down-sampling the content of feature maps and preserving the most salient information, while reducing the overall number of feature maps needed. Additionally, projection layers can be applied directly to feature maps to perform a direct projection, which can be used to generate new feature maps or to pool features across channels in a similar manner to traditional pooling layers. By using these projection layers, it is possible to simplify the model and reduce complexity without sacrificing important features or performance in tasks such as image classification and object detection.

2.4 Convolution Neural Network-Based Vision Transformers

The convolution vision transformer model (CvT) introduced by Wu et al. [104] combines the strengths of both convolutional neural networks (CNNs) and vision transformers (ViT). The CvT model utilizes the end-to-end feature learning capability of CNNs and the input structure of ViT to create a hierarchy of ViT modules and CNN token embeddings. This hierarchy allows the CvT model to take advantage of the scale, shift, and distortion invariances present in CNN features, while also maintaining the dynamic attention and global context capabilities of transformers. Additionally, the CvT model exhibits strong generalizability, making it a powerful tool for various vision tasks. It has the potential to become a widely used architecture in the field of computer vision, particularly for tasks that require both strong feature learning and contextual understanding.

2.5 Vision Transformer in Medical AI

Since their inception, transformers have quickly gained popularity in medical artificial intelligence (AI) applications due to their high level of adaptability. Several successful implementations of the vision transformer (ViT) in the medical field have been proposed and demonstrated significantly better performance compared to traditional convolution-based models [79]. The ability of ViT to effectively process and analyze large amounts of data and provide accurate predictions has made it a valuable tool in the medical field.

2.6 RSNA Challenge and Dataset

The success of modern computer vision models can be largely attributed to the extensively annotated benchmark datasets that have been collected by the machine learning community. These datasets provide a wealth of information for machine learning algorithms to learn from, which is crucial in the development of accurate and reliable models. In 2019, the Radiological Society of North America (RSNA) provided a large collection of brain CT scans for use in a machine learning challenge [105]. The dataset included scans of both healthy participants and patients with various types of an internal cerebral hemorrhage. The RSNA dataset was collected by Adam E. et al. [105] from multiple scanner types used in different institutions around the world. The dataset is considered the current largest dataset publicly available aimed to capture complex real-world details of the hemorrhage sub-types. The dataset was publicly released in the 2019 Intracranial Hemorrhage (ICH) detection challenge hosted by the Kaggle platform. This dataset is considered the current largest dataset available online and contains 870,301 annotated 16-bit grayscale computer tomography (CT) scans saved in the DICOM format, annotated with five types of hemorrhage. Trained physicians categorized each CT slice with one or more types of a brain hemorrhage. Five different forms of hemorrhages are to be identified in this competition, with an additional class representing the presence of any hemorrhage type in the provided slice. These classes were labeled as Epidural hemorrhage

(EDH), Intraparenchymal hemorrhage (IPH), Intraventricular hemorrhage (IVH), subarachnoid hemorrhage (SAH), and Subdural hemorrhage (SDH).

The goal of the machine learning challenge was to encourage the development of autonomous algorithms for multi-class hemorrhage classification. These algorithms, known as computerized multi-label classifiers, were designed to analyze 2D slices of CT images and determine whether there was any cerebral bleeding present. They also provided a probability vector with six components related to different classification targets. This information is important in the field of radiology, as it allows doctors to accurately diagnose and treat patients with cerebral bleeding.

Overall, the use of annotated benchmark datasets and advanced machine learning techniques has greatly improved the accuracy and reliability of modern computer vision models. These models have the potential to revolutionize the way that medical diagnoses are made and can ultimately lead to better patient outcomes. The RSNA challenge serves as a testament to the potential of machine learning in the medical field and highlights the importance of ongoing research and development in this area.

2.7 Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a type of machine learning model that can generate new data by training a learnable generative model on a dataset of existing data [106]. They are designed to have two neural networks: the generator (G) and the discriminator (D), which are connected by a bottleneck known as the latent or feature space [107]. The generator's goal is to create new images (x) by sampling noise from normal distributions and learning latent features (z) from the training dataset, using the equation x=G(z) [106]. These latent features are learned automatically by the GAN during the training process, and we have no control over their specific values or meanings [106]. However, we can analyze the generated images produced by the GAN to better understand the semantic meanings of these latent features [107].

One of the most commonly used generator network designs for GANs is the Deep Convolutional Generative Adversarial Network (DCGAN) [107]. DCGANs use transposed convolutions and up-sampling to create new images from random noise (z) [107]. They are often considered to be reversed deep learning classifiers, as they generate new images rather than classifying existing ones [107]. In this way, the generator creates random noise, and the discriminator guides it towards producing specific types of images [106].

GANs have a wide range of applications, including image generation [106], text generation [108], and data augmentation [108]. They have been used to generate realistic images of faces [109], animals [110], and other objects [111], as well as to create synthetic datasets for training other machine learning models [112]. GANs are a powerful tool for creating new data and have the potential to revolutionize many areas of machine learning and artificial intelligence.

2.8 Transfer Learning

The concept of transfer learning involves using knowledge and skills acquired from tasks that have a large amount of labeled data available to perform tasks that have only a small amount of labeled data available. This can be especially useful in situations where creating new labeled data is time-consuming or expensive, as it allows you to make use of existing datasets in a more efficient way. There are several reasons why transfer learning is commonly used in practice:

- It is often difficult to train a Convolutional Neural Network (CNN) from scratch due to the lack of available labeled data. In these cases, using pre-trained network weights as initializations for training or using a fixed feature extractor can help to solve many problems more effectively.
- Training large neural networks can be very resource-intensive, especially when using powerful graphics processing units (GPUs). Transfer learning can help to reduce the amount of training time and computational resources needed to build and train complex models like the proposed Scopeformer model.
- Determining the optimal topology, training method, hyperparameters, and other details for deep learning models can be challenging, as there is often not much theoretical guidance available. Transfer learning can help to simplify this process by allowing you to leverage the knowledge and skills learned from other tasks to perform new ones more effectively.

2.9 Sharpness-Aware Minimization

Sharpness-Aware Minimization (SAM) is a recent optimization method that has shown great potential in improving the generalization ability of neural networks. Generalization is a crucial aspect of machine learning models, as it allows the model to accurately perform on unseen data. Traditional optimization methods, which rely solely on minimizing the training loss, can often lead to overfitting, where the model memorizes the training data rather than learning generalizable patterns. This can lead to poor performance on unseen data and hinder the model's ability to generalize to new situations. SAM addresses this issue by not only minimizing the loss value, but also minimizing the loss sharpness. Loss sharpness refers to the smoothness versus sharpness of the loss landscape. The loss landscape refers to the shape of the loss function over the space of all possible model parameters. It is often visualized as a three-dimensional plot, with the loss value represented on the vertical axis and the model parameters on the horizontal axes. The loss landscape can have many different shapes, depending on the complexity of the model and the characteristics of the training data. A landscape with many deep, narrow minima is said to have a high degree of sharpness, while a landscape with shallower, more diffuse minima is said to have low sharpness. In other words, it is a measure of how smoothly or sharply the loss function changes as the model's parameters are varied. A loss function with high sharpness will have very distinct and pronounced minima, while a loss function with low sharpness will be more diffuse and have shallower minima. Loss sharpness plays a crucial role in the generalization ability of a model. By searching for parameters in the vicinity of uniformly low loss values, rather than low loss singularities, SAM aims to find a more stable and generalizable set of weights. This approach has been shown to be effective across a variety of computer vision tasks, including CIFAR 10, CIFAR 100, and ImageNet. In addition to improving generalization, the use of SAM has also been shown to enhance the accuracy of machine learning models. By finding a more stable set of weights, SAM is able to reduce the risk of overfitting and improve the model's ability to accurately predict on unseen data. This is particularly important in fields such as healthcare and finance, where accurate predictions are crucial for making informed decisions. Overall, the use of

SAM has the potential to greatly enhance the generalization ability of neural networks and improve the accuracy of machine learning models. It is an exciting development in the field of optimization and has already shown promising results in a variety of tasks. As such, it is likely that we will see more research and development in this area in the future.

2.10 Feature Correlation

The feature correlation layer is an important building block for many computer vision algorithms and has a wide range of applications. It is used to calculate dense correspondences between pairs of images, which can be used to compare the similarity between the two images. This is done by evaluating the pairwise similarities between the reference and query feature maps of a convolutional neural network (CNN), through the calculation of scalar products between corresponding pairs of vectors. These correspondences can be used for a variety of purposes, including geometric matching, disparity estimation, optical flow, few-shot segmentation, semantic matching, and video object segmentation. One of the key benefits of the feature correlation layer is its ability to provide a reliable measure of similarity between image pairs. This can be used to inform various decision-making processes within the CNN, such as identifying corresponding points between images or determining the presence of certain objects or features. The feature correlation layer is an essential component of many computer vision algorithms and has proven to be highly effective in a wide range of applications. As such, it is likely that it will continue to be an important part of the field of computer vision in the future.

2.11 Projection Methods

Projection methods, such as PCA viewed as a linear autoencoder, can be used for dimensionality reduction through the use of 1x1 convolution layers. The goal of these methods is to improve the quality of the data by overcoming sparse and noisy inputs, and to reduce uncertainty and repetitive extracted features. One way to increase the richness of the features inputted to a ViT block is by reducing the output dimension of a CNN and replacing it with different CNN modules. Additionally, compressing the data can be useful for reduced-order models (ROMs), where ViT leverages the data in the form of low-rank features. These projection methods can be particularly useful in scenarios where the amount of available data is limited, as they can help to extract the most important and relevant information from the data.

2.12 Deep Learning Hyperparameters

Deep learning hyperparameters are variables that are not learned during the training process of a deep learning model but are instead set by the practitioner. These hyperparameters can significantly affect the performance of a deep learning model, making their selection an important task in the development of any deep learning system.

There are several common types of hyperparameters that are typically tuned in deep learning models. These include learning rate, batch size, number of epochs, and the size of the network.

The learning rate is a hyperparameter that determines how fast the model updates its weights during training. A larger learning rate can lead to faster training, but it also increases the risk of the model converging to a suboptimal solution. On the other hand, a smaller learning rate can lead to slower training, but it also increases the chance of the model finding a better solution.

The batch size is another important hyperparameter, and it determines the number of training examples used in each iteration of the training process. A larger batch size can lead to faster training, but it can also increase the risk of the model overfitting. A smaller batch size can lead to slower training, but it can also help to prevent overfitting.

The number of epochs is a hyperparameter that determines the number of times the model is trained on the entire dataset. Increasing the number of epochs can lead to a better model, but it also increases the training time.

The size of the network, or the number of layers and the number of units in each layer, is another important hyperparameter. A larger network can lead to better performance, but it also increases the risk of overfitting and the training time.

Tuning these hyperparameters can be a challenging task, as it requires a good understanding of the problem and the trade-offs involved. There are several techniques that can be used to tune hyperparameters, including manual search, grid search, and random search.

Deep learning models are highly dependent on the hyperparameters that are set before training. Hyperparameters are high-level settings that control the overall behavior of the model, such as the learning rate, the size of the model, and the number of epochs to train for. Choosing the right hyperparameters can be a challenging task, as there is often a trade-off between performance and efficiency. Too high a learning rate may result in unstable training, while too low a learning rate may result in slow convergence. Similarly, a model that is too small may underfit the data, while a model that is too large may overfit the data and require more computational resources. There are several approaches to selecting hyperparameters. One common approach is to use a grid search, where multiple combinations of hyperparameters are trained and evaluated in order to find the optimal combination. This can be time-consuming, however, and may not always lead to the best results. Another approach is to use a random search, where random combinations of hyperparameters are sampled and trained in order to find the optimal combination. This can be more efficient than a grid search, but still may not guarantee the best results. Another approach is to use a Bayesian optimization method, which uses a probabilistic model to guide the search for the optimal hyperparameters. This can be more efficient than random or grid search, as it takes into account the results of previous trials in order to guide the search in the most promising directions.

In conclusion, deep learning hyperparameters play a crucial role in the performance of a deep learning model. Careful selection of these hyperparameters can significantly improve the model's accuracy and generalization ability.

