Project Report

Daniel Carrascoza
Electrical and Computer Engineering

Arnav Kamra
Computer Science and Engineering

Abstract

Our project aims to implement a method of Image Classification using ConvNet with self-attention/ Vision Transformers that will allow us to accurately identify cars. Our goal with this is to create a model that can be used in parking structures to survey and record cars that may be a part of crime. Our method aims to be cost efficient and accurate in comparison to other methods of surveillance.

1 Introduction

We aim to address the issue of not having proper image detection of cars going in and out of parking structures. This does not allow for a clear database of potential stolen vehicles being stored in parking structures or cars a part of criminal activity.

- The motivation of solving this problem: The motivation to solve this problem is due to risk imposed for people when parking in these structures. Unfortunately, 10 percent of property related crimes and 7.3 percent of violent crimes occur in parking areas.
- Due to costs, it would be hard to have surveillance across the entirety of the parking structure. However, by implementing a better system of tracking cars that enter and exit parking structures, we can try and make these areas safer.

2 Related Work

Vision Transformers branch out from traditional Convolutional Neural Networks. These neural networks apply the transformer architecture which allows the net to take information specific to certain areas of the spatial feature map across multiple convolutional layers.

Non-local Neural Networks Formula: $y_i = 1 + C(x) \sum_j f(x_i, x_j) g(x_j)$ [3]. In this formula, we can see the idea of Transformers in a non-local neural network. Our x_i value represents a specific location and j_i values refers to all locations where our feature may be. These locations are computed with a pairwise function, f. The unary function g computes a representation of g at the position j and the entire function is normalized by a factor of C.

This is the basis of a non-local neural network and the structure of this can be seen in Vision Transformers. When looking at the block structure of a non local neural network:

In figure 1, we see the block split the input into 3 parts which each carry out specific functions that are then combined in the end. This type of structure is common with non-local neural networks and is seen also in more straight-forward vision transformers.

Self Attention for Image Classification
For example, the model seen here:

Follows similar principle to the Non-local Neural Network. The input after a convolutional layer is split into two streams.
Stream 1 computes $\alpha(x_i, x_j) = \gamma(\delta(x_i, x_j))$ and Stream 2 computes $\beta(xj)$. The final output of the block is then given by $y_i = \sum_{j \in R(i)} \alpha(x_i, x_j) \odot \beta(x_j)$. [1]

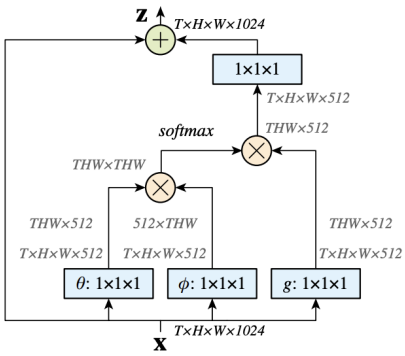

Figure 1: Non-local Neural Network block

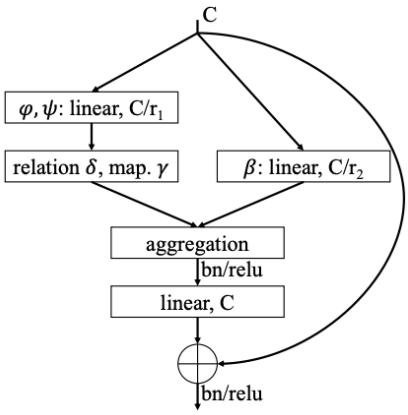

Figure 2: Self Attention block

Similar to the non-local neural network block, functions are split to calculate locations of features in the spatial map. The α calculation is are the attention weights. The δ refers to the relational function which we use as the Hadamard function. This particular block is simple, which allows it to be integrated in common convolutional networks like the ResNet models.

3 Method

The model we chose stems from the paper "Image Classification using ConvNet with self-attention/ Vision Transformers" by Alexey Dosovitskiy.[1]. Here is an image of the architecture:

Our model consists of 5 transitional layers and 4 Self Attention blocks. Each transitional layer expands the inputs dimension, while lowering the height and width dimensions. Inside these transitional layers are a 1x1 convolution with an augmentation of the output channels and a max-pool function. The self attention blocks follow the same structure as the part above: it splits the input into two streams and runs a pairwise function on one of them to have a spatial footing location in the feature map. It then aggregates them together with the Hadamard function and then outputted. After all transition blocks and SA blocks have been completed, We apply a global average pool to the number of classes we are looking for.

This technique strays from previous convolutional neural networks because of its ability to limit the weight sharing to specific parts of the feature map instead of sharing its weight accords the entire feature map. Through our pair-wise calculations, we are able to calculate the important regions of are image and apply different weights more effectively than a traditional convolutional net.

