Vision transformers with Inductive Bias introduced via self-attention regularization

2023 年 1 月

Luiz Henrique Barbosa Mormille

Discover your potential

Vision transformers with Inductive Bias introduced through self-attention regularization

by

Luiz Henrique Barbosa Mormille

A dissertation submitted

in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

Information Systems Science

Graduate School of Science and Engineering

Soka University, Hachioji, Tokyo

February 2023

This PhD dissertation has been partially supported by the Makiguchi Foundation for Education (*https://www.daisakuikeda.org/*) under the Soka University Makiguchi Foundation for Education, International Student Scholarship.

Dissertation submitted to the Graduate School of Science and Engineering, Soka University.

Vision transformers with Inductive Bias introduced through self-attention regularization

by

Luiz Henrique Barbosa Mormille

(18D5251)

Dissertation Review Committee

Prof. Naoya Torii	Graduate School of Science and Engineering, Soka University, Tokyo.
Prof. Tatsuo Unemi	Graduate School of Science and Engineering, Soka University, Tokyo.
Prof. Masayasu Atsumi	Graduate School of Science and Engineering, Soka University, Tokyo.

February 2023

Learning does not make one learned: there are those who have knowledge and those who have understanding. The first requires memory and the second philosophy.

—**Alexandre Dumas**

(The Count of Monte Cristo)

To lead meaningful and fulfilling lives, it's important that we continually return to the basics and consider the direction we should be heading. This means thinking about such questions as 'How do I best live my life?' 'What is my life's true purpose?' 'What are the key principles for realizing peace?' In other words, having the foundation of a solid philosophy is crucial.

—**Daisaku Ikeda**

(President of Soka Gakkai International)

To my beloved parents, with all my heart.

Acknowledgement

Firstly, I would like express my deepest gratitude to my mentor and the founder of Soka University, Dr. Daisaku Ikeda. Pursuing my doctoral degree here in Soka University is the greatest treasure of my life, and much more than the learning about AI, I was able to challenge my own tendencies and to grow into a better person. I know I still have a lot to learn, and I will never stop seeking knowledge. Thank you Ikeda Sensei for creating the opportunities, opening the doors and supporting me here in Soka University.

I would like to thank my supervisor, Professor Masayasu Atsumi, for all his support, advice and encouragement throughout my journey as a doctoral student. I would also like to thank my dear friends and lab mates Clifford Broni-Bediako and Koichi Nagatsuka for all the valuable discussions, collaboration and support on my research. And also, a big thanks to all my lab friends.

My doctoral course was partially supported by the Soka University Makiguchi Foundation for Education, through their International Student Scholarship, and also by the Japanese Ministry of Education, Culture, Sports, Science, and Technology, through their scholarship for international students. Thank you so much for the support that allowed me to actualize my dream of pursuing a doctoral degree here in Soka University. I also would like to thank all the staff at the International Students Office and the Graduate School of Science and Engineering Office. I'll never forget.

No matter how much I try, I will never be able to repay my debts of gratitude towards my mother and father. Thank you for always supporting me in the pursuit of my dreams. I could not be here if were not for you and all the sacrifices you made for me. Everything I am I owe to you. I love you. And also a big thanks to my siblings.

I also cannot fully express my gratitude towards my life partner, Cintia K. Shiratori. Thanks for always being the best partner, friend and supporter. You always inspire me to become a better person and every day I am able to learn more and more from you. Love you.

Moreover, my deepest gratitude to Editora Brasil Seikyo, for allowing me to apply what I study for such a beautiful and meaningful purpose, for opening my eyes for my mission, and for supporting me in this research. In special, I would like to thank my great friends there, Ricardo

Miyamoto, Edjan Santos, Cauê Barreira, Alessandra Miyagawa, Lygia Diniz and Luciano Leme.

I would also like to convey my deepest gratitude to my best friend, Daisuke Yano. Thank you for showing me the meaning of true friendship, for supporting me throughout my research, my Japanese studies and especially for helping me to prepare for my presentation in Japanese. Also, a big thanks to my close friends — Kenji, Koiti, Susan, Nim, Yda, Yssa, Chin, Shun and Seulgi — who have been my circle of undying support during these last couple of years and were always there for me when I needed most. Thank you all for being my family here in Japan, from the bottom of my heart.

My gratitude to Professor Renato Vicente, at the University of São Paulo, for being the intellectual mentor and friend who encouraged and inspired me to follow the dream of becoming a researcher. Furthermore, I would like to thank Professor Maria Guajardo for teaching me so much ever since I came here to Soka University. Thank you for inspiring to view the world from new perspectives, and for enabling me to grow as a human being.

And finally, I would also like to thank the mysteriously beautiful minds of Carl Sagan and Richard Feynman, whom I never had the pleasure to meet, but whose books and lectures spoken directly to me and sparkled the love for science in my heart. I would also like to thank, among others, Pink Floyd, Led Zeppelin, Iron Maiden, David Bowie, Sir Elton John, Sir Paul McCartney, Cat Stevens, James Taylor, Bruce Springsteen, Ennio Morricone, Hanz Zimmer and Joe Hisashi for presenting me with great music that encouraged me during my long hours of research.

Abstract

In recent years, the Transformer achieved remarkable results in computer vision related tasks, matching, or even surpassing those of convolutional neural networks (CNN). However, unlike CNNs, those vision transformers lack strong inductive biases and, to achieve state-of-the-art results, rely on large architectures and extensive pre-training on tens of millions of images. Approaches like combining convolution layers or adapting the vision transformer architecture managed to mitigate this limitation, however, large volumes of data are still demanded to attain state-of-the-art performance. Therefore, introducing appropriate inductive biases to vision transformers can lead to better convergence and generalization on settings with fewer training data. This work presents a novel way to introduce inductive biases to vision transformers: self-attention regularization. Two different methods of self-attention regularization were devised.

The first method proposed in this work is based on the two-dimensional spatial distance between image patches. In it, a newly proposed *distance loss* imposes a penalty over self-attention computation between each pair of patches based on their distance. When added to the task loss, the *distance loss* acts as a self-attention regularizer, in which the larger the distance between two patches, the greater the penalty attributed over self-attention computation.

Our second self-attention regularization method is based on the similarity between style representations of different regions on a image, taken from their gram matrices. A newly proposed *similarity loss* takes the pairwise distance between the style representation of all image regions in order to impose a penalty over self-attention computation. Similarly to the *distance loss*, the *similarity loss* is added to the task loss, acting as a self-attention regularizer. The intuition behind this method is that high self-attention values between two image patches with a similar style representation should result in a small loss; while high self-attention values between two patches whose style representation are distinct should result in a high loss.

Hence, on both self-attention regularization methods, without changing the global self-attention computation, inductive bias is introduced through minimizing their respective loss functions.

Furthermore, this work proposes ARViT, a novel vision transformer architecture, where both

self-attention regularization methods are deployed. The experimental results demonstrated that self-attention regularization leads to better convergence and generalization, especially on models pre-trained on mid-size datasets, with performance gains as big as 26% when fine-tuned on benchmark downstream classification tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

I am a philosophical naturalist. I am committed to the view that there's nothing in our brains that violates the laws of physics, there's nothing that could not in principle be reproduced in technology. It hasn't been done yet, we're probably quite a long way away from it, but I see no reason why in the future we shouldn't reach the point where a human made robot is capable of consciousness and of feeling pain. We can feel pain, why shouldn't they?

—**Richard Dawkins**

(Evolutionary biologist)

In this chapter, an overview of the of this PhD dissertation is presented, introducing the context, the motivation, the objectives, the propositions and the contributions of this work. In Section 1.1, a brief context of the research fields touched by this work is introduced. Section 1.2 presents the motivation of this dissertation. The main objectives of this work are described in Section 1.3. In Section 1.4, the contributions achieved by the work presented in this dissertation are listed. Lastly, an outline of the structure of this dissertation is presented in Section 1.5.

1.1 Context

This dissertation presents a research work in the field of Artificial Intelligence (AI), with emphasis on Deep Learning. This research investigation intersects the areas of *computer vision*, *transformers*, *self-supervised learning* (SSL) and *inductive bias*.

Deep learning is a subfield of Machine Learning (ML), which by itself is a subfield of AI concerned with comprehending and developing *learning* methods — methods that use data in order to improve their performance on a task, or a set of tasks [1, 2]. In ML, a model can learn through a supervised, semi-supervised, unsupervised or a reinforcement paradigm [3]. Within ML, Deep

Learning is a subfield centered around the concept of neural networks, which were introduced in 1943 by McCulloch and Pitts [4], and revolutionized the ML field. The term "deep learning" refers to the use of neural networks with numerous layers.

One of the areas most impacted by the advent of Deep Leaning was *computer vision*, a multidisciplinary research field that examines how computers can be programmed to extract high-level knowledge from digital pictures or videos. This area focuses on how computers can be taught or learn how to comprehend what is being shown to them. The goal is to find ways to reproduce tasks that can already be done by the human visual system. In other words, computer vision is the process of automatically extracting, analyzing, and comprehending useful information from a single image or a sequence of images. Computer vision has been revolutionized thanks to the development of deep learning based algorithms, with results that marginally exceeded that of earlier approaches on a variety of tasks, such as image classification [5–7], object detection [8–12], image segmentation [13], face recognition [14–16], person re-Identification [17–19], motion tracking [20–22], image colorization [23, 24] and image generation [25, 26].

Among deep neural networks architectures, when it comes to computer vision, Convolutional Neural Networks (CNN) [27] achieved remarkable results across all tasks, dominating the field for years. However, recently, other network architectures have been able to achieve results that compete or even outperform those of CNNs [28, 29]. One of such architectures is the *Transformer* [30]. Emerged from the natural language processing (NLP) field, at the core of the Transformer architecture is its self-attention mechanism, which computes the relation between all elements in a input sequence. The global characteristic of self-attention mechanism contrasts with the local nature of the convolution operation responsible for the tremendous success of CNNs. However, precisely due to global nature of self-attention, such "vision transformers" are deprived of the *inductive biases* inherent in CNNs, such as locality and translation equivariance [31].

Inductive biases are the set of assumption made by a model that make certain solutions (combination of parameter values) more preferable than others, and having the appropriate inductive biases for the task at hand allows a model to converge to better performing solutions. To overcome their lack of inductive biases, vision transformers often rely on very large architectures, pre-trained on hundreds of millions of images and over extensive training schedules. Such approach is associated with a high (and often prohibitive) computational cost.

The final disciplinary field touched by this research is *self-supervised learning* (SSL) [32], a paradigm that incorporate techniques for processing unlabeled data in order to produce representations that may be useful to subsequent learning tasks. The most important aspect of SSL approaches is that they do not need human-annotated labels, eliminating that high cost associated with manual annotation. Instead, in SSL the supervisory signals are obtained through

labels generated automatically from the data itself. During training, these labels function similarly to those of supervised learning tasks. Hence, SSL may be interpreted as a learning paradigm that fits between unsupervised and supervised learning.

The work presented in this dissertation can be placed within Deep Learning, at the intersection of computer vision, transformers, inductive bias, and self-supervised learning. The propositions of this work are to conceive and implement new inductive biases to vision transformers and to deploy and test those inductive biases on a newly proposed vision transformer architecture pre-trained using a self-supervised approach.

1.2 Motivation

Occam's razor, also known as the principle of parsimony, is the problem-solving principle in which, when dealing with competing solutions, the simpler one is considered the better [33, 34]. The Occam's razor is often viewed as an elementary *inductive bias*, where the simplest consistent hypothesis about the target function is to be preferred.

In the context learning systems, inductive biases, like the Occam's razor, are the set of assumptions made by a model in order to predict outputs from unseen data [35]. In other words, a potentially infinite number of hypotheses (combination of weight parameters) can exist for a finite set of training examples, and not all the hypotheses generalize well when tested on unseen data [36]. Learning models inherently tend to be biased toward a group of hypotheses encompassed in their architecture, and those hypotheses can be called an inductive bias.

In deep learning, the selection of models with appropriate inductive biases for the task at hand plays a crucial role in the effort to obtain better generalization [37, 38]. That is particularly true for settings where a small amount of training data is available. Accordingly, models with weak inductive biases tend to converge to the local optima when trained with limited data, therefore being unstable to changes in the initial states.

The success of Convolutional Neural Networks (CNN) is often attributed to its inductive biases, such as locality and translation equivariance. In recent years, studies with the Transformer [30] have also achieved remarkable success in computer vision [29, 39–43]. Nevertheless, state-of-the-art performance is only attained with very large architectures pre-trained on tens of millions of labeled data [44]. For example, the Vision Transformer (ViT) by Dosovitskiy *et al.* [45], when pre-trained on hundreds of millions of images, was able to achieve excellent results compared to CNNs on mid-sized or small image recognition tasks. The data hunger of vision transformers is precisely due to the fact that they have fewer inductive biases than CNNs, allowing them to search the hypothesis space more freely [46, 47]. Hence, when trained on small or medium scale datasets, such vision transformers will often converge to a local optima and generalize poorly on unseen data.

Therefore, such property of vision transformers eventually hinders its training on environments with low resources, due to the heavy computational cost for training a model with hundreds of millions of parameters on tens of millions of images.

Introducing appropriate inductive biases to vision transformers can lead to better convergence and generalization. One of the most common approaches is to introduce inductive biases from CNNs to vision transformers by combining convolution layers with self-attention layers [47–52]. Distilling convolved knowledge [53] and applying self-attention to local neighbors [29] also aim to address this challenge. Another approach revolves around adaptations to the vision transformer architecture. The Pyramid Vision Transformer (PVT) [54] progressively shrinks the transformer architecture incorporating the pyramid structure of CNNs. The Swin Transformer [55] has an hierarchical architecture and uses shifted windows to compute representations from images. Nevertheless, in order to match or surpass the performance of CNNs, such vision transformers are still pre-trained on large scale datasets, like the ImageNet-21K with 14.2 million images [56], or the JFT-300M with 303 million images [57].

Therefore, the main motivation of this work is to tackle the lack of inductive biases on vision transformers and the high computational cost necessary to overcome it (associated with the training of large-capacity models on large-scale datasets).

1.3 Research Aims and Objectives

Based on the aforementioned motivations, the main objectives of the research presented in this dissertation are:

1. To tackle the vision transformers' lack of inductive biases and to improve its ability to generalize well when pre-trained on smaller volumes of data.
2. To improve the ability of smaller capacity vision transformers to generalize well.
3. To reduce the computational cost associated with training high performance vision transformers.
4. To improve the ability of vision transformers when pre-trained on unlabelled data.

In order to achieve the main objectives, the specific objectives of this research are:

1. To investigate the lack of inductive biases on vision transformers and the high computational cost associated with overcoming it.
2. To explore the vision transformer architecture in order to investigate the viability of such models to achieve high performance with smaller capacity architecture designs.

3. To explore the possibilities of introducing new inductive biases to vision transformers, enabling a better and faster convergence when trained on mid-sized datasets, such as ImageNet [56].

4. To investigate the possibility of introducing novel inductive biases to vision transformers without modifying the self-attention operation, preserving its characteristic of being a global operation, and without introducing external components to its architectures, such as convolution layers.

5. To investigate how explicit regularization can be used to restrict the hypothesis space of the transformer towards solutions that generalize while demanding less training data.

6. To explore the possibility of introducing a regularization method to vision transformers based on a symbolic approach where external assumptions are made in order to favor certain distributions on model parameters.

7. To investigate the viability of combining a small capacity vision transformer architecture and a new inductive bias in order to obtain better representations and generalization when pre-trained on mid-sized datasets.

8. To explore pre-training vision transformers using self-supervised learning.

1.4 Contributions

The work presented in this dissertation tackles the lack of inductive biases on vision transformers, which is typically overcame by pre-training large capacity models on hundreds of millions of labeled data. To that end, this work, first, considered the trade-off between the connectionist and the symbolic approaches to artificial intelligence, where the former is focused on model learning, and the latter is based on knowledge derived from formal reasoning being passed down to an AI model.

The first two contributions of this research derived from this exercise are, two *self-attention regularization* methods. Regularization in machine learning has been a long used tool to improve a model's generalization and reduce overfitting [3]. In practice, regularization can be seen as a method to restrict the hypothesis space, favoring certain prior distributions on model parameters [58]. The work presented in this dissertation proposes two regularization methods as means to restrict the vision transformer's hypothesis space based on assumptions about the problem's solution. Hence, though logic is used to constrain the space of possible solutions, it does not impose a solution per se, and a network is still be able to learn within the available hypothesis space.

The first proposed self-attention regularization method is based on the pairwise two-dimensional distance between image patches, measured using the *the Manhattan distance*. The assumption made in this self-attention regularization method is that image patches that are spatially close

to each other should present higher self-attention scores than patches that are spatially distant to each other. In this method, a novel loss function, denoted *distance loss*, is introduced. The distance loss acts as a regularization term added to the vision transformer's training loss, and it encapsulates the logic of the method to penalize solutions based on the pairwise two-dimensional distance between image patches and the respective self-attention scores between them.

The second self-attention regularization method proposed in this research is based on the style similarity between different regions of an image. The style representation of a region of an image is obtained through its *gram matrix*, following the approach of Gatys *et al.* [59]. The assumption made in this method is that, the more similar the style of two regions are, the higher the self-attention scores between them should be. Contrarily, the least similar the style of two regions, the smaller the self-attention scores between them should be. To measure the distance between region styles, the pairwise mean squared error between their gram matrices is utilized. The core of our method is the *Similarity Loss*, designed to express the logical assumption of this method. It, as well, is added to the network training loss to act as a regularization term, penalizing solutions based on the pairwise similarity between image region styles and their respective self-attention scores.

The two self-attention regularization methods are de facto introducing new inductive biases to vision transformers, by restricting the hypothesis space and making certain solutions preferable.

Furthermore, this work introduces ARViT (which is short for Attention Regularized Vision Transformer), an architecture based on the ViT [45] and with changes inspired by Chen *et al.* [44]. ARViT is significantly smaller than the ViT, and in it, the proposed self-attention regularization methods are deployed.

Finally, all models developed in this researched were pre-trained on a self-supervised task using the ImageNet dataset [56], with approximately 1.3 million images.

Experimental results revealed that ARViT, without any self-attention regularization, when compared to a similar capacity vision transformer, already obtained performance gains as big as 23% on benchmark classification tasks. Moreover, the two proposed self-regularization methods further improved the performance of ARViT as far as up to 13% on those tasks. Finally, the models presented in this research also achieve competitive results with state-of-the-art visual transformers, in spite of being a smaller-capacity model and trained on a mid-size dataset.

The scientific contributions derived from the work in this dissertation are summarized as follows:

1. A method to introduce a new inductive bias to vision transformers through self-attention regularization based on the two-dimensional spatial distance between image patches, and a new loss function, denoted *distance loss*.

2. A new inductive bias to vision transformers introduced via a self-attention regularization method based on style similarity of distinct regions of an image, and a new loss function,

denoted *similarity loss*.

3. ARViT, a reduced variant of the ViT [45], with changes inspired by the work of Chen *et al.* [44]. ARViT has 6 encoder layers, 12 attention heads on each layer and 10M trainable parameters, which is a considerably smaller capacity when compared with the original ViT.

4. ARViT is pre-trained using a self-supervised learning approach with rotation estimation as a pretext-task [60], and evaluated on small benchmark downstream tasks, outperforming a similar capacity variant of the ViT by large margins on all tasks (up to 23%).

5. The deployment of the two proposed self-attention regularization methods on ARViT resulted in further marginal performance gains on all small benchmark downstream tasks (up to 13%).

1.5 Outline of the Dissertation

The work presented in this dissertation is divided into further 6 chapters. Chapter 2 describes the necessary background of the work in this dissertation which includes deep learning, the Transformer, inductive bias, model regularization and self-supervised learning. Furthermore, a review of relevant related work in mitigating the vision transformers' lack of inductive biases is presented. Chapter 3 presents a comprehensive description of our *two-dimensional distance based self-attention regularization* method, designed to introduce inductive biases to vision transformers. Chapter 4 introduces the *style similarity based self-attention regularization* method, including the intuition behind it, its algorithm and features. In Chapter 5, we introduce the newly proposed vision transformer architecture, the *Attention Regularized Vision Transformer* (ARViT), in which our self-attention regularization methods are deployed. Chapter 6 presents the experiments conducted in this work, detailing the implementation details, the evaluation methods, and the results. Finally, Chapter 7 provides the concluding remarks with suggestions for further work involving inductive bias on vision transformers.

1.6 References

[1] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*. Springer Science & Business Media, Apr. 2013.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[4] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943. [Online]. Available: http://link.springer.com/10.1007/BF02478259

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3065386

[6] C. Affonso, A. L. D. Rossi, F. H. A. Vieira, A. C. P. de Leon Ferreira *et al.*, "Deep learning for biological image classification," *Expert systems with applications*, vol. 85, pp. 114–122, 2017.

[7] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using imagenet pretrained networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 105–109, 2015.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[9] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast yolo: A fast you only look once system for real-time embedded object detection in video," *arXiv preprint arXiv:1709.05943*, 2017.

[10] Y. Yang and H. Deng, "Gc-yolov3: You only look once with global context block," *Electronics*, vol. 9, no. 8, p. 1235, 2020.

[11] Z. Zhou, J. Zhang, and C. Gong, "Automatic detection method of tunnel lining multi-defects via an enhanced you only look once network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 6, pp. 762–780, 2022.

[12] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv:1505.04597 [cs]*, May 2015, arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597

[14] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[15] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[16] M. Lai, "Deep learning for medical image segmentation," *arXiv preprint arXiv:1505.02000*, 2015.

[17] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2275–2284.

[18] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 6, pp. 2872–2893, 2021.

[19] K. Wang, H. Wang, M. Liu, X. Xing, and T. Han, "Survey on person re-identification based on deep learning," *CAAI Transactions on Intelligence Technology*, vol. 3, no. 4, pp. 219–227, 2018.

[20] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[21] G. Chandan, A. Jain, H. Jain *et al.*, "Real time object detection and tracking using deep learning and opencv," in *2018 International Conference on inventive research in computing applications (ICIRCA)*. IEEE, 2018, pp. 1305–1308.

[22] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, "Deep learning in multi-object detection and tracking: state of the art," *Applied Intelligence*, vol. 51, no. 9, pp. 6400–6429, 2021.

[23] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, and A. W. Muzaffar, "Image colorization: A survey and dataset," *arXiv preprint arXiv:2008.10774*, 2020.

[24] S. Huang, X. Jin, Q. Jiang, and L. Liu, "Deep learning for image colorization: Current and future prospects," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105006, 2022.

[25] T. Iqbal and H. Ali, "Generative adversarial network for medical images (mi-gan)," *Journal of medical systems*, vol. 42, no. 11, pp. 1–11, 2018.

[26] S. Minaee and A. Abdolrashidi, "Iris-gan: Learning to generate realistic iris images using convolutional gan," *arXiv preprint arXiv:1812.04822*, 2018.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2286–2296.

[29] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International conference on machine learning*. PMLR, 2018, pp. 4055–4064.

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

[31] C. Mouton, J. C. Myburgh, and M. H. Davel, "Stride and translation invariance in cnns," in *Southern African Conference for Artificial Intelligence Research*. Springer, 2021, pp. 267–281.

[32] Y. LeCun, "Self-Supervised Learning," 2020, Date Accessed: 2022-11-02. [Online]. Available: https://drive.google.com/file/d/1r-mDL4IX_hzZLDBKp8_e8VZqD7fOzBkF/view

[33] D. Stork, "Foundations of Occam's razor and parsimony in learning," in *NIPS 2001 Workshop*, 2001.

[34] J. Schaffer, "What Not to Multiply Without Necessity," *Australasian Journal of Philosophy*, vol. 93, no. 4, pp. 644–664, Oct. 2015. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/00048402.2014.992447

[35] T. M. Mitchell, *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research . . . , 1980.

[36] J. McCall, "Induction: From Kolmogorov and Solomonoff to De Finetti and Back to Kolmogorov - McCall - 2004 - Metroeconomica - Wiley Online Library." [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0026-1386.2004.00190.x?casa_token=CuZ-QV3GqDsAAAAA:8G2I9gNaZ7zTUJiqziPPHrajPpD-TyIEkRluqElVZfjSk2v2Tgh33eVHZuzfDBbOJ_oIgjr4Ya-jGDxk

[37] S. Ritter, D. G. T. Barrett, A. Santoro, and M. M. Botvinick, "Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study," Jun. 2017, arXiv:1706.08606 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1706.08606

[38] H. Hosseini, B. Xiao, M. Jaiswal, and R. Poovendran, "Assessing shape bias property of convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1923–1931, 2018.

[39] I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens, "Attention Augmented Convolutional Networks," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 3285–3294. [Online]. Available: https://ieeexplore.ieee.org/document/9010285/

[40] H. Zhao, J. Jia, and V. Koltun, "Exploring Self-Attention for Image Recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 10 073–10 082. [Online]. Available: https://ieeexplore.ieee.org/document/9156532/

[41] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[42] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," *arXiv:2005.12872 [cs]*, May 2020, arXiv: 2005.12872. [Online]. Available: http://arxiv.org/abs/2005.12872

[43] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," p. 13.

[44] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, "Visformer: The vision-friendly transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 589–598.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

[46] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, and Y. Xu, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, 2022, iSBN: 0162-8828 Publisher: IEEE.

[47] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, "Vitae: Vision transformer advanced by exploring intrinsic inductive bias," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 522–28 535, 2021.

[48] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 357–366.

[49] X. Chu, Z. Tian, B. Zhang, X. Wang, X. Wei, H. Xia, and C. Shen, "Conditional Positional Encodings for Vision Transformers," *arXiv:2102.10882 [cs]*, Mar. 2021, arXiv: 2102.10882. [Online]. Available: http://arxiv.org/abs/2102.10882

[50] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 908–15 919, 2021.

[51] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "LocalViT: Bringing Locality to Vision Transformers," *arXiv:2104.05707 [cs]*, Apr. 2021, arXiv: 2104.05707. [Online]. Available: http://arxiv.org/abs/2104.05707

[52] Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye, "Conformer: Local features coupling global representations for visual recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 367–376.

[53] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

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

[55] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

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

[57] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 843–852. [Online]. Available: http://ieeexplore.ieee.org/document/8237359/

[58] K. Weinberger, "Linear Regression," Date Accessed: 2022-07-15. [Online]. Available: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote08.html#map-estimate

[59] L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *arXiv:1508.06576 [cs, q-bio]*, Sep. 2015, arXiv: 1508.06576. [Online]. Available: http://arxiv.org/abs/1508.06576

[60] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

Chapter 2

Background and Related Work

I think it's much more interesting to live not knowing than to have answers which might be wrong. I have approximate answers and possible beliefs and different degrees of uncertainty about different things, but I am not absolutely sure of anything and there are many things I don't know anything about, such as whether it means anything to ask why we're here. I don't have to know an answer. I don't feel frightened not knowing things, by being lost in a mysterious universe...

— **Richard Feynman**

(Theoretical Physicist)

The work presented in this dissertation is contained within the field of deep learning, in the intersection of the areas of computer vision, transformers, inductive bias and self-supervised learning. In this chapter, basic concepts of the main areas of study relevant to this work are introduced. In section 2.1, a basic background and fundamental concepts of deep learning are presented. Section 2.2 briefly introduces the basic concepts of the transformer architecture, as well as its application in computer vision tasks through *vision transformers*. Section 2.3 introduces the fundamentals of inductive bias and its relevance in learning models. In section 2.4, the basic concepts of *regularization* are also introduced, for they are a fundamental building block of the method proposed in this research. Furthermore, in section 2.5, the self-supervised learning paradigm and its main differences and similarities with other learning paradigms are described. Finally, in section 2.6, relevant works related to the research presented in this dissertation are also briefly introduced.

2.1 Deep Learning

Deep Learning is a subfield of Machine Learning (ML), which by itself is a subfield of Artificial Intelligence (AI). It concerns the study of computational models that employ multiple processing layers to learn representations from data [1]. To create such computational models, these processing layers are built on the fundamental principles of artificial neural networks (ANNs). Hence, their denomination of Deep Neural Networks (DNNs) [2]. It has been more than 70 years since ANNs, also known as neural networks, first appeared [3]. With little practical applications on the first decades due to computational limitations, the research on neural networks gained new life with the remarkable growth in the processing capacity of computers in the early 2000s, which enabled the development of increasingly larger and more complex neural network architectures, giving birth to the field of Deep Learning [4]. The applications of Deep Neural Networks achieved state-of-the-art performance on a myriad of tasks across multiple areas, such as natural language processing (NLP) [5–7], speech recognition [8–10] and computer vision [11–13]. In this section, a historical background of Deep Learning and the basic concepts of DNNs are presented.

2.1.1 Historical background

In this section, a brief historical background of Artificial Intelligence, leading up to the advent of Deep Learning is presented. The first part is focused on the historical background of artificial intelligence, with emphasis on the aspects of *philosophy*, *formal reasoning*, *logic* and *mathematics* that laid the foundations for the development of modern computation and artificial intelligence. The second part of this section is focused on the two main approaches to AI — the symbolic approach, and the connectionist approach.

2.1.1.1 Historical foundations

At the core of Artificial Intelligence is the belief that the process of human reasoning can be reproduced by a machine. The study of mechanical reasoning, logic and the attempt to formalize the human thought process dates back as nearly 2500 years [14], and methods to represent it have been proposed and improved by philosophers over the years.

Philosophy and logic in China. The history of formal logic in China dates back to the Warring States period (475-221 BCE), during which the Chinese philosopher Mozi founded the Mohist school of thought and made significant contributions to the development of logic and reasoning in China [15]. Mozi's works emphasized the importance of impartial reasoning and argued for a universal love for all beings. He also formulated the concept of reductio ad absurdum, a method of proof by contradiction, which he used to criticize the claims of rival philosophers [16, 17].

In the Han dynasty (206 BCE-220 CE), the philosopher Gongsun Long wrote the *White Horse Dialogue*, which is considered one of the earliest examples of formal logic in China. The work explores the concept of language and meaning and provides a systematic treatment of the relationship between words and reality [18, 19].

During the Tang dynasty (618-907 CE), the development of formal logic in China reached new heights with the works of logicians such as Zhang Zai, who wrote the *Western Inscription* and made important contributions to the understanding of metaphysics and cosmology. Another notable figure from this period was Wei Liaoweng, who wrote the *Ten Problems*, a collection of paradoxes and logical puzzles that challenged the foundations of reasoning [18, 19].

In the Song dynasty (960-1279 CE), the development of formal logic in China continued with the works of logicians such as Cheng Hao, who emphasized the importance of intuition and experience in reasoning, and his brother Cheng Yi, who wrote the *Greater Learning* and developed a system of moral reasoning based on the principles of Confucianism [16, 18, 19].

Philosophy and logic in India. Indian logic can be traced back to the early Upanishads, ancient Hindu scriptures that contain philosophical discussions on knowledge and reality. The *Nyāya school* of philosophy, which was established around the 2nd century BCE, made significant contributions to the development of logic. Its founder, Akṣapāda Gautama, wrote the Nyāya Sutras, which lays down the foundations of Nyāya philosophy and provides a systematic treatment of the nature of knowledge, perception, inference, and fallacies [20, 21].

The Buddhist philosopher Dignaga, who lived in the 5th century CE, made important contributions to the development of Indian logic and epistemology. He wrote several works, including the *Pramāṇa-samuccaya*, which is considered one of the earliest comprehensive treatments of Indian logic [22]. Dignaga's works emphasized the importance of perception and inference in obtaining knowledge and introduced the concept of *valid cognition*, which refers to a cognitive state that is free from error and establishes the nature of reality [22].

The development of Indian logic continued into the medieval period with the works of logicians such as Bhartrhari and Uddyotakara, who wrote commentaries on the Nyāya Sutras and further developed the Nyāya system of logic [20, 21].

Philosophy and logic in Greece. Ancient Greek philosophy and logic played a seminal role in the development of Western philosophy and shaped the course of intellectual history for centuries to come.

One of the most influential figures in ancient Greek philosophy and logic was Aristotle, who was a student of Plato and a teacher of Alexander the Great. In his 350 BCE book *Prior Analytics* [23], Aristotle formalized the definition of syllogism, which is a logical argument that reaches a

conclusion by applying deductive reasoning over two presumably truth statements. A syllogism is often presented in three arguments — two propositions and a conclusion — as follows:

All men are mortal,

All kings are men,

Therefore, all kings are mortal [24]

Fifty years after the publication of *Prior Analytics*, Euclid published the mathematical treatise *Elements* [25], considered one of the most important works in the history of mathematics and one of the earliest comprehensive treatises on the subject. *Elements* is organized into 13 books containing mathematical and geometrical definitions, postulates, propositions, axioms and proofs, covering a wide range of mathematical topics, including geometry, number theory, and incommensurable magnitudes. It is considered one of the most influential models of formal reasoning in history [25].

al-Khwārizmī. al-Khwārizmī was an Persian mathematician and astronomer who lived in the 9th century CE. His works had a profound impact on the development of mathematics in the Islamic world and beyond [17, 26].

One of al-Khwārizmī's most famous works is the *Book of Calculation*, which introduced the decimal system and the use of the Hindu-Arabic numeral system [27].

Another important work by al-Khwārizmī is the *Book of Algebra*, which is considered one of the earliest treatises on algebra. In this work, al-Khwārizmī introduced the concept of algebra as a branch of mathematics and provided solutions to a wide range of mathematical problems. He also used algebra to solve practical problems, such as finding the value of unknown quantities and solving equations [26, 28, 29].

William of Ockham. William of Ockham was a 14th-century English logician, Franciscan friar, and theologian, being credited with the formulation of the principle of parsimony, also known as Ockham's Razor (commonly referred as Occam's Razor), which states that, among competing explanations, the simplest explanation is most likely the correct one [30–33].

In addition to his contributions to philosophy, William of Ockham made important contributions to the development of logic, with his most famous work being *Summa Logicae*, a comprehensive treatise on logic that covers a wide range of topics, including the nature of language, the theory of meaning, and the rules of inference and syllogism. In this work, he provided a systematic account of the principles of logic and the rules of reasoning, which had a profound impact on the development of logic and the study of language [30–33].

Ramon Llull. Ramon Llull was a 13th-century Majorcan writer, logician, and Franciscan lay brother who is considered one of the pioneers of computational theory. He is best known for his

work in the field of logic and the development of a system of thought, known as the *Ars Magna*, which aimed to provide a systematic method for combining ideas to generate new knowledge [34].

Llull's *Ars Magna* was a form of combinatorial logic that used simple diagrams and symbols to represent concepts and relationships between them. This system allowed for the manipulation of these symbols to generate new combinations of ideas and, in theory, lead to new knowledge. The *Ars Magna* was ahead of its time and paved the way for the development of modern computational theory and artificial intelligence [34, 35].

Gottfried Wilhelm Leibniz. Gottfried Wilhelm Leibniz was a 17th-century German philosopher, mathematician, and logician. He made significant contributions to a wide range of fields, including mathematics, physics, metaphysics, and the history of philosophy [14, 36].

In mathematics, Leibniz is best known for his invention of calculus, which he developed independently of Sir Isaac Newton. He also made important contributions to the development of the binary system, which is the basis of modern computer science [14, 36].

In the field of logic, Leibniz was a proponent of the principle of *non-contradiction*, a central tenet of classical logic, which states that, for any proposition, either it is true or its negation is true, but not both. In other words, something cannot both be and not be at the same time and in the same sense [14, 35–39] .

Leibniz saw the principle of non-contradiction as the foundation of all rational thinking, and argued that it is necessary for making any meaningful statement or argument. Without the principle of non-contradiction, it would not be possible to distinguish between truth and falsity, and all reasoning and debate would be rendered meaningless [14, 40–43].