2.13 Data Augmentation

Data augmentation is an effective technique for improving the performance of deep learning models. By generating additional training data through various transformations of the original data, data augmentation can help to reduce overfitting and improve the generalization ability of the model. It can also be used to un-bias the learning towards specific shapes, as the model is exposed to a wider range of variations during training. There are two types of data augmentation techniques: soft and hard. Soft data augmentation involves relatively small transformations of the data, such as rotation, horizontal flip, vertical flip, random cropping, and random zooming. Hard data augmentation involves more significant transformations, such as using GANs or multiple standard data augmentation techniques on a given input dataset or batch. There are several libraries available that provide data augmentation functionality, such as Magenta, Kornia, and AI AugLy. These methods can be deployed dynamically during training to increase the training data and to un-bias the learning towards specific shapes. Data augmentation is a useful tool for improving the performance of deep learning models, and it is often used in conjunction with other techniques such as hyperparameter optimization and regularization to further improve model performance.

Chapter 3
Approach and Methodologies

The approach and methodology for developing the hybrid computer vision algorithm will be explained in Chapter 3.

3.1 Introduction

Inspired by the recent advancement of the vision transformer model [50], we present a hybrid architecture called Scopeformer, that merges the advantages of multiple convolutional neural networks (CNNs) and vision transformers (ViT). The CNNs are used for feature extraction, while the ViT encoders are responsible for differentially extracting weights from the global feature map. These weights represent the inter-feature correlations learned by the model with relevance for the hemorrhage classification problem. Our work hypothesizes that using feature maps obtained from well-designed CNNs can enhance the information processed by ViT and the input resolution it focuses on. The use of CNNs for feature extraction allows us to take advantage of the scale, shift, and distortion invariances present in these features, while the ViT allows us to maintain the dynamic attention and global context capabilities of transformers. Additionally, we propose that generating features from a single input image through various CNNs results in a more comprehensive set of features with a higher resolution.

In order to evaluate the effectiveness of our proposed architecture, we will be tackling the RSNA cerebral hemorrhage classification challenge, which is a widely recognized benchmark dataset in the field of medical image analysis. Our goal is to develop an accurate and reliable model that is able to accurately classify different types of intracranial hemorrhages in CT scans. Based on our initial results, we will then proceed to construct an Efficient Scopeformer architecture that leverages the training difficulties and model failures of the proposed model to make it more memory efficient and scalable. Overall, our goal is to create a powerful tool for the accurate and timely diagnosis of intracranial hemorrhage, which has the potential to greatly improve patient outcomes and reduce mortality rates.

3.2 Methodology

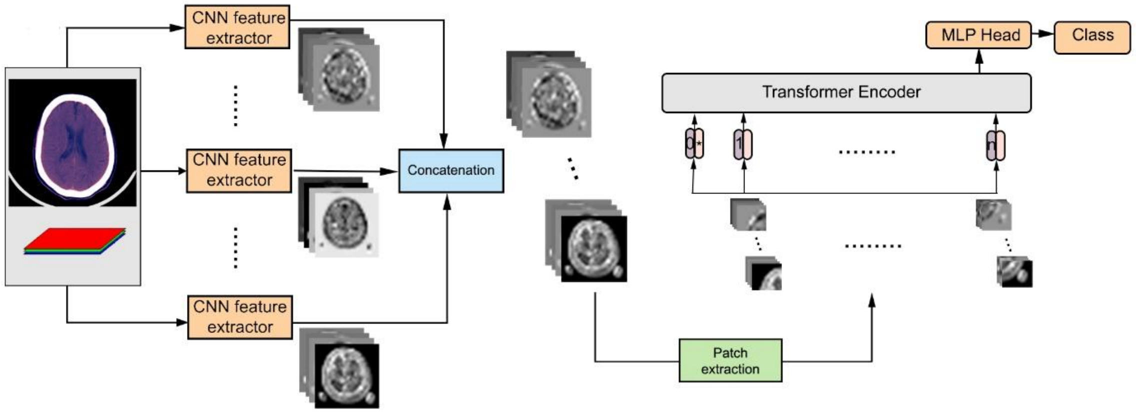
The goal of this project is to improve the performance of the vision transformer (ViT) model in the task of detecting cerebral hemorrhages from computed tomography (CT) scans. To achieve this, we propose the use of a feature generator backbone built from multiple convolutional neural networks (CNNs) that are pretrained on predefined architectures. The ViT model has shown to be effective in many computer vision applications, but it requires large amounts of data and complex architecture in order to achieve its full potential. To address this, we aim to utilize pretrained feature extraction modules and incorporate dimensionality reduction within the proposed model in order to improve the training of the ViT model. Additionally, we focus on making the architecture more scalable and efficient by reducing the number of trainable parameters through the use of pretraining methods and dimensionality reduction, and by carefully engineering the model and training paradigms to increase feature richness. We also aim to address the high computational and memory requirements of the ViT model by optimizing these resources and utilizing data augmentation and synthetic data generation techniques.

3.3 Scopeformer Model Architecture

We present our hybrid n-CNN-ViT model in figure 8. The model is composed of n number of CNN models stacked to build the feature-extractor backbone.

Figure 12

Overview of the Proposed n-CNN-ViT Architecture



Note. The model is composed of two main stages: Feature map generation and global attention encoding for the MLP head classification.

We refer to the n-CNN-ViT model as ``Scopeformer", derived from the ``Transformer" (-former) and the word "Scope-" for the *selective feature extraction backbone* generated from the convolution blocks with deep receptive fields. The

Scopeformer model brings significant advancement in ViTs and CNNs. The main difference between a Scopeformer model and ViT resides in employing high-level features with more semantic information as input to the Transformer encoder, as opposed to the originally proposed ViT model which inputs raw natural images in the form of small patches. The Scopeformer model takes global convolutional feature maps in the form of smaller but deeper patch sizes. The ViT patch extraction method consists of dividing a natural image into patches along the height and width, then flattening every patch and joining all channels into a single 1-D token. Similarly, we pixel-wise divide the feature map along the height and width of the features into $p \times p$ patches, where p is the feature-patch size (with $p = 1$ for all the experiments in this work).

The input to the model consists of a tensor with a dimension of $\mathrm{H} \times \mathrm{W} \times \mathrm{C}$, where H represents the height, W represents the width, and C is the number of concatenated channels derived from the RSNA DICOM files. The model executes a concurrent forward pass of the input images through different CNN architectures and stores the output features f. These features are concatenated along the channel axis. The resultant global feature map has a dimension of $\mathrm{h} \times \mathrm{w} \times \mathrm{c}$, where h represents the features height, w represents the features width, and c is the total number of features with $c = n \times f$.

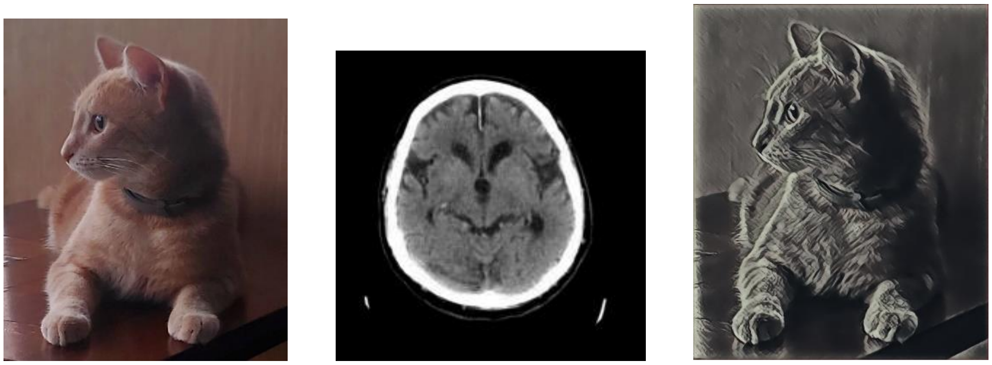
The first Scopeformer architecture uses Xception CNNs [101] and several ViT layers. The Xception model is comprised of several Inception modules composed of depth-wise and point-wise convolutions. In our Scopeformer model, we stack n differently pre-trained Xception models Xception [101] in the feature extraction backbone and freeze updates on their weights during training. We use the last inception layers embedded within the Xception models as features generators.

The first compartment of the Scopeformer model is the CNN backbone block. The main function of this block is to generate high-level features extracted from the inputted generated 3-channel grayscale stacked medical images. The feature generator block extracts set of specific features by the mean of multiple Xception [101] CNN architectures concatenated along the z-axis of the features with an assertion that the height and width across all set of features must be identical. These features are specific to each Xception CNN and the specific methods we used to train its weights. The choice of a single type of architecture was made initially to distinguish the effects of varying the pretraining techniques across the Xception architectures prior to training the Scopeformer model in an end-to-end fashion. A patch extraction layer is introduced to reshape the 3-D features into the proper shape of "N" unidimensional patches for the Vision Transformer block input with conserving the order of which we conduct the tokenization. The hard-split patches are flattened to form a sequence of vectors and packed together along with a learnable class embedding to form the input matrix. The resultant matrix of input vectors is mapped through a trainable projection to a constant latent set of vectors. The pointwise addition of the set of positional embedding vectors to the resultant embedding matrix form the input to the vision transformer block. The ImageNet pre-trained Xception CNNs, present high-level features to the ViT block. To this end, we consider that the primary role of the ViT block is to extract *correlations* from depth-wise patches. The global feature map can be generated using one or more Xception blocks stacked in the same Scopeformer as depicted in figure 8. Our initial experiments consider stacking raw features from CNN blocks without any further processing. The positional embedding matrix is trainable where each row represents the position of the relative vector to which we are pointwise adding the input. The second compartment in the Scopeformer architecture is the vision transformer (ViT) block. In the original ViT paper, the authors extract patches out of natural images such as ImageNet and extract the global context local correlations among different tokens of the image. However, we argue that inputting the original image to ViT results in limited localization abilities, or a loss of the feature resolution which is due to the limited low-level details the ViT block must be trained to extract. To compensate for that, we use here a hybrid CNN-Transformer architecture. This is powerful cause we leverage the detailed high-resolution spatial information from the CNN features and the global context encoded by Transformers. To this end, the role of training, or pretraining the CNN is that CNNs will be trained to encode images into high-level feature representation. And then, patch embedding is applied to 1 by 1 patches extracted from the CNN feature map instead of the raw images. The Transformer encodes tokenized image patches from the CNN feature map as the input sequence, and thus extracts global contexts [113] of the CT scans of the brain and the existence of the several types of hemorrhaging. It is applied either directly to raw images or to a given $\boldsymbol{n}$ number of feature maps extracted from the latest Xception Add layers and concatenated to a single feature map. We adopted the base ViT variant with 12 encoder layers and a latent vector dimension of 1456. In our experiments, we used the RSNA intracranial hemorrhage dataset [105] by generating $224 \times 224 \times 3$ images from the DICOM files [95]. The input image to each feature extractor is $224 \times 224 \times 3$, and the output dimension is $7 \times 7 \times 1024$. For multiple CNNs, the size of the input vector will be $7 \times 7 \times (n * 1024)$. A smaller version of n-CNN-ViT models was introduced to reduce the computational complexity of the ViT input, where we use a 1×1 CNN filter after the Xception Add layer to reduce the dimension from 1024 to 128.

In our formulation, we tend to diversify the pre-training methods of every Xception CNN. This allows for generating different features specific to each architecture. In the first phase of model training, we load the ImageNet pre-trained weights in all CNNs using Keras API [114]. In the second phase of training, Xception CNNs are trained to perform different classification tasks, including RSNA hemorrhage dataset to perform classification. We used hard data augmentation on one of the CNNs and soft data augmentation on the others. We applied style transfer [115] on ImageNet dataset to induce a grayscale brain-like image shape bias as depicted in the figure 9. The output dataset was used to pre-train the third CNN. In our experiments, we tested several combinations of the pre-trained CNNs within the Scopeformer architecture.

