# Attention Mechanisms for Image Classification

Deep Learning Course Project Report

Netanel Madmoni & Gil Ben-Or

July 25, 2023

**Abstract**

The attention mechanism is used in various field of deep learning to adjust the weights of certain parts of an input based on their importance for the task at hand. The usage of the attention mechanism in the field of computer vision has gained popularity in recent years, and attention modules are being used - either on their own or in conjunction with a convolutional neural network - for various computer vision tasks, such as image classification and object detection.

We propose a spatial attention module based on the architecture of an autoencoder, which compresses an input to a small spatial dimension and then attempts to restore it. The idea is to re-purpose the autoencoder architecture to keep certain parts of an image that are important for the image classification task. We also explore the combination of various attention modules in order to enhance their performance in the classification task.

By using only spatial attention, the CNN that had our autoencoder-inspired attention block achieved similar results on the CIFAR10 dataset to the netwotks that incorporated channel-based attention in them. It also achieved similar results to a deeper CNN while having about 17% less parameters. We believe that while not showing significantly better results, our work demonstrate the potential of our novel attention block. The code for our work can be found in the GitHub repository.

## 1 Introduction

### 1.1 The Attention Mechanism

The concept of *attention*, first introduced in [1] and later popularized by the Transformer architecture introduced in [2], is a core mechanism in state-of-the-art models for natural language processing (NLP) tasks. In recent years, more and more researchers in the field of computer vision are trying to incorporate this mechanism into their models, either in conjunction with a convolutional neural network (CNN) such as ResNet [3, 4], or on its own [5, 6].

The mechanism can be viewed as a dynamic selection procedure. This procedure is achieved by adaptively adjusting the weights of features based on the significance of the given input [7]. The purpose of the mechanism is to make a model 'focus' on parts of the input, which are the most informative to the task at hand, in order to enhance its performance.

In its simplest form, the general principle of attention consists of extracting weights (or 'scores') from an input, which are then multiplied by the input itself:

$$\mathbf{X}' = \mathbf{X} \circ f(\mathbf{X})$$

where $\mathbf{X}$ is the input to the module, $f$ is some function that returns weights for each part of the input, and $\circ$ is an element-wise product.

This makes certain parts of the input (hopefully the important ones for the task at hand) to have a more prominent representation before being processed by later parts of the model.

### 1.2 Some Existing Architectures

The idea of integrating attention into a CNN lead to several interesting architectural units ('blocks' or 'modules') that are designed to enhance the representational power of a CNN. Guo et al. [7] identified four main categories of attention in computer vision: *channel attention*: generate attention mask across channels and use it to select important channels (the 'what'); *spatial attention*: generate attention mask across spatial domains and use it to select important spatial regions (the 'where'); *temporal attention*: generate attention mask in time and use it to select important frames (the 'when'); *branch attention*: generate attention mask across branches and use it to select important parts from different paths of the model (the 'how').

In addition to these four categories, two additional hybrid categories are being used: (1) channel + spatial attention (2) temporal + spatial attention. In these architectures, different types of attention modules are applied to the input, either concurrently or sequentially.

In this work we focus on channel and spatial attention. We explain below two of the existing key architectures we looked at: the Convolutional Block Attention Module (CBAM) and the Efficient Channel Attention (ECA) Module.

*Convolutional Block Attention Module (CBAM)* - Proposed by Woo et al. [8], the module infers attention maps along the channel and spatial dimensions. The attention maps are then applied sequentially to the input for adaptive feature refinement. See Figure 1.