René Descartes. René Descartes was a 17th-century French philosopher, mathematician, and scientist. Descartes attempted to rebuild knowledge from first principles by using the philosophical method of systematic doubt. The aim of this method is to doubt all of one's beliefs, including one's most basic assumptions about the world, until only indubitable truths remain. From these indubitable truths, Descartes believed that all other knowledge could be deduced and built into a complete system of knowledge [14, 44].

The method of systematic doubt is based on the idea that the only way to arrive at certain knowledge is to begin by doubting everything, including one's own senses and the beliefs handed down by tradition. Descartes argued that this method of radical skepticism would allow him to identify and eliminate false beliefs, and arrive at a foundation of indubitable truths [14, 44].

Descartes used his method of systematic doubt to challenge the foundations of traditional Aristotelian philosophy, which he believed was based on uncertain and flawed foundations. He

believed that by doubting everything, he could arrive at a new, more certain foundation for knowledge that would not be based on uncritically accepted beliefs [14, 28, 44].

One of the most famous examples of Descartes' method of systematic doubt is his argument that he can be certain of his own existence, even though he may doubt everything else. He argued that since he is able to doubt his beliefs, he must exist as a thinking being in order to do so. From this indubitable truth, "Cogito, ergo sum" (I think, therefore I am), Descartes believed that he could deduce all other knowledge and build a complete system of knowledge [14, 27, 44].

Thomas Bayes. Thomas Bayes was an 18th-century English statistician and theologian who is best known for the proposition of the *Bayes' theorem* [45], a mathematical formulation that describes the relationship between the probability of an event and the prior probabilities of related events.

The theorem is expressed mathematically as follows:

$$P(A|B) = (P(B|A) * P(A))/P(B), \tag{2.1}$$

where:

- $P(A|B)$ is the probability of event A given that event B has occurred, known as the posterior probability [45].
- $P(B|A)$ is the probability of event B given that event A has occurred, known as the likelihood [45].
- $P(A)$ is the prior probability of event A, which represents our initial belief about the probability of the event before taking into account any new information [45].
- $P(B)$ is the marginal probability of event B, which represents the probability of event B, regardless of whether event A has occurred or not [45].

Bayes' theorem is particularly useful for dealing with problems where the information available is limited and uncertain, as it provides a way to update one's beliefs about the probability of an event based on new information. It is currently applied in many different fields, including artificial intelligence, computer science, economics and medicine [46–48].

More specifically, Bayesian methods are used extensively in machine learning, particularly in Bayesian deep learning, which is a type of machine learning that incorporates Bayesian principles into deep neural networks. In Bayesian deep learning, Bayes' theorem is used to estimate the probabilities of different model parameters given the data, and to make predictions about future data based on these probabilities [46].

Bayesian methods are also used in probabilistic graphical models [49], which are a type of probabilistic model used in machine learning and artificial intelligence. In probabilistic graphical models, Bayes' theorem is used to calculate the probability of an event based on the relationships between different events, which are represented as nodes in a graphical model.

Georg Cantor. Georg Cantor was a 19th-century German mathematician who is best known for his work in the field of set theory and the foundations of mathematics [50].

One of Cantor's most famous contributions was his development of the theory of infinite sets. Prior to Cantor's work, it was widely believed that there was only one kind of infinity, but Cantor showed through his *diagonalization proof* that there are more real numbers than there are natural numbers, and that therefore, there are different sizes of infinity and that some infinities were larger than others [50]. This discovery had far-reaching implications for the foundations of mathematics, causing controversy among mathematicians which were divided into *formalists*, who supported Cantor's proposition, and *intuitionists*, who opposed Cantor's proposition.

George Boole. George Boole was an English mathematician and logician who lived from 1815 to 1864. He is best known for his work in developing a system of mathematical logic that would later become known as Boolean algebra.

Boole's work was motivated by the idea of using algebraic symbols and methods to represent logical statements and to solve logical problems. He sought to create a system of symbolic logic that would be similar to traditional algebra, but that would apply to logic instead of arithmetic [51, 52].

In 1847, Boole published his first book, *The Mathematical Analysis of Logic* which introduced the basic ideas of his system of symbolic logic. In this book, Boole used symbols to represent logical concepts such as **AND**, **OR**, **NOT**, and **IMPLIES**, and he used algebraic operations to represent the relationships between these concepts [52].

Boole's work was widely influential, and his ideas formed the foundation for the modern field of symbolic logic. In particular, his concept of a binary system, where logical values are represented by either 1 or 0, has been fundamental to the development of digital computers, and it is still applied to areas such as computer science, engineering, and artificial intelligence [51, 52].

Gottlob Frege. Gottlob Frege was a German philosopher and logician who lived from 1848 to 1925. He is considered one of the most important figures in the development of modern logic and is often referred to as the "father of analytic philosophy."

Frege's work was motivated by a desire to provide a foundation for mathematics that would be based on logic. He saw that traditional logic was not well suited to this task, so he developed a new

system of logic that came to be known as *predicate logic*, introduced in his book *Begriffsschrift* [53].

In this system, Frege introduced the concept of a *function* and a *concept*, which would later become the basis for modern mathematical logic. He also introduced the idea of *quantifiers*, which allow us to make general statements about all objects of a certain type [53–55].

Frege's ideas had a profound impact on the development of modern logic and the foundations of mathematics. In particular, his work provided a new way of looking at mathematical concepts and paved the way for the development of modern set theory and model theory, still being widely studied and applied in areas such as logic, mathematics, and computer science [54–56].

Bertrand Russell and Alfred North Whitehead. Bertrand Russell was a British philosopher, logician, and mathematician who lived from 1872 to 1970. He is widely considered to be one of the most important figures in the development of modern logic and analytic philosophy.

Russell's work was motivated by a desire to provide a solid foundation for mathematics and to understand the nature of knowledge and truth. He saw that traditional logic was not well suited to this task, so he developed a new system of logic that was based on mathematical concepts and methods [29].

Alfred North Whitehead was a British philosopher and mathematician who lived from 1861 to 1947. He is best known for his work in the foundations of mathematics, metaphysics, and process philosophy.

Whitehead's work in mathematics was inspired by his belief that mathematics was not just a collection of symbols and formulas, but was a way of understanding the world. He was particularly interested in the foundations of geometry and worked to provide a more intuitive and philosophical understanding of mathematical concepts [57].

Together, Russell and Whitehead published their formal treatment of mathematical logic and the foundations of mathematics, *Principia Mathematica* [58], consisting of three-volumes.

One of the key innovations of *Principia Mathematica* was the introduction of the concept of a *type theory*, which separated mathematical concepts into different *types* based on their logical properties. This idea was later adopted by computer scientists and is now a fundamental component of modern type theory and programming languages [58].

The authors also introduced the idea of a *logical atom*, which is a basic proposition that cannot be reduced to more fundamental propositions. The logical atoms were used to build more complex propositions, and the authors showed how mathematical statements could be derived from these basic propositions [58].

Furthermore, Russell also exposed the paradox of self-reference that arises in set theory and logic, consisting of the assumption that a set of all sets that do not contain themselves can be formed [58]. If this set is called R, then the following paradoxical situation arises:

- If R contains itself, then by definition it is not a set that does not contain itself.
- If R does not contain itself, then by definition it is a set that does contain itself.

This paradox shows that basic assumptions about sets and their properties lead to a contradiction, calling into question the foundations of set theory.

Russell's paradox had a significant impact on the development of mathematical logic and set theory, and led to the development of alternative foundations for mathematics, such as *Zermelo-Fraenkel* set theory and type theory. These theories attempt to resolve the paradox by placing restrictions on the formation of sets and introducing new concepts, such as "types" of sets, to avoid the paradoxical situations that arise in traditional set theory.

David Hilbert. Following the impact of of Russell's and Whitehead's work, German mathematician David Hilbert initiated an endeavour to prove whether all mathematical reasoning could be formalized [38]. A formalist, Hilbert sought a more formal and rigorous system of mathematical proof, based on set theory, and that could solve all the issues that had appeared in math over the previous century.

Hilbert wanted to prove that the current mathematical system was *complete*, *consistent* and *decidable*. By complete, Hilbert meant that every true statement could be proved. By consistent, he meant that the mathematical system was free of contradictions. And by decidable, he meant that there is an algorithm that can always determine whether a statement follows from the axioms [38].

Kurt Gödel. Kurt Friedrich Gödel was a logician, mathematician, and philosopher who proposed two *incompleteness theorems* in the early 1930, demonstrating the inherent limitations of all formal systems for mathematics, thus addressing the questions raised by Hilbert regarding completeness and consistency [38, 59].

The first incompleteness theorem states that for any formal system powerful enough to describe the basic properties of the natural numbers, there exist mathematical statements that can neither be proven nor disproven within that system. In other words, any formal system that is powerful enough to capture the essence of arithmetic will necessarily be incomplete in the sense that there will always be some true mathematical statements that cannot be derived within the system [38, 60, 61].

The second incompleteness theorem states that if such a formal system is also consistent, meaning that it does not allow for any contradictions, then it cannot prove its own consistency. In other words, no formal system can prove its own consistency if it is indeed consistent. [38, 60, 61].

Alan Turing. Alan Turing was an English mathematician and computer scientist who is widely regarded as one of the pioneers of theoretical computer science and artificial intelligence [57]. Turing's most famous contribution to the field of computer science is his concept of the Universal

Turing machine, a theoretical device that is considered to be the theoretical basis for modern computers.

The Turing machine consists of an infinitely long tape, divided into cells, each of which can contain a symbol from a finite alphabet. The machine also has a read/write head that can read and write symbols on the tape, and a set of internal states that determine its behavior [62]. The machine operates by following a set of rules, or a program, which specify what action it should take based on the symbol it is currently reading and its current state [62].

The Turing machine is considered to be a universal model of computation, meaning that it can perform any operation that is computable. This makes it a powerful tool for studying the limits of what can and cannot be computed, and for understanding the nature of computation itself [62].

A system is considered *Turing complete* if it can perform any computation that can be performed by a Turing machine. This is an important concept in computer science and the theory of computation, as it provides a way of comparing and evaluating different computer systems and programming languages [57, 62].

The connection between Turing machines and *mathematical decidability* lies in the idea that a Turing machine can be used to determine whether or not a mathematical statement is true or false. In other words, a Turing machine can be used to perform a proof of a mathematical statement by following a set of rules to systematically determine whether the statement is true or false [57, 62]. If the statement can be proven to be true or false by a Turing machine, then it is said to be decidable.

The concept of decidability is important in mathematical logic and the theory of computation, because it provides a way of defining what is meant by a computable or solvable mathematical problem. A problem is considered computable or solvable if there exists a Turing machine that can determine its solution. It also has practical applications in the field of computer science, where it is used to determine the limits of what can be computed and to design more efficient algorithms for solving specific problems [38, 57].

Alan Turing is considered one of the most important founding figure in computer science. It can be said that all modern computers descended from his designs. And, in part, Turing's ideas about computability came from Hilbert's question on whether or not the systems of mathematics were decidable.

2.1.1.2 The Symbolic and Connectionist approaches

The advancements in mathematical logic laid the foundations for artificial intelligence to be developed, and the first modern computers emerged near the end of the first half of the 20th century, with the colossal machines developed to perform code breaking during World War II [63–69]. Together with other discoveries made at the same time in neurobiology [70], Claude Shannon's

information theory [71, 72], and Alan Turing's theory of computation [73, 74], researchers began to contemplate the possibility of making an electronic brain [36, 75–78], with the first work recognized as AI emerging with McCullouch and Pitts' 1943 Turing-complete formal design of "artificial neurons" [3].

Then, two approaches on how to develop AI begun to emerge from the 1950s — the *Symbolic* approach, and the *Connectionist* approach. The main idea behind the Symbolic approach is to develop a collection of high-level human-readable representations of problems, using tools such as logic programming and production rules in order to automate a variety of tasks [79–81]. In other words, the symbolic approach utilizes semantics and symbols to meaningfully model relationships and create intelligent programs. The majority of the early progress attained in the AI field were obtained through a symbolic approach [76, 82]. Nevertheless, the semantic nature of this approach limited its use to tasks that could be modeled within that semantics [83]. In other words, only tasks that could be addressed through human reasoning could be represented in a symbolic AI program [75, 76, 84].

The connectionist approach, on the other hand, aimed to develop artificial intelligence through learning [85]. Those who were adept to the connectionist approach were inspired by cognitive science and the connection of human neurons [85], proposing a theory where signal processing occurs simultaneously and distributively through numerical connections [86, 87]. In the connectionist approach, learning happens by modifying the weights of the connections based on the data processed by the model [88, 89]. Although is difficult to understand how they process information and to obtain a high-level understanding, connectionist AI programs' ability to be applied to a wide variety of tasks, structural closeness to real neurons and minimal requirements for innate structure are some of their main benefits [90–93]. Though the symbolic approach dominated the AI landscape for a few decades after the 1950s, most of the recent AI systems are based on a connectionist approach, such as deep learning based models [12, 13, 94–99] and a plethora of machine learning systems [100–105].

Authors have compared the symbolic and connectionist approaches to the human *mind* and the human *brain*, respectively [106]. Though there are inconsistencies in this comparison, it is capable to underline the basic differences between both approaches. Mind is generally interpreted as an expression of consciousness, intention, reasoning, logic and even emotion. Otherwise, the brain, in this comparison, means the intricate network of neurons and electric impulses that sustains the mind. Though the separation between mind and brain is still debated by diverse fields, such as neuroscience, philosophy and religions [75, 76, 107, 108], such definition is out of the scope of this research, and we adopted it merely from an allegorical perspective, indicating that the symbolic approach, analog to the human mind, starts from high level concepts that are transferred to an AI

program, whereas the connectionist approach, similarly to a human brain, starts from a low level emulation of a brain that gradually learns to represent high level concepts.

In theory, by adopting a connectionist approach, with an immensely large network and huge volumes of data, an AI functionality akin to human thinking can be attained [109–113]. However, in spite of the tremendous success of connectionist models, especially with deep learning, some authors argue that they lack in interpretability, pointing out to the potential consequences and dangers that could arise from transferring decision making to AI systems that lack an understandable train of logic [114–121].

Addressing such concerns are of immediate relevance, for AI systems are becoming increasingly interlaced with the fabrics of society, by driving autonomous vehicles [122–125], making credit decisions [126–130], tracking [131–133] and recognizing individuals [134–138], targeting personalized marketing [139–141] and content [142–149], and impacting a myriad of areas, such as agriculture [150–156], medicine and health-care [157–164], and the military industry [165–172]. Carl Sagan stated that "One [danger] is... that we've arranged a society based on science and technology in which nobody understands anything about science and technology, and this combustible mixture of ignorance and power, sooner or later, is going to blow up in our faces" [173]. For a society reliant on AI powered technologies, it is important to have a framework of logic, ethics and transparency [174].

Recently, there has been an increase in researches that aim at combining both the power of the connectionist approach and the logical framework of symbolism [175–179].

2.1.2 Basic concepts of Deep Neural Networks

Deep learning is a subfield of machine learning defined by the use of artificial neural networks (ANNs) comprised by multiple layers. A neural network itself, is model in machine learning inspired by the human brain and its biological network of neurons, and it is typically applied to address tasks that are hard to be defined through a symbolic approach [3, 180, 181]. Neural networks are considered to be universal approximators, meaning that they, depending on the amount of hidden neurons and connections, are capable of approximating any function, no matter how complex [182, 183]. The use of such "deep" neural networks was proven to be valuable on both regression and classification problems, and being capable of extracting powerful features from inputs of diverse natures, such as natural language, images, videos and sound. Usually, on a deep neural network (DNN), the deeper the layer, the higher the level of the representation obtained [184]. In this section we introduce the structure of an artificial neuron and some commonly used network architectures.

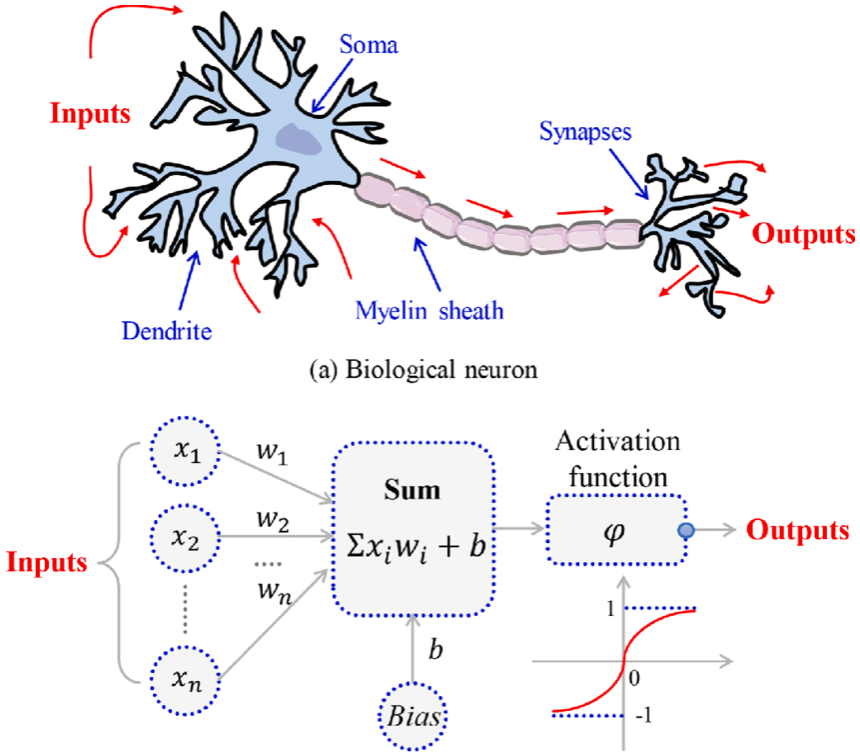


Figure 2.1: Comparison between biological neuron and artificial neuron. Figure from Karpathy *et al.* [186].

2.1.2.1 Artificial Neuron

Artificial neurons are the fundamental building block of artificial neural networks [185], and its conception was inspired by the neurobiological understanding of human-brain neurons [3, 180, 181].

Figure 2.1 illustrates a comparison between a biological neuron and an artificial neuron. First, in the biological neuron the *dendrites* are responsible for receiving signals from neighboring neurons. At the artificial neuron, the signals from the neighboring neurons are mimicked by an input vector, and the dendrites are represented by a weight vector that multiplies the inputs [187].

Secondly, the *soma* structure in a biological neuron has the function of adding the signals received through the dendrites [187]. In the artificial neuron, the soma is represented as the following summation function of all weighted inputs plus a bias:

$$\sum_{i=1}^{n}(\omega_i, x_i) + b, \tag{2.2}$$

where n is the length of the input vector, ω_i and x_i are respectively the weight value and the input value at position i, and b is a bias parameter.

Finally, at the right-most extremity of the biological neuron illustration, the region where the *synapses* occur is depicted. The region between the soma and the synapses is denoted *axon*, and it receives the signal from the summation occurred in the soma [187]. If the signal achieves an

"activation potential", the axon will open and transmit it to other neighboring neurons via synapses. In the artificial neuron, that "activation potential" is reproduced by an *activation function* (also referred to as transfer function) that takes in the summed weighted input, and if activated, outputs the value to be forwarded to the next hidden layer. It can be described as follows:

$$y = \varphi(\sum_{i=1}^{n}(\omega_i, x_i) + b), \tag{2.3}$$

where φ is the activation function and y is signal to be forwarded to the next hidden layer or as an output.

Though originally the activation function was a linear threshold function, recently most ANNs utilize nonlinear activation functions, such as the *sigmoid* function, the *hyperbolic tangent* function and the *rectified linear unit* function (ReLU) [181, 188, 189].

2.1.2.2 Common network architectures

Over the years, several distinct DNN architectures have been developed, each one of them with their respective set of inductive biases and preferred applications. Some notable DNN architectures are:

- Multilayer perceptrons (MLP) [190]. Develop in 1958 by Frank Rosenblatt, it is pointed as the oldest form of neural network;
- Convolutional neural networks (CNNs) [191]. A type of ANN that makes use of convolution in at least one of their layers. They were originally designed to process image and video.
- Recurrent neural networks (RNNs). Based on 1986 Rumelhart *et al.* [192], they are a category of ANNs characterized by its cycle like connections, where the output of certain nodes influence subsequent inputs of the same node. RNNs are mostly used for sequential data.
- Transformer [95]. A type o ANN that utilizes a self-attention mechanism in which the relation between all elements in the input sequence is computed. Emerged from the natural language processing context, it is now growing in popularity in the computer vision field as well.

2.2 The Transformer

The work of Vaswani *et al.*, 'Attention Is All You Need' [95], describes the *Transformer* and and its sequence-to-sequence architecture (or Seq2Seq), which transforms a given sequence of elements, such as the sequence of words in a sentence, into another sequence.

Emerged from the context of Natural Language Processing (NLP), the transformer consist of an Encoder and a Decoder. The former takes an input sequence and maps it into a n-dimensional

space. The latter then takes the n-dimensional representation produced of the Encoder and turns it into an output sequence. In the context o NLP, the input can, for example, be a sequence in a given language, and the output can be the sequence translated to a different language.

At the core of transformer lies its *self-attention* module. Self-attention is an operation that allows every element on the input sequence to interact with each other by computing their pairwise attention scores. This allow the transformer to capture long range dependencies within the input sequence, which is especially useful on NLP related tasks.

In this section, we describe the Transformer proposed by Vaswani *et al.* in his paper 'Attention Is All You Need' [95], detailing its main components. We further introduce *vision transformers*, which are variations of the transformer applied to computer vision tasks.

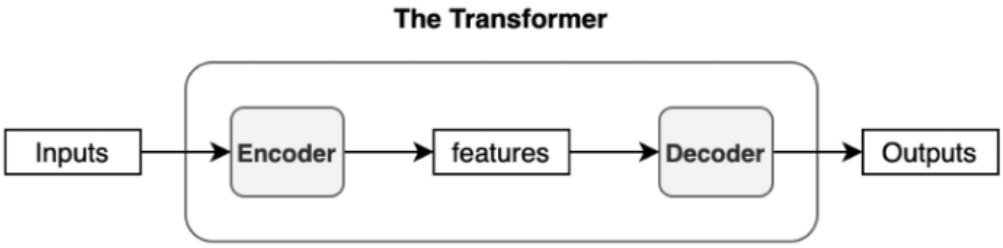


Figure 2.2: A basic diagram of the transformer [193].

2.2.1 The Transformer Architecture

The *transformer* is a DNN designed to take a sequence o feature vectors as an input, and to output another sequence. Emerged from the natural language processing context, one simple application example of the transformer would be *machine translation*, where a sentence in a given language is forwarded as an input to the transformer, and a sentence in a different language is expected as an output.

2.2.1.1 Encoder and Decoder

From a broader perspective, the transformer is composed of two distinct components connected between themselves: an *Encoder* component, and a *Decoder* component. Between those two components, information will only flow from the encoder to the decoder, as show in Figure 2.2.

The transformer encoder is composed of a stack of N *encoder blocks* (also referred to as encoder layers). Similarly, the transformer decoder is composed of a stack of decoder blocks (or decoder layers). The number of encoder blocks in the transformer encoder and the number of decoder blocks in the transformer decoder should be the same. In the original Transformer proposed by The work of Vaswani *et al.* [95], the number of encoder blocks and decoder blocks is of $N = 6$.

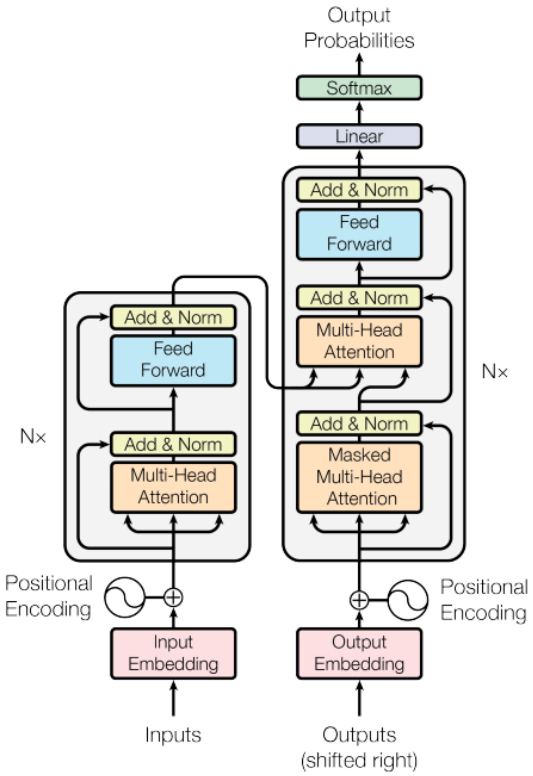


Figure 2.3: The encoder-decoder architecture. Figure from Vaswani *et al.* [95].

Encoder blocks are typically identical in structure, with each of them being composed of at least two basic components: a *self-attention* layer, and a position-wise fully connected feed forward network [95]. In addition, a residual connection followed by layer normalization is placed around both the self-attention layer and the feed forward layer. The structure inside an encoder layer shown in Figure 2.3. In this basic setting, information coming to an encoder block would first be processed by the self-attention layer, then forwarded to the feed forward layer.

A decoder block has a structure similar to that of the encoder block with only one extra layer to computed self-attention over the output of the transformer encoder. This layer is inserted between the self-attention layer and the feed forwards layer, and it also have a skip connection followed by layer normalization around it, as displayed in Figure 2.3.

A complete view of the Transformer proposed by Vaswani *et al.* [95] is shown in Figure 2.4.

2.2.1.2 Self-attention

Self-attention (also denoted Scaled Dot-Product Attention) is at the core of the transformer, being responsible for computing the pairwise relation between all elements in the input sequence. According to Vaswani *et al.*[95], the role attention function introduced is to map a "query and set of key-value pairs to an output." In this setting, Query, Keys, Values and output are all vectors.

Therefore, given an input sequence of length n, where x_i is the i-th element on the input

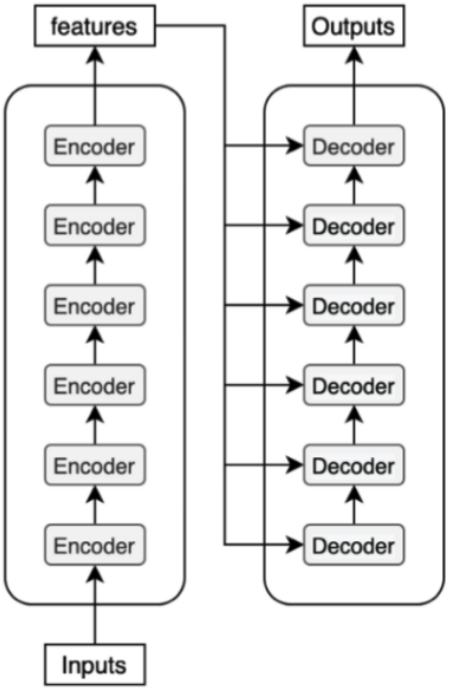


Figure 2.4: A diagram of the transformer displaying all the *encoder layers* and all the *decoder layers* [193].

sequence (a feature vector), the first step in self-attention is to compute a Query vector, a Key vector and a Value vector from each of the encoder's input vectors. Namely, q_i, k_i and v_i are Query vector, a Key vector and a Value vector computed from x_i.

In order to obtain q_i, k_i and v_i, x_i undergoes a matrix multiplication with three distinct weight matrices — W^Q, W^K and W^V, respectively — that are learned during training.

In the original paper, the length of x_i is 512 and the dimensions of W^Q, W^K and W^V are 512×64. The dimension of the query vector q_i and the key vector k_i is denoted d_k, and the dimension of the value vector v_i is denoted d_v. In the original paper, both d_k and d_v are equal to 64.

All query vectors, all key vectors and all values vectors for all elements in the input sequence are respectively arranged into three distinct matrices, Q, K and V. Then, self attention is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V. \tag{2.4}$$

In practice, the dot product between Q and K^T in Equation (2.4) is computing a score representing the the pairwise relation between all elements in the input sequence. The division of this dot product by $\sqrt{d_k}$ has the purpose of obtaining more stable gradients. The matrix resulting from $QK^T/\sqrt{d_k}$ then goes through a softmax layer to normalize all its values, so they are all positive and add up to 1, to then finally undergo a dot product with the value matrix V. A

visual representation of the self-attention computation is described in Figure 2.5, taken from the Transformer original work [95].

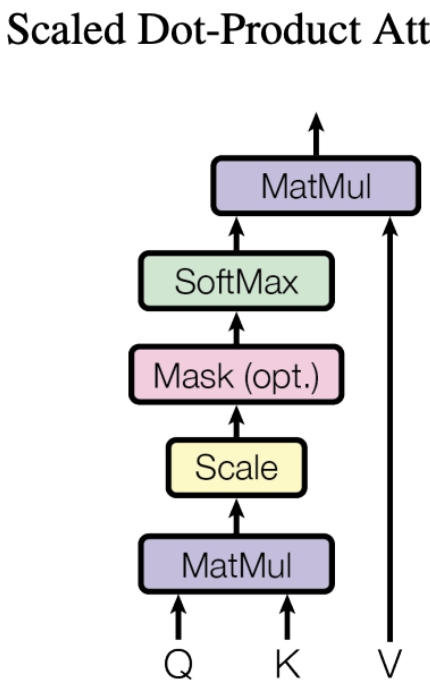


Figure 2.5: Scaled Dot-Product Attention (Self-attention), where Q is the query matrix, K is the key matrix and V is the values matrix. Figure from Vaswani *et al.* [95].

2.2.1.3 Multi-head attention

The Tranformer [95] further improved the self-attention mechanism by introducing the concept of *multi-headed attention*. Multi-head self-attention means that, instead of performing a single self-attention computation, the self-attention layer performs it h times in parallel, using different queries, keys and values.

In other words, instead of containing a single set of parameter matrices (W^Q, W^K, W^V), the self-attention layer is comprised of h sets (W_j^Q, W_j^K, W_j^V), where $1 \leq j \leq h$. The output of each attention head in then concatenated and projected through the dot product with with a parameter matrix W^O, as described in the following equation.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O, \\ \text{where head}_j &= \text{Attention}(Q_j, K_j, V_j). \end{aligned} \tag{2.5}$$

In this formulation, Q_j, K_j and V_j are respectively the query, key and value matrices produced by the dot product of the input sequence and W_j^Q, W_j^K and W_j^V. In the original work $h = 8$. A

illustrated representation of the multi-head self-attention computation is shown in Figure 2.6.

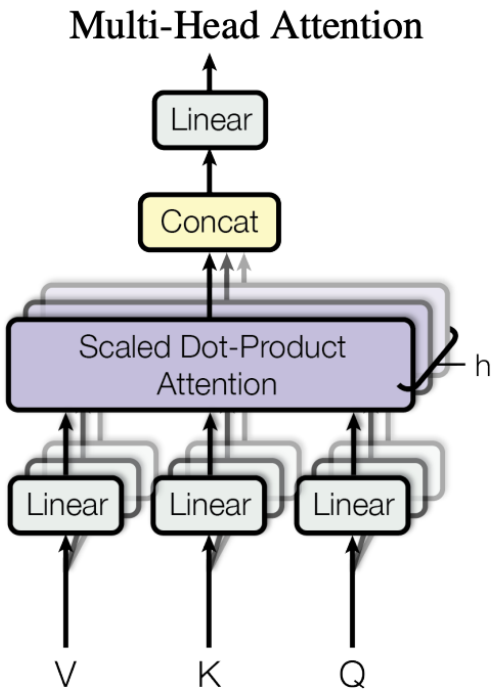


Figure 2.6: Multi-head self-attention computation. Figure from Vaswani *et al.* [95].

There are two main benefits from using a multi-head setup instead of a single self-attention function. Firstly, it enhances the ability of the model to "focus" on different parts of the input sequence. Secondly, it adds multiple representation subspaces to the model, thus improving the overall model's performance.

2.2.2 Vision Transformers

Over the course of the past decade, Convolutional Neural Networks (CNN) have dominated the landscape of computer vision, introducing a new era in this domain of study [191]. Being specifically designed for images, CNNs have strong inductive biases like translation invariance and locality [194].

Still, in spite of its remarkable success, CNNs also have its shortcomings. For instance, its design is domain-specific [195]. In other words, CNNs are not scalable to be domain agnostic. Furthermore, due to the design of the convolution operation, though CNNs learn to represent image features well, they lack a global understanding of the structural dependency between those features. On top of that, CNNs are also computationally expensive models [196].

On recent years, the Transformer, which achieve remarkable results on the domain of NLP, has also proven to be a reliable alternative to CNNs on computer vision related tasks [197, 198].

The original Vision Transformer (ViT), by Dosovitskiy *et al.* [199], function by splitting an input image into fixed-size patches, linearly embedding them, adding a positional encoding and forwarding the resulting sequence to a Transformer Encoder.

Similarly to CNNs, the computational cost of the ViT is also high. More specifically, the cost of self-attention is quadratic to the number of elements in the input sequence [199]. If each pixel of an image were to be embedded as an individual element in the input sequence, then self-attention would require each pixel to attend to every other pixel. Hence, the computational cost would be very high and would not scale to realistic input size on environments with limited resources. Thus, the input image is divided into patches and each patch is embedded as one element in the input sequence, therefore reducing the computational cost at self-attention [199]. Hence, in the self-attention layer, the pairwise attention score between each image patch is computed.

The ViT also adds an extra *classification token* to the patch embedding at the start of the sequence [199]. This token is ultimately used to perform the classification task at the model head (MLP with one hidden layer at pre-training time and a single linear layer for fine-tuning).

The Transformer Encoder itself consists of a set of encoder layers, each containing the following:

1. Multi-Head Self Attention Layer (MHSA) that concatenates the multiple attention outputs. The multiple attention heads enables the learning of both local and global dependencies in the image.

2. Multi-Layer Perceptrons (MLP) with two layers and Gaussian Error Linear Unit (GELU).

3. Two steps of Layer Norm(LN), applied before the MHSA layer and the MLP, respectively. The role of the layer norm is to improve both training time and generalization performance.

4. Residual connections subsequent to the MHSA layer and the MLP, enabling the gradients to flow through the network directly without passing through non-linear activations.

With this design, global features are learned at the higher encoder layers of the ViT, whereas at the lower layers, both global and local features are learned, allowing the ViT to learn more generic patterns [199].

Following the success of the ViT, many variations of vision transformers were proposed in the literature, tackling a myriad of tasks in the computer vision field.

2.2.2.1 Self-attention in Vision Transformers

Unlike the natural language processing domain, in the vision transformers, the input is not naturally a one dimensional sequence. Instead, an image is typically divided into a set of N two dimensional patches, and for each patch a linear encoding is produced. Those linear encodings are then arranged into a one-dimensional sequence and supplemented with an addition positional encoding (either

learned or fixed), with the goal of preserving the two-dimensional information between the patches [199].

The multi-head self-attention layer computes the attention map A, a $N \times N$ matrix containing pairwise relation between all patches. It is worth noting that the attention map A is not the final product of the self-attention layer, it is instead, a intermediate computation of that layer. More precisely, it is the matrix resulting from the following computation:

$$\text{AttentionMap}(Q, K) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}). \tag{2.6}$$

where Q and V are the queries and keys matrices, respectively. In other words, it the matrix obtained after the *softmax* computation and prior to the dot product with the with the values matrix V described in Equation 2.5. A scheme of an attention map A is shown in Figure 2.7.