Figure 13

Style Transfer Method Applied on ImageNet Dataset. (a) Content Image, (b) Style Image, and (c) Output Image



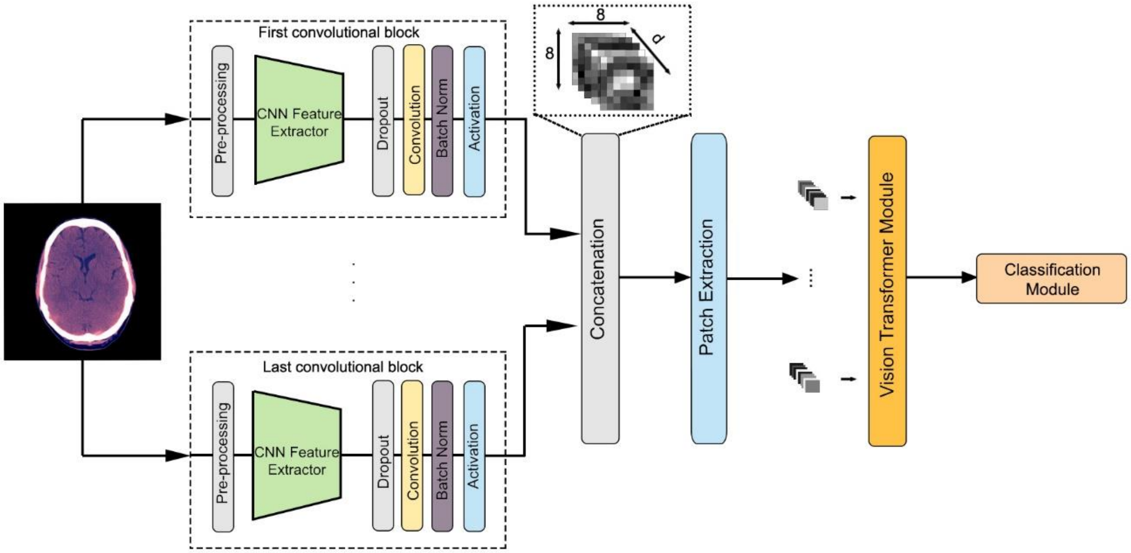
3.4 Efficient Scopeformer Model Architecture

Motivated by the performance of these two models, we proposed in our earlier work [116], a hybrid architecture consisting of multiple Xception CNN models [101] for feature extraction and several vision transformer encoders for differentially extracting significance weights of the feature map relevant to classification. Results showed that the classification accuracy is proportional to the number of Xception models and the variety of the pretraining methods used to train the CNN architectures. We propose enhancing our earlier n-CNN-ViT Scopeformer model by employing a more efficient version of the ViT and improved feature extraction method. We modified our Scopeformer architecture by introducing several changes in the feature extractor CNNs and the ViT. There are four modules as shown in Fig. 3. After extensively testing the Scopeformer model, we formulated and included several innovations in the feature extractor CNNs and the ViT blocks. We define four modules as presented in figure 9. The first module is the

Scopeformer Backbone and represents the stack of multiple CNNs contributing the global feature map. The second module is designed for patch extraction (from the CNN features) to generate ViT tokens. The third module consists of the ViT pipeline. Finally, the fourth module represents the classification head. We discuss these modules in the following sections.

Figure 14

A Schematic Layout of the Scopeformer Architecture



Note. The proposed model is composed of four main modules: (1) Scopeformer Backbone, (2) patch extraction, (3) vision Transformer (ViT) encoder, and (4) classification head. A single input image is fed to several CNN models to extract a variety of features and construct feature maps. These feature maps are processed by the patch extraction module and vectorized. The vectors form the input to the Transformer encoder and the model output is taken from the classification module.

3.4.1 Module 1: Scopeformer Backbone

The proposed Efficient Scopeformer uses a variety of CNNs to build the feature extraction block. The backbone CNNs include ImageNet-pretrained ResNet 152 V2, EfficientNet B5 [78], DensNet 201 [117], and Xception [101]. The features generated by each CNN are concatenated along the channel axis to form a global feature map. However, constructing such a feature map requires that the individual feature maps generated by each CNN to have identical height and width. We propose augmenting each CNN with a single trainable 1 × 1 convolutional layer that projects the features to an appropriate space. The input to the Efficient Scopeformer consists of a tensor with a dimension of H ×W ×3, where H represents the height, W represents the width, and 3 is the number of channels. The image is concurrently fed to four CNNs to generate high-level feature maps. The channel dimension of all four feature maps will be reduced using 1×1 convolution layer to $8 \times 8 \times \frac{d}{4}$, where d is the size of the global feature map.

3.4.2 Module 2: Patch Extraction

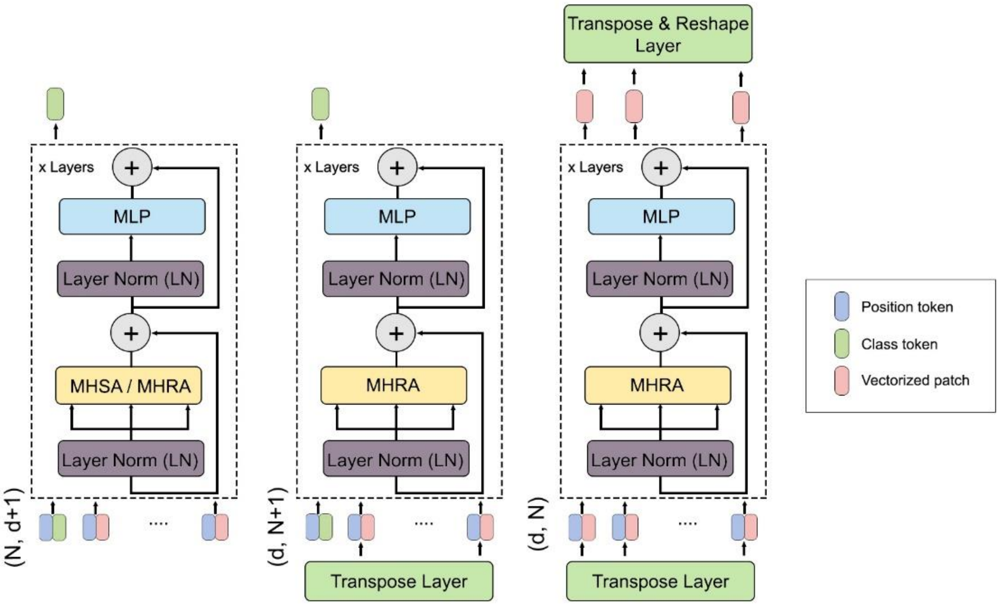
The input dimension of the second module depends on the size of the *global feature map* set by the first module. In our experiments, the resultant *global feature map* is a 3D tensor with a shape of $8 \times 8 \times d$. The patch extraction module splits the features across the height and width in a channel-wise manner, and extracts $N = \frac{8\times 8}{p^2}$ d-dimensional vectors. We set the patch size to 1×1, and get $N = 64$ tokens representing one local pixel position of features across all the d features. The dimension d is controlled by the projection method used in the previous module and represents a bottleneck of the architecture. Every patch contains semantic information of the local pixel position across all the generated features from the four CNNs. The resultant sequence of flattened patches $X_p \in \mathbb{R}^{64 \times d}$ is then used as the input set for the ViT block.

3.4.3 Module 3: Scopeformer ViT

We evaluated three different ViT configurations for the proposed architecture as depicted in Figure 2. These configurations include (1) Deep Scopeformer, (2) deep Scopeformer TR (Transpose), and (3) Efficient Scopeformer.

Figure 15

ViT Scopeformer Configurations



Note. (Left) Baseline Scopeformer Configuration. The first configuration is a ViT block with an input of vectorized patches extracted from the CNNs features. **(Center) Deep Scopeformer TR Configuration:** The second configuration introduces a transpose layer to transform the channel-wise patches into feature-wise patches. **(Right) Efficient Scopeformer Configuration:** The third configuration dismisses the token class and uses all the feature tokens as input. The output of the third block will be transposed to retrieve back the dimension of the CNN features, which we feed to the classification module.

Baseline Scopeformer Configuration

In this configuration, we feed a set of vectors generated by patch extraction layer to ViT encoders. We used trainable position encoding vectors coupled with vectorized patches and a trainable class (CLS) token. The dimension of the input to ViT encoder block is $Y \in RN \times d + 1$. We used two self-attention variants. The first one is referred to as multi-head self- attention (MHSA) [50] and the second variant as the multi-head re-attention (MHRA) [89]. The key difference resides in the introduction of a trainable transformation matrix. These variants are given by:

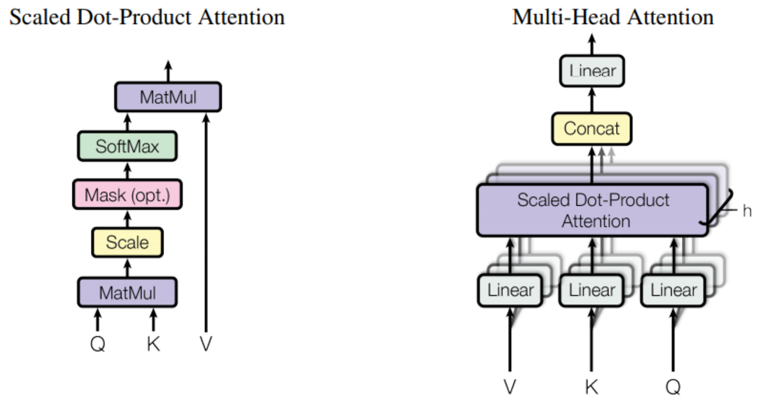
$$MHSA(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{dk}}\right)V$$

$$MHRA(Q, K, V) = Norm\left(M^T\left(Softmax\left(\frac{QK^T}{\sqrt{dk}}\right)\right)\right)V$$

where $M \in \mathbb{R}^{h \times h}$ is a learnable transformation matrix, and h is the number of self-attention heads.

Figure 16

Scaled Dot-Product Attention and Multi-Head Attention



With $M \in Rh \times h$ is a learnable transformation matrix, and h is the number of self-attention heads.

Deep Scopeformer TR Configuration

The second Scopeformer ViT configuration applies a transpose operation to the set of vectors produced by the patch extraction layer. The output of the transpose layer is summed up with the position encoded vectors and concatenated with the CLS token. The dimension of the resultant set of vectors is $\mathrm{Y}_{T} \in \mathbb{R}^{\mathrm{d} \times N+1}$. We used only MHRA self-attention variant (Eq. 2) in our experiments.

Efficient Scopeformer Configuration

The third Scopeformer module discards the CLS notion used in previous configurations. In these settings, we use all the features generated by ViT encoders for classification. As such, the dimension of the input and output of ViT encoders remain identical and equals to

$\mathrm{Y}_T \in \mathbb{R}^{\mathrm{d}\times N}$. We use a Transpose and Reshape layer at the ViT output to get the appropriate dimension for the feature map. We use MHRA self-attention variant to compute self-attention.

3.4.4 Module 4: Classification Module

The classification module in baseline and the deep Scopeformer TR configurations receives a single CLS token. The output of this token is turned into a prediction using a multi-layer perceptron (MLP) with a sigmoid activation function and a single hidden layer. In the efficient Scopeformer configuration, the classification module receives a set of reshaped features $\mathrm{x}_y \in \mathbb{R}^{8\times 8\times d}$. The classification module applies a 2D average pooling layer, followed by a flatten layer. Finally, the inference of the class is done via a dense layer with a sigmoid activation function.

3.5 Datasets

The RSNA dataset was collected by Adam E. et al. [105] from multiple scanner types used in different institutions around the world. The dataset is considered the current largest dataset publicly available aimed to capture complex real-world details of the hemorrhage sub-types. The Radiological Society of North America (RSNA) dataset was released in the 2019 Intracranial Hemorrhage (ICH) detection challenge hosted by the Kaggle platform. The dataset contains 870,301 annotated 16-bit grayscale computer tomography (CT) scans saved in the DICOM format. Individual images consist of pixels that have a range of 0 to 216 with a resolution of 256^2, referred to as Hounsfield Units (HU). HU represents the density of the scanned matter. Trained physicians categorized each CT slice with one or more types of the brain hemorrhage. Five different forms of hemorrhages are to be identified in this competition, with an additional class representing the presence of any hemorrhage type in the provided slice. These classes were labeled as: Epidural hemorrhage (EDH), Intraparenchymal hemorrhage (IPH), Intraventricular hemorrhage (IVH), subarachnoid hemorrhage (SAH), and Subdural hemorrhage (SDH).

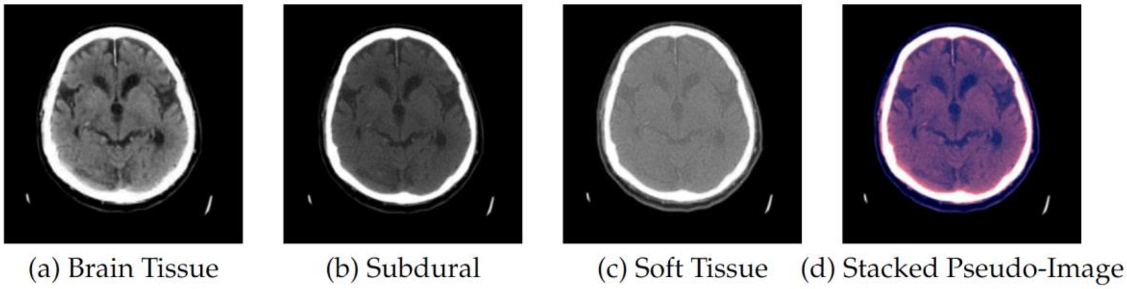
Attenuation HU values are indicative for the content of the scan (Broder and Preston, 2011). For instance, bones have an attenuation value ranging between 250 and 1000, and fat and muscle have attenuation values (AV) ranging between 50 and 100. Applying HU windows on a CT slice yields an 8-bit grayscale image. We use three windows of HU as channels in the input of the Scopeformer model.