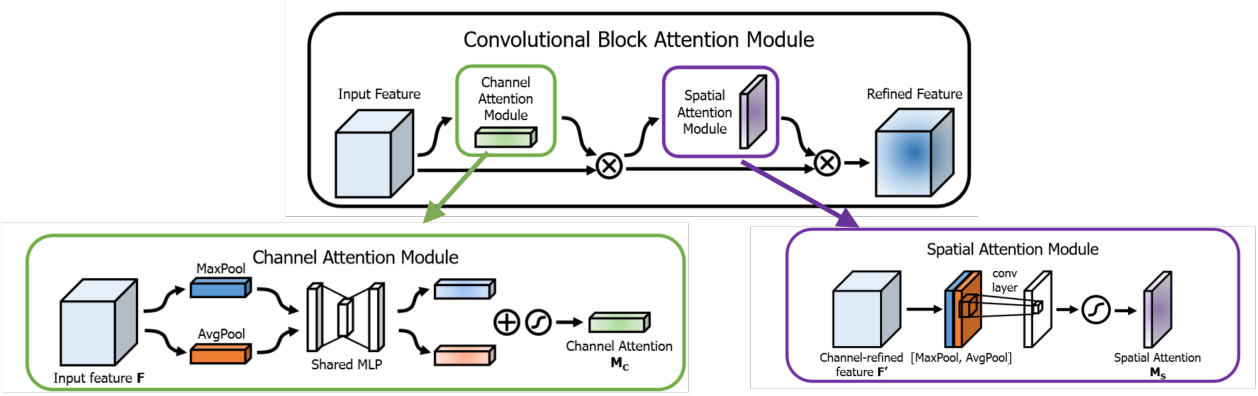


**Figure 1:** Diagram of the Convolutional Block Attention module (Adapted from [8]). For the channel attention map, average and max pooling operations are performed over the width and height of the input. the outputs of these operations are then being fed into a multilayer perceptron with one hidden layer. Finally they are added together and go through a sigmoid activation function. For the spatial attention map, the same pooling operations are being performed but over the channel space. The aggregated maps go through a $7 \times 7$ convolution and finally through the sigmoid activation function.

*Efficient Channel Attention (ECA) Module* - Proposed by Wang et al. [9], this module generates a channel attention map in an efficient manner, adding only a few parameters to the backbone architecture. See Figure 2

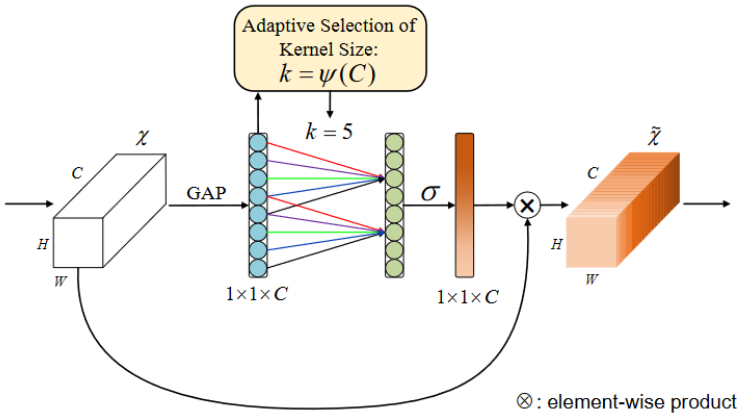


**Figure 2:** Diagram of the Efficient Channel Attention Module (Adapted from [9]). The input feature map is going through a global average pooling operation. ECA then generates channel weights by performing a 1D convolution. Kernel size is adaptively determined via a mapping of channel dimension: $k = \psi(C)$. The output of this convolution is going through the sigmoid activation function.

## 2 Proposed Work

In this work, we focus on the development and implementation of a novel attention mechanism and hybrid attention mechanisms. Our novel attention mechanism is inspired by and based on the architecture of autoencoders for capturing spatial relationships, while our hybrid attention mechanisms incorporate different existing models utilizing both spatial and channel relationships.

## 2.1 Autoencoder-Inspired Attention Block

Autoencoders were introduced by Hinton & Salakhutdinov in [10]. They are considered a generalization of principle component analysis, and are composed of two parts: an encoder module, which reduces the dimension of the input into a low-dimensional encoded version, and a decoder module, which recovers the data from the encoded version, reconstructing the input while trying to retain its most informative features.

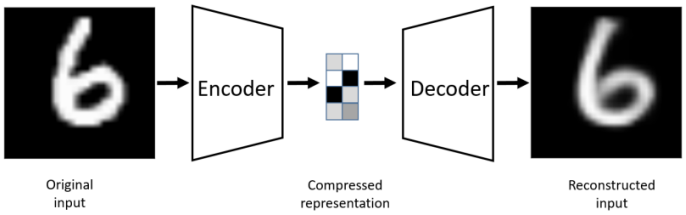


**Figure 3:** Example of an autoencoder architecture (Adapted from [11]). The input is compressed into a small representation by the encoder, and then reconstructed to its original size by the decoder.

In this work, We develop an autoencoder-inspired attention block and explore its effectiveness when incorporated into a CNN. The intuition for this is that an autoencoder-like architecture might be able to effectively extract the most 'important' features of an input feature map during the encoding-decoding process, which can be used to 'tell' the model in which areas of the feature map to focus for the classification task. Our block essentially re-purposes the autoencoder architecture to keep certain parts of an image that are important for the image classification task, rather than reconstructing an image perfectly.