We used a training function that is commonly used for convolutional networks and were able to compare the differences between a ResNet model and our Self Attention model. Used the Adam

Layers	Output Size	SAN10	SAN15	SAN19
Input	224×224×3	64-d linear		
Transition	112×112×64	2×2, stride 2 max pool → 64-d linear		
SA Block	112×112×64	[3×3, 16-d sa; 64-d linear] ×2	[3×3, 16-d sa; 64-d linear] ×3	[3×3, 16-d sa; 64-d linear] ×3
Transition	56×56×256	2×2, stride 2 max pool → 256-d linear		
SA Block	56×56×256	[7×7, 64-d sa; 256-d linear] ×1	[7×7, 64-d sa; 256-d linear] ×2	[7×7, 64-d sa; 256-d linear] ×3
Transition	28×28×512	2×2, stride 2 max pool → 512-d linear		
SA Block	28×28×512	[7×7, 128-d sa; 512-d linear] ×2	[7×7, 128-d sa; 512-d linear] ×3	[7×7, 128-d sa; 512-d linear] ×4
Transition	14×14×1024	2×2, stride 2 max pool → 1024-d linear		
SA Block	14×14×1024	[7×7, 256-d sa; 1024-d linear] ×4	[7×7, 256-d sa; 1024-d linear] ×5	[7×7, 256-d sa; 1024-d linear] ×6
Transition	7×7×2048	2×2, stride 2 max pool → 2048-d linear		
SA Block	7×7×2048	[7×7, 512-d sa; 2048-d linear] ×1	[7×7, 512-d sa; 2048-d linear] ×2	[7×7, 512-d sa; 2048-d linear] ×3
Classification	1×1×1000	global average pool → 1000-d linear → softmax		

Figure 3: Architecture of Model

optimizer with a learning rate of 1e-3 and paramaters that matched our model architecture. For Testing, we had a seperate dataset that allowed up to test on data the model had not seen before.

For training/optimizing the convolutional networks, we referenced text by the paper "Exploring Self-attention for Image Recognition" by Hengshuang Zhao.[2]. The article explores variations of the self-attention operator as the basic building block for image recognition models. It discusses two types of self-attention: pairwise self-attention and patchwise self-attention. Pairwise Self-Attention is a set operator rather than a sequence operator like convolution. It does not attach stationary weights to specific locations and is invariant to permutation and cardinality. The article presents variants of pairwise attention that have greater expressive power while retaining these invariance properties. Weight computation does not collapse the channel dimension, allowing feature aggregation to adapt to each channel. Patchwise Self-Attention on the other hand, similar to convolution, can uniquely identify specific locations within their footprint. They lack permutation or cardinality invariance but are more powerful than convolution. The experiments in the article show that both forms of self-attention are effective for building image recognition models. Pairwise self-attention networks match or outperform convolutional ResNet models with similar or lower parameter and computational budgets. The patchwise attention model even surpasses larger models like ResNet50 in accuracy while having significantly lower parameter and computational requirements. The article also suggests that self-attention networks may offer benefits overall, as indicated by their experiments. Hence for our model, we attempted to adopt the framework of the pairwise attention model (by implementing self-attention (transition) layers) to try and outperform the convolutional baseline from models like the ResNet50.

4 Experiments

The datasets we chose were the Stanford Cars Dataset [4] and its accompanying Stanford Cars Metadata Dataset. The dataset contains 3 major components:

- Annotation File: The annotation file is a .mat file containing the annotations and labels used for classifications of all images in the dataset.
- Cars Train Folder: The train folder contains 8144 images of 196 different car models.
- Cars Test Folder: The test folder contains 8041 images of 196 different car models.

The dataset consisted of 16,185 images and 196 classes. The images were RGB, however, were not uniform in dimensions. Our dataset pre-processing consisted of Transforming our data to 228x228 images (with a channel of 3) and creating a dictionary of the image values that are mapped to their respective index in the folder.

Process / Results
We started with a Self Attention with 2 SA blocks and transition layers to view our performance on the dataset. After the initial accuracy results from this, we moved onto our full Self Attention model.

This model had 5 SA blocks and 5 transitional layers. While training these models, it was evident that the training process required an immense amount of iterations and a high-end graphics card to speed up the training process. Although we did not fully train the model, we compared the results of the start of its training with a ResNet model on the same dataset and found that our Self Attention model preformed slightly better. However, our Self Attention model was very computationally expensive and has very difficult to train and modify parameters.

Room for Improvement
One aspect for improvement of our image classification project was the inherent complexity of the model. We aimed to experiment with a more intricate model to compare its performance against a simple ResNet model. However, we encountered limitations and challenges that impacted the accuracy and implementation of our model (related to cycle sizes and the training of weights). Due to computational constraints and the complexity of the model, longer cycles on a strong GPU were necessary to achieve more accurate results. However, this posed challenges in terms of resource allocation and training efficiency.

Additionally, we faced difficulties in debugging and optimizing the functions inside the Self-Attention Block. The complexity of the model made it challenging to identify and resolve issues efficiently, leading to suboptimal performance and slower progress in model development.Overall, the complexity of implementing a sophisticated model with self-attention or vision transformers posed significant challenges. Despite our efforts to experiment with advanced architectures, we were unable to iron out all the kinks and achieve the desired level of accuracy and robustness in image classification compared to the simpler ResNet model.

5 Supplementary Material

https://drive.google.com/file/d/1eEE-V1k5gf7ZXXgQRthZfv4zSSnCDLyx/view?usp=sharing

References

[1] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv:2010.11929, 2021.

[2] Hengshuang Zhao et al. *Exploring Self-attention for Image Recognition*. arXiv:2004.13621, 2020.

[3] Xiaolong Wang et al. *Non-local Neural Networks*. arXiv:1711.07971, 2018.

[4] JESSICA_LI. *Stanford Cars Dataset*. https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset, 2018.