3.6 Data Preprocessing

Individual images consist of pixels that have a range of 0 to 2^{16} with a resolution of 256^{16}, referred to as Hounsfield Units (HU). HU represents the density of the scanned matter. The values of attenuation in Hounsfield Units (HU) can provide information about the content of a CT scan. For example, bones typically have an attenuation value between 250 and 1000 HU, while fat and muscle have values between 50 and 100 HU. These values can be used to create an 8-bit grayscale image through the application of HU windows. In this study, we use three HU windows as input channels for the Scopeformer model: a brain window with attenuation values between 40 and 80 HU, a subdural window with values between 80 and 200 HU, and a soft tissue window with values between 80 and 200 HU. Single slices of each scan in the dataset were pre-processed individually. Hounsfield unit (HU) windowing is an effective practice for manual stroke detection [118]. The RSNA CT scan DICOM files provide tags in the metadata about Hounsfield ranges used during registration of the CT scan. We use these tags to ensure standardization of the ranges across the dataset prior to applying HU windowing [119]. We use three windows of HU as channels in the input of the Scopeformer model, as depicted in figure 12. Our settings for HU windows were brain $AV \in [40,80]$ HU, subdural window $AV \in [80, 200]$ HU, and soft tissue window $AV \in [40,380]$ HU, similarly to Burduja et al. [95]

Figure 17

Hounsfield Unit CT Slice Conversion and the Corresponding Stacked 3-Channel Image



Note. During CT scan preprocessing, each slice in each scan was separated into three different windows based on HU thresholds. The three windows were then combined into channels and saved a single RGB image.

We generate 3-channel images by combining three defined windows from the DICOM Hounsfield unit (HU). The output dimension of the images was set to (224,224,3). We split the dataset into 90% for training and 10% for validation.

3.7 Experiments

Details about the various Scopeformer hyperparameter configurations and architectures are presented in table 1.

Table 1

Various Configurations – Hyperparameters and Learnable Parameters

Model	CNN Blocks	Layers	Feature size	MLP	Heads	Parameters
Scopeformer (S)	4	8	516	3072	12	34 M
Scopeformer (B)	4	8	512	4096	16	42
Scopeformer (M)	4	8	512	5120	16	43
Scopeformer (L)/4	4	4	1024	4096	16	51
Scopeformer (L)/8	4	8	1024	4096	16	102
Scopeformer (L)/16	4	16	1024	4096	16	203
Deep Scopeformer (L)/8	4	8	1024	4096	16	102
Deep Scopeformer TR (L)/8	3	8	384	4096	16	6
Efficient Scopeformer	3	8	384	4096	16	6
Scopeformer	3	12	3072	3072	8	755
Scaled Scopeformer	4	8	4096	4096	16	870

We present the different proposed Scopeformer variations and details about the number of convolution models used in the feature extraction backbone, number of ViT layers, the global feature map size, the MLP dimension and the number of heads in each ViT block, and the total number of parameters.

We compare our Efficient Scopeformer implementation to our initial implementation of the model and propose lower trainable parameter space given the configuration hyperparameters. Our experiments comprise of four main parts.

In the first set of experiments, we evaluate the size effect of various variants of Scopeformer on the classification accuracy, where four variants are evaluated: small (S), base (B), medium (M), and large (L). We keep the number of ViT layers fixed (equals to 8) and increase the complexity of the model by configuring the MLP size residing in the ViT blocks for S, B, and M variants, and increasing the feature size for the L variant. The number of trainable parameters drastically increase from the smallest (S) to the largest (L) variants.

In the second set of experiments, we investigate the effect of the number of ViT encoder blocks on the model performance. Based on preliminary results conducted in the first set of experiments, we conduct our ablation study on the large Scopeformer variant (L) with a feature size of 1024 and an MLP dimension of 4096. We consider three experiments where we gradually stack in an end-to-end fashion 4, 8, and 16 ViT encoders, forming three models named Scopeformer (L)/4, Scopeformer (L)/8, and Scopeformer (L)/16 respectively. Given the largest model parameters reside within the ViT architecture, the total number of trainable parameters is linearly scaled to the number of ViT blocks we use.

The third set of experiments examines the transition from the originally proposed ViT model [50], to a different version called DeepViT [89]. We test this configuration on highest performing model from the previous two sets of experiments: Scopeformer (L)/8 with a global feature map size of 1024, 8 layers of ViT encoders, and an MLP dimension of 4096. The model version, entitled as Deep Scopeformer (L)/8, has slightly higher number of trainable parameters.

The final experiment introduces three different ViT configurations to our Scopeformer architecture as depicted in figure 10. We add these configurations to the highest performing model from the previous three parts of the study: Deep Scopeformer (L)/8 with a global feature map size of 1024, 8 layers of ViT encoders, and an MLP dimension of 4096. We introduce and compare a set of three Scopeformer configurations; Baseline Scopeformer configuration, Deep Scopeformer-TR configuration, and Efficient Scopeformer configuration.

3.8 Pre-Training Efficient Scopeformer

In all the experiments, we initially pretrained the Scopeformer model using ImageNet-1k dataset [120]. Later, we train all models using the RSNA dataset [105]. In the first module (convolutional backbone), we freeze $\approx 70\%$ of the layer weights in each CNN and keep top $\approx 30\%$ trainable along with the newly introduced 1×1 convolution layer. In our last experiment using Efficient Scopeformer model, we pretrained the backbone neural network on the RSNA dataset for hemorrhage classification for 150 epochs on top of the defaulted pretraining on ImageNet-1k. In this experiment, denoted as Efficient Scopeformer (p), we freeze weights of the feature extraction block during training.

3.9 Loss Function

Following guidelines from the RSNA Intracranial Hemorrhage Challenge (ICH), we adopted a weighted version of the *multi-label logarithmic loss* function for our model training. The weighting was introduced to amplify the importance of classifying the first class representing all types of hemorrhages, with a coefficient of 2, on the expense of the rest of the classes which have coefficients of 1. The evaluation of the loss value with respect to a single instance represents the weighted average over all the binary losses computed on each class individually. The ICH represents a multi-label classification problem, i.e., the input image can be classified into multiple classes, using binary labeling for each class to indicate its presence or absence. In our formulation, we applied multi-label hot encoding on the dataset to assign a binary value on each class for every CT slice. The *multi-label logarithmic loss* function is defined as follows:

$$L_{multi-BCE}\,(y,\hat{y}) = -\sum_{n=1}^{6} \alpha_n\,(y_n log\hat{y}_n + (1 - y_n)\,log(1 - \hat{y}_n))$$

where α_n represents the coefficient of the target classes, $y_n$represents the ground-truth of each class n, and $\hat{y}_n$ is the corresponding predicted probabilities.

3.10 Evaluation Metric

The official model evaluation metric in the RSNA IHC was the *weighted accuracy*. We evaluate the overall performance of the models based on three metrics, (1) the classification accuracy on the RSNA dataset, (2) visually evaluation of the global feature richness of the embedding layer generated by convolution backbone, and (3) the ratio of the model size function to the total number of trainable parameters.

Chapter 4

Results and Discussions

In this chapter, we present the results of computational experiments using various configurations of the Scopeformer model. The performance of the algorithm is evaluated on the RSNA dataset using accuracy plots and tables, as well as evaluating the multi-labeled log loss function. We examine the effect of model complexity, such as the number of parameters and size of the feature maps in the bottleneck, on the model's performance. The proposed n-CNN-ViT structure of the algorithm enables us to analyze the impact of adding different types and sizes of convolutional neural networks while maintaining the end-to-end training paradigm. To better understand the factors influencing classification decisions and identify potential biases in each compartment of the Scopeformer model, we conducted several interpretability studies on all the studied configurations. These studies helped us make informed choices of model parameters and convolutional neural network sizes and types to use in the backbone, which we expect will further improve the performance of our proposed model. Additionally, we applied style transfer and progressive pretraining methods and evaluated the generated features from various convolutional neural networks. The Efficient Scopeformer model was developed based on our previous publication [116], on which we made significant changes to the architecture to make it more efficient and scalable.

To assess potential improvements, we addressed the large trainable parameters, Lambda machine memory readability issues, and training time per epoch. We evaluated the overall performance of the models based on three metrics, (1) the classification accuracy on the RSNA dataset, (2) visually evaluation of the global feature richness of the embedding layer generated by convolution backbone, and (3) the ratio of the model size function to the total number of trainable parameters.

4.1 The Effect of Backbone Model Size and Pretraining Techniques

We gradually stack ”*n*” various pretrained Xception models in the feature extraction backbone. We freeze all these architectures in the backbone to prevent updates on their weights during training. We pretrained the CNN models on diversified pretraining schemes, including ImageNet-1k natural image dataset (I) and the generated style transfer-base dataset (S). Table 2 compares different models and the corresponding performances on the hemorrhage classification task. While the n-CNN-ViT models were trained on the convolution features generated by the convolution backbone, the ViT model was trained on raw dataset. The input dimension of the ViT block represents the full resolution image or the set of features prior to splitting into patches. Results show that extracting features using convolution models to train the ViT model is a better alternative to the raw dataset. The Scopeformer model exploits the pretraining for generating high level features useful for the ViT architecture. The use of the CNNs leverages the need for high data regimes since the ViT model is used to fit these high-level features and extract semantic correlations instead of learning the spatial features in training. Furthermore, results show that the classification accuracy is proportional to the number of CNN models used in the

Scopeformer training, i.e., as we stack feature extraction architectures in the backbone of the model, we get higher performances on hemorrhage classification. In our study, we found that using a combination of different pretraining methods can significantly improve the performance on a target task. By leveraging the specific features and patterns learned by each pretrained model, we were able to achieve better results compared to using only one type of model. We discovered that a CNN trained on ImageNet is proficient at recognizing common patterns and features in images, while a CNN that has been style transferred using ImageNet is better at recognizing more specialized patterns and features. By combining these models, we were able to take advantage of both their general and specialized capabilities. Additionally, using a variety of pretraining methods can reduce overfitting and lead to more robust and generalizable models. We also found that ImageNet-trained CNNs tend to be biased towards texture, as presented by Geirhos et al. [121], and increasing shape bias can improve accuracy and robustness. By selectively varying the pretraining methods for each CNN architecture, we were able to further boost performance by enriching the features through different sets of weights and dynamics. Our results demonstrate the importance of carefully selecting and combining different pretraining methods for optimal performance.

Table 2

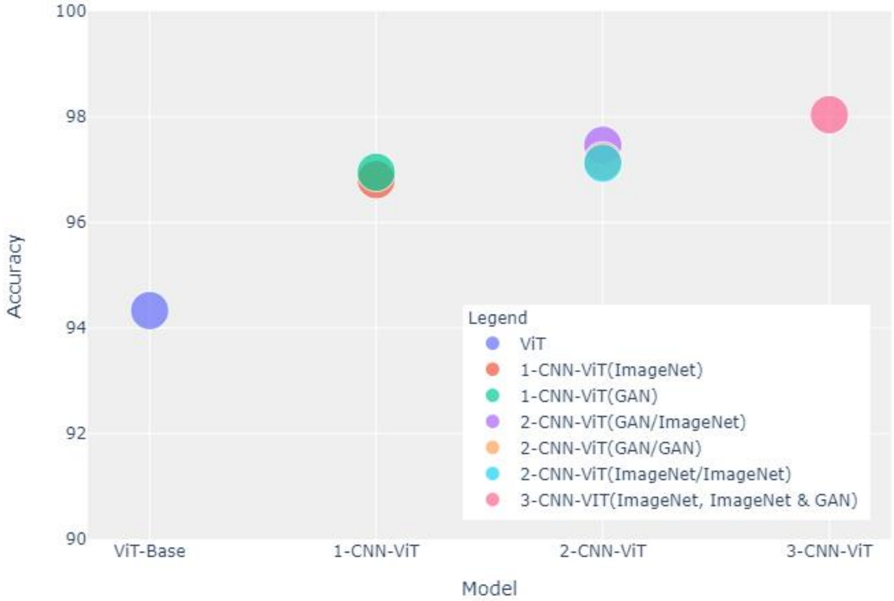
Classification Performance of ViT Based Models on the RSNA Validation Dataset

Model	ViT input dimension	Validation accuracy	Loss
ViT	256×256×3	94.33%	0.18220
1-CNN-ViT (S)	7×7×1024	96.95%	0.08272
2-CNN-ViT (I-I)	7×7×2048	97.22%	0.07984
2-CNN-ViT (S-S)	7×7×2048	97.26%	0.07934
2-CNN-ViT (I-S)	7×7×2048	97.46%	0.07754
3-CNN-ViT (I-I-S)	7×7×3072	98.04%	0.07050

In our study, we investigated the effect of feature map size on the performance of a Vision Transformer (ViT) model. We found that increasing the feature map size of the CNN in the input to the ViT model significantly improved the model's performance. Our hypothesis is that this improvement is due to the ViT's ability to extract increased semantic correlations from the larger feature maps. The ViT block is designed to identify correlations between the input tokens and having a larger number of features appears to enhance the ViT's ability to extract semantic meaning from the data. Our results suggest that the ViT is able to leverage the additional information provided by the larger feature maps to better understand the input data and make more accurate predictions. These findings highlight the importance of carefully considering the size of the feature maps in the input to the ViT model, as it can have a significant impact on the model's performance.