We explore the usage of two types of autoencoder architectures: a convolutional autoencoder and a linear autoencoder [12]. While both having the same general encoder-decoder structure, the convolutional autoencoder uses convolutions and transposed convolutions (also known as deconvolutions) for changing the dimensionality of the input, while a linear autoencoder does so with linear, fully connected layers. The structure of our version of the convolutional-autoencoder-based attention block is presented in Figure 4.

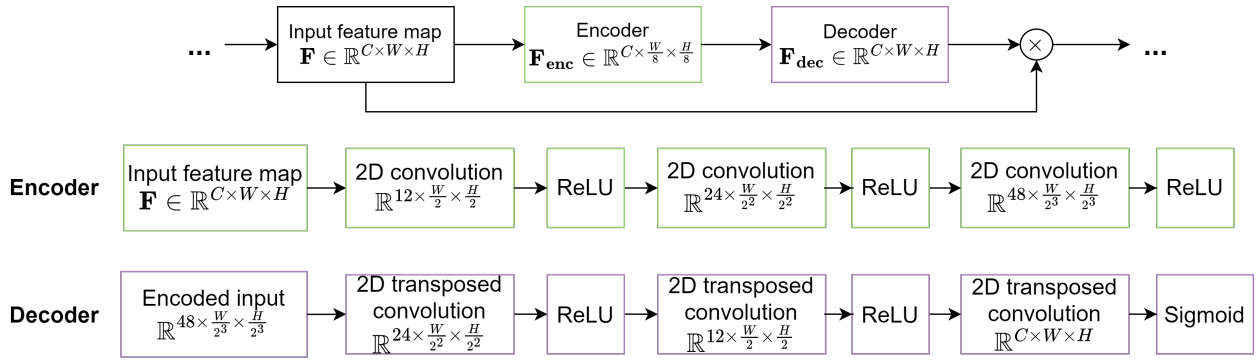


**Figure 4:** Architecture of our convolutional-autoencoder-inspired attention module. The encoder is a sequence of convolutional layers, each followed by a ReLU activation function. The convolutional layers progressively increase the number of channels while simultaneously reducing the spatial dimensions of the input. This process allows the encoder to capture the spatial relationships within the input, focusing on 'where' the informative parts are. The decoder is responsible for reconstructing the original input from the encoded feature space. Using transposed convolutional layers, it progressively decreases the number of channels, while simultaneously increasing the spatial dimensions of the feature maps. the sigmoid function outputs a value between 0 and 1, which can be interpreted as the importance or weight of each feature in the input. The output of the autoencoder is then element-wise multiplied by the input - applying an attention map that emphasizes the important features and suppresses the less important ones. In the linear version of the autoencoder, all convolution and transposed convolutions are replaced with linear blocks.

## 2.2 Hybrid models

We also propose the usage of hybrid models that combine different attention mechanisms. These hybrid models are designed to leverage the unique advantages of each individual mechanism, potentially improving the overall performance of the model.

We explore two types of hybrid attention mechanisms: (1) ECA channel attention + CBAM Spatial Attention (2) ECA channel attention + autoencoder spatial attention. Similarly to other existing hybrid attention models such as CBAM, in both of these models we aim that the ECA mechanism will focus on 'what' content is present

in the input, while the spatial component - either our new autoencoder-based attention module or CBAM spatial attention module - will focus on 'where' the informative parts are. By combining these two mechanisms, we aim to create a more comprehensive attention mechanism that considers both 'what' and 'where' aspects of the input.

# 3 Experiments & Results

To test our new modules, we used a simple CNN network with two hidden layers as a backbone model, in which the modules will be incorporated. A diagram of the model is presented in Figure 5. for the data, we used the CIFAR10 dataset [13].

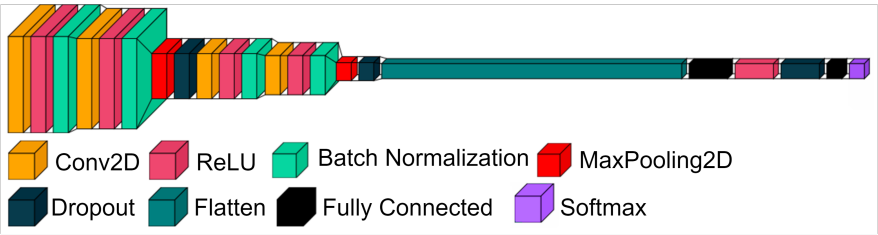


**Figure 5:** A schematic representation of the CNN used as a backbone for testing our modules. The network consists of two convolutional layers, followed by a fully connected layer and a softmax layer for output. Each convolutional layer is a sequence of operations including 2D convolution, ReLU activation, batch normalization, max pooling, and dropout.