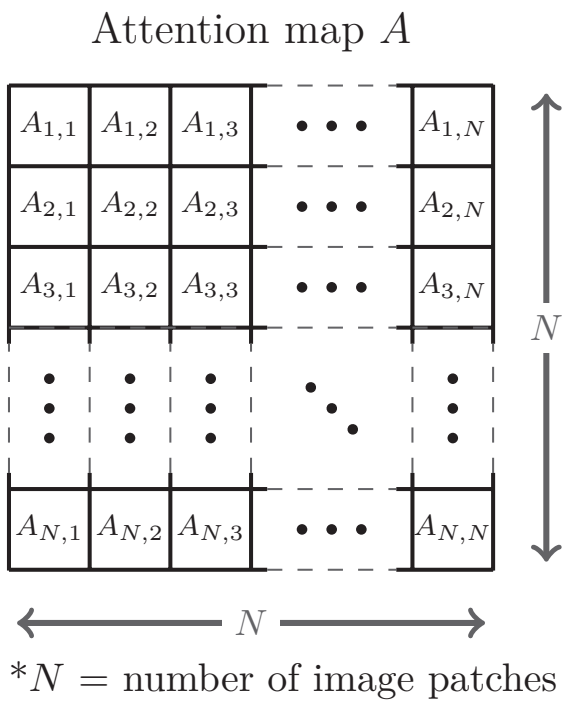


Figure 2.7: The Attention map A. It expresses the pairwise self-attention between all patches in a given input image. A row in A, for example A_1, indicates the self-attention between patch x_1 and all other patches. Image by the author.

Any given row $A_i \in A$ indicates the self-attention score between the i-th patch and all other patches. To better visualize which patches are being attended by an specific patch, and to perform a qualitative inspection of the attention map, each row in A can be individually reshaped to two-dimensions, as shown in Figure 2.8.

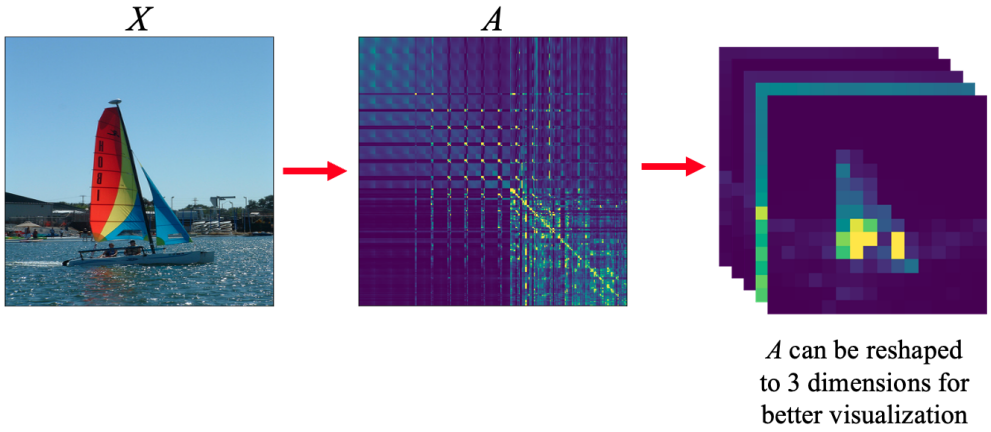


Figure 2.8: A real attention map A, computed from an input image X. On the right side, a example of the attention map reshaped to 3 dimensions — meaning that individual rows in A are being reshaped to 2 dimensions — in order to perform a qualitative analysis of the attention map. Image by the author.

2.3 Inductive Bias

In machine learning problems — and by consequence, in deep learning problems — choosing a model with the appropriate induction biases is crucial in the pursue for better generalization [200]. In short, inductive biases are the set of assumptions inherent to a model's architecture that favors certain solutions (combination of weight values) in detriment of others. In this section, we briefly describe the basic concepts of inductive bias, from the logical perspective of inductive reasoning, the role of inductive bias in machine learning, and finally the inductive biases of CNNs and vision transformers.

2.3.1 Inductive reasoning

Humans can induce a number of hypothesis given a single observation. For example, on the observation showed in Figure 2.9, if one would consider a scenario where the observer is travelling Japan for the first time in his life, and the temple on the photo is the first temple he has seen since arriving in Japan, he could draw a number of hypothesis based on that single observation. For example, he may assume the hypothesis that all temples in Japan are golden, or that all are located in the nature, or even that all are surrounded by a lake. This process of hypothesizing a general rule from examples is denoted *inductive reasoning* (or inductive inference) [201], and it starts with one observation (eg. a temple in Japan), and leads to a generalization hypothesis.

As showed by the example on Figure 2.9, it is possible to define a set of hypotheses based on a single observation. Nevertheless, one of the main properties of inductive reasoning is that a valid observation may lead to different hypothesis, which can be either true or false [202].

The biggest challenge of inductive reasoning is how to select a single hypothesis. Returning to

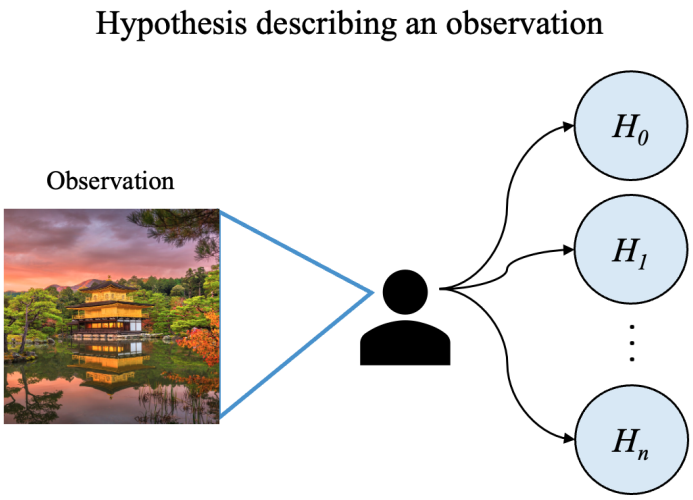


Figure 2.9: Hypothesis describing an observation. Image by the author.

the example on Figure 2.9, one way to select an hypothesis would be to select the simplest one: There are temples in Japan. This hypothesis is the simplest because it adds no extra constraints based on properties of the observation, such as color, nature and lakes. In other words, it is a simple generalization.

This method of selecting a hypothesis is denoted "Occam's razor" [203, 204]. It derived from a philosophical perspective and can be viewed as a basic inductive bias, for it determines that the simplest hypothesis that describes an observation should be chosen.

These processes of inductive reasoning are a fundamental and ubiquitous component of intelligent behavior. Formal studies of these processes raised a number of questions, which however are out of the scope of this work [201].

2.3.2 Inductive biases in Machine Learning

In Machine Learning (ML) tasks — and Deep Learning tasks — a model is trained on a subset of observations with the goal of creating a generalization rule from them [200]. Ideally, this generalization rule has to be valid for both seen and unseen data. That is to say, the goal is to obtain a rule that works for the whole population from a limited sample. In that scheme, the set of observations is the training dataset, and the hypotheses are the ML algorithm and all parameters that can be learned.

However, there is a infinite number of hypothesis for a finite number of examples, and choosing one hypothesis over another without a set a assumption would be extremely difficult. The prioritization of a set hypotheses (restriction of hypothesis space) is denoted *inductive bias*.

Choosing the model with the appropriate induction bias for the task at hand leads to better generalization, especially in a low data setting. The less training data, the stronger inductive biases

should be so the model can better generalize on unseen data. Contrarily, in a rich data setting, it may be preferable to avoid strong induction biases to allow the model to be less constrained and search through the hypothesis space freely [205, 206].

2.3.3 Inductive biases on CNNs

The remarkable success of CNNs on computer vision are often attributed to its inductive biases. Inductive biases like locality and weight sharing are biases specific to CNNs architecture. Locality implies that pixels spatially close to each other are related. Weight sharing implies that the search for specific patterns should be processed the same way on different parts of an image. Furthermore, the inductive biases of translation invariance and translation equivariance can be introduce by either using pooling layers or not, respectively [207].

2.3.4 Inductive biases on vision transformers

Transformers lack strong inductive biases, on one hand allowing them to search the hypotheses space more freely while on the other making them more data-hungry models [208, 209]. Therefore, on a data rich setting, vision transformers can find a better optimum, outperforming CNNs on several tasks [210–213]. Nevertheless, such vision transformers perform poorly on low data settings, and the injection of inductive biases might provide advantages to these models [214].

2.4 Regularization

In machine learning, regularization can be defined as a method that aims to restrict the hypothesis space, thus changing the resulting solution. It is commonly used to prevent overfitting during training and/or to improve the results on ill-posed problems [2, 215].

Regularization techniques can be roughly divided into two categories: *explicit* and *implicit*. The former refers to the practice of adding a regularization term to the objective function of a machine learning problem. The regularization term, also referred to as penalty term, has the function of imposing a cost to objective function being optimized, constraining the hypothesis space towards solutions that achieve an optimum trade-off between the objective function and the regularization term [216]. It is mostly used to improve the optimization of ill-posed problems. Implicit regularization, on the other hand, encompasses all the remaining forms of regularization. In this section, we introduce the basic concepts of explicit regularization, as well as its most commonly known techniques.

2.4.1 Explicit regularization

In machine learning, the objective of a classifier C, for example, is to correctly classify any given x. In order to do so, the classifier C attempts to approximate the function the best classifies any x by reducing a *loss function* given a limited set of training examples, as described in Equation (2.7).

$$\min_{C} \sum_{i=1}^{n} L(C(X_i), y_i), \tag{2.7}$$

where C is the classifier's function, L is the loss function, and X_i and y_i are the input data and its corresponding label.

Commonly, the way explicit regularization works is by adding a regularization term P to the loss function, with the purpose of imposing a penalty to the loss computation [215, 217], as described in Equation (2.8).

$$\min_{C} \sum_{i=1}^{n} L(C(X_i), y_i) + \lambda P(C), \tag{2.8}$$

where P is the regularization term and λ is a hyperparameter assigned to control the trade-off between the loss of predicting $C(x)$ and the penalty imposed by $P(C)$. The penalty is often computed as a product of the complexity of the classifier's weights and biases. In practice, regularization terms that penalize a function based on the complexity of their parameters can be interpreted as a method to enforce the Occam's razor inductive bias, by making simpler solutions more favorable through a smaller penalty [215, 218]. Nonetheless, from a Bayesian perspective, depending on what definitions are specified at the regularization term, it can be interpreted as a way of favoring certain prior distribution on model parameters [219]. Two regularization techniques widely used are the *Tikhonov regularization* [218] and the *Lasso regularization* [217], in which the former adopts the L_2-norm of the model's weights vector as a penalty term, and the latter takes the L_1-norm of the model's weights vector as a regularizer.

Aside from the effect of imposing certain prior distributions on a model, another application of regularization methods is reducing overfitting [219]. In other words, models can benefit from regularization to generalize better on unseen data.

2.5 Learning paradigms

Deep learning is a large field whose core is a neural network algorithm. The size of the neural network is determined by millions or even billions of trainable parameters [2]. Nevertheless, in spite of the neurobiological inspiration of DNNs, the sample efficiency in deep learning is much smaller than in humans or animal [220]. In other words, much more trials are necessary for a machine to

learn something when compared to a human. One of the main reasons for that, is that humans are able to draw background knowledge about the world, which allows them to learn new tasks much quicker than machines [221].

In this section we present more information about the performance gap between *learning* in humans and in machines, while briefly introducing the main learning paradigms of deep learning, with a focus on the self-supervised learning paradigm, which is adopted in this research.

2.5.1 Supervised Learning

Though supervised-learning has notably been the most explored learning paradigm, it often require large amounts of labeled data to train systems for specific use cases [215]. One could interpreted supervised learning as analogous to a classroom where the model is a student which is "taught" by a teacher through annotated examples.

In supervised learning, given a deep neural network E, an algorithm aims at finding a function f (combination of weights parameters in E) in a given hypothesis space F, such that $f : X \mapsto Y$, where X and Y are the input and the output space, respectively.

In a training set o N examples, a single data sample is a denoted (x_i,y_i), where x_i and y_i is are respectively the input data (i.e., feature vector) of the i-th example and its corresponding label.

Then, the learning happens by updating the the weight values in f using a gradient-based method in order to minimize a specified loss function L, as described in equation (refeq:supervised).

$$\min_{f} \sum_{i=1}^{N} L(f(X_i), y_i). \tag{2.9}$$

Up to now, most of the practical success in machine learning were attained through supervised learning. Its main applications in computer vision include image classification [13, 222, 223], object detection [12, 224–227], image segmentation [228], face recognition [229–231], person re-Identification [138, 232, 233], motion tracking [131–133], image colorization [234, 235] and image generation [236, 237]. Nevertheless, one of the main drawbacks in supervised learning is that it usually requires too many samples to predict a very small amount of information [220].

2.5.2 Reinforcement Learning

Reinforcement learning is a method used to train AI agents to learn environment behavior in specific contexts using reward feedback policy [238]. In other words, in reinforcement learning, unlike supervised learning, the feedback loop does not provide a correct answer to the model, instead, it just provides a signal of how good or bad the model's inference was.

Though reinforcement learning has proven to be great to perform tasks that can be simulated, such as games, the amount of trials necessary to achieve high levels of performance on those tasks is very high. For instance, using reinforcement learning, a machine might need as much as 80 hours to learn to play an *Atari* game to the same level a human would take only 15 minutes [239]. In Tian *et al.* [240], 20 million self-play games (running on 2000 GPUs for 14 days) were necessary to teach a machine to play the game *Go* at e world class level, which is in practice, many more games than any human can perform within a lifetime. Another study showed that, through reinforcement learning, a machine took an equivalent of 200 years of play time in order to learn how to play the game *Starcraft* at a high level [241]. Furthermore, another research using reinforcement learning took the equivalent of 10,000 years of simulation for a machine to learn how to solve the *Rubik's cube* [242].

The greatest drawback of reinforcement learning is that it requires too many trials, and in the real world that might not be a feasible possibility, since it is not possible to run the real world faster than real time [220]. Moreover, when applying reinforcement learning in the real world, unlike simulated environments, the negative signal for bad assumptions made by a model is also accompanied by real world consequences. For instance, a self driving car can cause an accident if learning in the real world. Therefore, reinforcement learning for real-life tasks often require "real world level" simulations.

2.5.3 Self-supervised learning

In 2019, Yann LeCun raised the question of how do humans and animals learn so quickly [238]. Infants are capable of learning a diverse spectrum of concepts, from language to intuitive physics, just by observing the world around them [243, 244]. For instance, Figure 2.10 illustrates the time line of the development of certain language skills on infants. At as early as 4 months of age, infants are capable of distinguishing vowels and even proper names. With 6 months of age, babies can perceive frequent words and typical consonants. And at 12 months they are already capable of producing their first words.

Moreover, Figure 2.11 displays the same timeline view regarding the learning of intuitive physics concepts. For instance, infants learn about object permanence by 3 months old, and are able to understand how gravity works by 9 months old.

Studies indicate that infants are capable of learning how the world works by observation [221]. The concept of *learning* is often connected to the ability to predict [238]. Prediction is the essence of intelligence, and humans learn models of the world by predicting [238]. In other words, humans develop a model of the world that allows them to predict the consequences of their actions.

One example, is the fact that humans can are capable to learn how to drive reasonably well in

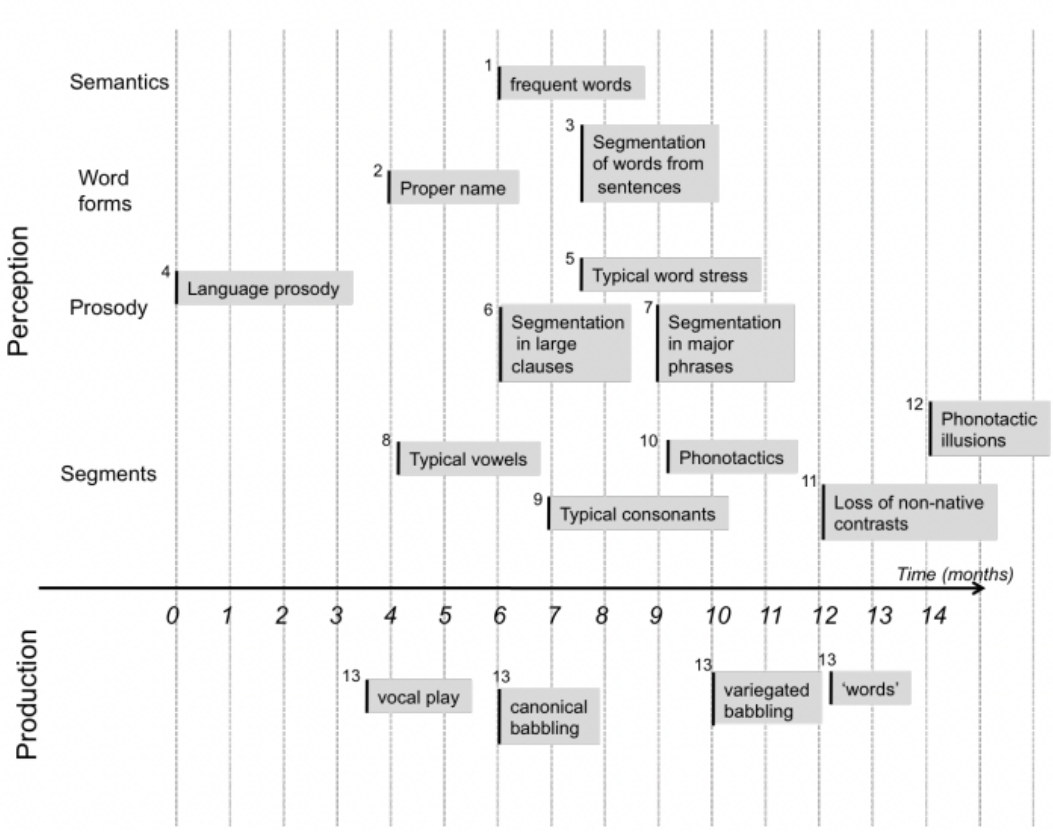


Figure 2.10: Studies illustrating the timeline of language development on infants. Boxes above the timeline indicate perception skills, and boxes bellow the timeline indicate production skills. The left edge of each box is aligned to the earliest age at which the result has been documented. Figure by Emmanuel Dupoux [243].

about 22 hours of practical training, mainly because of their background knowledge [245]. On the other hand, autonomous vehicles can take hundreds of thousands of hours on simulators to achieve a similar skill level [220].

In the context of deep learning, self-supervised learning can also be interpreted as *learning by prediction* [220]. In other words, it is not focused on learning a task, like classifying objects or segmenting a sentence. Self-supervised learning aims at learning the structure of the world just by observing the world, or as LeCun stated, self-supervised leaning aims to "predict everything from everything else" [238].

When compared to other learning paradigms, the main difference of self-supervised learning is the amount of feedback information received by the model at every trial. As described by LeCun (2019) [238], in reinforcement learning, the model is trained to predict a scalar reward amounting for only a few bits of information. In supervised learning, the machine attempts to predict a piece of human supplied data, such as categories or sequences. In this case, the amount of feedback information is usually within the range of 10 to 10,000 bits per trial. Finally, on self-supervised learning, the model is trained to predict any part of its input that has been omitted from it, such as words in a sequence, a region of an image, or future frames in a video. Therefore, on self-supervised

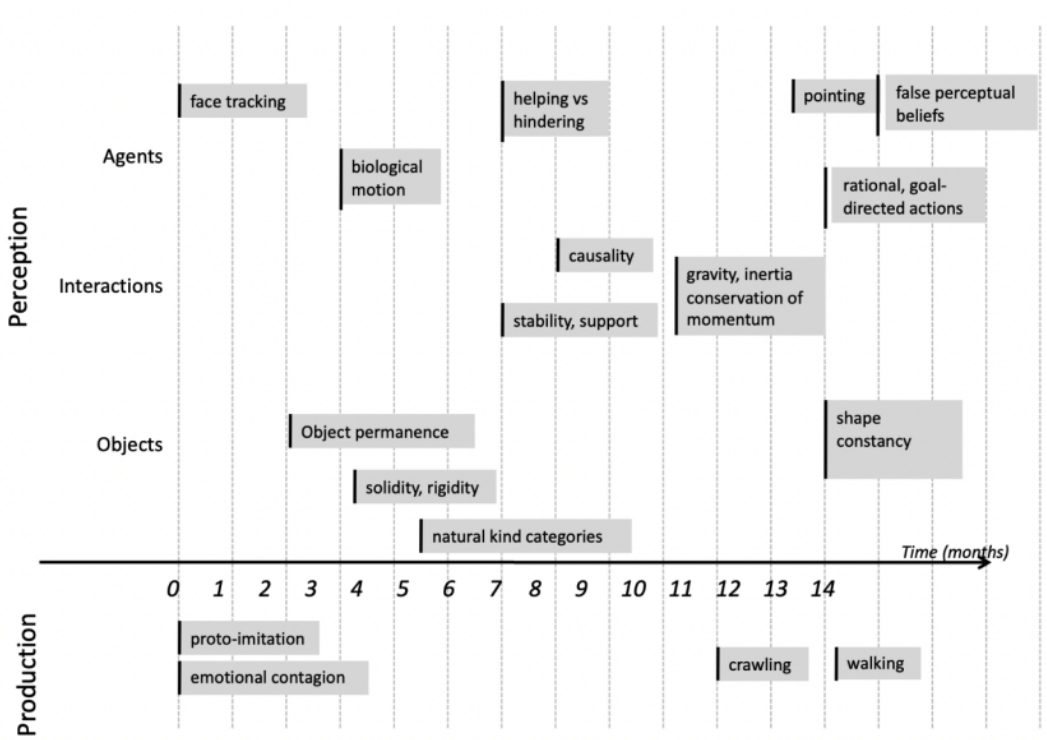


Figure 2.11: Landmark of intuitive physics acquisition in infants. Each box is an experiment showing a particular ability at a given age. Figure by Ronan Riochet [244].

learning, the feedback information typically amounts to millions of bits per sample.

Therefore, self-supervised learning (SSL) can be defined as the paradigm where the model trains itself to learn one part of the input — which is, by design, omitted during training — from another part of the input [246, 247]. Hence, it is also commonly referred to as predictive or pretext learning.

From a practical point of view, in this process, an unsupervised problem is transformed into a supervised problem through the *labels* that were generated from the input data. This method is also useful to take advantage of large quantities of unlabeled data [246]. However, its success is highly dependent on setting the right learning objectives in order to obtain strong supervised signals from the input data itself [248].

SSL first became popular in natural language processing (NLP) [249–253] and recent studies in computer vision tasks showed that it can be competitive with or even outperform supervised methods that used fully manually annotated datasets [254–259].

The process of the SSL method is to identify an hidden part of the input from any unhidden part of the input. The design of which information from the inputed data is omitted to be used as a label, and which information is used for training denotes a *pretext text*.

Over the past few years, several pretext-tasks have already been proposed in computer vision, such as image colorization [260–262], rotation estimation [263], inpainting (Figure 2.12) [264, 265], exemplar [266] and relative patch position [267].

Currently, one of the most successful approaches in SSL is pre-training a model using *pretext-tasks*

[268], then transferring the trained model to a supervised downstream task [269].

Figure 2.12: Semantic Inpainting results for context encoder trained jointly using reconstruction and adversarial loss. First four rows contain examples from Paris StreetView Dataset, and bottom row contains examples from ImageNet. Figure by Pathak *et al.* [265].

2.6 Related work

Emerged from natural language processing (NLP), the Transformer [95] is capable of computing the relationship between all elements in a given input sequence. Following the success in NLP, it has also achieved remarkable results in the field of computer vision by addressing diverse tasks such as recognition and classification [210, 211], detection [212] and segmentation [213].

When using the Transformer in NLP, a linear embedding, denoted as *token*, is generated from each word in the input sentence. Then, as the Transformer processes a given token in the sequence, *self-attention* takes information from all other tokens as well [95]. Hence, self-attention is a global operation, and by directly applying it to an image, Transformers processes it as a one dimensional sequence of embeddings of pixels. Therefore, in spite of its ability to capture long-range dependencies, the Transformer inherently lacks some inductive biases that efficiently aggregate contextual information, which is attributed to CNNs' success through translation equivariance and locality [208, 209].

By training on large-scale datasets and with longer training schedules, the vision transformer (ViT) proposed in Dosovitskiy *et al.* [199] overcame the architecture's lack of inductive biases, achieving excellent results on benchmark image recognition tasks. Considering its 3 variants (ViT-

Base, ViT-Large and ViT-Huge), the ViT ranges from 86M to 632M parameters, and state-of-the-art results were obtained when pre-training on the ImageNet-21k with 21k classes and 14M images [270], and the JFT-300 with 18k classes and 303M images [271]. Therefore, both the model capacity and the amount of training data impose a high computational cost.

Other works with vision transformers have both directly and indirectly dealt with its lack of inductive biases. Chu *et al.* [272] proposed a conditional positional encoding (CPE), which is dynamically generated and conditioned on the local neighborhood of the input tokens, which assisted in maintaining the desired translation-invariance in image classification tasks. In order to introduce CNNs' inductive biases into vision transformers, the combination of self-attention with convolution layers was also proposed in several works [273–275]. The LocalViT [276] introduces locality in vision transformer by adding a depth-wise convolution into the feed-forward network. The Conformer [277] fuses local features and global representations under different resolutions. The work of Touvron *et al.* [278] proposed to distill convolved knowledge by using a CNN model as a teacher network. Other works explored adding inductive biases by applying self-attention to local neighbors [211, 279]. In the work of Han *et al.* [280], self-attention is not only computed globally between image patches, but also locally, between smaller sub-patches inside each image patch.

Inductive biases can also be introduced in vision transformers through adaptations and modifications to its architecture. The LeViT, by Graham *et al.* [281], takes inspiration on the success of CNNs to introduce convolutional components to the transformer, replacing the column-like structure of the Transformer with a pyramid-like structure with pooling. The Pyramid Vision Transformer (PVT) [282] is also a progressive shrinking pyramid style transformer which reduces the computations of large feature maps. On the PVT, unlike regular vision transformers, for each query, self-attention is computed only with a sampled version of the input tokens. The Swin Transformer [283] uses a hierarchical representation that starts from small-sized patches which are gradually merged with their neighboring patches in deeper transformer layers. Then, self-attention is computed only on non-overlapping spatially grouped patches, in a approach that helps to introduce the inductive biases of locality, hierarchy and translation invariance [284, 285]. The Twins, proposed by Chu *et al.* [286], brings on advantages of both CPE and PVT to devise two novel spatial designs for vision transformers. Furthermore, the Vision Transformer Advanced by Exploring intrinsic Inductive Bias (ViTAE) introduces intrinsic scale invariance through a series of spatial pyramid reduction modules combined with multiple convolutions with different dilation rates, thus obtaining robust feature representation for objects at various scales [209].

2.7 References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943. [Online]. Available: http://link.springer.com/10.1007/BF02478259

[4] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009, publisher: Toronto, ON, Canada.

[5] L. Deng and Y. Liu, *Deep learning in natural language processing*. Springer, 2018.

[6] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[7] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 604–624, 2020.

[8] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8599–8603.

[9] L. Deng and J. Platt, "Ensemble deep learning for speech recognition," in *Proc. interspeech*, 2014.

[10] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, pp. 1–28, 2018.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3065386

[14] C. Prodi and T. Out, "History of artificial intelligence," *golden years*, vol. 1956, pp. 3–1, 1974.

[15] C. Fraser, "School of names," 2005.

[16] P. B. Ebrey, P. B. Ebrey, P. B. Ebrey, and P. B. Ebrey, *The Cambridge illustrated history of China*. Cambridge University Press Cambridge, 1996, vol. 1.

[17] I. P. McGreal, "Great thinkers of the eastern world the major thinkers and the philosophical and religious classics of china, india, japan, korea, and the world of islam," 1995.

[18] C. Hansen, "Language and logic in ancient china." Advanced Reasoning Forum, 2020.

[19] J. Needham and C. A. Ronan, *The Shorter Science and Civilisation in China: Volume 2*. Cambridge University Press, 1978, vol. 2.

[20] K. K. Chakrabarti, *Classical Indian philosophy of induction: the Nyaya viewpoint*. Lexington Books, 2010.

[21] K. K. Chakrabarti, *Classical Indian philosophy of mind: The Nyaya dualist tradition*. SUNY Press, 1999.

[22] D. Keown, "A dictionary of buddhism (oxford paperback reference s.)," 2004.

[23] R. Smith, *Prior analytics*. Hackett Publishing, 1989.

[24] J. S. Mill, *A System of Logic, Ratiocinative and Inductive: Being a Connected View of the Princilples of Evidence and the Methods of Scientific Investigation*. Harper and brothers, 1869.

[25] T. L. Heath *et al.*, *The thirteen books of Euclid's Elements*. Courier Corporation, 1956.

[26] H. Corbin, *The voyage and the messenger: Iran and philosophy*. North Atlantic Books, 1998.

[27] S. L. Montgomery and A. Kumar, *A history of Science in world cultures: voices of knowledge*. Routledge, 2015.

[28] A. Ben-Menahem, *Historical encyclopedia of natural and mathematical sciences*. Springer Science & Business Media, 2009.

[29] C. A. Pickover, *The math book: from Pythagoras to the 57th dimension, 250 milestones in the history of mathematics*. Sterling Publishing Company, Inc., 2009.

[30] P. King, "William of ockham: Summa logicae," in *Central Works of Philosophy*. Routledge, 2015, pp. 242–269.

[31] G. Priest and S. Read, "The formalization of ockham's theory of supposition," *Mind*, vol. 86, no. 341, pp. 109–113, 1977.

[32] J. Corcoran and J. Swiniarski, "Logical structures of ockham's theory of supposition," *Franciscan Studies*, vol. 38, pp. 161–183, 1978.

[33] J. Corcoran, "Ockham syllogistic semantics," in *Journal of Symbolic Logic*, vol. 46, no. 1. ASSN SYMBOLIC LOGIC INC 1325 SOUTH OAK ST, CHAMPAIGN, IL 61820, 1981, pp. 197–198.

[34] J. R. i Balaguer, "Historia de la filosofía española: Filosofía cristiana de los siglos xiii al xv, por tomás carreras y artau [y] joaquín carreras y artau. obra laureada por la asociación con el premio moret. tomos i-ii. madrid, asociación española para el p," *Estudis Romànics*, pp. 274–282, 1949.

[35] A. Bonner, "Doctor illuminatus: a ramón llull reader," 1993.

[36] P. McCorduck and C. Cfe, *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*. CRC Press, 2004.

[37] P. R. Norvig and S. A. Intelligence, "A modern approach," *Prentice Hall Upper Saddle River, NJ, USA: Rani, M., Nayak, R., & Vyas, OP (2015). An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. Knowledge-Based Systems*, vol. 90, pp. 33–48, 2002.

[38] D. Berlinski, *The advent of the algorithm: The idea that rules the world*. Houghton Mifflin, 2000.

[39] B. G. Buchanan, "A (very) brief history of artificial intelligence," *Ai Magazine*, vol. 26, no. 4, pp. 53–53, 2005.

[40] G. W. Leibniz, "Gottfried wilhelm leibniz," *Technology*, vol. 4, p. 1, 2011.

[41] M. Dascal, "Leibniz. language, signs and thought," *Leibniz. Language, Signs and Thought*, pp. 1–215, 1987.

[42] E. N. Zalta, *A (Leibnizian) theory of concepts*. na, 2000.

[43] W. Lenzen, "Leibniz: Logic," 2014.

[44] W. Durant and A. Durant, *The Age of reason begins: a history of European civilization in the period of Shakespeare, Bacon, Montaigne, Rembrandt, Galileo, and Descartes: 1558-1648*. Simon and Schuster, 1961, vol. 7.

[45] J. Joyce, "Bayes' theorem," 2003.

[46] S. B. McGrayne, *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, & Emerged Triumphant from Two Centuries of C.* Yale University Press, 2011.

[47] A. Edwards, "Commentary on the arguments of thomas bayes," *Scandinavian Journal of Statistics*, pp. 116–118, 1978.

[48] J. A. Paulos, "The mathematics of changing your mind," *New York Times (US)*, 2011.

[49] L. Mormille and F. Cozman, "Learning probabilistic relational models: A simplified framework, a case study, and a package," in *Proceedings of 5th Symposium on Knowledge Discovery, Mining and Learning, Uberlndia, Minas Gerais, Brazil*, 2017, pp. 129–136.

[50] P. E. Johnson, "The genesis and development of set theory," *The Two-Year College Mathematics Journal*, vol. 3, no. 1, pp. 55–62, 1972.

[51] J. Corcoran, "Aristotle's prior analytics and boole's laws of thought," *History and Philosophy of Logic*, vol. 24, no. 4, pp. 261–288, 2003.

[52] G. Boole, "The laws of thought," *Collected Logical Works*, vol. 2, 1952.

[53] G. Frege *et al.*, "Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought," *From Frege to Gödel: A source book in mathematical logic*, vol. 1931, pp. 1–82, 1879.

[54] W. Demopoulos, "Frege and the rigorization of analysis," *Journal of philosophical logic*, vol. 23, no. 3, pp. 225–245, 1994.

[55] D. Schlimm, "On frege's begriffsschrift notation for propositional logic: Design principles and trade-offs," *History and Philosophy of Logic*, vol. 39, no. 1, pp. 53–79, 2018.

[56] R. L. Mendelsohn, *The Philosophy of Gottlob Frege*. Cambridge University Press, 2005.

[57] P. McCorduck, M. Minsky, O. G. Selfridge, and H. A. Simon, "History of artificial intelligence." in *IJCAI*, 1977, pp. 951–954.

[58] A. N. Whitehead and B. Russell, *Principia Mathematica*. Cambridge University Press, 1925–1927.

[59] B. Buldt, "The scope of gödel's first incompleteness theorem," *Logica Universalis*, vol. 8, no. 3, pp. 499–552, 2014.

[60] K. Gödel, "Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i," *Monatshefte für mathematik und physik*, vol. 38, no. 1, pp. 173–198, 1931.

[61] M. Davis, "The incompleteness theorem," *Notices of the American Mathematical Society*, vol. 53, no. 4, pp. 414–418, 2006.

[62] A. Turing, "Turing machine," *Proc London Math Soc*, vol. 242, pp. 230–265, 1936.

[63] A. M. Turing, "Turing's treatise on enigma," *Unpublished Manuscript*, 1940.

[64] C. Severance, "Alan turing and bletchley park," *Computer*, vol. 45, no. 6, pp. 6–8, 2012.

[65] J. J. P. Eckert and J. W. Mauchly, "Electronic numerical integrator and computer," Feb. 4 1964, uS Patent 3,120,606.

[66] M. H. Weik, "The eniac story," *Ordnance*, vol. 45, no. 244, pp. 571–575, 1961.

[67] B. J. Copeland, "Colossus: its origins and originators," *IEEE Annals of the History of Computing*, vol. 26, no. 4, pp. 38–45, 2004.

[68] F. Carter, *Codebreaking with the Colossus Computer*. Bletchley Park Trust, 1996.

[69] T. Sale, "Colossus," in *Encyclopedia of Computer Science*, 2003, pp. 233–235.

[70] E. Jones, E. Miller, and W. Moodie, *How the Mind Works*. G. Allen & Unwin, 1945.

[71] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[72] C. Shannon, "Claude shannon," *Information Theory*, vol. 3, p. 224, 1948.

[73] B. J. Copeland, *The essential turing*. Clarendon Press, 2004.

[74] G. Dodig-Crnkovic, "Significance of models of computation, from turing model to natural computation," *Minds and Machines*, vol. 21, no. 2, pp. 301–322, 2011.