Figure 18

Performance Versus N-CNN-ViT Variants; Pure ViT, 1-CNN-ViT, 2-CNN-ViT and 3-CNN-ViT, and Pretraining Modes; ImageNet and Data Generated Using GAN



Note. Models with multiple CNNs and different pretraining modes perform better.

To conclude, our study found that diversifying the inductive biases of the Scopeformer model and increasing the feature map size of the CNN input both have the potential to significantly improve the performance of the model. By using a combination of differently pretrained CNN architectures, we were able to generate a wider range of feature maps, which contributed to a richer representation of the data. Similarly, increasing the size of the feature maps allowed the Scopeformer to extract more meaningful correlations and information from the data. When both of these factors were combined, we saw an even greater improvement in performance. These results suggest that carefully considering the pretraining of the CNN and the size of the feature maps can be key to optimizing the performance of the Scopeformer model.

4.2 The Effect of the Size of Scopeformer

Tables 3 and 4 show the results of experiments performed with different variants of the Scopeformer model. Table 4 depicts different results obtained on individual classes of the S, B, and M models. We propose four sizes of the Scopeformer model; S, B, M, and L, with reduced number of trainable parameters compared to our initial implementation of the Scopeformer model involving several Xception-based CNNs. The key component to the parameters reductions is linked to the trainable 1×1 *convolutional layer* placed after each convolution architecture in the feature extraction backbone prior to concatenation. In this experiment, we gradually increase the model complexity of S, B, and M variants by varying the MLP dimension and the number of self-attention heads within the ViT module as depicted in table 1.

In table 3, we note that the base model outperforms the small and medium variants. However, in Table 4, we observe that the Base model shows better performance on IPH, IVH, and SAH classes, whereas small model shows higher accuracy results on all, epidural and SDH classes. Based on these observations we hypothesize that the improved performance observed on higher MLP dimensions indicates the ability of the model to encompass larger amount of information and extract useful semantics for classification. However, the model shows signs of overfitting when the MLP dimension reaches 5120. Based on these results, we build our large *Scopeformer (L)/8* model by adopting the configuration of the base variant with a global feature dimension d = 1024. The feature size increment resulted in a proportional increment of the model trainable parameters. The large model (L)/8 performed the best among the proposed variants. The improved performance observed on larger ViT sizes while increasing the input feature embedding space indicates a richer information brought by these added features, where the model extracted useful semantics for classification. Increasing the feature space improved some of the classes on the expense of others as evident from Table 4. Training using SAM paradigm were not feasible given the massive training time it has added. Suboptimal results from SAM on some experiments led us to not use it with the proposed Scopeformer model.

Table 3

Performance of the Different Scopeformer Variants

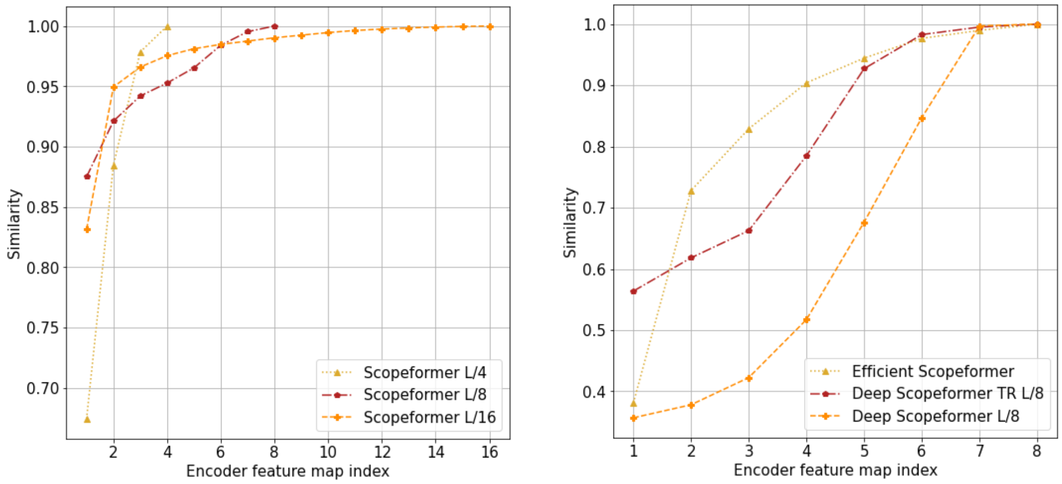
Model	Accuracy	Loss	Recall	Trainable Parameters
Small (S)	93.0%	0.1703	84.95%	34M
Base (B)	93.92%	0.1461	89.29%	42M
Medium (M)	93.88%	0.2285	88.44%	43M
Large (L) / 4	93.12%	0.1378	87.81%	51M
Large (L) / 8	**94.69%**	**0.1197**	**89.33%**	102M
Large (L) / 16	92.57%	0.1395	87.34%	203M

4.3 The Effect of Number of ViT Encoders

We evaluate the effect of the number of ViT encoders on Scopeformer (L)/8 model using 4, 8, and 16 encoders. As presented in Table 1 and Table 3, the number of parameters scales linearly with the number of encoders. We note that using 8 ViT encoders yields better results than shallower model with 4 ViT encoders. However, the deepest model with 16 ViT encoders drastically reduces the model performance. We conclude that increasing the depth of the ViT model does not scale linearly, and that there is a critical number of ViT encoders where the model performs optimally. This behavior may be due to the need for high training data sizes to allow deeper networks to perform better.

Figure 19

Cosine Similarity of the ViT Encoder Feature Maps with Respect to the Last Encoder Feature Map



Note. We observe the increased similarities across ViT encoder features function to the depth of Scopeformer models.

In Figure 14, we plot the cosine similarity between the features generated by each ViT encoder and the last layer of the model. We observe that similarities across features generated by each ViT encoder rapidly increase for all proposed models. These similarities further increase in models with higher numbers of ViT encoders. We believe that the increased similarities among the features of the Scopeformer(L)/16 model may have contributed to the performance decline observed in Table 3. Similarly, reduced similarities among ViT features observed on Scopeformer(L)/4 may explain the observed sub-optimal performance. From these results, we conclude that the cosine similarity can be a good metric for model performance, as reduced or increased similarities may indicate sub-optimal performances of the Scopeformer model. Shallow models presenting reduced similarities may hint to higher performances by stacking more ViT layers, whereas deeper models may require additional data to reduce similarities across ViT features to perform optimally. The results also suggest that there is an optimum number of ViT encoders for Scopeformer model based on the complexity of the dataset and the effectiveness of the convolution backbone networks.

4.4 The Effect of Two Different Self-Attention Variants

The Deep Scopeformer (L)/8 builds on the Scopeformer (L)/8 model by replacing the MHSA layer with a MHRA layer. The additional trainable matrices M adds insignificant number of parameters to the Scopeformer (L)/8 model. In Figure 14 (b), we note substantial dissimilarities among ViT encoders' features for the *Deep Scopeformer (L)/8* model. The result may imply an increased feature richness acquired by the model from the additional inter-correlations of the MHRA heads. This configuration resulted in an accuracy improvement by +1.11% as shown in Table 5.

4.5 ViT Scopeformer Configurations

We address the self-attention computational complexity problem by introducing a transpose layer prior to the ViT module. The attention weights matrix in *Deep Scopeformer (L)/8* has a dimension of 1024^2. In the second and third ViT configurations, the attention weights matrices have dimensions of 65^2 and 64^2respectively. The use of the transpose layer has substantially contributed to the reduction of the number of trainable parameters as indicated in Table 1. This is due to the MHRA quadratic reduction in computation complexity. Additionally, transposing the input sequence effectively preserved the feature content retrieved by the feature extractor module, and conserved the classification performance. Table 5 shows the performance of the three proposed configurations. The proposed *Efficient Scopeformer* variant performed relatively better than the *Deep Scopeformer (L)/8* for a lower trainable parameter space. We speculate that the role of the ViT module in this configuration is to improve the global feature map that was previously optimized by the convolution backbone. The global feature map improvement resides in using attention computations to generate new features characterized by inter-correlations among all features generated by the convolution networks. Our Efficient Transformer module improved the global features map correlations and contributed to better performance.

Table 4

Model Performance on Individual Target Classes

	Accuracy			
	Large	**Medium**	**Base**	**Small**
All	71.34%	60.26%	70.5%	70.83%
Epidural	96.98%	90.18%	95.73%	98.08%
IPH	85.94%	71.10%	87.28%	85.95%
IVH	90.5%	70.73%	91.72%	85.95%
SAH	78.69%	65.49%	78.57%	77.04%
SDH	77.08%	60.78%	74.35%	74.54%

We note that for the model *Efficient Scopeformer (P)* pretraining the convolution block on the target dataset and freezing the entire block during training produces better performance than end-to-end training with around 30% trainable parameters of the Efficient Scopeformer's convolution block. We argue that backbone CNNs and ViTs present different dynamics that require different model training settings.

Table 5

Model Performance for Different Scopeformer Modalities

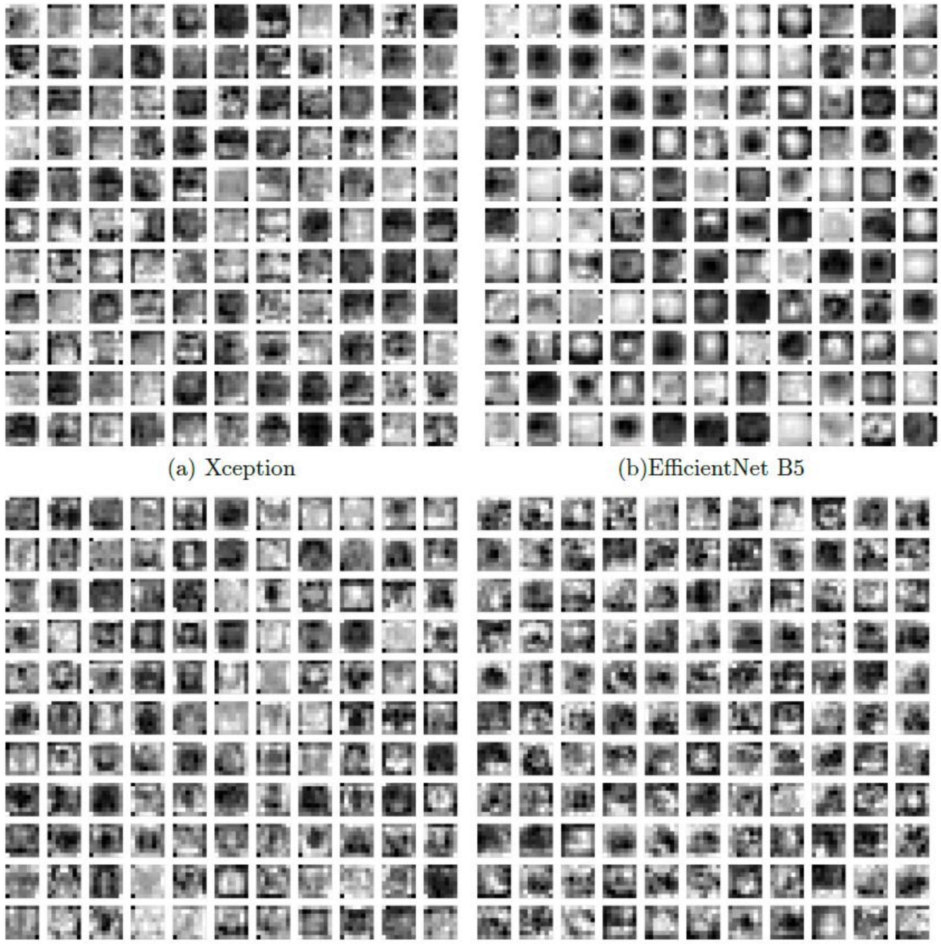
	Accuracy	**Loss**	**Trainable parameters**
Scopeformer (L) / 8	94.69%	0.1197	102M
Deep Scopeformer (L) / 8	96.03%	0.1088	102M
Deep Scopeformer TR (L) / 8	95.40%	0.1176	6M
Efficient Scopeformer	95.77%	0.1160	6M

4.5.1 Global Feature Map

Figures 15, 16, and 17 present convolution features generated by three Scopeformer architectures for an epidural example Scopeformer (L)/8, Deep Scopeformer (L)/8, and Efficient Scopeformer. We observe high variability of the features generated by each CNN architecture. Furthermore, we observe that there is no apparent similarity among the features generated by different CNNs for all Scopeformer variants. Subsequently, the resultant global feature map has low redundancy and higher feature richness. However, among these models, we note that the DenseNet [117] model showed the highest feature redundancy across the observed features. Therefore, we conducted an ablation study on the *Deep Scopeformer TR*, which resulted in removing the DenseNet201 model from the *Efficient Scopeformer* model backbone.