Every attention module was inserted immediately after each convolutional layer. The intuition behind this is that the attention will work on the 'raw' feature maps generated by the input. Although for future research we recommend testing several other locations in the network. For the hybrid models, the channel attention module was inserted before the spatial attention module, as it was shown to yield better results in [8].

In order to control randomness as much as possible in our experiments, we decided to train each model multiple times ('runs'). We then averaged the results for a more robust estimation of model performance. All training hyperparameters were identical across models as well as across runs. Models were trained for 10 epochs $\times$ 16 runs, with a learning rate of 0.1. In each epoch, the models were trained on a training set and evaluated on a validation set (validation size 5,000). At the end of each run, we selected the model who performed best on the validation set for each architecture (16 models for each architecture). We than averaged these models' performance on the test set.

For comparison purposes, we also included a deeper CNN model, which is similar to the one described above, but with three hidden layers rather than two.

All of our code was implemented in PyTorch. For the CBAM and ECA modules, we used the code from their official GitHub repositories [1].

Experiments results are presented in Table 1.

| Model | Test Set Accuracy | | Number of parameters (M) |
|---|---|---|---|
| | mean | std | |
| CNN | .667 | .018 | **2.62** |
| CNN + ECA | .667 | .023 | 2.62 |
| CNN + CBAM | .642 | .026 | 2.63 |
| CNN with three hidden layers | .665 | .026 | 3.51 |
| CNN + Conv. Autoencoder | **.675** | .020 | 2.92 |
| CNN + Conv. Autoencoder + ECA | .665 | .018 | 2.92 |
| CNN + Linear Autoencoder | .602 | .026 | 2.72 |
| CNN + Linear Autoencoder + ECA | .607 | .032 | 2.72 |
| CNN + CBAM Spatial Attention + ECA | .669 | .014 | 2.63 |

**Table 1:** Results of experiments

Without using any channel-based attention, the model with the autoencoder-inspired blocks achieved similar results to the models that had channel-based attention modules in them. It also performed as well as a deeper

[1] CBAM: CBAM official repository by Jongchan Park on GitHub; ECA: The official PyTorch code for ECANet by Banggu Wu on GitHub.

CNN model, while having only around 83% of the number of its parameters. As for the hybrid models, however, we did not see an improvement over the other models.

While the convolutional-autoencoder-based attention block yielded relatively good results, the case was different for the linear-autoencoder-based attention block, which performed even worse than the base CNN model. We believe that it is because this architecture does not suffice for capturing 'important' regions in images, due to its linear nature.

# 4 Conclusions & Future Research

In this project, we explored various attention mechanisms for image classification tasks. We proposed and implemented a novel autoencoder-inspired spatial attention module, as well as hybrid attention modules combining channel and spatial attention.

Our experiments show that the convolutional autoencoder attention module, which uses only spatial attention, can achieve comparable results to other networks that use channel-based attention modules, while also achieving similar results to a deeper backbone CNN by using fewer parameters. The hybrid attention mechanisms resulted did not show an improvement compared to the performance of their individual components. In this case, further investigation is needed. Overall, we believe that our results demonstrate the potential of autoencoder-inspired spatial attention to improve image classification performance in a relatively efficient manner.

For future research, we suggest investigating and comparing different configurations of the autoencoder attention module, such as number of layers, layer sizes, and activation functions, which could improve the performance of the network. Locating the block in different parts of the network should also be considered for testing, as well as evaluating the attention mechanisms on additional, more complex CNN backbones and larger datasets, since their effectiveness may vary across architectures. It might also be worth testing our module for a different computer vision task, such as object detection.

We believe our work provides a solid base framework for continued research for exploring the importance of attention in the context of image classification, and could help develop a deeper understanding of how to best leverage attention for computer vision tasks.

# References


[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.

[4] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," 2018.

[5] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," 2019.

[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," 2021.

[7] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, vol. 8, pp. 331–368, mar 2022.

[8] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," 2018.

[9] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," 2020.

[10] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[11] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2021.

[12] Y. Zhang, "A better autoencoder for image: Convolutional autoencoder," in *ICONIP17-DCEC. Available online: http://users. cecs. anu. edu. au/Tom. Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58. pdf (accessed on 23 March 2017)*, 2018.

[13] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." http://www.cs.toronto.edu/ kriz/cifar.html.