[75] D. Crevier, *AI: the tumultuous history of the search for artificial intelligence*. Basic Books, Inc., 1993.

[76] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[77] H. Moravec, *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.

[78] R. Cordeschi, *The discovery of the artificial: Behavior, mind and machines before and beyond cybernetics*. Springer Science & Business Media, 2002, vol. 28.

[79] H. A. Simon and A. Newell, "Computer science as empirical inquiry: symbols and search," *Communications of the ACM*, vol. 19, no. 3, pp. 11–126, 1976.

[80] L. Shustek, "An interview with ed feigenbaum," *Communications of the ACM*, vol. 53, no. 6, pp. 41–45, 2010.

[81] D. Lenat and E. Feigenbaum, "On the thresholds of knowledge," *Foundations of Artificial Intelligence, MIT Press, Cambridge, MA*, pp. 185–250, 1992.

[82] G. Kolata, "How can computers get common sense? two of the founders of the field of artificial intelligence disagree on how to make a thinking machine," *Science*, vol. 217, no. 4566, pp. 1237–1238, 1982.

[83] H. Kautz, "The third ai summer, aaai robert s. engelmore memorial lecture, thirty-fourth aaai conference on artificial intelligence, new york, ny, february 10, 2020," 2020.

[84] J. Howe, "Artificial intelligence at edinburgh university: A perspective," *Archived from the original on*, vol. 17, 2007.

[85] E. N. Zalta, U. Nodelman, C. Allen, and J. Perry, "Stanford encyclopedia of philosophy," 1995.

[86] P. Smolensky, "Grammar-based connectionist approaches to language," *Cognitive Science*, vol. 23, no. 4, pp. 589–613, 1999.

[87] J. W. Shavlik, "Combining symbolic and neural learning," *Machine Learning*, vol. 14, no. 3, pp. 321–331, 1994.

[88] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart, "Predicting the future: A connectionist approach," *International journal of neural systems*, vol. 1, no. 03, pp. 193–209, 1990.

[89] P. M. Todd, "A connectionist approach to algorithmic composition," *Computer Music Journal*, vol. 13, no. 4, pp. 27–43, 1989.

[90] D. E. Rumelhart, "The architecture of mind: A connectionist approach," *Mind readings*, pp. 207–238, 1998.

[91] S. E. Fahlman and G. E. Hinton, "Connectionist architectures for artificial intelligence," *Computer*, vol. 20, no. 01, pp. 100–109, 1987.

[92] P. Smolensky, "Connectionist ai, symbolic ai, and the brain," *Artificial Intelligence Review*, vol. 1, no. 2, pp. 95–109, 1987.

[93] G. E. Hinton, "Connectionist learning procedures," in *Machine learning*. Elsevier, 1990, pp. 555–610.

[94] B. Mitchell and J. Sheppard, "Deep structure learning: Beyond connectionist approaches," in *2012 11th International Conference on Machine Learning and Applications*, vol. 1. IEEE, 2012, pp. 162–167.

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

[96] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

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

[98] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[99] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[100] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021.

[101] T. M. Mitchell and T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997, vol. 1, no. 9.

[102] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[103] Y.-Y. Song and L. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[104] J. Yadav and M. Sharma, "A review of k-mean algorithm," *Int. J. Eng. Trends Technol*, vol. 4, no. 7, pp. 2972–2976, 2013.

[105] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[106] J. Manyika, "Getting ai right: Introductory notes on ai & society," pp. 5–27, 2022.

[107] H. Robinson, "Dualism," 2002.

[108] R. M. Young, "The mind–body problem," in *Companion to the history of modern science*. Routledge, 2020, pp. 702–711.

[109] S. Cash and R. Yuste, "Linear summation of excitatory inputs by ca1 pyramidal neurons," *Neuron*, vol. 22, no. 2, pp. 383–394, 1999.

[110] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current opinion in neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[111] D. L. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nature neuroscience*, vol. 19, no. 3, pp. 356–365, 2016.

[112] U. Güçlü and M. A. van Gerven, "Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream," *Journal of Neuroscience*, vol. 35, no. 27, pp. 10 005–10 014, 2015.

[113] M. Zorzi and A. Testolin, "An emergentist perspective on the origin of number sense," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 373, no. 1740, p. 20170043, 2018.

[114] D. Dubhashi and S. Lappin, "Ai dangers: Imagined and real," *Communications of the ACM*, vol. 60, no. 2, pp. 43–45, 2017.

[115] S. Makridakis, "The forthcoming artificial intelligence (ai) revolution: Its impact on society and firms," *Futures*, vol. 90, pp. 46–60, 2017.

[116] K. Karger, "Will superintelligence take over?: A critical reflection on the apparent and actual dangers of artificial intelligence," in *Artificial Intelligence*. Brill mentis, 2020, pp. 87–103.

[117] M. L. Littman, I. Ajunwa, G. Berger, C. Boutilier, M. Currie, F. Doshi-Velez, G. Hadfield, M. C. Horowitz, C. Isbell, H. Kitano *et al.*, "Gathering strength, gathering storms: The one

hundred year study on artificial intelligence (ai100) 2021 study panel report," *arXiv preprint arXiv:2210.15767*, 2022.

[118] B. M. Smith, "Fight fire with fire: Fend off the dangers of artificial intelligence by designing regulations that employ ai run inspections," 2022.

[119] D. Acemoğlu, D. Autor, J. Hazell, P. Restrepo, E. Calvano, G. Calzolari, V. Denicolò, S. Pastorello, A. Agrawal, J. Gans *et al.*, "Dangers of unregulated artificial intelligence," 2021.

[120] D. J. Fuchs, "The dangers of human-like bias in machine-learning algorithms," *Missouri S&T's Peer to Peer*, vol. 2, no. 1, p. 1, 2018.

[121] A. Burkert, "Ethics and the dangers of artificial intelligence," *ATZ worldwide*, vol. 119, no. 11, pp. 8–13, 2017.

[122] F. Duarte and C. Ratti, "The impact of autonomous vehicles on cities: A review," *Journal of Urban Technology*, vol. 25, no. 4, pp. 3–18, 2018.

[123] A. Faisal, M. Kamruzzaman, T. Yigitcanlar, and G. Currie, "Understanding autonomous vehicles," *Journal of transport and land use*, vol. 12, no. 1, pp. 45–72, 2019.

[124] Y. Wiseman, "Autonomous vehicles," in *Research Anthology on Cross-Disciplinary Designs and Applications of Automation*. IGI Global, 2022, pp. 878–889.

[125] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[126] A. Ghodselahi and A. Amirmadhi, "Application of artificial intelligence techniques for credit risk evaluation," *International Journal of Modeling and Optimization*, vol. 1, no. 3, p. 243, 2011.

[127] D. Van Thiel and W. F. F. Van Raaij, "Artificial intelligence credit risk prediction: An empirical study of analytical artificial intelligence tools for credit risk prediction in a digital era," *Journal of Risk Management in Financial Institutions*, vol. 12, no. 3, pp. 268–286, 2019.

[128] D. Mhlanga, "Financial inclusion in emerging economies: The application of machine learning and artificial intelligence in credit risk assessment," *International Journal of Financial Studies*, vol. 9, no. 3, p. 39, 2021.

[129] M. Islam, L. Zhou, F. Li *et al.*, "Application of artificial intelligence (artificial neural network) to assess credit risk: a predictive model for credit card scoring," 2009.

[130] S. Bhatore, L. Mohan, and Y. R. Reddy, "Machine learning techniques for credit risk evaluation: a systematic literature review," *Journal of Banking and Financial Technology*, vol. 4, no. 1, pp. 111–138, 2020.

[131] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[132] G. Chandan, A. Jain, H. Jain *et al.*, "Real time object detection and tracking using deep learning and opencv," in *2018 International Conference on inventive research in computing applications (ICIRCA)*. IEEE, 2018, pp. 1305–1308.

[133] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, "Deep learning in multi-object detection and tracking: state of the art," *Applied Intelligence*, vol. 51, no. 9, pp. 6400–6429, 2021.

[134] S. Balaban, "Deep learning and face recognition: the state of the art," *Biometric and surveillance technology for human and activity identification XII*, vol. 9457, pp. 68–75, 2015.

[135] W. Wang, J. Yang, J. Xiao, S. Li, and D. Zhou, "Face recognition based on deep learning," in *International Conference on Human Centered Computing*. Springer, 2014, pp. 812–820.

[136] H. Wang and L. Guo, "Research on face recognition based on deep learning," in *2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*. IEEE, 2021, pp. 540–546.

[137] P. S. Prasad, R. Pathak, V. K. Gunjan, and H. Ramana Rao, "Deep learning based representation for face recognition," in *ICCCE 2019*. Springer, 2020, pp. 419–424.

[138] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2275–2284.

[139] D. Koehn, S. Lessmann, and M. Schaal, "Predicting online shopping behaviour from clickstream data using deep learning," *Expert Systems with Applications*, vol. 150, p. 113342, 2020.

[140] N. S. Reddy, "Optimal feature selection and hybrid deep learning for direct marketing campaigns in banking applications," *Evolutionary Intelligence*, vol. 15, no. 3, pp. 1969–1990, 2022.

[141] A. Micu, A. Capatina, D. S. Cristea, D. Munteanu, A.-E. Micu, and D. A. Sarpe, "Assessing an on-site customer profiling and hyper-personalization system prototype based on a deep learning approach," *Technological Forecasting and Social Change*, vol. 174, p. 121289, 2022.

[142] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 627–636.

[143] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017.

[144] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 278–288.

[145] A. Singhal, P. Sinha, and R. Pant, "Use of deep learning in modern recommendation system: A summary of recent works," *arXiv preprint arXiv:1712.07525*, 2017.

[146] A. Da'u and N. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2709–2748, 2020.

[147] H. T. Nguyen, M. Wistuba, J. Grabocka, L. R. Drumond, and L. Schmidt-Thieme, "Personalized deep learning for tag recommendation," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 186–197.

[148] M. Schedl, "Deep learning in music recommendation systems," *Frontiers in Applied Mathematics and Statistics*, p. 44, 2019.

[149] F. Fessahaye, L. Perez, T. Zhan, R. Zhang, C. Fossier, R. Markarian, C. Chiu, J. Zhan, L. Gewali, and P. Oh, "T-recsys: A novel music recommendation system using deep learning," in *2019 IEEE international conference on consumer electronics (ICCE)*. IEEE, 2019, pp. 1–6.

[150] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and electronics in agriculture*, vol. 147, pp. 70–90, 2018.

[151] L. Santos, F. N. Santos, P. M. Oliveira, and P. Shinde, "Deep learning applications in agriculture: A short review," in *Iberian Robotics conference*. Springer, 2019, pp. 139–151.

[152] N. Zhu, X. Liu, Z. Liu, K. Hu, Y. Wang, J. Tan, M. Huang, Q. Zhu, X. Ji, Y. Jiang *et al.*, "Deep learning for smart agriculture: Concepts, tools, applications, and opportunities," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 4, pp. 32–44, 2018.

[153] Q. Zhang, Y. Liu, C. Gong, Y. Chen, and H. Yu, "Applications of deep learning for dense scenes analysis in agriculture: A review," *Sensors*, vol. 20, no. 5, p. 1520, 2020.

[154] Y.-Y. Zheng, J.-L. Kong, X.-B. Jin, X.-Y. Wang, T.-L. Su, and M. Zuo, "Cropdeep: the crop vision dataset for deep-learning-based classification and detection in precision agriculture," *Sensors*, vol. 19, no. 5, p. 1058, 2019.

[155] T. T. Nguyen, T. D. Hoang, M. T. Pham, T. T. Vu, T. H. Nguyen, Q.-T. Huynh, and J. Jo, "Monitoring agriculture areas with satellite images and deep learning," *Applied Soft Computing*, vol. 95, p. 106565, 2020.

[156] M. H. Saleem, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," *Precision Agriculture*, vol. 22, no. 6, pp. 2053–2091, 2021.

[157] X. Liu, L. Faes, A. U. Kale, S. K. Wagner, D. J. Fu, A. Bruynseels, T. Mahendiran, G. Moraes, M. Shamdas, C. Kern *et al.*, "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis," *The lancet digital health*, vol. 1, no. 6, pp. e271–e297, 2019.

[158] G. Hinton, "Deep learning—a technology with the potential to transform health care," *Jama*, vol. 320, no. 11, pp. 1101–1102, 2018.

[159] S. A. Bini, "Artificial intelligence, machine learning, deep learning, and cognitive computing: what do these terms mean and how will they impact health care?" *The Journal of arthroplasty*, vol. 33, no. 8, pp. 2358–2361, 2018.

[160] L. Faes, S. K. Wagner, D. J. Fu, X. Liu, E. Korot, J. R. Ledsam, T. Back, R. Chopra, N. Pontikos, C. Kern *et al.*, "Automated deep learning design for medical image classification by health-care professionals with no coding experience: a feasibility study," *The Lancet Digital Health*, vol. 1, no. 5, pp. e232–e242, 2019.

[161] J.-G. Lee, S. Jun, Y.-W. Cho, H. Lee, G. B. Kim, J. B. Seo, and N. Kim, "Deep learning in medical imaging: general overview," *Korean journal of radiology*, vol. 18, no. 4, pp. 570–584, 2017.

[162] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on mri," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.

[163] M. Kim, J. Yun, Y. Cho, K. Shin, R. Jang, H.-j. Bae, and N. Kim, "Deep learning in medical imaging," *Neurospine*, vol. 16, no. 4, p. 657, 2019.

[164] T. Liu, E. Siegel, and D. Shen, "Deep learning and medical image analysis for covid-19 diagnosis and prediction," *Annual Review of Biomedical Engineering*, vol. 24, 2022.

[165] S. Das, L. Jain, and A. Das, "Deep learning for military image captioning," in *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018, pp. 2165–2171.

[166] M. Zhang, H. Li, G. Xia, W. Zhao, S. Ren, and C. Wang, "Research on the application of deep learning target detection of engineering vehicles in the patrol and inspection for military optical cable lines by uav," in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1. IEEE, 2018, pp. 97–101.

[167] H. Li, L. Yu, J. Zhang, and M. Lyu, "Fusion deep learning and machine learning for heterogeneous military entity recognition," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[168] X. Wang, R. Yang, Y. Lu, and Q. Wu, "Military named entity recognition method based on deep learning," in *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*. IEEE, 2018, pp. 479–483.

[169] Y. Wang, X. Ning, B. Leng, and H. Fu, "Ship detection based on deep learning," in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2019, pp. 275–279.

[170] A. Aouto, T. Huynh-The, J.-M. Lee, and D.-S. Kim, "Pose-based identification using deep learning for military surveillance systems," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2019, pp. 626–629.

[171] D. Namiot, E. Ilyushin, and I. Chizhov, "Military applications of machine learning," *International Journal of Open Information Technologies*, vol. 10, no. 1, pp. 69–76, 2021.

[172] S. McElwee, J. Heaton, J. Fraley, and J. Cannady, "Deep learning for prioritizing and responding to intrusion detection alerts," in *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 2017, pp. 1–5.

[173] C. Sagan, *Conversations with Carl Sagan*. Univ. Press of Mississippi, 2006.

[174] D. Greene, A. L. Hoffmann, and L. Stark, "Better, nicer, clearer, fairer: A critical assessment of the movement for ethical artificial intelligence and machine learning," 2019.

[175] R. Sun and F. Alexandre, *Connectionist-symbolic integration: From unified to hybrid approaches*. Psychology Press, 2013.

[176] M. Garnelo and M. Shanahan, "Reconciling deep learning with symbolic artificial intelligence: representing objects and relations," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 17–23, 2019.

[177] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence: Current trends," *arXiv preprint arXiv:2105.05330*, 2021.

[178] A. d. Garcez, S. Bader, H. Bowman, L. C. Lamb, L. de Penning, B. Illuminoo, H. Poon, and C. Gerson Zaverucha, "Neural-symbolic learning and reasoning: A survey and interpretation," *Neuro-Symbolic Artificial Intelligence: The State of the Art*, vol. 342, p. 1, 2022.

[179] V. Golovko, A. Kroshchanka, M. Kovalev, V. Taberko, and D. Ivaniuk, "Neuro-symbolic artificial intelligence: Application for control the quality of product labeling," in *International Conference on Open Semantic Technologies for Intelligent Systems*. Springer, 2020, pp. 81–101.

[180] D. O. Hebb, "Animal and Physiological Psychology," *Annual Review of Psychology*, vol. 1, no. 1, pp. 173–188, 1950.

[181] S. Haykin, *Neural Networks and Learning Machines, 3/E*. Pearson Education India, 2010.

[182] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[183] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[184] K. Gurney, *An introduction to neural networks*. CRC press, 2018.

[185] R. A. Alzahrani and A. C. Parker, "Neuromorphic circuits with neural modulation enhancing the information content of neural signaling," in *International Conference on Neuromorphic Systems 2020*, 2020, pp. 1–8.

[186] A. Karpathy *et al.*, "Convolutional neural networks for visual recognition," *Notes accompany the Stanford CS class CS231*, 2017.

[187] J. B. Ahire, "The Artificial Neural Networks Handbook: Part 4 - DZone AI," Date Accessed: 2022-12-10. [Online]. Available: https://dzone.com/articles/the-artificial-neural-networks-handbook-part-4

[188] H. N. Mhaskar and C. A. Micchelli, "How to Choose an Activation Function," in *Proceeding of Advances in Neural Information Processing Systems*, vol. 6, 1994.

[189] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[190] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[191] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[192] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[193] N. Shibuya, "Transformer's Encoder-Decoder: Let's Understand The Model Architecture," Date Accessed: 2022-12-28. [Online]. Available: https://kikaben.com/transformers-encoder-decoder/

[194] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2286–2296.

[195] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?" *arXiv preprint arXiv:1805.12177*, 2018.

[196] M. V. Valueva, N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and computers in simulation*, vol. 177, pp. 232–243, 2020.

[197] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[198] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[199] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

[200] T. M. Mitchell, *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ..., 1980.

[201] D. Angluin and C. H. Smith, "Inductive Inference: Theory and Methods," *ACM Computing Surveys*, vol. 15, no. 3, pp. 237–269, Sep. 1983. [Online]. Available: https://dl.acm.org/doi/10.1145/356914.356918

[202] J. McCall, "Induction: From Kolmogorov and Solomonoff to De Finetti and Back to Kolmogorov - McCall - 2004 - Metroeconomica - Wiley Online Library." [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0026-1386.2004.00190.x?casa_token=CuZ-QV3GqDsAAAAA:8G2I9gNaZ7zTUJiqziPPHrajPpD-TyIEkRluqElVZfjSk2v2Tgh33eVHZuzfDBbOJ_oIgjr4Ya-jGDxk

[203] D. Stork, "Foundations of Occam's razor and parsimony in learning," in *NIPS 2001 Workshop*, 2001.

[204] J. Schaffer, "What Not to Multiply Without Necessity," *Australasian Journal of Philosophy*, vol. 93, no. 4, pp. 644–664, Oct. 2015. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/00048402.2014.992447

[205] S. Ritter, D. G. T. Barrett, A. Santoro, and M. M. Botvinick, "Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study," Jun. 2017, arXiv:1706.08606 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1706.08606

[206] H. Hosseini, B. Xiao, M. Jaiswal, and R. Poovendran, "Assessing shape bias property of convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1923–1931, 2018.

[207] C. Mouton, J. C. Myburgh, and M. H. Davel, "Stride and translation invariance in cnns," in *Southern African Conference for Artificial Intelligence Research*. Springer, 2021, pp. 267–281.

[208] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, and Y. Xu, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, 2022, iSBN: 0162-8828 Publisher: IEEE.

[209] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, "Vitae: Vision transformer advanced by exploring intrinsic inductive bias," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 522–28 535, 2021.

[210] H. Zhao, J. Jia, and V. Koltun, "Exploring Self-Attention for Image Recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 10 073–10 082. [Online]. Available: https://ieeexplore.ieee.org/document/9156532/

[211] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[212] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," *arXiv:2005.12872 [cs]*, May 2020, arXiv: 2005.12872. [Online]. Available: http://arxiv.org/abs/2005.12872

[213] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," p. 13.

[214] K. Morrison, B. Gilby, C. Lipchak, A. Mattioli, and A. Kovashka, "Exploring Corruption Robustness: Inductive Biases in Vision Transformers and MLP-Mixers," Jul. 2021, arXiv:2106.13122 [cs]. [Online]. Available: http://arxiv.org/abs/2106.13122

[215] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[216] P. Bühlmann and S. Van De Geer, *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[217] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[218] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[219] K. Weinberger, "Linear Regression," Date Accessed: 2022-07-15. [Online]. Available: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote08.html#map-estimate

[220] Y. LeCun, "Self-Supervised Learning," 2020, Date Accessed: 2022-11-02. [Online]. Available: https://drive.google.com/file/d/1r-mDL4IX_hzZLDBKp8_e8VZqD7fOzBkF/view

[221] S. J. Hespos and K. VanMarle, "Physics for infants: Characterizing the origins of knowledge about objects, substances, and number," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 3, no. 1, pp. 19–27, 2012.

[222] C. Affonso, A. L. D. Rossi, F. H. A. Vieira, A. C. P. de Leon Ferreira *et al.*, "Deep learning for biological image classification," *Expert systems with applications*, vol. 85, pp. 114–122, 2017.

[223] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using imagenet pretrained networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 105–109, 2015.

[224] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast yolo: A fast you only look once system for real-time embedded object detection in video," *arXiv preprint arXiv:1709.05943*, 2017.

[225] Y. Yang and H. Deng, "Gc-yolov3: You only look once with global context block," *Electronics*, vol. 9, no. 8, p. 1235, 2020.

[226] Z. Zhou, J. Zhang, and C. Gong, "Automatic detection method of tunnel lining multi-defects via an enhanced you only look once network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 6, pp. 762–780, 2022.

[227] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[228] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv:1505.04597 [cs]*, May 2015, arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597

[229] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[230] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[231] M. Lai, "Deep learning for medical image segmentation," *arXiv preprint arXiv:1505.02000*, 2015.

[232] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 6, pp. 2872–2893, 2021.

[233] K. Wang, H. Wang, M. Liu, X. Xing, and T. Han, "Survey on person re-identification based on deep learning," *CAAI Transactions on Intelligence Technology*, vol. 3, no. 4, pp. 219–227, 2018.

[234] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, and A. W. Muzaffar, "Image colorization: A survey and dataset," *arXiv preprint arXiv:2008.10774*, 2020.

[235] S. Huang, X. Jin, Q. Jiang, and L. Liu, "Deep learning for image colorization: Current and future prospects," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105006, 2022.

[236] T. Iqbal and H. Ali, "Generative adversarial network for medical images (mi-gan)," *Journal of medical systems*, vol. 42, no. 11, pp. 1–11, 2018.

[237] S. Minaee and A. Abdolrashidi, "Iris-gan: Learning to generate realistic iris images using convolutional gan," *arXiv preprint arXiv:1812.04822*, 2018.

[238] Y. LeCun, "Energy-Based Self-Supervised Learning," 2019, Date Accessed: 2022-09-06. [Online]. Available: http://helper.ipam.ucla.edu/publications/mlpws4/mlpws4_15927.pdf

[239] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[240] Y. Tian, J. Ma, Q. Gong, S. Sengupta, Z. Chen, J. Pinkerton, and L. Zitnick, "Elf opengo: An analysis and open reimplementation of alphazero," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6244–6253.

[241] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[242] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[243] E. Dupoux, "Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner," *Cognition*, vol. 173, pp. 43–59, 2018.

[244] R. Riochet, M. Y. Castro, M. Bernard, A. Lerer, R. Fergus, V. Izard, and E. Dupoux, "Intphys 2019: A benchmark for visual intuitive physics understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[245] D. Press, "How Long Does It Take You to Learn How to Drive?" Date Accessed: 2022-12-28. [Online]. Available: https://drivingpress.com/how-long-does-it-take-to-learn-to-drive/

[246] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

[247] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *arXiv preprint arXiv:2206.13188*, 2022.

[248] M. C. Schiappa, Y. S. Rawat, and M. Shah, "Self-supervised learning for videos: A survey," *ACM Computing Surveys*, 2022.

[249] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019, arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805

[250] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv:1802.05365 [cs]*, Mar. 2018, arXiv: 1802.05365. [Online]. Available: http://arxiv.org/abs/1802.05365

[251] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," p. 12.

[252] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," p. 24.

[253] A. M. Dai and Q. V. Le, "Semi-supervised Sequence Learning," *arXiv:1511.01432 [cs]*, Nov. 2015, arXiv: 1511.01432. [Online]. Available: http://arxiv.org/abs/1511.01432

[254] C. Doersch and A. Zisserman, "Multi-task Self-Supervised Visual Learning," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 2070–2079. [Online]. Available: http://ieeexplore.ieee.org/document/8237488/

[255] T. H. Trinh, M.-T. Luong, and Q. V. Le, "Selfie: Self-supervised Pretraining for Image Embedding," *arXiv:1906.02940 [cs, eess, stat]*, Jul. 2019, arXiv: 1906.02940. [Online]. Available: http://arxiv.org/abs/1906.02940

[256] M. Minderer, O. Bachem, N. Houlsby, and M. Tschannen, "Automatic Shortcut Removal for Self-Supervised Representation Learning," *arXiv:2002.08822 [cs]*, Jun. 2020, arXiv: 2002.08822. [Online]. Available: http://arxiv.org/abs/2002.08822

[257] I. Misra and L. van der Maaten, "Self-Supervised Learning of Pretext-Invariant Representations," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 6706–6716. [Online]. Available: https://ieeexplore.ieee.org/document/9156540/

[258] O. J. Hénaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, and S. M. A. Eslami, "Data-Efficient Image Recognition with Contrastive Predictive Coding," p. 11.

[259] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *2020 IEEE/CVF Conference on Computer Vision and*

Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, Jun. 2020, pp. 9726–9735. [Online]. Available: https://ieeexplore.ieee.org/document/9157636/

[260] R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorization," *arXiv:1603.08511 [cs]*, Oct. 2016, arXiv: 1603.08511. [Online]. Available: http://arxiv.org/abs/1603.08511

[261] A. Deshpande, J. Rock, and D. Forsyth, "Learning Large-Scale Automatic Image Colorization," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, Dec. 2015, pp. 567–575. [Online]. Available: http://ieeexplore.ieee.org/document/7410429/

[262] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning Representations for Automatic Colorization," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 577–593.

[263] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[264] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks," p. 9.

[265] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[266] I. Katircioglu, H. Rhodin, V. Constantin, J. Spörri, M. Salzmann, and P. Fua, "Self-supervised Segmentation via Background Inpainting," *arXiv:2011.05626 [cs]*, Nov. 2020, arXiv: 2011.05626. [Online]. Available: http://arxiv.org/abs/2011.05626

[267] M. Noroozi and P. Favaro, "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles," *arXiv:1603.09246 [cs]*, Aug. 2017, arXiv: 1603.09246. [Online]. Available: http://arxiv.org/abs/1603.09246

[268] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised Visual Representation Learning by Context Prediction," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, Dec. 2015, pp. 1422–1430. [Online]. Available: http://ieeexplore.ieee.org/document/7410524/

[269] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and Benchmarking Self-Supervised Visual Representation Learning," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 6390–6399. [Online]. Available: https://ieeexplore.ieee.org/document/9010709/

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

[271] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 843–852. [Online]. Available: http://ieeexplore.ieee.org/document/8237359/

[272] X. Chu, Z. Tian, B. Zhang, X. Wang, X. Wei, H. Xia, and C. Shen, "Conditional Positional Encodings for Vision Transformers," *arXiv:2102.10882 [cs]*, Mar. 2021, arXiv: 2102.10882. [Online]. Available: http://arxiv.org/abs/2102.10882

[273] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing Convolutions to Vision Transformers," *arXiv:2103.15808 [cs]*, Mar. 2021, arXiv: 2103.15808. [Online]. Available: http://arxiv.org/abs/2103.15808

[274] H. Yan, Z. Li, W. Li, C. Wang, M. Wu, and C. Zhang, "ConTNet: Why not use convolution and transformer at the same time?" *arXiv:2104.13497 [cs]*, May 2021, arXiv: 2104.13497. [Online]. Available: http://arxiv.org/abs/2104.13497

[275] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, "Incorporating Convolution Designs into Visual Transformers," *arXiv:2103.11816 [cs]*, Apr. 2021, arXiv: 2103.11816. [Online]. Available: http://arxiv.org/abs/2103.11816

[276] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "LocalViT: Bringing Locality to Vision Transformers," *arXiv:2104.05707 [cs]*, Apr. 2021, arXiv: 2104.05707. [Online]. Available: http://arxiv.org/abs/2104.05707

[277] Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye, "Conformer: Local features coupling global representations for visual recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 367–376.

[278] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," p. 11.

[279] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International conference on machine learning*. PMLR, 2018, pp. 4055–4064.

[280] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 908–15 919, 2021.

[281] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze, "Levit: a vision transformer in convnet's clothing for faster inference," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 259–12 269.

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

[283] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[284] Z. Xie, Y. Lin, Z. Yao, Z. Zhang, Q. Dai, Y. Cao, and H. Hu, "Self-supervised learning with swin transformers," *arXiv preprint arXiv:2105.04553*, 2021.

[285] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3202–3211.

[286] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9355–9366, 2021.

Chapter 3

Two-dimensional distance based Self-attention regularization

In opposition to the foolish ignorabimus our slogan shall be: "We must know - we will know!"

—**David Hilbert**

(Mathematician)

A computer would deserve to be called intelligent if it could deceive a human into believing that it was human.

—**Alan Turing**

(Mathematician and Computer Scientist)

This chapter presents a new *self-attention regularization* method designed with the purpose of introducing a new inductive bias on vision transformers. This method is based on the pairwise two-dimensional distance between image patches, which is measured through the *Manhattan distance*. Section 3.1 introduces the conception of the method. Two basic concepts of vision transformers are introduced in section 3.2. Section 3.3 details the assumption made by this method. Section 3.4 formally describes the two-dimensional distance based self-attention regularization method. And finally, section 3.5 presents a summary of this method.

3.1 Conception of the method

Regularization is a technique in machine learning that has been utilized for decades with the purpose of reducing overfitting and improving a model's generalization [1]. In practice, regularization can also be interpreted as a mean to restrict the hypothesis space by penalizing certain solutions, thus favoring certain prior distributions on model parameters [2]. A typical approach to regularization

Parts of the content of this chapter were previously published in Mormille *et al.*, 2022 (see Appendix C)

is to add a regularization term (penalty term) to the objective function of a machine learning problem, as shown in Equation (3.1).

$$\min_{C} \sum_{i=1}^{n} L(C(X_i), y_i) + \lambda P(C), \tag{3.1}$$

where and X_i and y_i are the input data and its corresponding label, C is the objective function, L is a loss function, P is the regularization term and λ is a hyperparameter assigned to control the trade-off between the loss of predicting $C(X)$ and the penalty imposed by $P(C)$.

Therefore, the goal of the regularization term is to, based on a criteria, impose a cost (penalty) to the objective function being optimized. By controlling the trade-off between the objective function and the regularization term, solutions that would eventually result in a high penalty are avoided, creating a inductive bias towards solutions that result in smaller penalties [3].

One common approach to regularization is that in which the regularization term P penalizes solutions based on the complexity of their parameters, which can also be seen as a method to enforce the Occam's razor inductive bias [1, 4]. The *Tikhonov regularization* and the *Lasso regularization* are two popular methods to penalize solutions based on the complexity of their parameters, with the former taking the $L2$-norm of the model's weights vector as a penalty term [4], and the latter taking the $L1$-norm of the model's weights vector as a penalty term [5].

Nevertheless, the regularization method presented in this chapter proposes not to restrict the hypothesis space based on the complexity of the solution. Instead, it favors certain solutions in detriment of others based on assumptions. In other words, an assumption (or set of assumptions) about the problem's solution is translated into a computer program encapsulated by a regularization term to be added to machine learning's objective function, as shown in Figure 3.1.

This method takes partial inspiration from the *Symbolic approach* to Artificial Intelligence (AI), in which high level symbolic representations of formal reasoning are passed down to an AI system. However, this method is not considered a combination of the *symbolic* and *connectionist* approaches, since a symbolic representation of an assumption is not forcibly imposed on a solution. Instead, it is merely used to introduce a bias in a learning system that can still explore the hypothesis space for the best possible solution, as shown in Figure 3.2.

3.2 Basic concepts of vision transformers

Before introducing the assumptions translated into a regularization term in this proposed method, we introduce two fundamental concepts of vision transformers that are at the core of that assumption: image patches and attention maps.

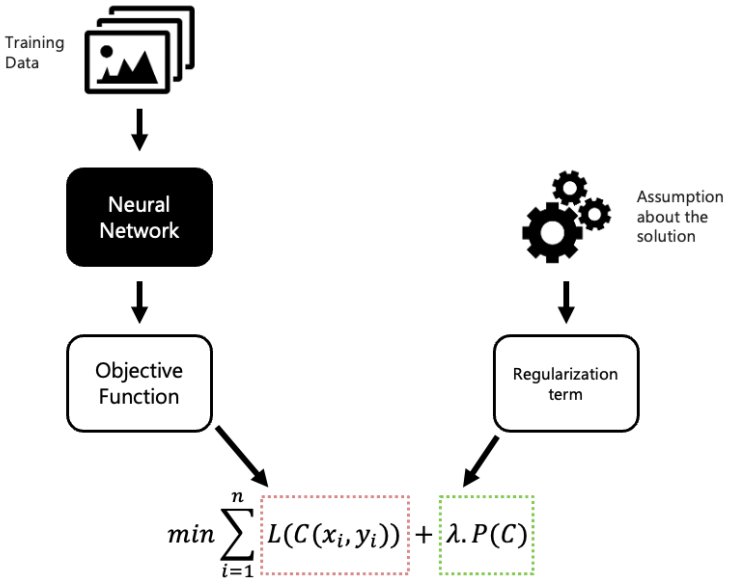


Figure 3.1: Illustration of the method's conception process. The left side shows a machine learning problem, where a neural network is trained set of training data to optimize a certain objective function (loss function). On the right side, an assumption over the problem's solution is made and formalized in the form of a regularization term to be added to the objective function. In this scheme, L is the objective function; C is the network; x_i and y_i are the i-th sample and its respective label; n is the size of the training set; P is the regularization term; and λ is the hyperparameter used to control de trade-off between loss function and penalty. Figure by the author.

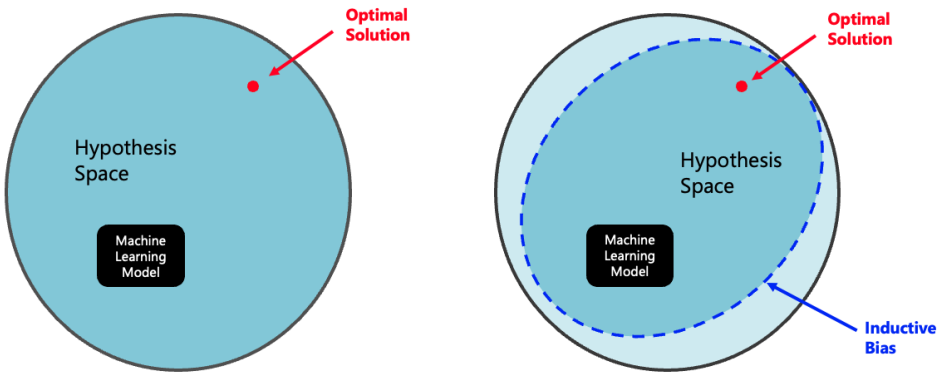


Figure 3.2: On the left side: illustration of the hypothesis space of a machine learning problem. On the right side: the hypothesis space limited by an inductive bias. The line representing the inductive bias is dashed to indicate that, although the inductive bias make certain solutions more preferable, it is still possible for a model to converge to a minimum outside that space. Ideally, the optimal solution would be inside the hypothesis space a model is biases towards, but that is not always the case.