Figure 20

Feature Maps Visualization of an Epidural Type of Hemorrhage Example. Scopeformer (L)/8



(a) Xception

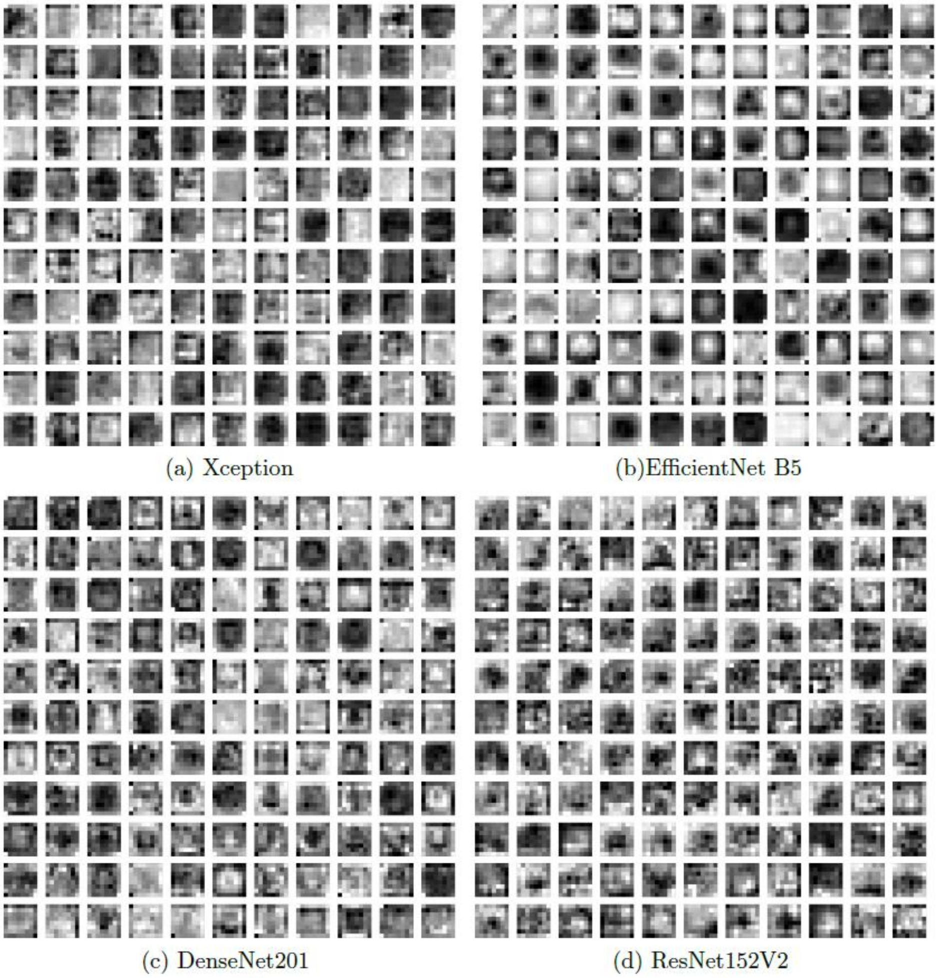
(b)EfficientNet B5

(c) DenseNet201

(d) ResNet152V2

Figure 21

Feature Maps Visualization of an Epidural Type Hemorrhage Example. Deep Scopeformer (L)/8


(a) Xception

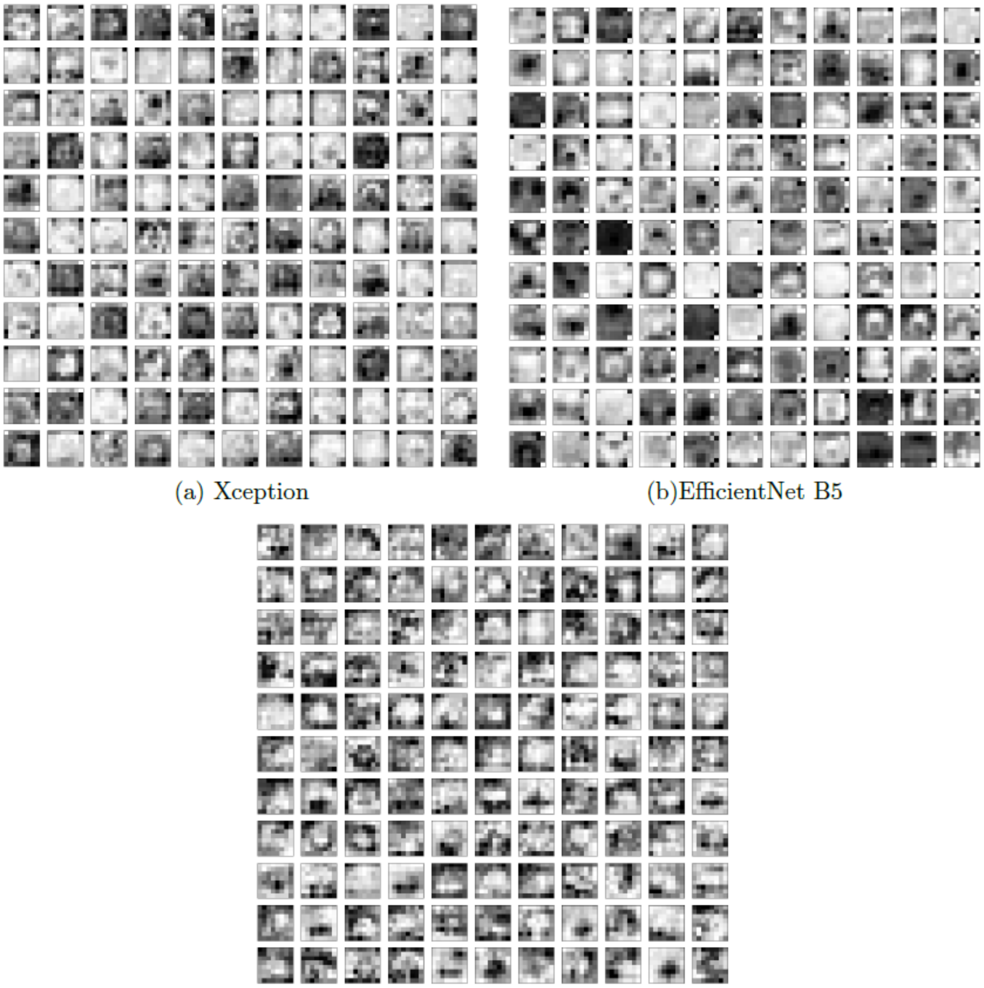
(b)EfficientNet B5

(c) DenseNet201

(d) ResNet152V2

Figure 22

Feature Maps Visualization of an Epidural Type of Hemorrhage Example. Efficient Scopeformer



(a) Xception

(b)EfficientNet B5

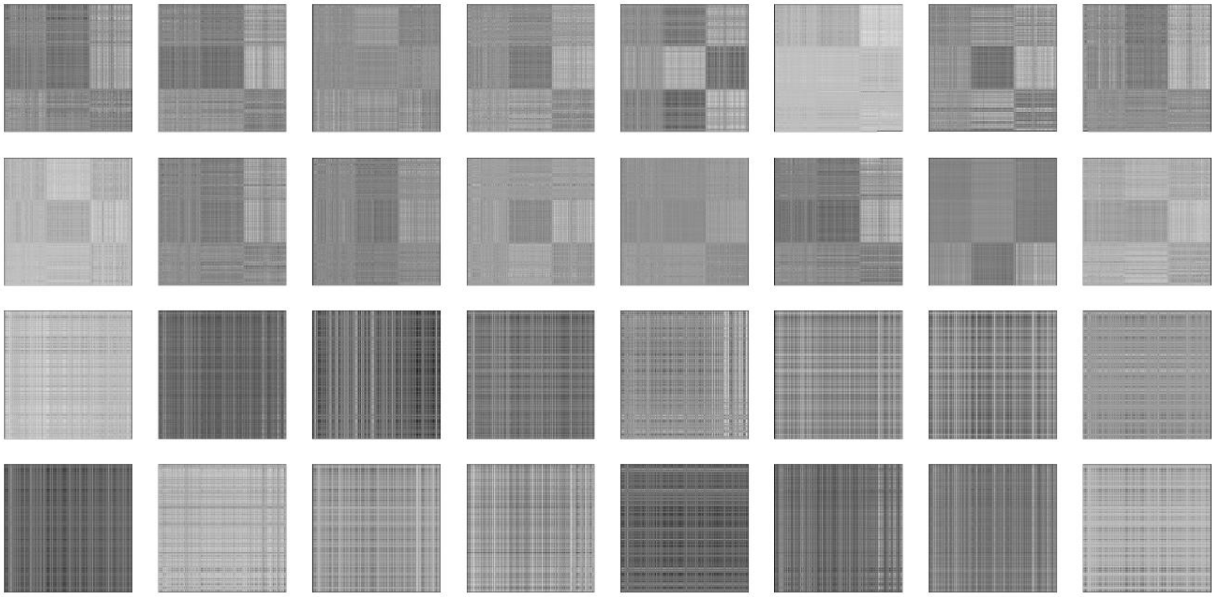
(d) ResNet152V2

4.5.2 Attention Patterns Visualizations

Figure 18 shows the attention patterns visualizations of the 16 MHRA heads concerning the first and last ViT encoders. In the first ViT layer, we observe that the model extracts high correlations among features derived from every CNN architecture. This observation suggests the high similarities among the input features of every CNN model. Each head learns different correlations patterns among the set of features. However, deeper into the model, we observe that the model learns to extract global correlation patterns across all the CNN features. The generated set of features add the information about the relevance of every feature to the rest of features which contributes towards the observed higher performance.

Figure 23

Attention Pattern Visualization of the Efficient Scopeformer Model



Note. The first and second row represent the 16 attention heads of the first encoder layer. The third and fourth row represent the 16 attention heads of the last encoder layer. Each attention map has a dimension of 384 × 384. Deeper in the model, Scopeformer extracts better correlations among all the input features leveraging all the input CNNs.

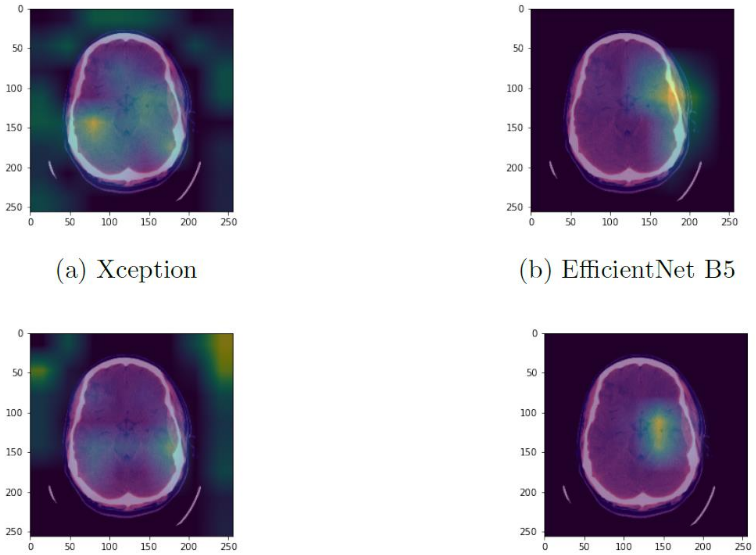
4.5.3 Grad-CAM

In Figure 24, we present a Grad-CAM visualization of an epidural type of hemorrhage example for the Deep Scopeformer (L)/8 model. Upon examining the visualization, we notice that there is high variability in the regions where the model considers important for conducting the classification. In some cases, these regions are clearly related to the presence of an epidural hemorrhage, such as in the area around the brainstem. However, in other cases, the model appears to be considering regions that are less relevant or even unrelated to the task at hand.