3.2.1 Image Patches

Transformers are designed to receive a one-dimensional sequence of feature vectors as input. Nevertheless, images are typically represented in a two-dimensional arrange of pixels with three color channels. Though an intuitive approach to tackle this apparent "incompatibility" between

images and transformers would be to *flatten* an image's pixels into one dimension and to compute a linear embedding for each pixel, two limitations arise from this approach.

Firstly, similarly to CNNs, the computational cost of the transformer is also high, and the cost of self-attention, particularly, is quadratic to the number of elements in the input sequence [6]. Therefore, if each pixel were to be individually embedded, then every self-attention layer on the transformer would require each pixel to attend to every other pixel. Hence, even in a low resolution image, the computational cost of self-attention would be enormously high, impacting its usage on environments with limited resources. The second limitation is that, by flatting two-dimensions into a single one, the information of the 2D relation between each pixel is lost.

To deal with the first challenge, a common approach in vision transformers is to first, split an input image into same-size two-dimensional *patches*; then to compute a linear embedding for each patch; and finally to rearranged those patch embeddings into a one-dimensional input vector [6].

In this research work, we follow this approach and split an input image $X \in \mathbb{R}^{3\times H\times W}$ into N patches, where (H, W) are the height and width of a 3 channels image. A patch is denoted as $x_n \in \mathbb{R}^{3\times C\times C}$, and (C, C) is the resolution of each patch, as shown in Figure 3.3.

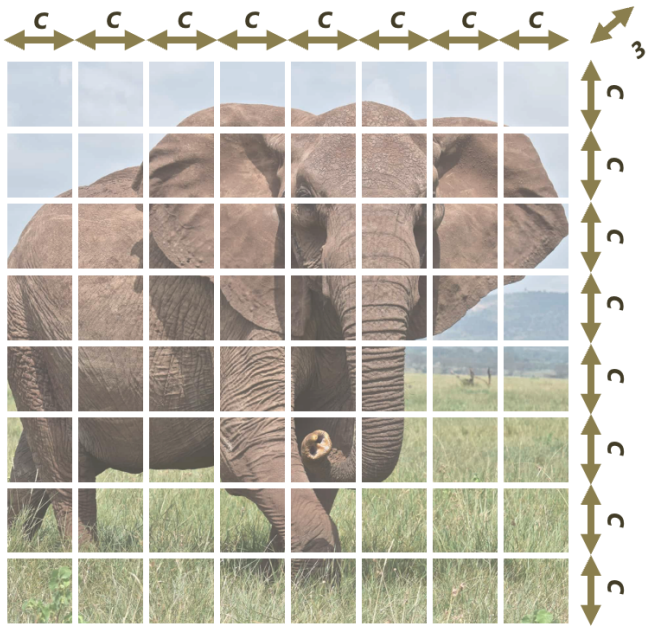


Figure 3.3: Illustration of an image with three color channels being divided into a set of patches with resolution (C,C). In this example, the total number of patches $N = 64$. Figure by the author.

Afterwards, a linear embedding is computed from each patch. Those linear embeddings are then arranged in a sequence compatible with the transformer. Furthermore, in order to deal with the loss of information on the pairwise 2D relation between image patches, we adopt a standard practice on vision transformers, which is to supplement each individual embedding with a positional

encoding [6]. [1]

3.2.2 Attention maps

At the core of the Transformer [7] is its *self-attention mechanism*, which is also called Scaled Dot-Product Attention. As describe in section 2.2.1.3, the multi-head self-attention (MHA) layer of an encoder is responsible for computing the pairwise relation between all elements of the transformer's input sequence. The input of vision transformers is typically a sequence of linear embeddings of images patches, as described in section 3.2.1. Therefore, the self-attention layer of the encoder layer computes the pairwise self-attention score between all image patches [8].

To perform self-attention, three parameter matrices are learned per attention-head. They are the denoted W^Q, W^K and W^V and each of them produces a different representation from the input sequence — the query matrix Q, key matrix K and the value matrix V, respectively [7]. Then, from these representations, self-attention is computed as described in Equation (2.4).

The attention map A is an intermediate output of the self-attention layer (Eq. 2.6), and in vision transformers, they express the pairwise self-attention scores between all image patches. Therefore, the attention map A is a $N \times N$ matrix-like representation containing the pairwise relation between all elements in a given input sequence.

As shown in Fig. 3.4, we refer to the attention map produced by an encoder layer as $A \in \mathbb{R}^{N \times N}$, where N is the number of patches into which an image is divided. A *row* $A_n \in \mathbb{R}^N$ contains the pairwise self-attention between patch x_n and all other patches [9].

3.3 Method's assumption

In this self-attention regularization method, an assumption about the learning problem is derived from a logical perspective and translated into a regularization term to be added to machine learning's objective function.

The assumption is that a pair image patches that, from a two-dimensional perspective, are close to each other should typically present a higher self-attention score between them than a pair patches that are spatially distant to each other.

Figure 3.5 exemplifies the simple idea behind this assumption. First, there is a small 2D distance between the pair of patches highlighted near the top left corner of the image. Therefore, the self-attention scores between them is expected to be high. In contrast, the 2D distance between the other highlighted pair of patches is significantly large. Hence, the self-attention scores between them is expected to be small.

[1] For more details on the linear embeddings and the positional encoding, please refer to Chapter 5.

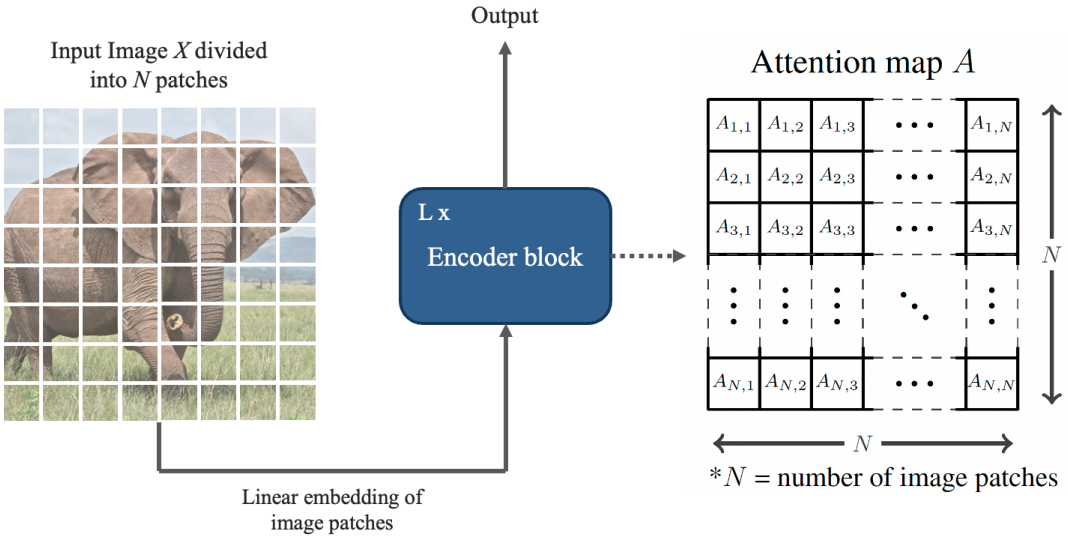


Figure 3.4: Overview of the attention map computation. First, an input image X is divided into N patches (in this example $N = 64$); Then, a linear embedding is computed for each patch; Then a sequence of embeddings is forwarded to the first of L encoder blocks (layer), with the subsequent encoder blocks always receiving the output of the previous one; at the MHA layer of each encoder block, an attention map A, with dimension $N \times N$, is computed. The attention map A expresses the pairwise self-attention between all patches in a given input image. A row in A, for example A_1, indicates the self-attention between patch x_1 and all other patches. Image by the author.

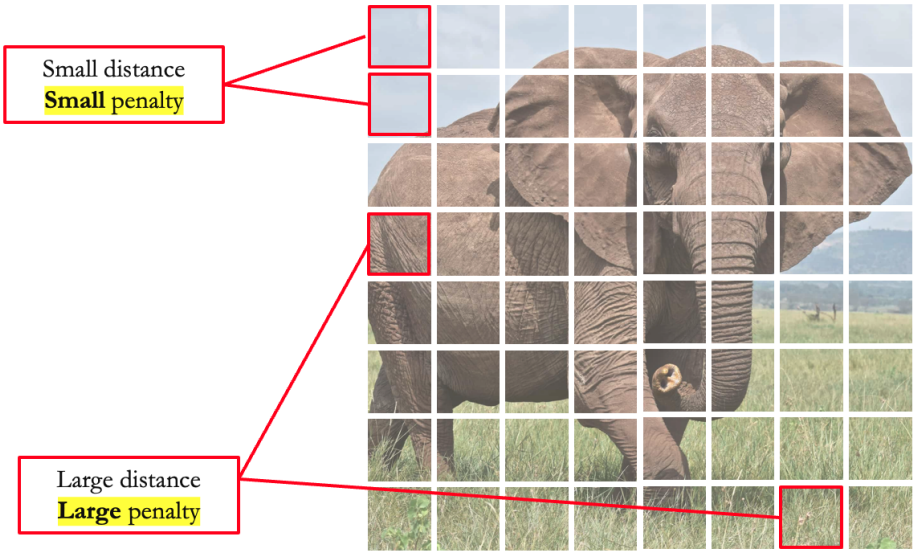


Figure 3.5: Example of the expected attention scores according to the assumption made by this regularization method. Image by the author.

In this method, a novel regularization term in the form of a loss function is devised, and it is denoted *distance loss*. The distance loss formalizes the logic of this assumption, and acts as a regularization term added to the vision transformer's loss function, where it penalizes solutions based on the pairwise two-dimensional distance between image patches and the respective self-attention scores between them. By penalizing a solution based on the attention-maps, the learning

of weights in the W^Q, W^K and W^V parameter matrices of a regularized layer is expected to be biased towards solutions that reduce the penalty imposed by the distance loss.

3.4 Method

We describe our self-attention regularization method based on the 2D spatial distance between image patches, and the regularization term denoted *Distance Loss*, designed for the vision transformer encoder. The *Distance Loss* penalizes the self-attention computation between image patches based on the *Manhattan distance* between them. The intuition behind it is that the more spatially distant two image patches are, the greater the penalty attributed over self-attention computation. In this section, we introduce the method and its specific elements.

3.4.1 Penalty Matrix

Firstly, we introduce the first two elements of this method: the *distance matrix* D, containing the pairwise Manhattan distance between all image patches[2]; and the *penalty matrix* P, obtained directly from D and whose role is to impose a penalty on self-attention computation between each pair of patches based on their respective Manhattan distances.

The distance matrix D is computed as follows. First, the pairwise Manhattan distance between all patches is computed and arranged on a distance matrix D. A row $D_n \in D$ indicates the pairwise Manhattan distance between a patch x_n and all other patches. Therefore, D has dimensions $N \times N$, and it is a hollow and symmetric matrix, with all of its off-diagonal entries being positive. That is, $\forall n \in N$: $D_{n,n} = 0$, $\forall n, m \in N : D_{n,m} = D_{m,n}$, and $\forall n, m \in N$: $D_{n,m} > 0$ if $n \neq m$. An overview of the distance matrix computation is shown in Fig. 3.6, with the example of an input image divided into 9 patches.

After computing the distance matrix D, the penalty matrix P can be obtained directly from it. More specifically, each element $P_{n,m} \in P$ is obtained through the following equation:

$$P_{n,m} = \frac{D_{n,m} - \frac{\max D_n}{\alpha}}{\sqrt{(D_{n,m} - \frac{\max D_n}{\alpha})^2 + \beta}}, \tag{3.2}$$

where $D_{n,m} \in D$ is the element in n^{th} row and the m^{th} column of the distance matrix and $\max D_n$ is the maximum value in the row D_n. Moreover, Equation (3.2) is a function with two horizontal asymptotes controllable through hyperparameters $\alpha \geq 1$ and $\beta > 0$.

An example of the computation of a single element $P_{1,2} \in P$ through Equation (3.2) is shown in Figure 3.7. In it, in prder to compute $P_{1,2} \in P$, equation (3.2) takes the element in D at

[2]The Manhattan distance is the distance between two points measured along axes at right angles.

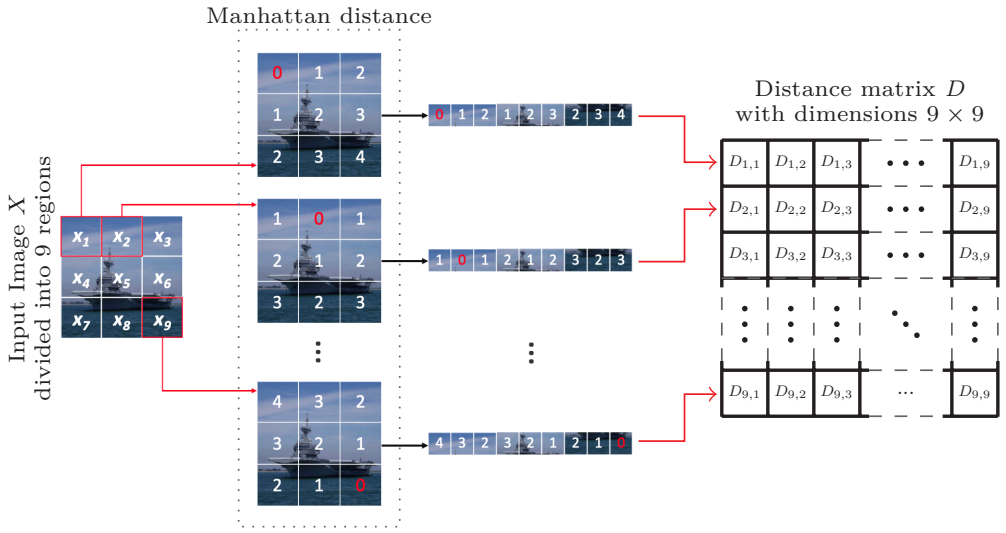


Figure 3.6: Overview of the computation of the distance matrix through the example of an image divided into 9 patches. The pairwise Manhattan distance is computed between all patches then arranged in a distance matrix D. A row in D, for example D_1, indicates the Manhattan distance between patch x_1 and all other patches.

the corresponding position, which in this case in $D_{1,2}$, as well as the maximum value at the corresponding row in D, which in this case is D_1.

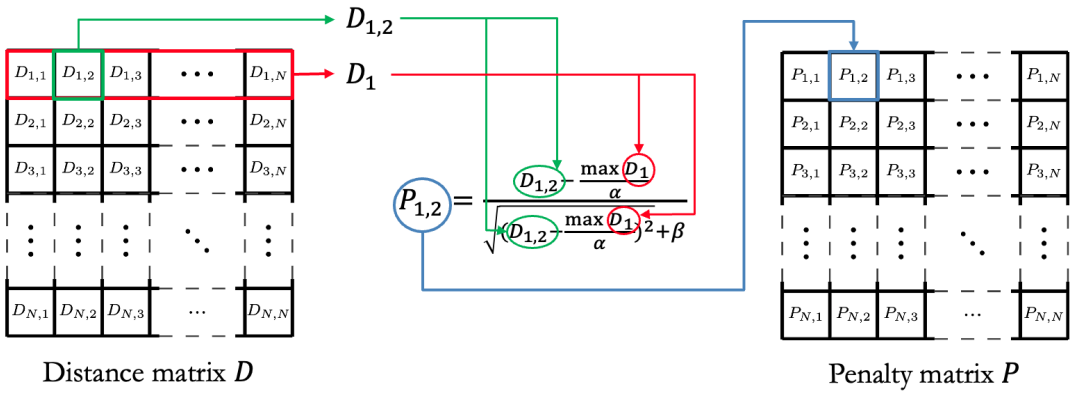


Figure 3.7: Example of computation of a single element $P_{1,2} \in P$ through Equation (3.2). Image by the author.

The penalty matrix P maintains the same dimensions as D, which is $N \times N$. Furthermore, P is still symmetric, since $\forall n, m \in N : P_{n,m} = P_{m,n}$. However, it is neither hollow and it migh present negative values. More specifically, when $D_{n,m}$, which is the manhattan distance between patches x_n and x_m, is smaller than $\frac{\max D_n}{\alpha}$, the computed value for $P_{n,m}$ is negative. Still, the fact that Equation (3.2) is a function with two horizontal asymptotes defined through hyperparameters allows for the control of the values contained in P.

When introduced to the distance loss, the penalty matrix P can be interpreted as a matrix containing the pairwise penalty factor between all image patches, which are used to impose a cost on self-attention computation. For instance, $P_{i,j} \in P$ expresses the factor used to penalize the

self-attention score between image patches x_i and x_j.

3.4.2 Loss Computation

The *distance loss*, that acts as a regularization term when added to the network objective function, is computed over 2 elements: the attention map A, and the penalty matrix P.

For N image patches into which an input image X is divided, at the MHA layer of the encoder layer, the attention map $A \in \mathbb{R}^{N \times N}$, containing the pairwise self-attention between all patches is computed, as described in section 3.4.2. Furthermore, as described in section 3.4.1, the penalty matrix $P \in \mathbb{R}^{N \times N}$ is obtained.

Then, the distance penalty l_X for a single image X can be computed through the following total sum of the pointwise product of the attention map A and the penalty matrix P:

$$l_X = \sum_{n=1}^{N} \sum_{m=1}^{N} A_{n,m} \times P_{n,m}, \tag{3.3}$$

where N is the number of patches, A is the attention map and P is the penalty matrix. A visual representation of this process is also shown in Figure 3.8. It is possible to notice that for any pair $n, m \in N$, a position $A_{n,m} \in A$ represents the self-attention score between patches x_n and x_m. Similarly, a position $P_{n,m} \in P$ represents the 2D-distance based penalty score between patches x_n and x_m. Hence the pointwise product between A and P.

However, because there is the possibility of the penalty matrix P (computed in Equation (3.2)), presenting negative values, when obtaining l_X through the summatory of the pointwise product between P and A described in Equation (3.3), there is also the possibility that l_X can eventually result in a negative value. Thus, to avoid the scenario where a negative penalty is obtained (which would constitute a reward), l_X undergoes the following conditional:

$$l_X^* = \begin{cases} l_X, & \text{if } l_X \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Then, the distance loss — which is the regularization term — over a set of I training examples can be computed as follows:

$$\mathcal{L}_D = \sum_{i=1}^{I} \log\left(l_{X_i}^* + 1\right). \tag{3.5}$$

Finally, with the regularization term, the last step of the process is to add it to the vision transformer's objective function. Typically, a network like the vision transformer is trained to address a given task by minimizing a loss function $\mathcal{L}_T$ described as:

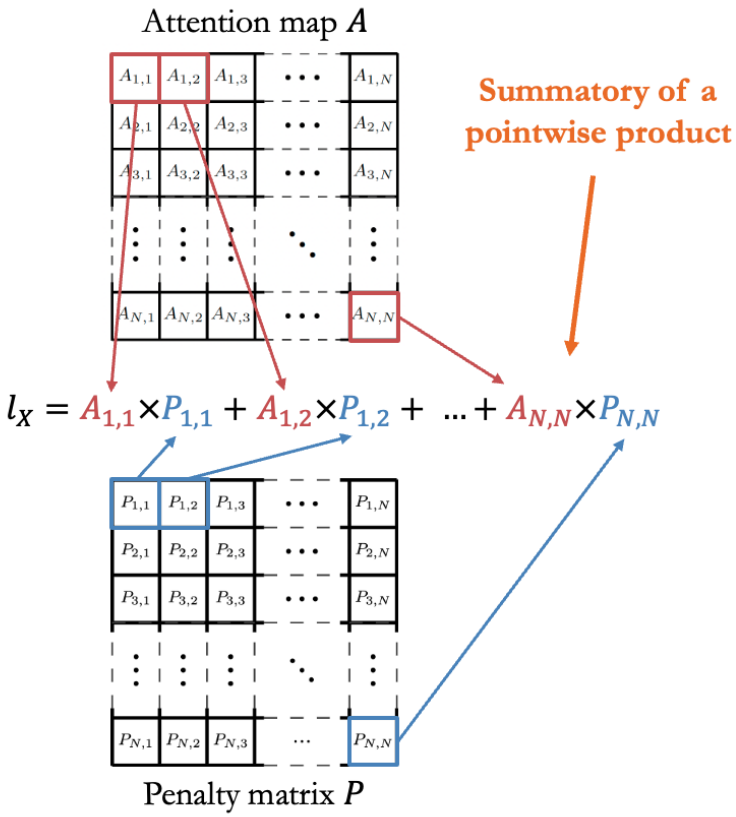


Figure 3.8: Visual representation of the computation the distance penalty l_X for a single input image X through Equation (3.3). Image by the author.

$$\mathcal{L}_T = \sum_{i=1}^{I} L_t(E(X_i), y_i), \tag{3.6}$$

where L_t is any loss function suitable to the task, E is the network, I is the number of training examples, and X_i and y_i are the input and the label, respectively.

The distance loss $\mathcal{L}_D$ defined in Equation (3.5) is added to $\mathcal{L}_T$ where it acts as a self-attention regularizer, and the total regularized loss is obtained from $\mathcal{L}_T$ and $\mathcal{L}_D$ as follows:

$$\mathcal{L}_R = \mathcal{L}_T + \lambda \mathcal{L}_D, \tag{3.7}$$

where $\lambda > 0$ is a hyperparameter assigned to control the balance between $\mathcal{L}_T$ and $\mathcal{L}_D$. An overview of the loss computation is shown in Figure 3.9. The penalty matrix P is pre-computed once before training and can be used for all training examples, and the Attention map A is extracted from the encoder layer being regularized.

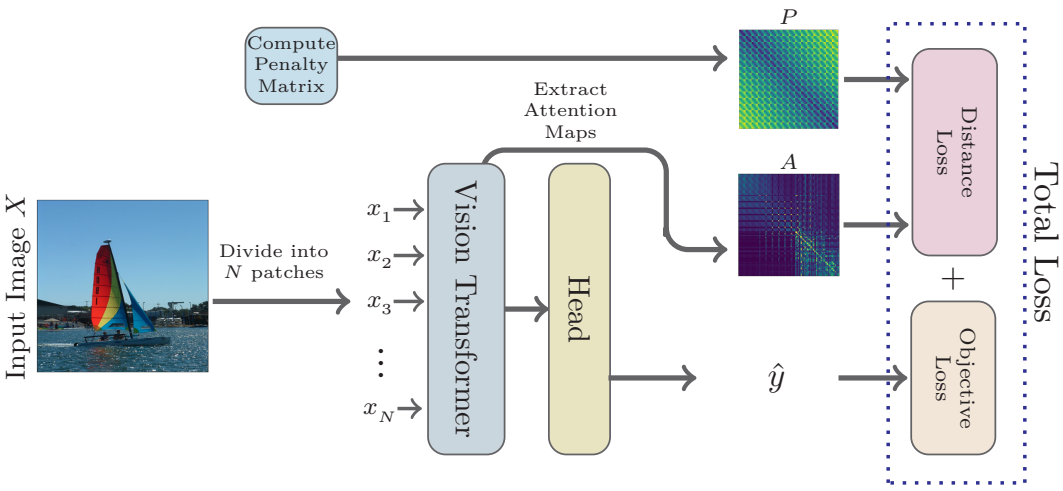


Figure 3.9: Overview of the regularization method on a vision transformer. The penalty matrix P and the attention map A are used to computed the distance loss, which acts as a regularization term when added to the vision transformer's objective loss. The penalty matrix P is pre-computed prior to training the model. The attention map A is extracted from an encoder layer.

3.5 Summary

Without modifying the global self-attention computation, the proposed distance loss encompasses the assumption made by this method, and acts as a self-attention regularizer when minimized alongside the vision transformer's objective function. The main assumption of the this method is that, the larger the distance between two patches $x_n \in X$ and $x_m \in X$, the smaller should be their self-attention scores, and therefore, the greater the penalty attributed over self-attention computation. In other words, minimizing the distance loss induces an attention map A to present low values in positions where the penalty matrix P presents high values. Hence, inductive bias is induced on self-attention maps by minimizing the distance loss.

3.6 References

[1] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[2] K. Weinberger, "Linear Regression," Date Accessed: 2022-07-15. [Online]. Available: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote08.html#map-estimate

[3] P. Bühlmann and S. Van De Geer, *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[4] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16

Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

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

[8] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, "Visformer: The vision-friendly transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 589–598.

[9] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2286–2296.

Chapter 4

Style similarity based Self-attention regularization

You don't have to be a mathematician to have a feel for numbers.

—**John Nash**

(Mathematician)

Science is not only compatible with spirituality; it is a profound source of spirituality.

—**Carl Sagan**

(Astronomer, astrophysicist, astrobiologist, author, and science communicator)

This chapter, similarly to chapter 3, presents a novel *self-attention regularization* method with the objective of introducing a new inductive bias on vision transformers. In this method, the regularization is carried out based on the pairwise style similarity between image regions, which are measured using the *mean square error* (MSE) between regions gram matrices. Section 4.1 briefly introduces the background of the method. Two newly introduced concept of *image region* is described in section 4.2. Section 4.3 introduces the assumption made by this method. Section 4.4 formally describes the style similarity based self-attention regularization method. And finally, section 4.5 presents a summary of this method.

4.1 Background

This self-attention regularization method shares the same conception as the method presented in section 3.1. In this method as well, the goal of devising a regularization term is to, based on a assumption, impose a penalty on the objective function being optimized. Then, by controlling

Parts of the content of this chapter were previously published in Mormille *et al.*, 2023 (see Appendix D)

the balance between the objective function and the regularizer, solutions whose combinations of parameters result in a high penalties are avoided, thus creating a inductive bias towards solutions that result in smaller penalties. In other words, in this method as well, a assumption is translated into a computer program that acts as a regularization term to be added to a machine learning's objective function.

Furthermore, two basic concepts of vision transformers — *image patches* and *attention maps* — are a central part of this method as well. A formal definition of image patches is presented in section 3.2.1, and a description of attention maps and how they are obtained in presented in section 3.4.2.

4.2 Image regions

In this method, we introduce a the concept of *image regions*. In concept, image regions are akin to image patches, and they can be defined as a two-dimensional slice of an image with three color channels. Moreover, all image regions are set to have the same dimensions.

Regarding **image patches**, this method adopts the same definition presented in section 3.2.1, where an image $X \in \mathbb{R}^{3\times H\times W}$ is divided into Npatches, and (H, W) are the height and width of a 3 channels image. Furthermore, an individual patch is denoted as $x_n \in \mathbb{R}^{3\times C\times C}$, and (C, C) is the resolution of each patch [1].

The definition of **image regions** is very similar, with an input image $X \in \mathbb{R}^{3\times H\times W}$ being divided into R *regions*, where an individual *region* $x^r \in \mathbb{R}^{3\times G\times G}$ is a 2D slice of the input image with 3 channels and resolution (G, G).

A very important aspect of image regions, is that their resolution G has to be an integer multiple of C (where C is the resolution of the image patches). An illustration of both image regions and patches is shown on Figure 4.1.

4.3 Method's assumption

In this region similarity based self-attention regularization method, an assumption about the learning problem is made and translated into a regularization term to be added to vision transformer's objective function.

The assumption is that a pair image patches that have a similar *style* should typically present a higher self-attention score between them than a pair patches that have seemingly different styles. Therefore, the intuition behind this self-attention regularization method is that high self-attention values between two image patches with a similar style representation should result in a small loss;

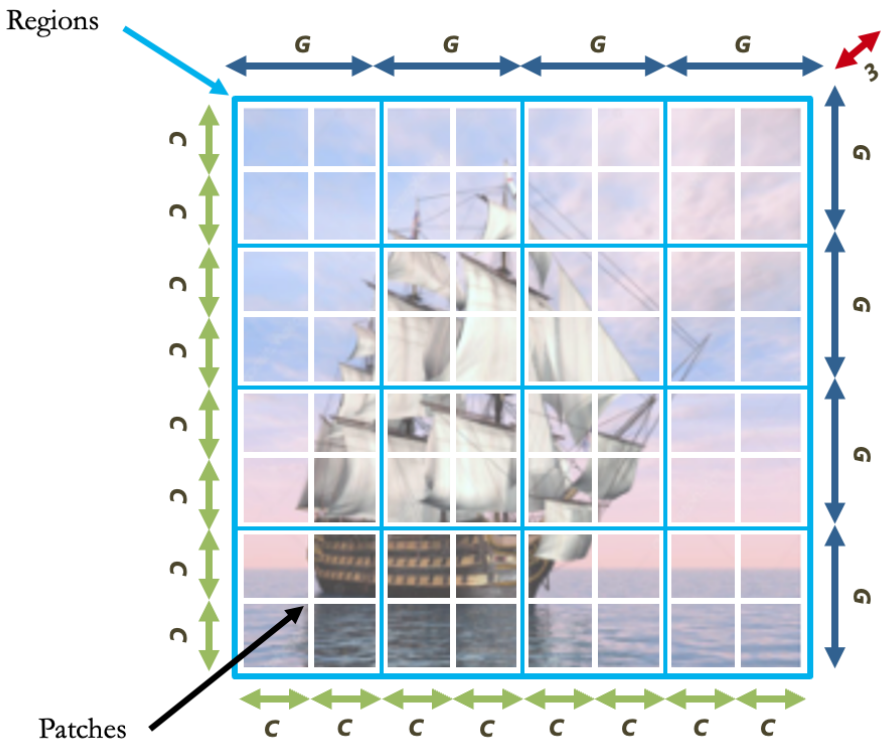


Figure 4.1: Illustration of image regions and image patches on the same three color channels image. Image regions are represented in blue color, and have resolution (G,G). Image patches are represented in white, with resolution (C,C). In this example, $G = 2 \times C$, and the total number of regions is $R = 16$, and the total number of patches is $N = 64$. Figure by the author.

while high self-attention values between two patches whose style representation are very distinct should result in a high loss.

There are two key aspects to this assumption. The first is about how to represent style on an image. To address that, we take inspiration on Gatyl *et al.* [2] style representation of an image. In this work, the *gram matrix* of an image's representation is used to capture its style and subsequently recreate it on a different image. Therefore, to represent the style of an image with three color channels, this method also adopts the use of gram matrices.

The second key element of this method is the granularity in which the style representation is obtained. A straightforward approach would be to compute the style of every image patch $x_n \in X$ and then to compute the pairwise similarity between the styles of each patch. Nevertheless, in order to experiment with different levels of granularity without having to alter the resolution of the image patches, the concept of image region described in section 4.2 was introduced and henceforth, a style representation is computed for each individual image region $x^r \in X$.

The resolution of an image region G has to be equal to a multiple of the patches resolution C. So, if for example, the patch resolution is $C = 16$, then the region resolution has to be $G = 16, 32, 48...$ We define this relation as $G = \omega \times C$, where $\omega \in \mathbb{Z}$ and $\omega \geq 1$. Therefore, because of the condition,

it is guaranteed that each region will contain at least on image patch within itself and a patch will always be part of one and only one region.

The assumption of this method is that, image patches contained in regions with similar style representations are imposed a small penalty on their self-attention scores. Whereas image patches contained in regions with very distinct style representations are imposed a high penalty on their self-attention scores. Furthermore, patches contained within the same image region are not imposed any penalty on their self-attention scores whatsoever. In other words, the assumption is that patches contained in regions with similar style are expected to have high self-attention scores between them, while patches contained in regions with seemingly different styles are expected to present low self-attention scores between them. An illustration of this assumption through three pair of image patches is shown in Figure 4.2.

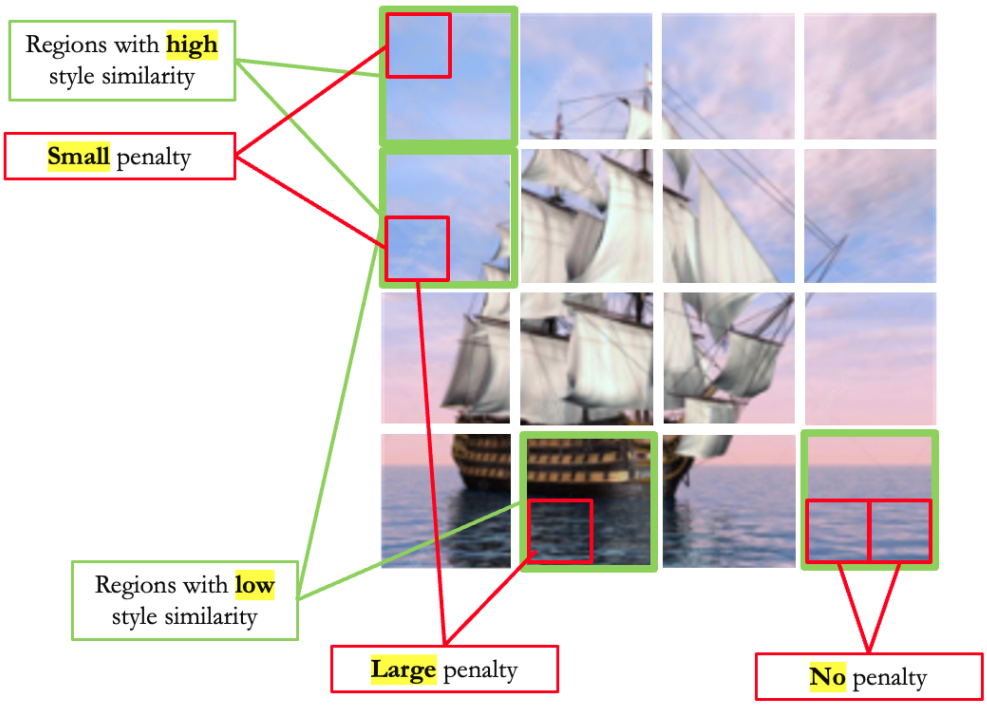


Figure 4.2: Example of the expected penalties on self-attention scores through a regularization term deploying the assumption of this method. In this example, three pair of patches are compared based on the style of the image regions containing them. Image by the author.

In this method, a novel regularization term in the form of a loss function is devised, and it is denoted *similarity loss*. The similarity loss formalizes the logic of this assumption, and acts as a regularization term added to the vision transformer's loss function, where it penalizes solutions based on the similarity of the style representation between image patches and the respective self-attention scores between them. By penalizing a solution based on the self-attention scores produced by the MHA layer, the learning of weights in the W^Q, W^K and W^V parameter matrices of a regularized layer is expected to be biased towards solutions that reduce the penalty imposed

by the distance loss.

4.4 Method

We describe our self-attention regularization method based on style similarity, and the regularization term denoted *Similarity Loss*, designed for the vision transformer encoder. The *Similarity Loss* penalizes the self-attention score between image patches based on the *Mean Square Error* (MSE) between the style representations of the image regions that contain them. The intuition behind this method is that, the more similar the style of two regions are, the higher the self-attention scores between patches contained in them should be. Contrarily, the least similar the style of two regions, the smaller the self-attention scores between patches contained in them should be. In this section, we introduce the method and its specific elements.

4.4.1 Similarity matrix

Firstly, we introduce the first element of this method: the *similarity matrix* S, containing the pairwise similarity between regions in an input image. In order to compute the similarity between two regions $x^r \in X$ and $x^s \in X$, we propose a distance function based on the Gram matrix which was introduced as an image style representation in Gatys *et al.* [2].

The correlation between the three color channels of each individual region is obtained by computing its gram matrix. The gram matrix of an individual patch $x^r \in \mathbb{R}^{3\times G\times G}$ is denoted $\mapsto g_r \in \mathbb{R}^{3\times 3}$ and can be obtained through the following matrix multiplication:

$$g_r = x^{r*\top} x^{r*}, \tag{4.1}$$

where $x^{r*} \in \mathbb{R}^{3\times G^2}$ is obtained by flattening each color channel in a region x^r. The gram matrix g_r is a style representation of the image region x^r.

After the style representation (gram matrix) for all individual image regions is obtained, in order to assemble the similarity matrix S, the pairwise mean square errors between all gram matrices are taken, its values are then normalized and then arranged to create the similarity matrix $S \in \mathbb{R}^{R\times R}$.

In short, S contains the pairwise style similarity between all image regions. An overview of the computation of the similarity matrix is shown in Figure 4.3. It shows that, first, each individual image region $x^r \in X$ is mapped to a style representation g_r through the computation of their gram matrix. Then, the MSE score between each pair of image regions is computed and arranged in the similarity matrix S. The similarity matrix S has resolution (R,R), where R is the number of image regions. Furthermore, $\forall r, s \in R$: $leqS_{r,s} = MSE(g_r, g_s)$. Also, since the MSE scores were normalized, $\forall r, s \in R$: $0 \leq S_{r,s} \leq 1$.

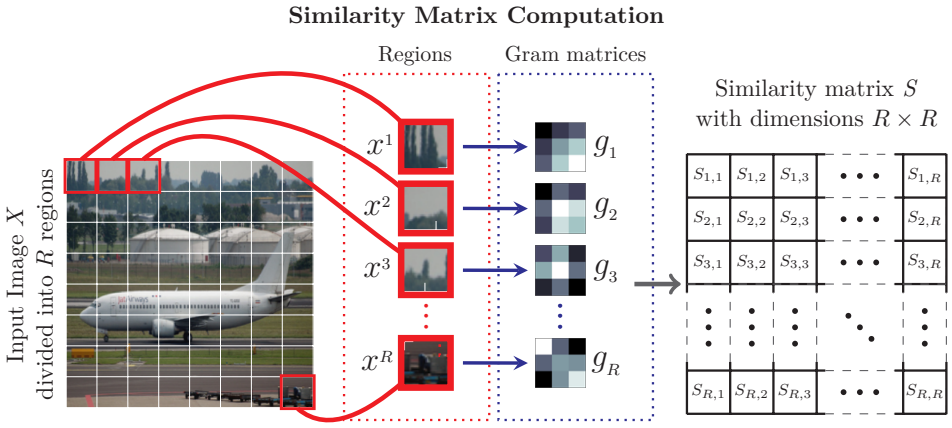


Figure 4.3: Overview of the computation of the similarity matrix S. An image is divided into fixed-size regions; the gram matrix of each individual region is computed; then a similarity matrix S containing the pairwise mean square errors between all gram matrices is built. S is a symmetric non-negative hollow matrix.

Since the similarity between the styles of any two regions x^r and x^s is taken using the mean square error, it means that, the higher the MSE score between them, the less similar they are. Furthermore, S is a hollow matrix and all the off-diagonal entries are positive, that is, $\forall r \in R$: $MSE(g_r, g_r) = 0$, and $\forall r, s \in R$: $(S_{r,s} > 0$ if $r \neq s)$. Moreover, S is also symmetric, that is, $\forall r, s \in R : S_{r,s} = DSs, r$.

4.4.2 Self-attention matrix

The second element introduced in this method is the *self-attention matrix* M_{sa}. The self-attention matrix is obtained directly from the attention map A, which is accessed at the multi-had attention (MHA) layer of each encoder layer. The process of obtaining the self-attention matrix M_{sa} can be interpreted as scaling the attention map A to match the dimensions of the similarity matrix S.

An attention map produced by the MHA layer of an encoder layer is denoted as $A \in \mathbb{R}^{N \times N}$, where N is the number of patches into which an input image X is divided. The attention map is a 2-dimensional matrix-like representation of the self-attention scores between all possible pairs of patches in the input sequence. A is non-negative, but unlike the similarity matrix S, it is neither hollow nor symmetric.

An overview of the encoder layer and the attention map is shown in Figure 4.4. A *row* $A_n \in \mathbb{R}^N$ indicates the pairwise self-attention scores between patch x_n and all other patches. For a better visualization of the attention map, each row in A can be reshaped into 2 dimensions so that a set of N two-dimensional attention maps $\{A_i : i = 1...N\}$ is obtained, as further depicted in Figure 4.4.

The self-attention matrix $M_{sa} \in \mathbb{R}^{G \times G}$ is computed by scaling-down the dimensions of an attention map $A \in \mathbb{R}^{N \times N}$ to match the dimensions of the similarity matrix $S \in \mathbb{R}^{G \times G}$.

The intuition behind this process is to aggregate the self-attention scores between pairs of

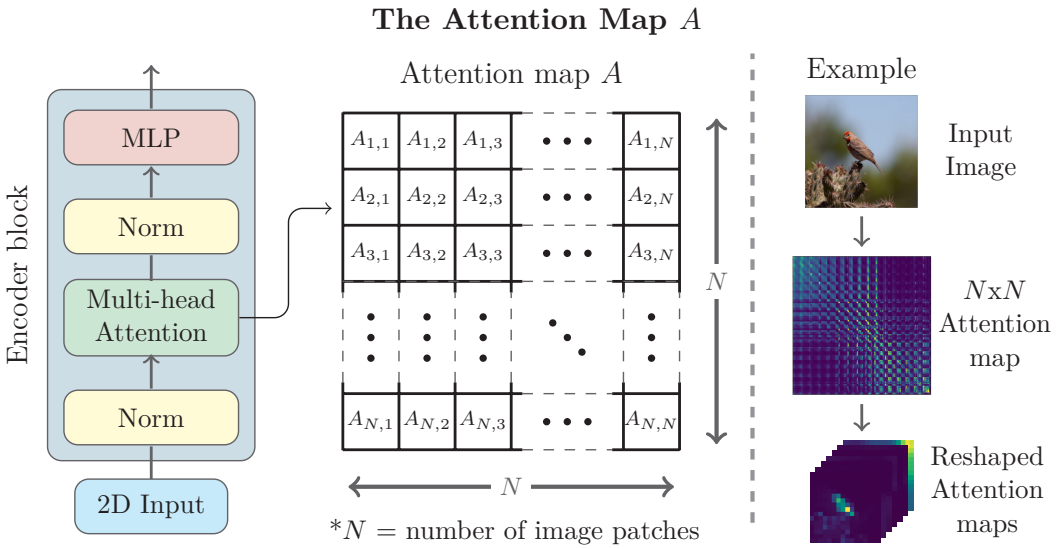


Figure 4.4: In an encoder block (also referred to as encoder layer), the attention map A for a given input image is generated by the multi-head attention (MHA) layer. It can be interpreted as a matrix containing the pairwise self-attention value between all patches from the input image. Every row in A can be further reshaped and visualized as an individual 2D attention map.

patches contained in the same region through a average pooling operation. However, in order to perform this operation, simply applying a single average pooling operation of kernel and stride equal to $\psi = \lfloor G/P \rfloor$ would result in aggregations between pair of patches contained in different regions.

Therefore, in order to obtain the self-attention matrix M_{sa}, a 10-step process is required, involving operations such as *reshaping*, *permutation*, *normalization* and two *average pooling* operations with a kernel and stride size $\psi = \lfloor G/P \rfloor$.

The 10 steps to encode the self-attention matrix M_{sa} for an input image $X \in \mathbb{R}^{3\times H\times H}$ divided in to N patches with resolution (C,C), and R regions of resolution (G,G), are as follows:

1. **1st step -** to access the attention map $A \in \mathbb{R}^{N\times N}$ at the MHA layer of the encoder layer;
2. **2nd step -** to *reshape* the attention map A from 2 dimensions (N,N) to 3 dimensions, obtaining $A_2 \in \mathbb{R}^{N\times\sqrt{N}\times\sqrt{N}}$.
3. **3rd step -** to perform an *average pooling* on A_2 with kernel and stride of $\psi = \lfloor G/P \rfloor$, resulting in $A_3 \in \mathbb{R}^{N\times\frac{\sqrt{N}}{\psi}\times\frac{\sqrt{N}}{\psi}}$.
4. **4th step -** to *reshape* A_3 from 3 dimensions (N, $\frac{\sqrt{N}}{\psi}$, $\frac{\sqrt{N}}{\psi}$) back to two dimensions, obtaining $A_4 \in \mathbb{R}^{N\times\frac{N}{\psi^2}}$.
5. **5th step -** to *permute* the two dimensions of A_4, obtaining $A_5 \in \mathbb{R}^{\frac{N}{\psi^2}\times N}$.
6. **6th step -** to *reshape* A_5 from 2 dimensions ($\frac{N}{\psi^2}, N$) to 3 dimensions, obtaining $A_6 \in \mathbb{R}^{\frac{N}{\psi^2}\times\sqrt{N}\times\sqrt{N}}$.

7. **7th step -** to perform an *average pooling* on A_6 with kernel and stride of $\psi = \lfloor G/P \rfloor$, resulting in $A_7 \in \mathbb{R}^{\frac{N}{\psi^2} \times \frac{\sqrt{N}}{\psi} \times \frac{\sqrt{N}}{\psi}}$.

8. **8th step -** to *reshape* A_7 from 3 dimensions $(\frac{N}{\psi^2}, \frac{\sqrt{N}}{\psi}, \frac{\sqrt{N}}{\psi})$ back to two dimensions, obtaining $A_8 \in \mathbb{R}^{\frac{N}{\psi^2} \times \frac{N}{\psi^2}}$.

9. **9th step -** to *permute* the two dimensions of A_8, obtaining $A_9 \in \mathbb{R}^{\frac{N}{\psi^2} \times \frac{N}{\psi^2}}$.

10. **10th step -** to *normalize* all values in A_9 to be between 0 and 1, thus obtaining $M_{sa} \in \mathbb{R}^{\frac{N}{\psi^2} \times \frac{N}{\psi^2}}$, where $\frac{N}{\psi^2} = R$.

For a better comprehension on how to compute M_{sa} from A, the 10 steps are depicted through an example in Figure 4.5, with an input image with resolution (3x256x256), $P = 16$, $G = 32$, and $\psi = 32/16 = 2$. In this example, the input image is divided in $N = 256$ patches and $R = 64$ regions, resulting in $A \in \mathbb{R}^{256 \times 256}$ and $M_{sa} \in \mathbb{R}^{64 \times 64}$.

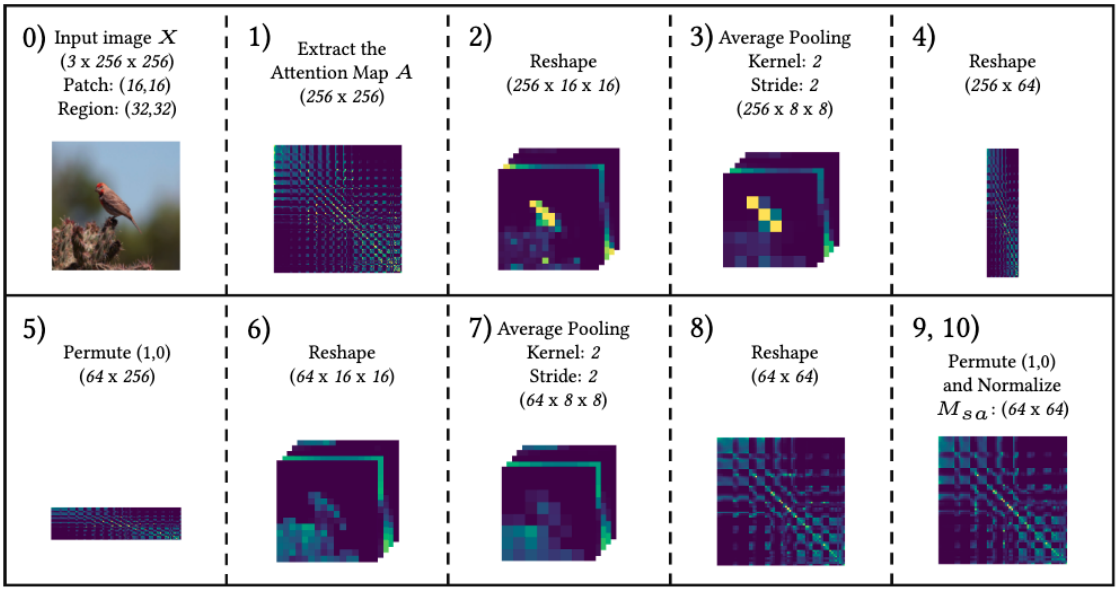


Figure 4.5: Overview of the 10-step computation of the self-attention matrix M_{sa} demonstrated through an example; (0) input image $X \in \mathbb{R}^{3 \times 256 \times 256}$ is divided into 256 patches with resolution (16,16) and 64 regions with resolution (32,32); (1) the first step is to compute the attention map $A \in \mathbb{R}^{256 \times 256}$; (2) A is reshaped to 3 dimensions of $256 \times 16 \times 16$; (3) an average pooling with kernel and stride $\psi = \lfloor 32/16 \rfloor = 2$ is performed, reducing the resolution of the representation on the latter two dimensions; (4) the representation is again reshaped to two dimensions of 256×64; (5) and its two dimensions are permuted; (6) again, the representations is reshape to three dimensions of $64 \times 16 \times 16$; (7) and an average pooling operation reduces the resolutions of the latter two dimensions; (8) the representation is again reshaped to two dimensions of 64×64; (9 and 10) and finally, $M_{sa} \in \mathbb{R}^{64 \times 64}$ is obtained by permuting back the two dimensions and normalizing the values of the representation. Image by the author.

In summary, the self-attention matrix M_{sa} is a scaled-down version of the attention map A to match the dimensions of the similarity matrix S. One important observation is that, in case $G = P$, it is not necessary to scale-down A through the aforementioned 10-step, and $M_{sa} = A$.

4.4.3 Loss Computation

The *similarity loss*, that acts as a regularization term when added to the network objective function, is computed over 2 elements. The self-attention matrix $M_{sa} \in \mathbb{R}^{R \times R}$, introduced in section 4.4.2, and obtained from the attention map $A \in \mathbb{R}^{N \times N}$. The second element and the similarity matrix $S \in \mathbb{R}^{R \times R}$, described in section 4.4.2, and containing the pairwise style similarity between image regions.

The similarity penalty for a single training example X is computed from the total sum of the pointwise product between the similarity matrix S and the self-attention matrix M_{sa}. It is denoted l_{S_X} and is expressed as follows:

$$l_{S_X} = \sum_{r=1}^{R} \sum_{s=1}^{R} M_{sa_{r,s}} \times (S_{r,s} + \rho), \tag{4.2}$$

where R is the number of regions an image is divided into, and $\rho \geq 0$ is a distance bias hyperparameter, whose purpose is to control the learning process. A visual representation of this process is also shown in Figure 4.6. It is possible to notice that for any pair $r, s \in R$, a position $M_{sa_{r,s}} \in M_{sa}$ represents an aggregation of self-attention scores. Similarly, a position $S_{r,s} \in S$ represents the style-similarity penalty weight between regions x^r and x^s. Hence the pointwise product between M_{sa} and S.

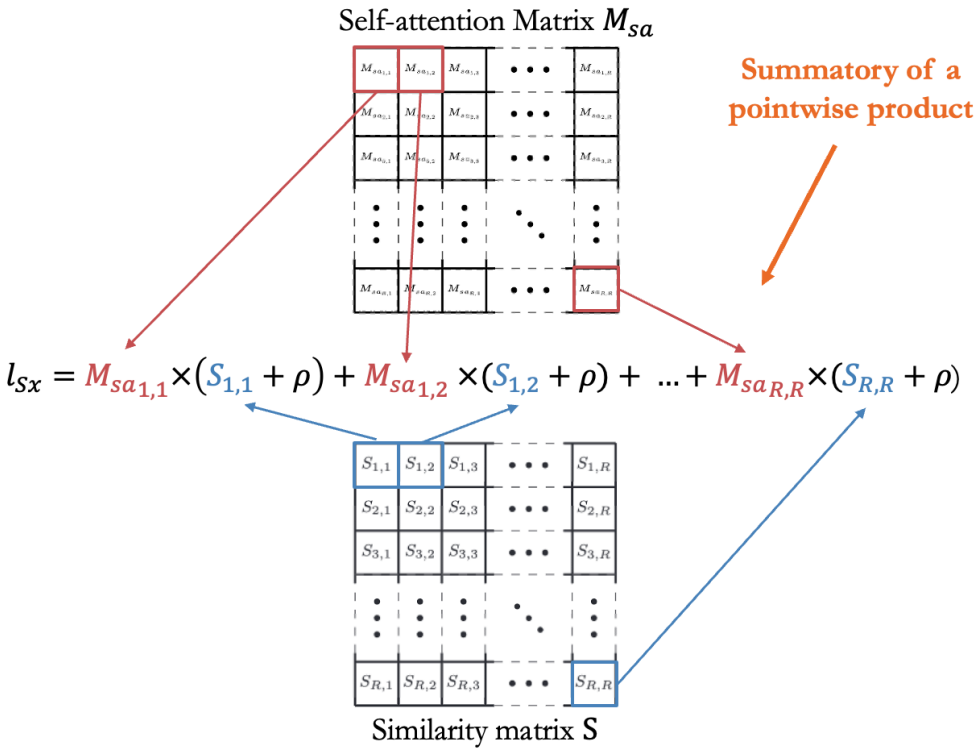


Figure 4.6: Visual representation of the computation the similarity penalty l_{S_X} for a single input image X through Equation (4.2). Image by the author.

Then, the *Similarity Loss* over a set of I training examples can be computed as follows:

$$\mathcal{L}_S = \sum_{i=1}^{I} \log\left(l_{S_{X_i}} + 1\right), \tag{4.3}$$

Finally, with the similarity loss $\mathcal{L}_S$ — which is the regularization term — defined, the last step of the process is to add it to the vision transformer's objective function. Commonly, a network such as the vision transformer is trained to address a task by minimizing a loss function, which we denote $\mathcal{L}_T$, and describe as:

$$\mathcal{L}_T = \sum_{i=1}^{I} L_t(E(X_i), y_i), \tag{4.4}$$

where L_t is any loss function suitable to the task, E is the network, I is the number of training examples, and X_i and y_i are the input and the label, respectively.

Finally, the similarity loss $\mathcal{L}_S$ (Eq. 4.3) is added to the task loss $\mathcal{L}_T$ (Eq. 4.4), acting as a self-attention regularizer restricting the hypothesis space by penalizing solutions that produce high self-attention values between patches within regions whose gram matrices' distance is large. Hence, the total loss:

$$\mathcal{L}_R = \mathcal{L}_T + \lambda\mathcal{L}_S, \tag{4.5}$$

where the hyperparameter $\lambda > 0$ is used to control the trade-off between the task loss $\mathcal{L}_T$ and the similarity loss $\mathcal{L}_S$.

An overview of method is displayed in Figure 4.7. Unlike the penalty matrix P used at the two-dimensional distanced based regularization, that can be pre-computed once before training and used for all training examples (Section 3.4), a similarity matrix S_i has to be computed for each individual training example x_i during the forward pass. Furthermore, the self-attention matrix M_{sa} is computed from the attention map A extracted from the encoder layer being regularized.

4.5 Summary

To summarize, without altering self-attention, which remains a global operation, the proposed similarity loss captures the assumption made in this method, acting as a self-attention regularizer when minimized together with the vision transformer's objective function. The logic behind this method is that, the more distant the style of two regions x^r and x^s), represented by their gram matrices g_r and g_s, the greater the penalty attributed on the similarity loss over the self-attention scores between two patches x_n and x_m, located in x^r and x^s respectively. Therefore, the similarity loss acts as a self-attention regularizer that induces the self-attention matrix M_{sa} to present low values in positions where S presents high values, and vice-versa. Hence, an inductive bias derived

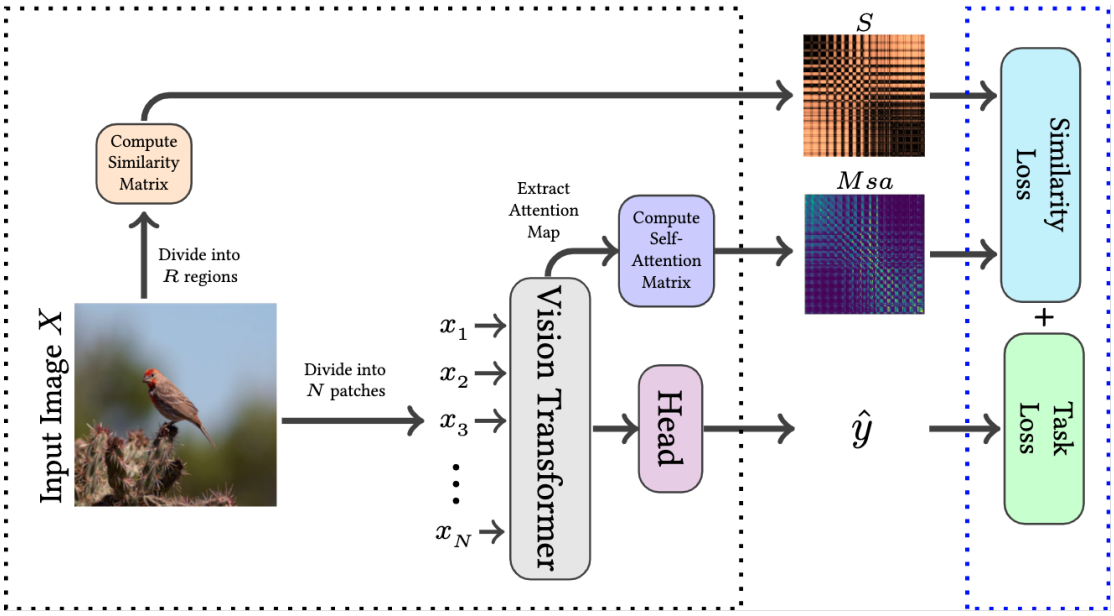


Figure 4.7: Overview of the style-similarity based self-attention regularization method with the Similarity Loss.

from an assumption is induced on the vision transformer through minimizing the similarity loss.

4.6 References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

[2] L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *arXiv:1508.06576 [cs, q-bio]*, Sep. 2015, arXiv: 1508.06576. [Online]. Available: http://arxiv.org/abs/1508.06576

Chapter 5

ARViT: Attention Regularized Vision Transformer

If you wish to make an apple pie from scratch, you must first invent the universe.

— **Carl Sagan**
(Astronomer, astrophysicist and science communicator)

The first principle is that you must not fool yourself and you are the easiest person to fool.

— **Richard Feynman**
(Theoretical Physicist)

In order to achieve state-of-the-art performance, vision transformers often tend to be composed by large scale architectures, with hundreds of millions of trainable parameters. Such characteristic derives mainly from their inherent lack of inductive biases. However, training such large models might hinder their use on environments with limited resources. Therefore, adapting the vision transformer to be able to learn good representations with a smaller capacity architecture might open new possibilities. In this chapter we introduce Attention Regulated Vision Transformer (ARViT), a proposed variant of the ViT [1], with significantly less trainable parameters and changes inspired by the work of Chen *et al.* [2]. In section 5.1 we introduce the ViT by Dosovitskiy *et al.* and in section 5.2 we describe ARViT's architecture.

5.1 The ViT Architecture

On recent years, the Transformer [3] achieved remarkable results on the domain of natural language processing and has also proven to be a reliable alternative to CNNs on computer vision related tasks [4, 5].

The original Vision Transformer (ViT) was introduced by Dosovitskiy *et al.* [1] in the groundbreaking work "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". As overview of the ViT architecture is shown in Figure 5.1.

The original Transformer [3] takes a one-dimensional sequence of token embeddings as input. To encode two-dimensional images, the ViT first split the images into a set of N patches of equal resolution (P,P). The number of patches N also indicates the length of the Transformer input

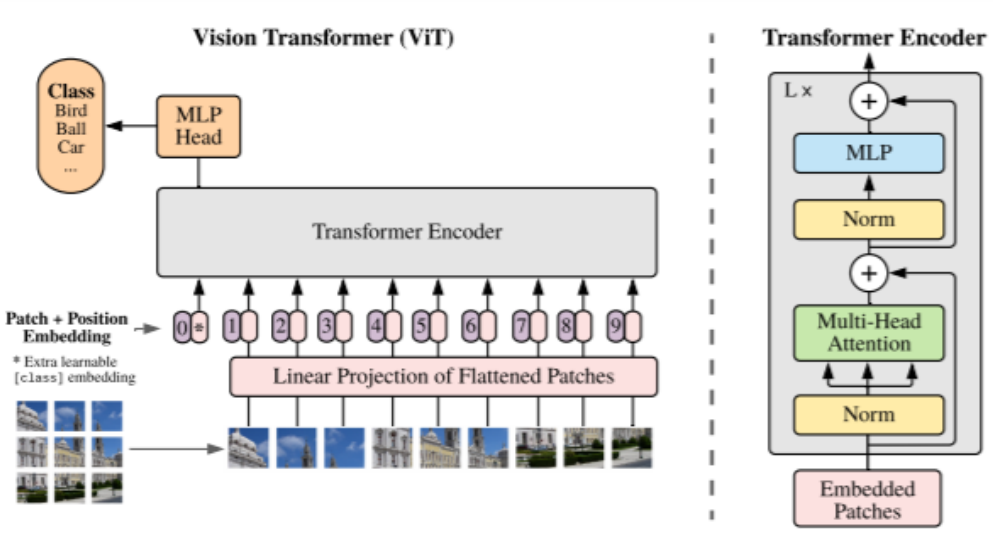


Figure 5.1: The Vision Transformer architecture. Figure from Dosovitskiy *et al.* [1]

sequence. The sequence of N patches is flattened and embedded to D dimensions through a linear projection. The output of the linear projection is denoted *patch embeddings* [1].

An additional learnable embedding is then attached to the patch embeddings. This extra embedding is denoted *class embedding* and its output state is solely forwarded to the model's head to perform classification both during pre-trainig and fine-tuning [1].

Furthermore, the patch embeddings are supplemented by *position embeddings* to preserve positional information. The ViT uses learnable position embeddings. Then, the sequence of embeddings is forwarded to the transformer encoder as input [1].

The "backbone" of the ViT architecture consists solely of a transformer encoder with L layers. Each encoder layer consists of: a Multi-Head Self Attention Layer (MHSA), that concatenates the multiple attention outputs; a Multi-Layer Perceptron (MLP) with two layers and Gaussian Error Linear Unit (GELU); two steps of Layer Norm, applied before the MHSA layer and the MLP, respectively; and residual connections subsequent to the MHSA layer and the MLP [1].

Finally, the classification head is a MLP with one hidden layer during pre-training, and a single linear layer during fine-tuning. The ViT head takes only the output state of the lass embedding to perform classification [1].

In their work, Dosovitskiy *et al.* [1] proposed three variants of the ViT. The first one is the ViT-Base, with 12 encoder layers, 12 attention heads per layer, hidden dimension $D = 768$ and 86 million trainable parameters. The second variant is denoted ViT-Large, with 24 encoder layers, 16 attention heads per layer, hidden dimension $D = 1024$ and 307 million trainable parameters. Finally, the third variant is denoted ViT-Huge, with 32 encoder layers, 16 attention heads per layer, hidden dimension $D = 1280$ and 632 million trainable parameters [1].

5.2 ARViT's architecture

5.2.1 Motivation

In spite of the remarkable success of the ViT, its architectures and variants have some limitations. Possibly the largest one regards the high computational cost associated with the large capacity of the networks [1]. Since the cost of self-attention is quadratic to the number of elements in the

input sequence, having a large number of MHA layers with, each with large number of attention heads, scales to hundreds of millions of trainable parameters.

Another limitation pointed at by the work of Dosovitskiy *et al.* [1] is the lack of inductive biases on the ViT. This allows the ViT to thrive once pre-trained with over tens of millions of images with datasets like theImageNet-21k, with 21k classes and 14M images [6], and the JFT-300 [7], with 18k classes and 303M images. However, due to the lack of inductive biases, the ViT struggles when pre-trained on mid-size datasetes, like the ImageNet dataset with approximately 1.3 million images [8].

Therefore, motivated by those two limitations of the ViT, this research work introduces ARViT — the Attention Regularized Vision Transformer. The main objectives of ARViT are to, first, provide a smaller capacity vision transformer which is still capable of performing well on classification downstream tasks. And secondly, ARViT is developed to deploy the self-attention regularization methods proposed in Chapter 3 and Chapter 4. Therefore, ARViT aims at both reducing the computational cost associated with vision transformers and to tackle their inherent lack of inductive biases.

5.2.2 Architecture

The backbone architecture of our Attention Regulated Vision Transformer (ARViT) can be interpreted as a reduced version of the ViT [1] with changes inspired by the work of Chen *et al.* [2]. An overview of ARViT's architecture is displayed in Figure 5.2.

5.2.2.1 Modification 1

ARViT follows the approach of Dosovitskiy *et al.* [1] of splitting an input image $X \in \mathbb{R}^{3\times H\times W}$ into N patches with resolution (C, C), where (H, W) are the height and width of a 3 channels image. However, the first modification of ARViT in relation to the ViT comes both on the method used split the image into patches, as well as the way to encode the *patch embeddings*.

Inspired by the work of Chen *et al.* [2], ARViT splits the input image into patches and encode the patch embeddings in a single operation — a convolution with kernel size $C \times C$, stride equals to C and output channels equal to D. With this set up, the area of of each patch is convolved to a spatial size of 1×1 with D dimensions. Then, the 3D output of the convolution operations is flattened into a sequence of N patch embeddings.

5.2.2.2 Modification 2

The second modification in comparison to the ViT comes in regard to the *position embeddings*. While the ViT uses learnable position emebddings [1], ARViT patch embeddings are supplemented with fixed 2D sinusoid position embeddings.

5.2.2.3 Modification 3

The third modification of ARViT in relation to the ViT was also inspired by Chen *et al.* [2]. In its architecture, the ViT attaches a class embedding to the input sequence, and the state of this class embedding at the output of the encoder serves as the only image representation used by the model head to perform classification. On ARViT, the class embedding is removed, and the model head utilizes the representations encoded for all embeddings in the input sequence.

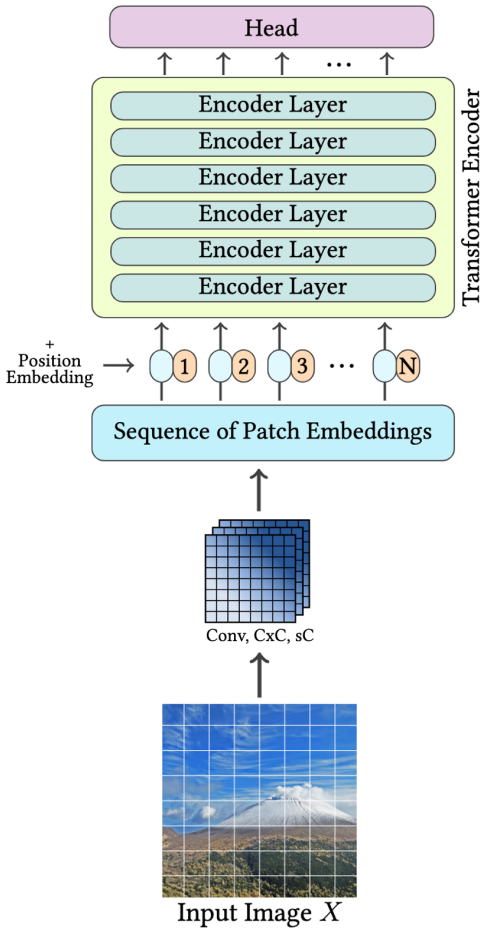


Figure 5.2: ARViT architecture overview. ARViT splits the input image into N patches and encode the *patch embeddings* with a convolution operation with kernel size $C \times C$ and stride size C. ARViT flattens the patch embeddings into a sequence supplemented with *position embeddings* before forwarding the input sequence to the Transformer encoder. The Transformer encoder is composed of six encoder layers, and the output of the last layer is fed to the model head. Image by the author.

5.2.2.4 Modification 4

The fourth modification is made inside the encoder layer. On the ViT, the encoder layer is formed by a MHS layer, a MLP with one hidden layer, two steps of Layer Norm, and residual connections subsequent to the MHSA layer and the MLP [1]. On ARViT, we replace the MLP with one hidden layer with a single fully connected layer.

5.2.2.5 Modification 5

The fifth and final modification is directly related with the model's capacity. ARViT scales down the number of encoder layers, attention heads and hidden dimensions. The ViT variants can have between: 12 and 32 encoder layers; 12 to 16 attention heads; and 768 to 1280 hidden dimensions. ARViT, however, reduces the number of encoder layers to 6, maintains the number of attention heads at the ViT lower limit of 12, and reduces the hidden dimensions to 516.

5.2.3 Comparison between ARViT and ViT variants

A comparison between the variants of the ViT and ARViT is shown in Table 5.1. With the changes proposed in section 5.2.2, ARViT has and architecture with 6 encoder blocks (layers), 12 attention heads, and 516 hidden dimensions. Other modifications also provided a further reduction in in the model capacity, and ARViT has a total of 10 million trainable parameters.

For effect of comparison, this work also introduces the ViT-Tiny, a ViT architecture also with 6 encoder layers, 12 attention heads and 516 hidden dimensions. The ViT-Tiny has 18 million trainable parameters — 8 million more than ARViT. This gap between ARViT and the ViT-Tiny can be explained by modifications 2 (section 5.2.2.2) and 4 (section 5.2.2.4) .

Table 5.1: Comparison between variants of the ViT and ARViT.

Model	Layers	Hidden-Dim	Heads	Params
ViT-B [1]	12	768	12	86M
ViT-L [1]	24	1024	16	304M
ViT-H [1]	32	1280	16	632M
ARViT	6	516	12	10M
ViT-Tiny	6	516	12	18M

5.3 References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

[2] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, "Visformer: The vision-friendly transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 589–598.

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

[4] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[5] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[6] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," *arXiv preprint arXiv:2104.10972*, 2021.

[7] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 843–852. [Online]. Available: http://ieeexplore.ieee.org/document/8237359/

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

Chapter 6

Experiments

If it disagrees with experiment, it's wrong. In that simple statement is the key to science. It doesn't make any difference how beautiful your guess is, it doesn't matter how smart you are who made the guess, or what his name is ... If it disagrees with experiment, it's wrong. That's all there is to it.

—**Richard Feynman**
(1918–1988)

This chapter describes the experiments and results with the two-dimensional distance based self-attention regularization, the style similarity based self-attention regularization, and with ARViT, the proposed vision transformer architecture. Section 6.1 describes the implementation details. Section 6.2 introduces the evaluation methods. Section 6.3 presents the results of the experiments. And finally, section 6.4 provides a discussion and analysis of the experiments and results.

6.1 Implementation details

In this section we introduce the implementation details of our experiments, all proposed ARViT variants, hyperparameter definitions, pre-training details and hardware specifications.

6.1.1 ARViT variants

To evaluate the performance of ARViT and the effects of the self-attention regularization methods on it, we pre-trained a total of 19 different variants. All variants, however, share the same input specifications, which are the resolution of the input images, which is defined as (256,256), and the *image patches* resolution, which is defined as $C = 16$.

The main distinction between each variant is which encoder layer had its self-attention regularized, and with which regularization method. Furthermore, suffixes are utilized to identify each ARViT variant.