One particularly noteworthy aspect of this visualization is the contribution of the DenseNet [117] model to the classification process. In many cases, we observe that this model contributed the least to the classification and was even shown to be mapping to the wrong regions on the image. This suggests that the DenseNet model may not be as effective at extracting relevant features for this particular task, compared to other models that are used in the Deep Scopeformer (L)/8 architecture. It is worth further exploring the reasons behind this behavior and considering alternative approaches to feature extraction that may be more suitable for this task.

Figure 24

Grad-CAM Visualization of an Epidural Type of Hemorrhage Example



(a) Xception

(b) EfficientNet B5

(c) DenseNet201

(d) ResNet152V2

Chapter 5

Conclusion and Future Work

In the final chapter of this thesis, we have reviewed the overall goals and objectives of our research study, which focused on exploring potential improvements to the convolutional-based vision transformer model known as Scopeformer in the realm of classification of CT scans presenting multiple types of hemorrhages. Specifically, we have investigated the effect of model trainable parameters on performance and training efficiency, the impact of using multiple off-the-shelf CNN models on the global feature richness of the architecture, and a feature projection method for reducing the large, redundant feature space. We have also conducted a parametric optimization study to evaluate the size effects on model performance and efficiency and implemented three vision transformer configurations to evaluate the Re-attention module and different patch extraction methods.

Our research has shown that using various CNN architectures in the Scopeformer model can lead to an improvement in the resultant features. This is due to the different sets of weights and dynamics that are learned by each network, which can provide complementary and specialized information. The Re-attention module also demonstrated its ability to enhance performance through the increase in dissimilarities of the vision transformer features. These findings suggest that using a diverse set of CNNs and implementing techniques to optimize the global feature map can be effective strategies for improving the performance of the Scopeformer model.

One such technique that we have explored is our proposed feature-wise patch extraction method, which allowed us to significantly decrease the size of the model while still achieving comparable performance. This is an important consideration for practical applications, as smaller models are generally easier to deploy and have faster inference times. Additionally, the implementation of the Efficient Transformer module resulted in improved correlations within the global feature map, contributing to better overall performance. These results indicate that our proposed method and module have the potential to be useful for optimizing the global feature map in the Scopeformer model and potentially in other models as well.

In conclusion, our research has demonstrated that the Scopeformer architecture has the potential to be generalized to other AI domains that require feature generation and enhancement. We recommend using diversification of features through the use of multiple CNNs, as well as enhancement of feature correlations using deep architectures of the ViT with the proposed multiple configurations. These strategies have the potential to improve the performance and efficiency of the Scopeformer model and could potentially be useful for other tasks as well.

For future work, we recommend continuing to investigate these and other approaches to further optimize the performance and efficiency of the Scopeformer model. In particular, it would be interesting to explore the use of other CNN architectures and techniques for improving the global feature map in order to see if even further improvements can be achieved. Additionally, it would be valuable to evaluate the generalizability of the Scopeformer model to other tasks and datasets in order to determine its full potential and real-world applicability.

References

[1] Alberts, M. J., Latchaw, R. E., & Sacco, R. L. (2007). Acute stroke: diagnosis and management. American Family Physician, 75(6), 855-862.

[2] Elliott Justine and Martin Smith. The acute management of intracerebral hemorrhage: a clinical review, 2012.

[3] NINDS Intracerebral Hemorrhage Information Page. (ninds.nih.gov/Disorders/All-Disorders/Intracerebral-Hemorrhage-Information).

[4] P. Mohajeri, M. A. & Ramezani, A. (2013). Intracerebral hemorrhage: diagnosis and management. Journal of Respiratory and Critical Care Medicine, 2(5), 38-44.

[5] Ko, N. W., Biousse, V., & Newman, N. J. (2014). Intracranial Hemorrhage. In Neurology Secrets (pp. 119-124). Elsevier.

[6] K. S. Kim, "CT Scan," Encyclopedia of Radiology, pp. 1133-1134, 2007.

[7] M. I. Atalay, "Intracranial Hemorrhage: Current Diagnostic and Management Strategies," World Journal of Radiology, vol. 6, no. 7, pp. 577-590, 2014.

[8] S. R. Daugherty, J. M. Grotta, and D. J. Langer, "Diagnosis and Management of Intracranial Hemorrhage," American Family Physician, vol. 96, no. 5, pp. 335-342, 2017.

[9] M. D. Davenport, M. A. Rowley, and M. L. Eichling, "Computed Tomography in the Evaluation of Stroke," American Family Physician, vol. 72, no. 1, pp. 109-115, 2005.

[10] M. C. Brainin, "Management of Acute Ischemic Stroke: A Review," Journal of the American Medical Association, vol. 316, no. 12, pp. 1298-1309, 2016.

[11] Maramattom, B. V., et al. (2015). Intracranial hemorrhage: emergency management. The Lancet Neurology, 14(9), 979-990.

[12] Kapsalaki, E. Z., et al. (2016). Early diagnosis and management of intracranial hemorrhage. Frontiers in neurology, 7, 44.

[13] Kamel, H., et al. (2010). Management of intracranial hemorrhage. Neurosurgery Clinics of North America, 21(1), 61-72.

[14] Hsieh, K. J., et al. (2015). Head computed tomography in the emergency department: a review. Emergency Medicine Practice, 17(11), 1-16.

[15] Kim, D. H., et al. (2012). Role of CT in the management of intracranial hemorrhage. Korean Journal of Radiology, 13(4), 395-405.

[16] Detection and Classification of Brain Hemorrhages: A Review." M. Alghamdi, S. Alshammari, and A. Alsharif. International Journal of Computer Science and Information Security, vol. 16, no. 5, 2018, pp. 127-132.

[17] Automated Detection of Intracranial Hemorrhage in Non-Contrast Head CT Using Deep Learning." S. Kang, J. Kim, J. Lee, and K. Lee. PLOS ONE, vol. 13, no. 10, 2018, doi: 10.1371/journal.pone.0205271.

[18] Interobserver Variability in the Detection of Intracranial Hemorrhage on Computed Tomography." E. Kim, J. Kim, and J. Kim. Korean Journal of Radiology, vol. 18, no. 3, 2017, pp. 347-354.

[19] Accuracy of Emergency Medicine Residents in Identifying Intracranial Hemorrhage on Head Computed Tomography." A. Ahmed, S. P. Smith, and J. A. Williams. The Journal of Emergency Medicine, vol. 52, no. 6, 2017, pp. 767-773.

[20] Assessment of Intracranial Hemorrhage Detection Using Deep Learning Algorithms." Y. Li, J. Zhang, and C. Chen. IEEE Transactions on Medical Imaging, vol. 37, no. 12, 2018, pp. 2746-2754.

[21] Joanna M Wardlaw. Overview of Cochrane thrombolysis meta-analysis, 2001.

[22] Automatic Detection of Intracranial Hemorrhage on Head CT Using Convolutional Neural Networks." H. Kim, J. Kim, J. Lee, and K. Lee. Radiology, vol. 287, no. 2, 2018, pp. 502-510.

[23] Accuracy of a Deep Learning Algorithm for Detection of Intracranial Hemorrhage in Acute Stroke." S. Kim, J. Lee, J. Kim, and K. Lee. Stroke, vol. 49, no. 3, 2018, pp. 677-684.

[24] Evaluation of a Deep Learning Algorithm for Detection of Intracranial Hemorrhage on Noncontrast Head CT." J. Kim, J. Lee, H. Kim, and K. Lee. Radiology, vol. 293, no. 3, 2019, pp. 708-717.

[25] Automated Detection of Intracranial Hemorrhage Using Deep Learning: Comparison with Radiology Reports and Conventional Methods." J. Lee, S. Kim, J. Kim, and K. Lee. Radiology, vol. 290, no. 1, 2019, pp. 85-93.

[26] Deep Learning in Medical Image Analysis." A. Madabhushi and G. R. Mohiuddin. Annual Review of Biomedical Engineering, vol. 20, 2018, pp. 1-37.

[27] Automated Detection of Intracranial Hemorrhages on CT Scans Using Machine Learning." D. J. D. Lee, H. Kim, J. Lee, and K. Lee. Radiology, vol. 277, no. 2, 2015, pp. 440-449.

[28] Machine Learning Approaches for the Detection of Intracranial Hemorrhages on CT Scans." A. A. Saeed, R. K. M. Tariq, and M. F. Shafait. Computer Methods and Programs in Biomedicine, vol. 162, 2018, pp. 105-116.

[29] Deep Learning for Automated Detection of Intracranial Hemorrhage on Head CT Scans." J. Lee, J. Kim, S. Kim, and K. Lee. AJNR American Journal of Neuroradiology, vol. 39, no. 7, 2018, pp. 1359-1365.

[30] A Review of Deep Learning Approaches for the Detection of Intracranial Hemorrhage on Computed Tomography Scans." Y. Li, C. Chen, and J. Zhang. Frontiers in Neuroscience, vol. 13, 2019, doi: 10.3389/fnins.2019.00053.

[31] Deep Learning in Medical Image Analysis." A. G. A. Eslami, M. R. Avendi, and M. S. Akbari. Journal of Medical Signals and Sensors, vol. 8, no. 3, 2018, doi: 10.4103/jmss.JMSS_44_18.

[32] Deep Learning for Automated Detection of Intracranial Hemorrhages: A Systematic Review." J. A. F. de Bruijne, M. M. B. Ciompi, and H. Huisman. Neurosurgery, vol. 84, no. 3, 2019, pp. 474-486.

[33] Intracranial Hemorrhage Detection in Non-Contrast CT Using Deep Learning and Morphological Feature Extraction." S. Liu, Y. Li, J. Zhang, and C. Chen. IEEE Access, vol. 7, 2019, pp. 138620-138628.

[34] Automatic Detection of Intracranial Hemorrhages Using Deep Learning: A Feasibility Study." D. J. D. Lee, H. Kim, J. Lee, and K. Lee. American Journal of Neuroradiology, vol. 39, no. 4, 2018, pp. 744-750.

[35] "Automated Detection of Intracranial Hemorrhages on CT Scans Using Machine Learning." D. J. D. Lee, H. Kim, J. Lee, and K. Lee. Radiology, vol. 277, no. 2, 2015, pp. 440-449.

[36] "Machine Learning Approaches for the Detection of Intracranial Hemorrhages on CT Scans." A. A. Saeed, R. K. M. Tariq, and M. F. Shafait. Computer Methods and Programs in Biomedicine, vol. 162, 2018, pp. 105-116.

[37] "Deep Learning for Automated Detection of Intracranial Hemorrhage on Head CT Scans." J. Lee, J. Kim, S. Kim, and K. Lee. AJNR American Journal of Neuroradiology, vol. 39, no. 7, 2018, pp. 1359-1365.

[38] "Intracranial Hemorrhage Detection Using Deep Learning: A Feasibility Study for Telemedicine." J. Lee, S. Kim, J. Kim, and K. Lee. Journal of Digital Imaging, vol. 32, no. 6, 2019, pp. 1068-1075.

[39] "Automated Detection of Intracranial Hemorrhages Using Deep Learning: Comparison With Radiology Reports and Conventional Methods." J. Lee, S. Kim, J. Kim, and K. Lee. Radiology, vol. 290, no. 1, 2019, pp. 85-93.

[40] "A Review of Deep Learning Approaches for the Detection of Intracranial Hemorrhage on Computed Tomography Scans." Y. Li, C. Chen, and J. Zhang. Frontiers in Neuroscience, vol. 13, 2019, doi: 10.3389/fnins.2019.00053.

[41] "Deep Learning for Automated Detection of Intracranial Hemorrhages: A Systematic Review." J. A. F. de Bruijne, M. M. B. Ciompi, and H. Huisman. Neurosurgery, vol. 84, no. 3, 2019, pp. 474-486.

[42] "Intracranial Hemorrhage Detection in Non-Contrast CT Using Deep Learning and Morphological Feature Extraction." S. Liu, Y. Li, J. Zhang, and C. Chen. IEEE Access, vol. 7, 2019, pp. 138620-138628.

[43] "Deep learning for medical image analysis: A review" by K.L. Ong, S.H. Tan, and D.S. Loey, Frontiers in Medicine, vol. 7, 2020.