The first variant introduced is *ARViT-Base*, which corresponds to the model trained without regularizing self-attention on any layer whatsoever. *ARViT-Base* can be seen as the baseline model used as a basis of comparison for all other models.

All other 18 variants trained with different permutations of layers regularized through the self-attention regularization methods proposed in Chapters 3 and 4.

6.1.1.1 ARViT variants regularized with the two-dimensional distance based method

Using the two-dimensional distance based self-attention regularization method, a total of 6 models were experimented with. They are: ARViT-L1, ARViT-L2, ARViT-L3, ARViT-L4, ARViT-L5 and ARViT-L6. The suffix of these models can be described as follows: firstly, a capital letter "L" is used to identify that the regularization method used was the two-dimensional distance based one; then, the integer ranging between 1 and 6 is used to identify which encoder layer is being regularized. Hence:

1. **ARViT-L1** is the variant of ARViT where the first encoder layer is regularized using the two-dimensional distance based method.
2. **ARViT-L2** is the variant where the second encoder layer is regularized using this method.
3. **ARViT-L3** is the variant where the third encoder layer is regularized using this method.
4. **ARViT-L4** is the ARViT variant where the fourth encoder layer is regularized using this method.
5. **ARViT-L5** is the variant in which the fifth encoder layer is regularized using the two-dimensional distance based method.
6. **ARViT-L6** is the variant where the sixth encoder layer is regularized using this method.

A visual representation of the encoder of each of the ARViT variants utilizing the two-dimensional distance based self-attention regularization method is shown in Figure 6.1.

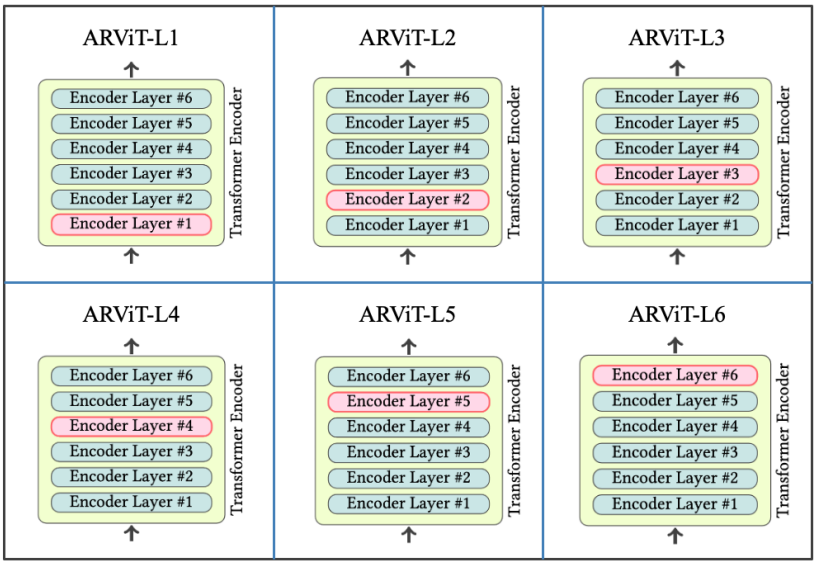


Figure 6.1: Illustration of the encoders (each containing 6 encoder layers) of ARViT-L1, ARViT-L2, ARViT-L3, ARViT-L4, ARViT-L5 and ARViT-L6. The encoder layers in pink indicate the layer being regularized with the two-dimensional distance based self-attention regularization method. Image by the author.

6.1.1.2 ARViT variants regularized with the style similarity based method

Using the style similarity based self-attention regularization method, a total of 12 models were developed. They are: ARViT-R1-1, ARViT-R1-2, ARViT-R1-3, ARViT-R1-4, ARViT-R1-5, ARViT-R1-6, ARViT-R2-1, ARViT-R2-2, ARViT-R2-3, ARViT-R2-4, ARViT-R2-5 and ARViT-R2-6. The

suffix of these models have three elements, each representing one specification. The first element of the suffix is the letter "R", which indicates that the regularization method used was the style similarity based one.

The second element in the suffix is an integer number immediately after the letter "R". This number represents the parameter $\psi = G/P$, where G is the *image region* resolution, and C is the resolution of the *image patches*. Since the image patches of all ARViT variants have resolution $(16, 16)$, when the resolution of the image regions was defined as 16 then $\psi = 16/16 = 1$. Moreover, when the resolution of the image regions was defined as 32 then $\psi = 32/16 = 2$.

Finally, the third element of this suffix is yet another integer, ranging between 1 and 6. This integer is used to identify which encoder layer is being regularized and it immediately follows the integer indicating the value of ψ, with a single hyphen dividing them. Hence:

1. **ARViT-R1-1** is the variant of ARViT where the first encoder layer is regularized using the style similarity based method with region size $G = 16$.
2. **ARViT-R1-2** is the variant of ARViT where the second encoder layer is regularized using the style similarity based method with region size $G = 16$.
3. **ARViT-R1-3** is the variant of ARViT where the third encoder layer is regularized using the style similarity based method with region size $G = 16$.
4. **ARViT-R1-4** is the variant of ARViT where the fourth encoder layer is regularized using the style similarity based method with region size $G = 16$.
5. **ARViT-R1-5** is the variant of ARViT where the fifth encoder layer is regularized using the style similarity based method with region size $G = 16$.
6. **ARViT-R1-6** is the variant of ARViT where the sixth encoder layer is regularized using the style similarity based method with region size $G = 16$.
7. **ARViT-R2-1** is the variant of ARViT where the first encoder layer is regularized using the style similarity based method with region size $G = 32$.
8. **ARViT-R2-2** is the variant of ARViT where the second encoder layer is regularized using the style similarity based method with region size $G = 32$.
9. **ARViT-R2-3** is the variant of ARViT where the third encoder layer is regularized using the style similarity based method with region size $G = 32$.
10. **ARViT-R2-4** is the variant of ARViT where the fourth encoder layer is regularized using the style similarity based method with region size $G = 32$.
11. **ARViT-R2-5** is the variant of ARViT where the fifth encoder layer is regularized using the style similarity based method with region size $G = 32$.
12. **ARViT-R2-6** is the variant of ARViT where the sixth encoder layer is regularized using the style similarity based method with region size $G = 32$.

6.1.2 Hyperparameter choice

For the ARViT variants trained with self-attention regularized using the two-dimensional distance based approach, in order to optimize the balance between the task loss $\mathcal{L}_T$ and the distance loss $\mathcal{L}_D$, we experimented with different values of λ in Equation (3.7), obtaining the best results with $\lambda = 1e^{-3}$. Furthermore, the hyperparameters in Equation (3.2) were set as $\alpha = 4$ and $\beta = 0.5$.

In addition, on ARViT variants trained with the region style similarity based self-attention regularization method, to optimize the trade off between $\mathcal{L}_T$ and $\mathcal{L}_S$, distinct values of λ (Eq. 4.5) were tested, with the best outcomes achieved when $\lambda = 5e^{-3}$. We further experimented with a wide range of values for the distance bias ρ (Eq. 4.2), with the best results obtained when $\rho = 3e^{-1}$.

6.1.3 Pre-training

We pre-train all ARViT variants on the ImageNet dataset with approximately 1.3 million images [1]. However, instead of utilizing the labels of the 1000 categories in the Imagenet Dataset, all models were trained on a self-supervised learning pretext-task. The chosen pretext-task was the *rotation estimation* [2]. In this task, an input image is rotated into one of four possible angles — 0^o; 90^o; 180^o; and 270^o — and the objective of the network is to correctly predict the angle of the performed rotation.

Furthermore, all ARViT variants are pre-trained for 90 epochs, using the *adam* optimizer with a base learning rate of 0.0001. The batch size was defined as 80 images and, following the approach of Caron *et al.* [3], color jittering, Gaussian blur, solarization and multi-crop [4] data augmentations were performed on all images of the training set.

Finally, all models are pre-trained distributed over four NVIDIA GEFORCE GTX 2080 Ti GPUs, with a capacity of 11 Gb each.

6.2 Evaluation methods

To evaluate ARViT variants, we report the Top-1 accuracy after fine-tuning them on the following benchmark classification downstream tasks:

1 - CIFAR-10 [5]. A dataset containing 60000 small resolution (32x32) color images, divided into 10 classes, with 6000 images per class. The classes are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

2 - CIFAR-100 [5]. Similar to CIFAR-10, it contains 60000 small resolution (32x32) color images of 100 classes with 600 images per class.

3 - Oxford 102 Flowers [6]. A dataset containing 6552 large-scale images, divided into 102 categories of flowers common to the United Kingdom. Each class consists of 40 to 258 images.

4 - IMAGENETTE. A subset of Imagenet [1] containing 10716 images and 10 easily classified classes: tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute.[1]

[1] Available at: https://github.com/fastai/imagenette

5 - IMAGEWOOF. A subset of Imagenet [1] containing 10364 images divided into 10 harder to classify classes of dog breeds: Australian terrier, Border terrier, Samoyed, Beagle, Shih-Tzu, English foxhound, Rhodesian ridgeback, Dingo, Golden retriever and Old English sheepdog.[1]

6.3 Results

In this section, the results of the conducted experiments are reported.

6.3.1 ARViT baseline

The Top-1 accuracy of ARViT-Base on the five downstream tasks is shown in Table 6.1. For a matter of comparison, we pre-trained a ViT-Tiny, variant of the ViT [7] introduced in section 5.2.3. The ViT-Tiny also has 6 encoder layers, 12 attention heads and 516 hidden dimensions. However, the ViT-Tiny has 18 million trainable parameters against the 10 million of ARViT.

The ViT-Tiny was also pre-trained on the Imagenet dataset with 1.3 million images on a rotation estimation pretext-task, in batches of 80 images, using the *adam* optimizer with base learning rate 0.0001, and using the same data augmentations as ARViT-Base.

After pre-training, the ViT-Tiny was fine-tuned on the five downstream tasks: CIFAR-10, CIFAR-100, Flowers-102, Imagenette and Imagewoof. A substantial improvement can be observed when comparing the Top-1 accuracy of ARViT-Base with the ViT-Tiny. On CIFAR-10, ARViT-Base outperforms ViT-Tiny by roughly 10%. On CIFAR-100, the improvement is greater than 23%. On the Flowers-102 dataset, ARViT-Base surpasses ViT-Tiny by approximately 8%. On Imagenette, ARViT-Base outperforms the ViT-Tiny by roughly 6%. And finally on the Imagewoof dataset, the improvement obtained by ARViT-Base is of approximately 11%.

Regarding the pre-training time, for ARViT-Base, each epoch was trained on an average of 16m44s, whereas for ViT-Tiny, it took an average of 22m45s for each epoch to be trained. As a matter of comparison, ViT-B, the smaller of the ViTs from Dosovitskiy *et al.* [7], took an average of 1h49m12s to train each epoch on the same machine.

Table 6.1: Top-1 accuracy of ARViT-Base and ViT-Tiny on five different downstream tasks. ViT-Tiny is as a variant of the ViT [7] included in the table for effects of comparison.

Model	CIFAR-10	CIFAR-100	Flowers	Imagenette	Imagewoof
ViT-Tiny	73.30%	53.55%	73.91%	84.72%	61.77%
ARViT-Base	83.13%	76.27%	81.61%	91.18%	73.01%

6.3.2 ARViT with self-attention regularized using a 2D spatial distance based approach

The Top-1 accuracy on the 5 downstream tasks attained by the ARViT variants pre-trained using the two-dimensional distance based self-attention regularization approach is shown in Table 6.2. Although no substantial improvements could be observed on ARViT-L1, ARViT-L2, ARViT-L4, ARViT-L5 and ARViT-L6, when regularizing self-attention on the third encoder layer, significant gains in accuracy were observed on all five tasks. Namely, ARViT-L3 improves the performance of ARViT-Base by 1.63% on CIFAR-10, 0.36% on CIFAR-100, 1.47% on Flowers-102, 0.87% on Imagenette and 2.89% on Imagewoof.

Table 6.2: Comparison between the performance of ARViT on different downstream tasks when regularized with the two-dimensional distance based method. Values in bold indicate the model with best performance on each downstream task.

Model	CIFAR-10	CIFAR-100	Flowers	Imagenette	Imagewoof
ViT-Tiny	73.30%	53.55%	73.91%	84.72%	61.77%
ARViT-Base	83.13%	76.27%	81.61%	91.18%	73.01%
ARViT-L1	82.02%	75.89%	80.12%	90.58%	72.92%
ARViT-L2	83.63%	76.20%	81.27%	91.56%	74.81%
ARViT-L3	**84.76%**	**76.63%**	**83.08%**	**92.05%**	**75.90%**
ARViT-L4	83.31%	76.52%	81.98%	91.88%	75.24%
ARViT-L5	83.70%	76.01%	81.21%	91.30%	73.46%
ARViT-L6	79.81%	74.84%	79.14%	88.77%	70.44%

As for the training time, for ARViT-Base, and ViT-Tiny, the reported training times per epoch were 16m44s and 22m45s, respectively. For the ARViT variants pre-trained using the two-dimensional distance based self-attention regularization method — namely ARViT-L1, ARViT-L2, ARViT-L3, ARViT-L4, ARViT-L5 and ARViT-L6 — the average training time per epoch was 18m20s.

6.3.2.1 Comparison with State of the Art

So far, studies on self-supervised vision transformers, like the SiT [8], MoCo v3 [9] and DINO [3], were conducted using variants of the original ViT [7]. ARViT's capacity is significantly smaller than these models, with only 6 encoder layers, 12 attention heads, and roughly 10M trainable parameters. In Table 6.3, the performance of ARViT was compared with other self-supervised vision transformers on two small downstream classification tasks: CIFAR-10 and CIFAR-100. ARViT-L3 shows an improvement of 3.56% on CIFAR-10 and 20,66% on CIFAR-100 when compared with the SiT [8], a model with roughly 9 times more parameters than ours. However, MoCo v3 [9] and DINO [3] still marginally outperform ARViT on these tasks.

Table 6.3: Linear evaluation of vision transformers after self-supervised pre-training. Accuracy reported on CIFAR-10 and CIFAR-100. ARViT-L3 has its self-attention regularized on the 3th encoder layer, and it is the best performing ARViT variant using the two-dimensional distance based regularization method.

Model	CIFAR-10	CIFAR-100
SiT[8]	81.20%	55.97%
MoCo v3 (ViT-B) [9]	98.9%	90.5%
MoCo v3 (ViT-L) [9]	99.1%	91.1%
MoCo v3 (ViT-H) [9]	99.1%	91.2%
DINO (ViT-S/16) [3]	99.0%	90.5%
DINO (ViT-B/16) [3]	99.1%	91.7%
ARViT-Base (ours)	83.13%	76.27%
ARViT-L3 (ours)	84.76%	76.63%

6.3.3 ARViT with self-attention regularized using a region similarity based approach

When applying the two-dimensional spatial distance regularization to the 3rd encoder layer of ARViT (ARVIT-L3), it achieved the highest Top-1 accuracy on all 5 downstream classification tasks, surpassing ARViT-Base and the ViT-Tiny by margins as big as 3% and 23%, respectively.

Experiments with the style similarity based self-attention regularization method deployed on ARViT demonstrate marginal improvements in the Top-1 accuracy on all 5 downstream classification tasks. Those gains in performance occur in comparison with both the baseline architecture (ARViT-Base), as well with the ARViT variants with self-attention regularized using the two-dimensional spatial distance method. Table 6.4 reports the Top-1 accuracy on each downstream task of the models pre-trained using the style-similarity based self-attention regularization method.

Table 6.4: Comparison between the performance of ARViT on different downstream tasks when regularized with the style similarity based method. Values in bold indicate the model with best performance on each downstream task.

Model	CIFAR-10	CIFAR-100	Flowers	Imagenette	Imagewoof
ViT-Tiny	73.30%	53.55%	73.91%	84.72%	61.77%
ARViT-Base	83.13%	76.27%	81.61%	91.18%	73.01%
ARViT-L3	84.76%	76.63%	83.08%	92.05%	75.90%
ARViT-R2-1	86.97%	84.85%	83.75%	95.18%	80.27%
ARViT-R2-2	87.76%	87.25%	85.33%	**95.81%**	**83.28%**
ARViT-R2-3	88.29%	88.45%	85.52%	95.66%	81.93%
ARViT-R2-4	87.06%	86.09%	84.72%	94.99%	81.08%
ARViT-R2-5	87.00%	86.17%	85.33%	95.29%	81.73%
ARViT-R2-6	87.40%	84.80%	84.91%	94.51%	80.81%
ARViT-R1-1	88.29%	88.63%	85.58%	95.48%	82.31%
ARViT-R1-2	86.99%	86.40%	85.15%	94.28%	79.61%
ARViT-R1-3	87.17%	84.91%	85.33%	95.36%	82.20%
ARViT-R1-4	87.54%	86.13%	85.31%	95.48%	82.93%
ARViT-R1-5	**88.45%**	**89.65%**	**85.82%**	95.40%	82.70%
ARViT-R1-6	87.55%	87.31%	85.15%	95.78%	81.15%

On CIFAR-10, the best performing model was ARViT-R1-5, with a Top-1 accuracy of 88.45%. That represents a gain of roughly 4% in accuracy compared to ARViT-L3, and 5% in comparison with ARViT-Base.

ARViT-R1-5 also achieved the best performance on CIFAR-100 and the Flowers dataset [6]. On CIFAR-100 it achieved a Top-1 accuracy of 89.65%, surpassing ARViT-Base and ARViT-L3 by roughly 13%. On the Flowers dataset the gains were around 4% and 3% in relation to ARViT-Base and ARViT-L3, respectively.

When finetuned on Imagenette and Imagewoof, the model that achieve the best Top-1 accuracy was ARViT-R2-2. Again, it surpasses ARViT-Base and ARViT-L3 by large margins. On Imagenette, the accuracy gain is a bit short of 5% compared to ARViT-Base, and 4% compared to ARViT-L3. On the Imagewoof classification task, the accuracy improvement achieved in comparison with those models was roughly 10% and 8%, respectively.

Furthermore, when deploying our style similarity based self-attention regularization method to ARViT and adopting the same training specifications, the observed training time was of 18m44s per epoch.

6.3.3.1 Regularization effects

We also evaluate changes in attention loss of each encoder layer during training. Figure 6.2 compares the regularized (similarity loss is minimized) and non-regularized (ARViT-Base) similarity loss of each encoder layer. It can be noticed that even on ARViT-Base, during the learning process, the network implicitly learns self-attention in such way that the similarity loss decreases. Furthermore, when self-attention is regularized, the similarity loss is reduced faster and converges at a lower value.

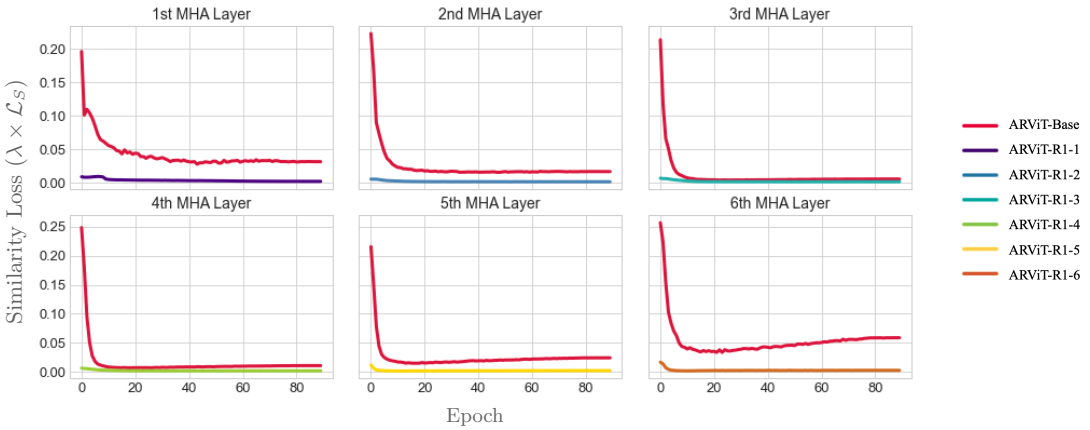


Figure 6.2: Overview of the Similarity Loss (multiplied by λ) during self-supervised pre-training of ARViT's variants on the ImageNet dataset [1], with region resolution 16x16. Each graph compares the similarity loss on a layer of the ARViT-Base against the ARViT variant in which the same layer was regularized.

We also perform a visual inspection of the attention maps to qualitatively analyse the attention obtained, with visualizations displayed in Figure 6.3. Regularized layers covers the target of the image more precisely and attend less to the background, regardless of the target size. This demonstrates that introducing self-attention regularization indeed mitigates absence of locality and scale-invariance in self-attention of visual transformers.

6.3.3.2 Comparison with State of the Art

The style-similarity based self-attention regularization method was deployed on ARViT, a small vision transformer with 6 encoder layers and 8 attention heads. Currently, state-of-the-art self-supervised vision transformers possess marginally larger capacities, with the SiT [8], MoCo v3 [9] and DINO [3] ranging from 12 to 32 encoder layers. In addition, ARViT also has a significantly smaller number of trainable parameters.

In Table 6.5, the performance of ARViT-Base, ARViT-L3 (the best performing ARViT variant using the two-dimensional distance based regularization method), and ARViT-R1-5 (the best performing ARViT variant using the style similarity based regularization method) are compared with other self-supervised visual transformers on two small size classification downstream tasks: CIFAR-10 and CIFAR-100. ARViT-R1-5 shows an improvement of 7.25% on CIFAR-10 and 33.68% on CIFAR-100 when compared with the SiT [8], a model with roughly 9 times more parameters than ARViT. Furthermore, ARViT-R1-5 also achieved competitive results on CIFAR-100 when compared to very large architectures, like MoCo v3 [9] and DINO [3]. Such results were obtained mainly due to self-attention regularization, which increased the ARViT-Base performance by more than 13%. However, MoCo v3 [9] and DINO [3] still marginally outperform ARViT-R1-5 on CIFAR-10.

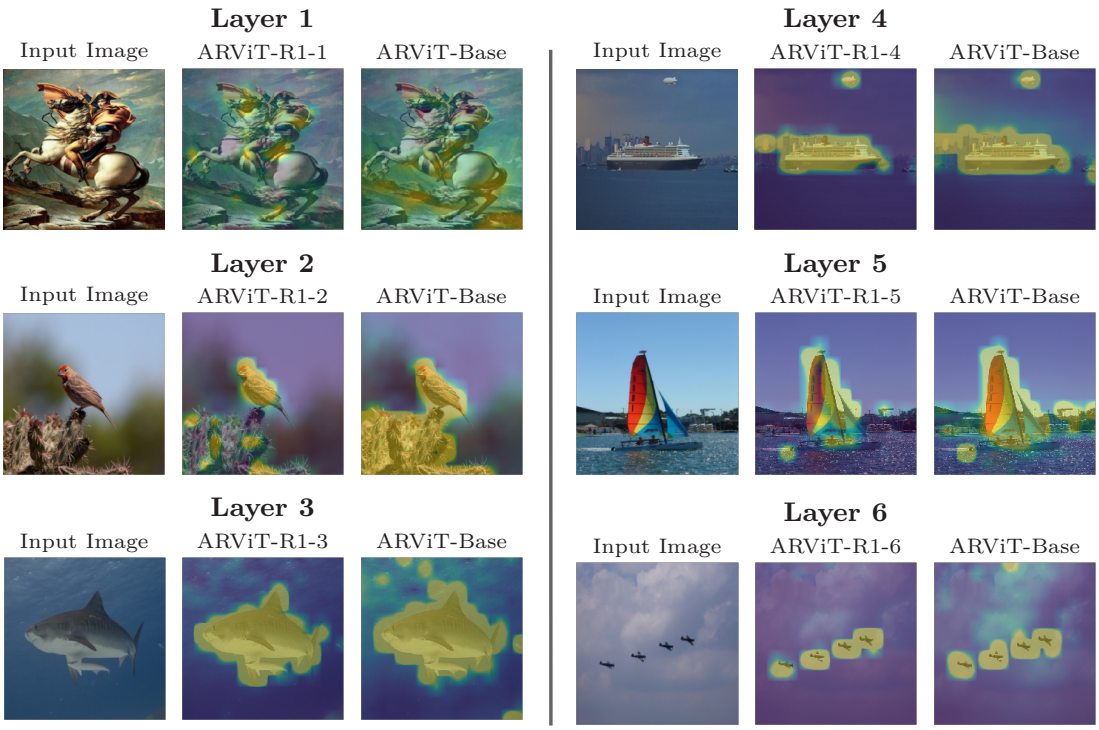


Figure 6.3: Attention maps produced by: the first encoder layer on ARViT-Base and ARViT-R1-1; the second encoder layer on ARViT-Base and ARViT-R1-2; the third encoder layer on ARViT-Base and ARViT-R1-3; the fourth encoder layer on ARViT-Base and ARViT-R1-4; the fifth encoder layer on ARViT-Base and ARViT-R1-5; the sixth encoder layer on ARViT-Base and ARViT-R1-6. For each layer, both attention maps indicate the self-attention values for the same image patch. The resolution of the attention maps was enhanced via interpolation to match that of the images.

Table 6.5: Linear evaluation of vision transformers after self-supervised pre-training. Accuracy reported on CIFAR-10 and CIFAR-100. ARViT-R1-5 has its self-attention regularized on the 5th encoder layer, and it is the best performing ARViT variant on these tasks using the style similarity based regularization method.

Model	CIFAR-10	CIFAR-100
SiT[8]	81.20%	55.97%
MoCo v3 (ViT-B) [9]	98.9%	90.5%
MoCo v3 (ViT-L) [9]	99.1%	91.1%
MoCo v3 (ViT-H) [9]	99.1%	91.2%
DINO (ViT-S/16) [3]	99.0%	90.5%
DINO (ViT-B/16) [3]	99.1%	91.7%
ARViT-Base (ours)	83.13%	76.27%
ARViT-L3 (ours)	84.76%	76.63%
ARViT-R1-5 (ours)	88.45%	89.65%

6.4 Discussion

This chapter described the the experiments conducted and the results attained during the research presented in this dissertation. The experiments aimed at tackling the lack of inductive biases on vision transformers, addressing the ability of smaller capacity vision transformers to generalize well, reducing the computational cost associated with training high performance vision transformers, and to improve the ability of vision transformers when pre-trained on unlabeled data.

The experiments involved the exploration of a new vision transformer architecture, denoted

ARViT, and two novel self-attention regularization methods — the first based on the pairwise two-dimensional distance between image patches, and the second based on the style similarity of image regions. Those experiments also aimed at observing if regularization can be an effective tool to encapsulate an assumption about the problem's solution in order to promote better generalization.

6.4.1 ARViT analysis

Experimental results with ARViT-base (without any self-attention regularization) indicate that the baseline architecture obtained marginal improvements in the Top-1 accuracy on all downstream tasks when compared to the ViT-Tiny, which is a vision transformer with similar dimensions. The gains obtained by ARViT-Base over the ViT-Tiny were as big as 23%, and such expressive results indicate that, the changes performed in the ViT architecture that led up to ARViT have a major role in improving the model's performance. Furthermore, though ARViT-Base and ViT-Tiny share the same number of encoder layers, attention-heads and hidden dimensions, because of the architectural changes, ARViT has only 10 million trainable parameters, as opposed to the 18 million in ViT-Tiny, demonstrating that it is, indeed, possible to improve performance even with a smaller capacity model.

6.4.2 Two-dimensional distance based self-attention regularization analysis

Experiments deploying the *two-dimensional distance based self-attention regularization* on ARViT indicated that, in most cases, the method had little to no effect on the models performance. Most especially, when applied to the first, second, fourth, fifth and sixth layers there were no real performance gains attained. Nevertheless, when applying the two-dimensional distance based self-attention regularization method to the third layer of ARViT (namely ARViT-L3), the model achieved notable gains (up to 2.89%) in the Top-1 accuracy of all five downstream tasks. This indicates that the assumption made in this regularization method is not valid for most encoder layers, being useful only in specific cases.

6.4.3 Style similarity based self-attention regularization analysis

Experiments with ARViT deploying the *style similarity based self-attention regularization* showed that the method was able to lead ARViT's performance to marginal gains when compared to its baseline model, ARViT-Base. In fact, all 12 variants of ARViT regularized using this method had major improvements in Top-1 accuracy on all 5 downstream tasks, with gains as big as 13%.

This experiments also measured the impact of changing the granularity of the style representation in this regularization method. Two sets of 6 models each were trained using *image regions* of resolution 16 and 32, respectively. It observed that, models trained with a smaller style granularity (region resolution $G = 32$), promoted the largest gains on downstream tasks containing images with higher resolution — Imagenette and Imagewoof. On the other hand, a larger style granularity (region resolution $G = 16$) was responsible for the larger accuracy gains on downstream tasks with images of small resolution — CIFAR-10, CIFAR-100 and Flowers 102.

6.4.4 Summary

The experimental results indicate that the two self-attention regularization methods are in fact introducing new inductive biases to vision transformers, thus restricting the hypothesis space and making certain solutions preferable, leading to the conclusion that regularization can in fact be a tool that helps conciliate both the connectionist and symbolic approaches to AI.

Though ARViT and its variants still unperformed when compared to the very large state-of the art architectures pre-trained on tens of millions of images, it was able to lower this gap by marking a considerable improvement in relation to vision transformers of similar capacity. A comparison between all models learned in the experiments is shown in Table 6.6.

Finally, from a qualitative point of view, the attention maps also learned to represent the self-attention scores according to assumption embedded in the regularization term. Appendix A displays a myriad of attention maps, taken from different images and generated by different models.

Table 6.6: Comparison between the performance of ARViT on different downstream tasks when regularized with different methods. Furthermore, ViT-Tiny is as a variant of the ViT [7] included in the table for effects of comparison. It has the same number of encoder layers and attention heads as ARViT. Values in bold indicate the model with best performance on each downstream task.

Model	Regularization Type	Regularized Layer	Region resolution	CIFAR-10	CIFAR-100	Flowers	Imagenette	Imagewoof
ViT-Tiny	-	-	-	73.30%	53.55%	73.91%	84.72%	61.77%
ARViT-Base	-	-	-	83.13%	76.27%	81.61%	91.18%	73.01%
ARViT-L1	2D distance	1st layer	-	82.02%	75.89%	80.12%	90.58%	72.92%
ARViT-L2	2D distance	2nd layer	-	83.63%	76.20%	81.27%	91.56%	74.81%
ARViT-L3	2D distance	3rd layer	-	84.76%	76.63%	83.08%	92.05%	75.90%
ARViT-L4	2D distance	4th layer	-	83.31%	76.52%	81.98%	91.88%	75.24%
ARViT-L5	2D distance	5th layer	-	83.70%	76.01%	81.21%	91.30%	73.46%
ARViT-L6	2D distance	6th layer	-	79.81%	74.84%	79.14%	88.77%	70.44%
ARViT-R2-1	Region similarity	1st layer	32	86.97%	84.85%	83.75%	95.18%	80.27%
ARViT-R2-2	Region similarity	2nd layer	32	87.76%	87.25%	85.33%	**95.81%**	**83.28%**
ARViT-R2-3	Region similarity	3rd layer	32	88.29%	88.45%	85.52%	95.66%	81.93%
ARViT-R2-4	Region similarity	4th layer	32	87.06%	86.09%	84.72%	94.99%	81.08%
ARViT-R2-5	Region similarity	5th layer	32	87.00%	86.17%	85.33%	95.29%	81.73%
ARViT-R2-6	Region similarity	6th layer	32	87.40%	84.80%	84.91%	94.51%	80.81%
ARViT-R1-1	Region similarity	1st layer	16	88.29%	88.63%	85.58%	95.48%	82.31%
ARViT-R1-2	Region similarity	2nd layer	16	86.99%	86.40%	85.15%	94.28%	79.61%
ARViT-R1-3	Region similarity	3rd layer	16	87.17%	84.91%	85.33%	95.36%	82.20%
ARViT-R1-4	Region similarity	4th layer	16	87.54%	86.13%	85.31%	95.48%	82.93%
ARViT-R1-5	Region similarity	5th layer	16	**88.45%**	**89.65%**	**85.82%**	95.40%	82.70%
ARViT-R1-6	Region similarity	6th layer	16	87.55%	87.31%	85.15%	95.78%	81.15%

6.5 References

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

[2] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[3] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9630–9640. [Online]. Available: https://ieeexplore.ieee.org/document/9709990/

[4] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9912–9924, 2020.

[5] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009, publisher: Toronto, ON, Canada.

[6] M.-E. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image*

Processing. Bhubaneswar, India: IEEE, Dec. 2008, pp. 722–729. [Online]. Available: http://ieeexplore.ieee.org/document/4756141/

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

[8] S. Atito, M. Awais, and J. Kittler, "SiT: Self-supervised vIsion Transformer," *arXiv:2104.03602 [cs]*, Apr. 2021, arXiv: 2104.03602. [Online]. Available: http://arxiv.org/abs/2104.03602

[9] X. Chen, S. Xie, and K. He, "An Empirical Study of Training Self-Supervised Vision Transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9620–9629. [Online]. Available: https://ieeexplore.ieee.org/document/9711302/

Chapter 7

Conclusions and Future Work

The highest forms of understanding we can achieve are laughter and human compassion.

—**Richard Feynman**

(Theoretical Physicist)

For small creatures such as we the vastness is bearable only through love.

—**Carl Sagan**

(Astronomer, astrophysicist and science communicator)

The work presented in this dissertation addressed the lack of inductive biases on vision transformers, a limitation usually tackled by pre-training large capacity models pre-trained on hundreds of millions of labeled data. In parallel, this work also explored the ability of smaller capacity vision transformers to generalize well, reducing the computational cost associated with training high performance vision transformers, and aimed at improving the ability of vision transformers when pre-trained on unlabeled data. This chapter provides the summary of this work's main propositions and contributions. This chapter also circles back to the research objectives presented in chapter 1, summarizing how the results of this research contributed to them. Finally, possible future research directions to further extend this line of research are also presented.

7.1 Conclusion

In this work, inspiration was drawn from the trade-off between the connectionist and the symbolic approaches to artificial intelligence in order to address the lack of inductive biases on vision transformers. The connectionist approach is concerned with model learning, and the symbolic approach focus on knowledge derived from formal reasoning being passed down to an AI model.

From that perspective, the first contributions of this research were two *self-attention regularization* methods. Regularization is a tool in machine learning used to reduce overfitting and

improve a model's generalization [1]. Effectively, regularization can be interpreted as a mean to restrict a machine learning problem's hypothesis space, thus favoring specific prior distributions on model parameters [2]. Therefore, the proposed two self-attention regularization methods were designed to be means of restricting the vision transformer's hypothesis space based on assumptions about the learning problem. Hence, though logic was used to constrain the space of possible solutions, it did not impose a solution by itself, and the vision transformer was still able to learn within the available hypothesis space, and therefore, this method cannot be formally considered a combination between the connectionist and symbolic approaches per se.

The first proposed self-attention regularization method was based on the pairwise two-dimensional distance between image patches, measured using the *the Manhattan distance*. The assumption made by this self-attention regularization method was that image patches that are spatially close to each other should present higher self-attention scores than patches that are spatially distant to each other. In this method, a novel loss function, denoted *distance loss*, was introduced. The distance loss acts as a regularization term added to the vision transformer's training loss, and it encapsulates the logic of the method to penalize solutions based on the pairwise two-dimensional distance between image patches and the respective self-attention scores between them.

The second self-attention regularization method proposed in this research was based on the style similarity between different regions of an image. The style representation of a region of an image was obtained through its *gram matrix*, following the approach of [3]. The assumption made in this method was that, the more similar the style of two regions are, the higher the self-attention scores between them should be. Contrarily, the least similar the style of two regions, the smaller the self-attention scores between them should be. To measure the distance between region styles, the pairwise mean squared error between their gram matrices was utilized. The core of our method is the *Similarity Loss*, designed to express the assumption of this method. It, as well, was added to the network training loss to act as a regularization term, penalizing solutions based on the pairwise similarity between image region styles and their respective self-attention scores.

Furthermore, this work introduced ARViT (Attention Regularized Vision Transformer), a vision transformer architecture inspired by the ViT [4] and with changes inspired by Chen *et al.* [5]. ARViT is significantly smaller than the ViT, and in it, the proposed self-attention regularization methods were deployed.

At the experiments, all models developed in this researched were pre-trained on a self-supervised task using the ImageNet dataset [6], with approximately 1.3 million images. Results revealed that ARViT, without any self-attention regularization, when compared to a similar capacity vision transformer, already obtained performance gains as big as 23% on benchmark classification tasks.

Moreover, the two proposed self-regularization methods further improved the performance of ARViT as far as up to 13% on those tasks, thus indicating that the two self-attention regularization methods are in fact introducing new inductive biases to vision transformers, by restricting the hypothesis space and making certain solutions preferable.

Finally, the models presented in this research also achieve competitive results with state-of-the-art visual transformers, in spite of being a smaller-capacity model and trained on a mid-size dataset.

The scientific contributions derived from the work in this dissertation are summarized as follows:

1. A method to introduce a new inductive bias to vision transformers through self-attention regularization based on the two-dimensional spatial distance between image patches, and a new loss function, denoted *distance loss*.
2. A new inductive bias to vision transformers introduced via a self-attention regularization method based on style similarity of distinct regions of an image, and a new loss function, denoted *attention loss*.
3. ARViT, a reduced variant of the ViT [4], with changes inspired by the work of Chen *et al.* [5]. ARViT has 6 encoder layers, 12 attention heads on each layer and 10M trainable parameters, which is a considerably smaller capacity when compared with the original ViT.
4. ARViT's was pre-trained using a self-supervised learning approach with rotation estimation as a pretext-task [7], and evaluated on small benchmark downstream tasks, outperforming a similar capacity variant of the ViT by large margins on all tasks (up to 23%).
5. The deployment of the two proposed self-attention regularization methods on ARViT resulted in further marginal performance gains on all small benchmark downstream tasks (up to 13%).

7.2 Future work

Given the positive results obtained in the experiments conducted in this research, the following possibilities of future works were opened:

1. Further explore the use of regularization — on vision transformers and other network architectures — as a method to introduce inductive biases based on assumptions about the learning problem.

2. Experiments with the two self-attention regularization methods proposed in this work on other vision transformer architectures.

3. Explore the concept of self-attention regularization on vision transformers pre-trained on different self-supervised learning pretext-tasks. Though rotation estimation provides a good supervision signal, it provides the same amount of feedback information as a standard supervised classification task. Therefore, pretext-tasks that "mask" more information from the input could be explored.

7.3 References

[1] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[2] K. Weinberger, "Linear Regression," Date Accessed: 2022-07-15. [Online]. Available: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote08.html#map-estimate

[3] L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *arXiv:1508.06576 [cs, q-bio]*, Sep. 2015, arXiv: 1508.06576. [Online]. Available: http://arxiv.org/abs/1508.06576

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

[5] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, "Visformer: The vision-friendly transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 589–598.

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

[7] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

Appendices

Appendix A

Figures of Attention Maps

This appendix contains figures with different representations of attention maps produced by ARViT variants.

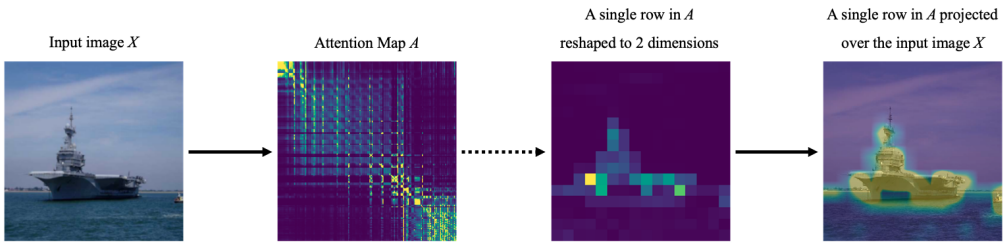


Figure A.1: Attention map — example (1). Attention map generated by the 3rd layer of ARViT-R1-3 for an input image X. A single row A_p can be reshaped into two dimensions.

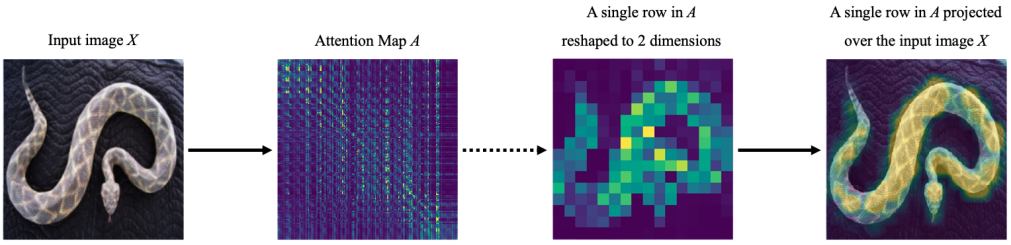


Figure A.2: Attention map — example (2). Attention map generated by the 3rd layer of ARViT-R1-3 for an input image X. A single row A_p can be reshaped into two dimensions.

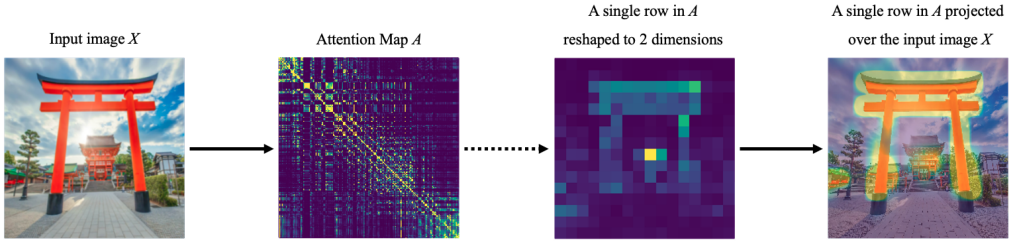


Figure A.3: Attention map — example (3). Attention map generated by the 3rd layer of ARViT-R1-3 for an input image X. A single row A_p can be reshaped into two dimensions.

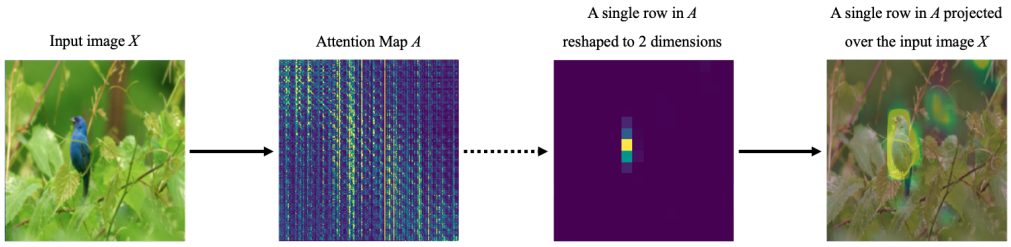


Figure A.4: Attention map — example (4). Attention map generated by the 5rd layer of ARViT-R1-5 for an input image X. A single row A_p can be reshaped into two dimensions.

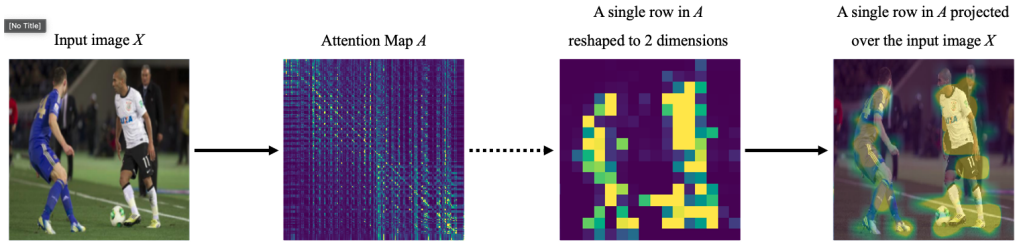


Figure A.5: Attention map — example (5). Attention map generated by the 5rd layer of ARViT-R1-5 for an input image X. A single row A_p can be reshaped into two dimensions.

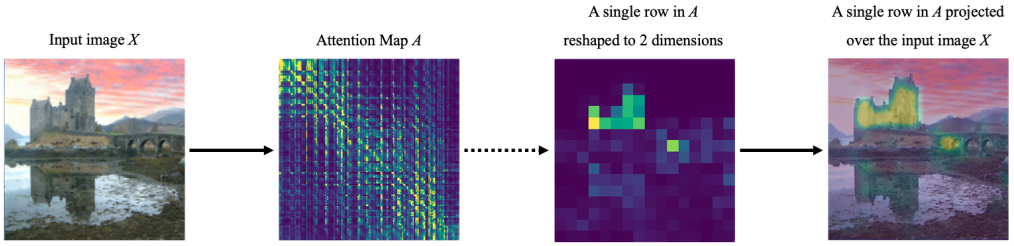


Figure A.6: Attention map — example (6). Attention map generated by the 5rd layer of ARViT-R1-5 for an input image X. A single row A_p can be reshaped into two dimensions.

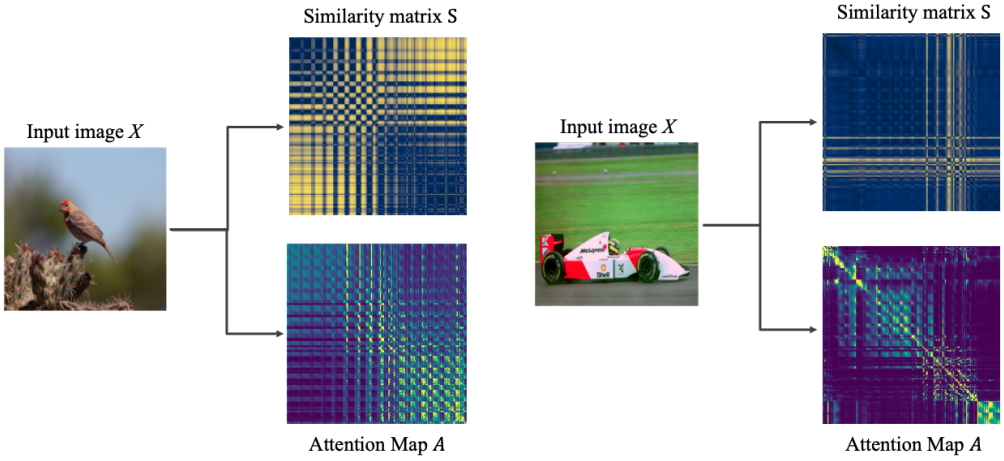


Figure A.7: Similarity matrix S and attention map A. Example (1) (left): input image X, similarity matrix with region size $G = 16$, and attention map generated by the third layer of ARViT-R1-3. Example (2) (right): input image X, similarity matrix with region size $G = 16$, and attention map generated by the third layer of ARViT-R1-3.

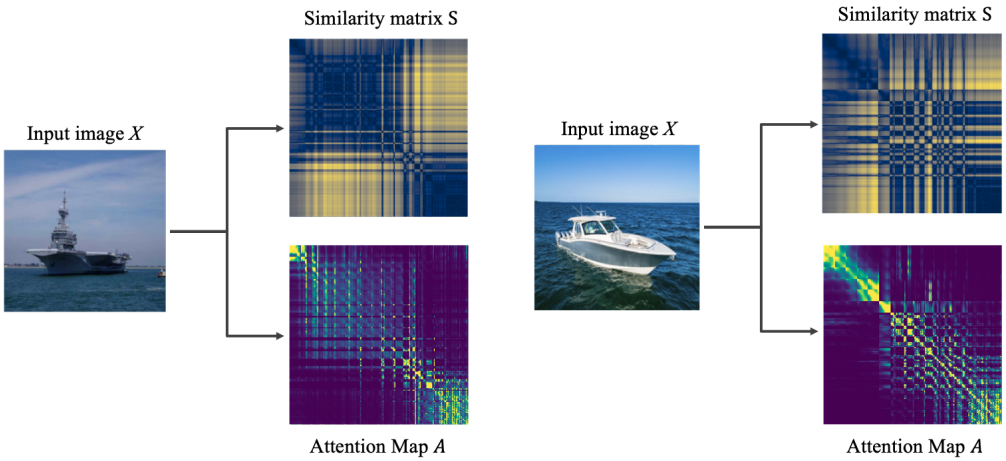


Figure A.8: Similarity matrix S and attention map A. Example (3) (left): input image X, similarity matrix with region size $G = 16$, and attention map generated by the fifth layer of ARViT-R1-5. Example (4) (right): input image X, similarity matrix with region size $G = 16$, and attention map generated by the fifth layer of ARViT-R1-5.

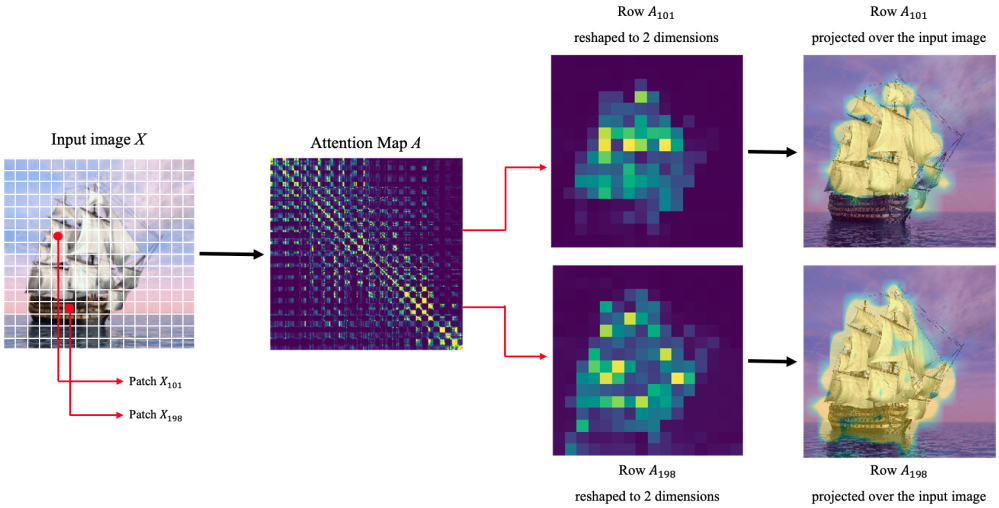


Figure A.9: Visual comparison between two distinct rows on the attention map A. On the left, an input image A is divided into 256 patches. From it, an attention map A is generated by the 5th layer of ARViT-R1-5 (regularized using style similarity). The input image contains a sail-ship. Row A_{101} contains the pairwise self-attention values between the highlighted patch X_{101} and all other patches. Row A_{198} contains the pairwise self-attention values between the highlighted patch X_{198} and all other patches. By reshaping rows A_{101} and A_{198}, it is possible to visualize a 2D attention map showing how much patches X_{101} and X_{198}, respectively, attend all other patches. Patch X_{101} mainly contains "sails" and A_{101} mostly attends to patches containing sails. Patch X_{198} mainly contains the hull of the ship, however, A_{198} attends all patches containing the ship (including the sails).

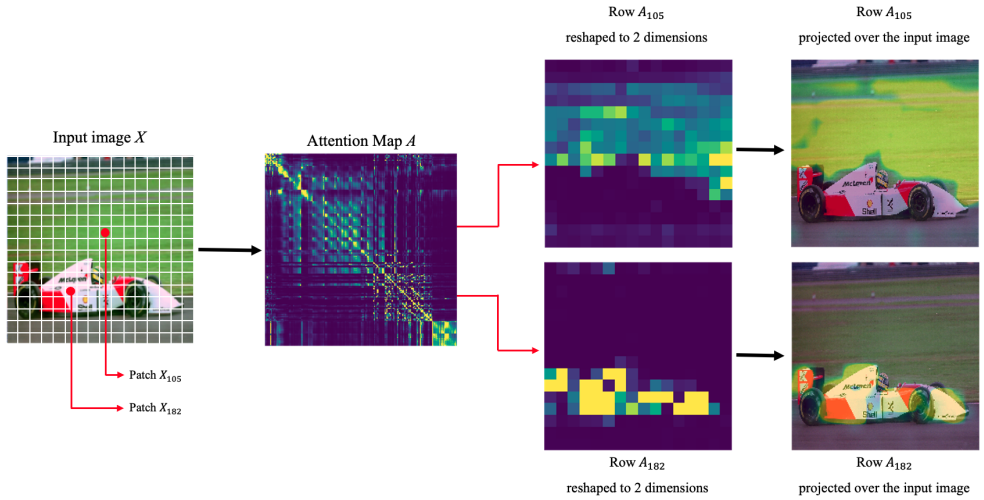


Figure A.10: Visual comparison between two distinct rows on the attention map A. On the left, an input image A is divided into 256 patches. From it, an attention map A is generated by the 5th layer of ARViT-R1-5 (regularized using style similarity). The input image contains a F1 car. Row A_{105} contains the pairwise self-attention values between the highlighted patch X_{105} and all other patches. Row A_{182} contains the pairwise self-attention values between the highlighted patch X_{182} and all other patches. By reshaping rows A_{105} and A_{182}, it is possible to visualize a 2D attention map showing how much patches X_{105} and X_{182}, respectively, attend all other patches. Patch X_{105} mainly contains "grass" and A_{105} mostly attends to patches containing grass. Patch X_{182} mainly contains the F1 car, and A_{182} attends all patches containing the F1 car.

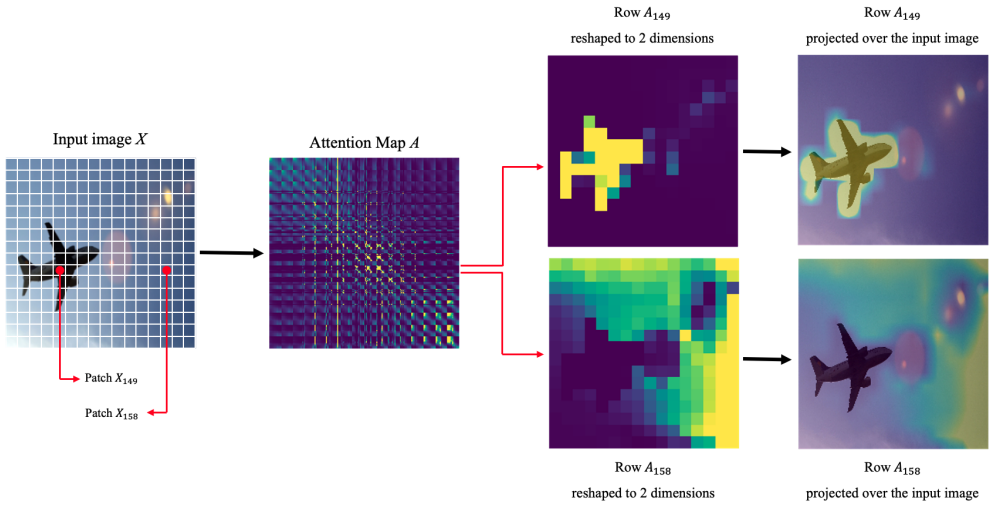


Figure A.11: Visual comparison between two distinct rows on the attention map A. On the left, an input image A is divided into 256 patches. From it, an attention map A is generated by the 5th layer of ARViT-R1-5 (regularized using style similarity). The input image contains an airplane. Row A_{149} contains the pairwise self-attention values between the highlighted patch X_{149} and all other patches. Row A_{158} contains the pairwise self-attention values between the highlighted patch X_{158} and all other patches. By reshaping rows A_{149} and A_{158}, it is possible to visualize a 2D attention map showing how much patches X_{149} and X_{158}, respectively, attend all other patches. Patch X_{149} mainly contains "airplane" and A_{149} attends to patches containing mostly "airplane". Patch X_{158} mainly contains the sky, and A_{158} attends mainly the patches containing the sky.

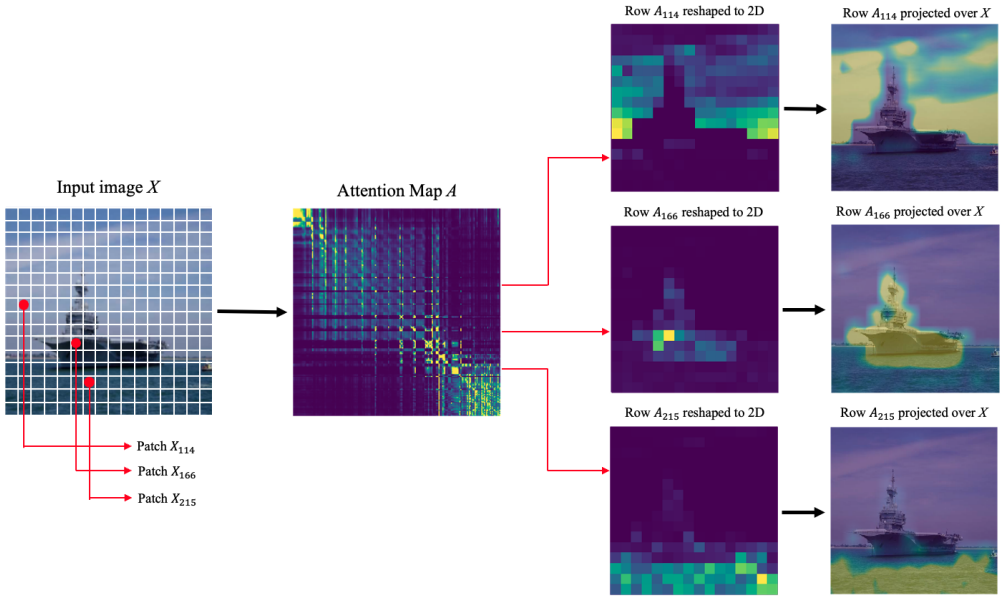


Figure A.12: Visual comparison between three distinct rows on the attention map A. On the left, an input image A is divided into 256 patches. From it, an attention map A is generated by the 5th layer of ARViT-R1-5 (regularized using style similarity). The input image contains an aircraft carrier ship. Row A_{114} contains the pairwise self-attention values between the highlighted patch X_{114} and all other patches. Row A_{166} contains the pairwise self-attention values between the highlighted patch X_{166} and all other patches. Row A_{215} contains the pairwise self-attention values between the highlighted patch X_{215} and all other patches. By reshaping rows A_{114}, A_{166} and A_{215}, it is possible to visualize a 2D attention map showing how much patches X_{114}, X_{166} and X_{215}, respectively, attend all other patches. Patch X_{114} mainly contains the sky and A_{114} attends to patches containing mostly sky. Patch X_{166} mainly contains the aircraft carrier ship, and A_{166} attends mainly the patches containing the the ship. Patch X_{215} mainly contains the sea, and A_{166} attends mainly the patches containing the sea.

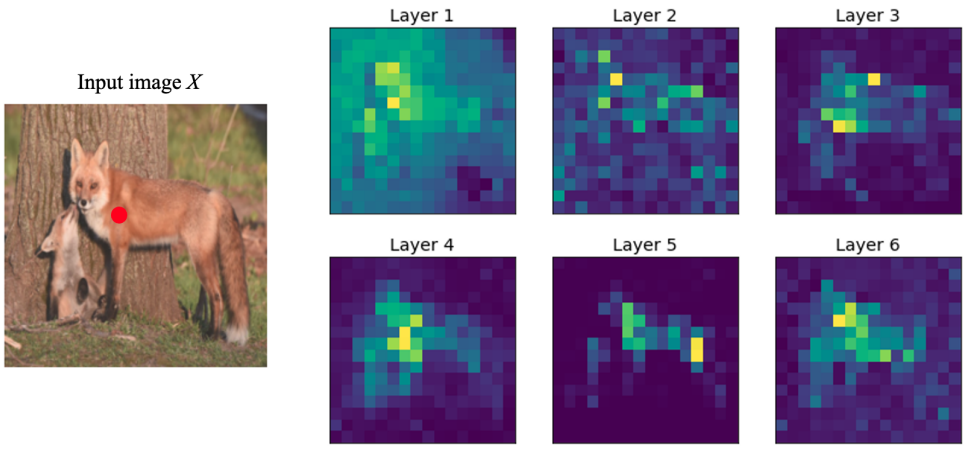


Figure A.13: Differences between attention maps generated by different encoder layers - example (1). Six 2D representations of a row p in attention maps produced by each encoder layer of ARViT-R1-5. The attention scores on row p indicate how much patch X_p (highlighted in red) attend to all other patches.

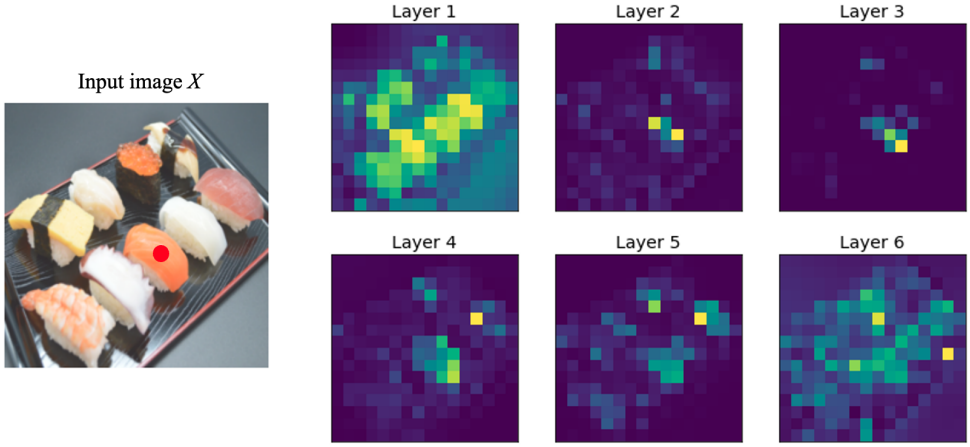


Figure A.14: Differences between attention maps generated by different encoder layers - example (2). Six 2D representations of a row p in attention maps produced by each encoder layer of ARViT-R1-3. The attention scores on row p indicate how much patch X_p (highlighted in red) attend to all other patches.

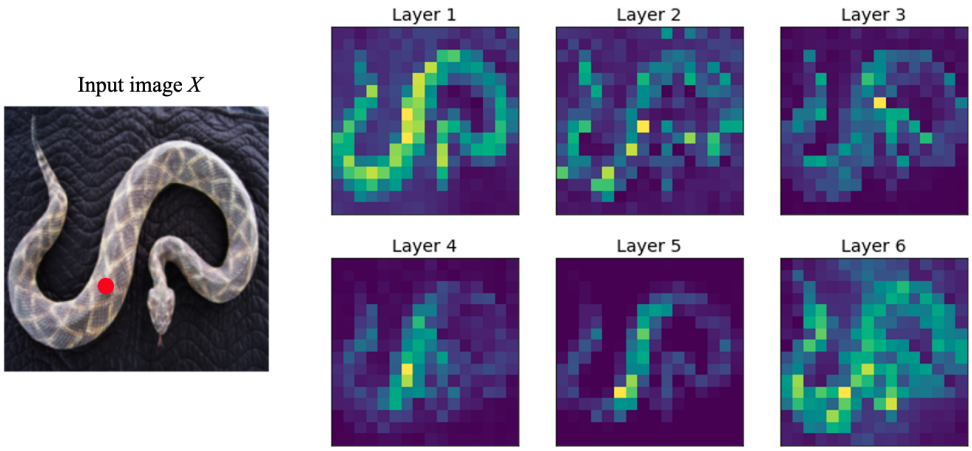


Figure A.15: Differences between attention maps generated by different encoder layers - example (3). Six 2D representations of a row p in attention maps produced by each encoder layer of ARViT-R1-5. The attention scores on row p indicate how much patch X_p (highlighted in red) attend to all other patches.

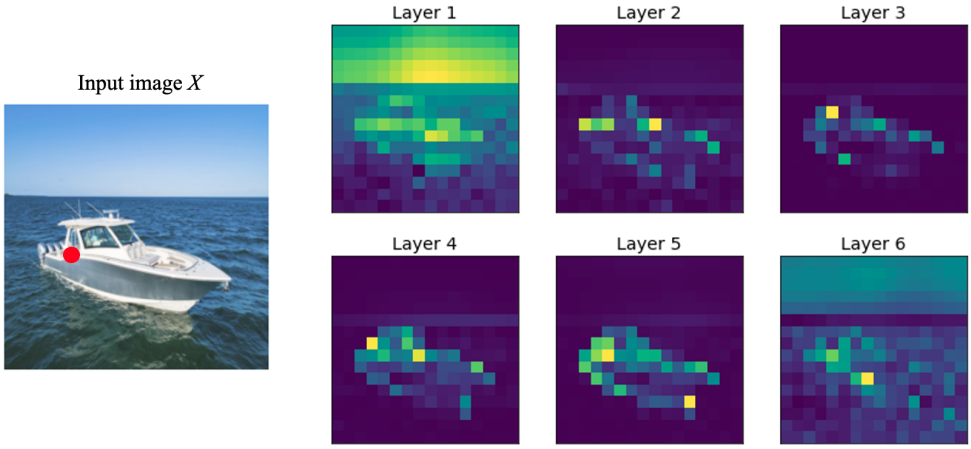


Figure A.16: Differences between attention maps generated by different encoder layers - example (4). Six 2D representations of a row p in attention maps produced by each encoder layer of ARViT-R1-5. The attention scores on row p indicate how much patch X_p (highlighted in red) attend to all other patches.

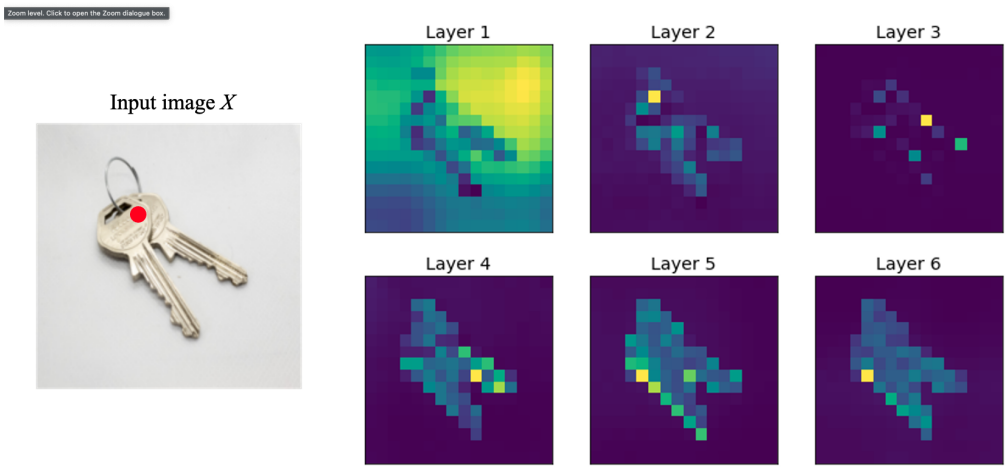


Figure A.17: Differences between attention maps generated by different encoder layers - example (5). Six 2D representations of a row p in attention maps produced by each encoder layer of ARViT-R1-5. The attention scores on row p indicate how much patch X_p (highlighted in red) attend to all other patches.

Appendix B

Source Code

The *Two-dimensional distance based self-attention regularization method*, the *Style similarity based self-attention regularization method* and the ARViT architecture were implemented using *Python* as a programming language, the `PyTorch`[1] deep learning library, and the `Fast.ai`[2] deep learning library. Though the codes for both self-attention regularization methods were developed from scratch, the implementation of ARViT architecture was made by modifying the ViT codes in `pytorch-image-models`[3] (a collection of Image Models implemented in Pytorch).

The source code is available at *https://github.com/mormille/self-attention-regularization.git*. The documentation of the source code is not provided at the moment, but the author will make it available in a near future. Furthermore, this code available in the aforementioned url is intended to be used only for academic research and teaching purposes.

[1] https://pytorch.org/
[2] https://www.fast.ai/
[3] https://github.com/rwightman/pytorch-image-models

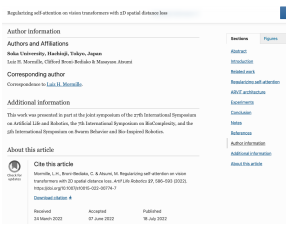
Appendix C

Paper I

Regularizing self-attention on vision transformers with 2D spatial distance loss

Luiz H. Mormille, Cliford Broni-Bediako, Masayasu Atsumi

Published in *Artificial Life and Robotics* (AROB), 27, no. 3,
August 2022, pages 586–593.
DOI: 10.1007/S10015-022-00774-7.



Paper Abstract

Recently, the vision transformer (ViT) achieved remarkable results on computer vision-related tasks. However, ViT lacks the inductive biases present on CNNs, such as locality and translation equivariance. Overcoming this deficiency usually comes at high cost, with networks with hundreds of millions of parameters, trained over extensive training routines and on large-scale datasets. Although one common alternative to mitigate this limitation involves combining self-attention layers with convolution layers, thus introducing some of the inductive biases from CNNs, large volumes of data are still necessary to attain state-of-the-art performance on benchmark classification tasks. To tackle the vision transformer's lack of inductive biases without increasing the model's capacity or requiring large volumes of training data, we propose a self-attention regularization mechanism based on two-dimensional distance information on an image with a new loss function, denoted Distance Loss, formulated specifically for the transformer encoder. Furthermore, we propose ARViT, an architecture marginally smaller than state-of-the-art vision transformers, in which the self-attention regularization method is deployed. Experimental results indicate that the ARViT, pre-trained with a self-supervised pretext-task on the ILSVRC-2012 ImageNet dataset, outperforms a similar capacity Vision Transformer by large margins on all tasks (up to 24%). When comparing with large-scale self-supervised vision transformers, ARViT also outperforms the SiT (Atito et al. in SiT: self-supervised vision transformer, 2021), but still underperforms when compared to MoCo (Chen et al. in: 2021 IEEE/CVF international conference on computer vision (ICCV), Montreal, 2020) and DINO (Caron et al. in: 2021 IEEE/CVF international conference on 354 computer vision (ICCV), Montreal, 2021).

Appendix D

Paper II

Introducing inductive bias on Vision Transformers through Gram matrix similarity based regularization

Luiz H. Mormille, Cliford Broni-Bediako, Masayasu Atsumi

In *Artificial Life and Robotics* (AROB)
Accepted on November 30, 2022
DOI: 10.1007/S10015-022-00845-9.

Paper Abstract

In recent years, the Transformer achieved remarkable results in computer vision related tasks, matching, or even surpassing those of convolutional neural networks (CNN). However, unlike CNNs, those vision transformers lack strong inductive biases and, to achieve state-of-the-art results, rely on large architectures and extensive pre-training on tens of millions of images. Approaches like combining convolution layers or adapting the vision transformer architecture managed to mitigate this limitation, however, large volumes of data are still demanded to attain state-of-the-art performance. Therefore, introducing the appropriate inductive biases to vision transformers can lead to better convergence and generalization on settings with fewer training data. To that end, we propose a self-attention regularization method based on the similarity between different image regions. At its core is the Attention Loss, a new loss function devised to penalize self-attention computation between image patches based on the similarity between gram matrices, leading to better convergence and generalization, especially on models pre-trained on mid-size datasets. We deploy the method on ARViT, a small capacity vision transformer and, after pre-training with a self-supervised pretext-task on the ILSVRC-2012 ImageNet dataset, our self-attention regularization method improved ARViT's performance by up to 13on benchmark classification tasks and achieved competitive.