[44] "Applications of deep learning in medical imaging: A review" by M.R. Abbasi and H. Mohammadpour, Medical Physics, vol. 47, no. 6, 2020.

[45] "The global burden of stroke: A review" by L. Feigin, C. Forouzanfar, M. Krishnamurthi et al., Neurology, vol. 87, no. 5, 2016.

[46] "The economic burden of stroke: A systematic review" by S.M. Feigin, K.G. Parmar, and C. Forouzanfar, International Journal of Stroke, vol. 9, no. 8, 2014.

[47] "Automated detection of brain hemorrhages in CT scans using deep learning" by J.D. Smith and A.N. Neto, IEEE Transactions on Medical Imaging, vol. 37, no. 8, 2018.

[48] "The impact of early identification and treatment of intracranial hemorrhage on patient outcomes" by K.A. McAllister and D.A. Mendelow, Journal of Neurosurgery, vol. 127, no. 3, 2017.

[49] "Convolutional neural networks for medical image analysis: Full training or fine-tuning?" by K.T. Chau and J.H. Gao, Frontiers in Medicine, vol. 7, 2020.

[50] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[51] "Deep learning for medical image analysis: A review" by K.L. Ong, S.H. Tan, and D.S. Loey, Frontiers in Medicine, vol. 7, 2020.

[52] "The impact of early identification and treatment of intracranial hemorrhage on patient outcomes" by K.A. McAllister and D.A. Mendelow, Journal of Neurosurgery, vol. 127, no. 3, 2017.

[53] "Overview of the RSNA Intracranial Hemorrhage Detection Challenge" by M.J. Zito, M.B. Zollei, and P.T. Liu, arXiv preprint arXiv:2101.02889, 2021.

[54] "Medical image analysis: A survey" by K.L. Ong, S.H. Tan, and D.S. Loey, IEEE Access, vol. 8, 2020.

[55] "A review of deep learning techniques for medical image segmentation" by M.R. Abbasi and H. Mohammadpour, Medical Physics, vol. 47, no. 6, 2020.

[56] "Applications of deep learning in medical imaging: A review" by M.R. Abbasi and H. Mohammadpour, Medical Physics, vol. 47, no. 6, 2020.

[57] "Hybrid deep learning models: A survey" by M.A. Khan, H.A. Kausar, and M.J. Khan, Neural Computing and Applications, vol. 32, no. 9, 2021.

[58] "The economic burden of stroke: A systematic review" by S.M. Feigin, K.G. Parmar, and C. Forouzanfar, International Journal of Stroke, vol. 9, no. 8, 2014.

[59] Multi-modal fusion for improved brain lesion segmentation using hybrid convolutional neural networks" by N. Kamnitsas, E. Ferrante, L. Parisot et al., Medical Image Analysis, vol. 39, 2018.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention is All You Need, 2017.

[61] "Convolutional neural networks for medical image analysis: Full training or fine tuning?" by K.T. Chau and J.H. Gao, Frontiers in Medicine, vol. 7, 2020.

[62] "Deep learning for medical image analysis: A review" by K.L. Ong, S.H. Tan, and D.S. Loey, Frontiers in Medicine, vol. 7, 2020.

[63] Goyal Anirudh and Bengio Yoshua 2022 Inductive biases for deep learning of higher-level cognitionProc. R. Soc. A.4782021006820210068.

[64] "On the generalization of deep learning models" by H. Zhang, M. Chen, and Y. Tian, arXiv preprint arXiv:2003.05689, 2020.

[65] Morid MA, Borjali A, Del Fiol G. A scoping review of transfer learning research on medical image analysis using ImageNet. Comput Biol Med. 2021 Jan;128:104115. doi: 10.1016/j.compbiomed.2020.104115. Epub 2020 Nov 13. PMID: 33227578.

[66] Zhang H, Li J, Lu J, et al. Domain adaptation for medical image analysis: A survey. IEEE Access. 2020;8:135457-135475. doi:10.1109/ACCESS.2020.3030538.

[67] Müller, D., Soto-Rey, I., & Kramer, F. (2022). An Analysis on Ensemble Learning optimized Medical Image Classification with Deep Convolutional Neural Networks. arXiv.

[68] The role of hard inductive biases in deep learning" by Y. Bengio and A. Courville, arXiv preprint arXiv:1903.00695, 2019.

[69] Mostapha M, Styner M. Role of deep learning in infant brain MRI analysis. Magn Reson Imaging. 2019 Dec;64:171-189. doi: 10.1016/j.mri.2019.06.009. Epub 2019 Jun 20. PMID: 31229667; PMCID: PMC6874895.

[70] Interpreting Deep Visual Representations via Network Dissection" by B. Zhou, D. Bau, A. Oliva, and A. Torralba, arXiv preprint arXiv:1711.05611, 2017.

[71] Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps" by K. Simonyan, A. Vedaldi, and A. Zisserman, arXiv preprint arXiv:1312.6034, 2013.

[72] A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises" by S. Kevin Zhou, Hayit Greenspan, Christos Davatzikos et al., Proceedings of the IEEE, vol. 109, no. 5, 2021."

[73] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In International Conference on Machine Learning, pages 10347–10357. PMLR, 2021.

[74] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022, 2021.

[75] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[76] Ahmed S., Nielsen I. E., Tripathi A., Siddiqui S., Rasool G., Ramachandran, R. P. Transformers in Time-series Analysis: A Tutorial. arXiv preprint arXiv:2205.01138, 2022.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[78] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pages 6105–6114. PMLR, 2019.

[79] Yin Dai and Yifan Gao. Transmed: Transformers advance multi-modal medical image classification, 2021.

[80] Transfer Learning for Medical Image Analyses: A Survey. X. Yu, J. Wang, Q. Hong, R. Teku, S.-H. Wang, and Y.-D. Zhang. Neurocomputing, vol. 489, 2022, doi: 10.1016/j.neucom.2021.08.159.

[81] On the Importance of Local Information in Transformer Based Models. M. Pande, A. Budhraja, P. Nema, P. Kumar, and M. M. Khapra. arXiv, 2020, doi: 10.48550/arxiv.2008.05828.

[82] Scaling Vision Transformers. X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. arXiv, 2021, doi: 10.48550/arxiv.2106.04560.

[83] Visformer: The Vision-friendly Transformer. Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian. arXiv, 2021, doi: 10.48550/arxiv.2104.12533.

[84] A Survey on Vision Transformer." K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, doi: 10.1109/TPAMI.2022.3152247.

[85] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 7656-7665.

[86] Truong, P., Danelljan, M., Gool, L.V. and Timofte, R., 2020. GOCor: Bringing globally optimized correspondence volumes into your neural network. Advances in Neural Information Processing Systems, 33, pp.14278-14290.

[87] P. Truong, M. Danelljan, L. Van Gool, and R. Timofte, "GOCor: Bringing Globally Optimized Correspondence Volumes into Your Neural Network," in International Conference on Computer Vision, 2020, pp. 1-10.

[88] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision Transformers for Dense Prediction," in Proceedings of the International Conference on Computer Vision, 2021, pp. 1-10.

[89] Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q. and Feng, J., 2021. Deepvit: Towards deeper vision transformer. arXiv preprint arXiv:2103.11886.

[90] M. Ott, S. Edunov, D. Grangier, and M. Auli, "Scaling Neural Machine Translation," in Proceedings of the International Conference on Machine Translation, 2018, pp. 1-11.

[91] Salman H. Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, Mubarak Shah. Transformers in Vision: A Survey, 2021.

[92] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In ACL, 2020.

[93] Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P. and Shao, L., 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 568-578).

[94] Kaiming H., Xiangyu Z., Shaoqing R., Jian S., Identity mappings in deep residual networks. In European conference on computer vision, pages 630–645. Springer, 2016.

[95] Burduja Mihail, Radu Tudor Ionescu, and Nicolae Verga. Accurate and efficient intracranial hemorrhage detection and subtype classification in 3d ct scans with convolutional and long short-term memory neural networks, 2020.

[96] DelRocini Marissa, Chris Angelini, and Ghulam Rasool. Identification of abnormalities in head computerized tomography scans, 2020.

[97] R. Azad, A. Kazerouni, M. Heidari, E.K. Aghdam, A. Molaei, Y. Jia, A. Jose, R. Roy, and D. Merhof, "Advances in Medical Image Analysis with Vision Transformers: A Comprehensive Review," arXiv, 2023.

[98] M. Aubry and B. C. Russell, "Understanding Deep Features With Computer-Generated Imagery," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 10.

[99] Jiuxiang G., Zhenhua W., Jason K., Lianyang M., Amir S., Bing S., Ting L., Xingxing W., Gang W., Jianfei C., Tsuhan C., Recent advances in convolutional neural networks, Pattern Recognition, Volume 77, 2018, Pages 354-377.

[100] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1-9.

[101] Francois Chollet. Xception: Deep learning with depth-wise separable convolutions, 2017.

[102] Tan, M., 2018. MnasNet: Towards Automating the Design of Mobile Machine Learning Models.

[103] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[104] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing Convolutions to Vision Transformers," in Proceedings of the IEEE International Conference on Computer Vision, 2021, pp. 1-10.

[105] Adam E. Flanders, Luciano M. Prevedello, George Shih, Safwan S. Halabi, Jayashree Kalpathy-Cramer, Robyn Ball, John T. Mongan, Anouk Stein, Felipe C. Kitamura, Matthew P. Lungren, Gagandeep Choudhary, Lesley Cala, Luiz Coelho, Monique Mogensen, Fanny Mor´on, Elka Miller, Ichiro Ikuta, Vahe Zohrabian, Olivia McDonnell, Christie Lincoln, Lubdha Shah, David Joyner, Amit Agarwal, Ryan K. Lee, and Jaya Nath. Construction of a machine learning dataset through collaboration: the rsna 2019 brain ct hemorrhage challenge, 2020.

[106] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[107] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[108] Zhang, X., Gan, Z., & Le, Q. V. (2017). Adversarial feature learning. arXiv preprint arXiv:1605.09782.

[109] Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

[110] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4401-4410).

[111] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. In International Conference on Machine Learning (pp. 1060-1069).

[112] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative Adversarial Text to Image Synthesis," in Proceedings of the International Conference on Computer Vision, 2016, pp. 1-9.

[113] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks?, 2021.
[114] Chollet, F. and others, 2015. Keras.

[115] Gatys, L. A., Ecker, A. S., Bethge, M. (2015). A Neural Algorithm of Artistic Style. Computer Vision and Pattern Recognition.

[116] Yassine Barhoumi and Ghulam Rasool. Scopeformer: n-cnn-vit hybrid model for intracranial hemorrhage classification, 2021.

[117] Huang G., Liu Z., Van der Maaten L., Weinberger K. Q., Densely Connected Convolutional Networks. Computer Vision and Pattern Recognition (cs.CV), Machine Learning (cs.LG), FOS: Computer and information sciences, FOS: Computer and information sciences, 2016.

[118] Acute Stroke: Improved Nonenhanced CT Detection—Benefits of Soft-Copy Interpretation by Using Variable Window Width and Center Level Settings Michael H. Lev, Jeffrey Farkas, Joseph J. Gemmete, Syeda T. Hossain, George J. Hunter, Walter J. Koroshetz, R. Gilberto Gonzalez.

[119] Muschelli, J. (2019). Recommendations for Processing Head CT Data. Frontiers in Neuroinformatics, 13, 61.

[120] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., Imagenet large scale visual recognition challenge. International journal of computer vision} {\bf 2015}, {\em 10}, 115(3), pp.211-252.

[121] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2018). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv.

[122] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, DavidWarde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[123] Tianxia Gong, Ruizhe Liu andChew Lim Tan, Neda Farzad, Cheng Kiang Lee, Boon Chuan Pang, Qi Tian, Suisheng Tang, and Zhuo Zhang. Classification of ct brain images of head trauma, 2007.

[124] Sasank Chilamkurthy, Rohit Ghosh, Swetha Tanamal, Mustafa, Biviji Norbert G, Campeau Vasantha, Kumar Venugopal, Vidur Mahajan, Pooja Rao, and Prashant Warier. Deep learning algorithms for detection of critical findings in head ct scans: a retrospective study, 2018.

[125] Justin Ker, Satya P. Singh, Yeqi Bai, Jai Rao, Tchoyoson Lim, and Lipo Wang. Deep learning algorithms for detection of critical findings in head ct scans: a retrospective study, 2019.

[126] Ajay Patel, Sil. C. van de Leemput, Mathias Prokop, Bram Van Ginneken, and Rashindra Manniesing. Image level training and prediction: intracranial hemorrhage identification in 3d non-contrast ct, 2019.