Giorgos Dimopoulos

# From Idea to Implementation

## Introduction

The main goal of this task is to demonstrate the process of designing and running an experiment that compares different SOTA computer vision models and to discuss the advantages and disadvantages of using one architecture over the other.

The experiment aims to not only compare the performance of the models but also explain how each solution is expected to improve the solution of the problem at hand.

## Experiment Overview

The experiment will use the popular dataset MNIST [1] for training and evaluation. Spatial Transformer Networks (STN) [2] will be used to create the baseline model. Next, the STN network will be enhanced by replacing the traditional convolution layers in the STN architecture with CoordConv layers [3] in order to improve the performance of the baseline.

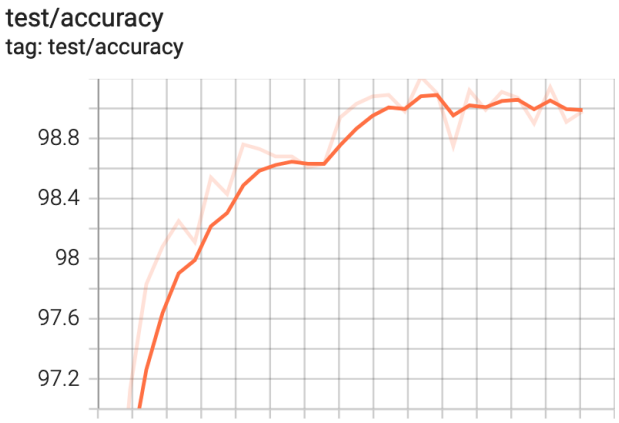
In the following step of the experiment the baseline and its improved version will be compared against two different SOTA solutions. The first solution will be using Group Equivariant Convolutional Networks (G-CNN) [3] and the second Vision Transformers (ViT) [4].

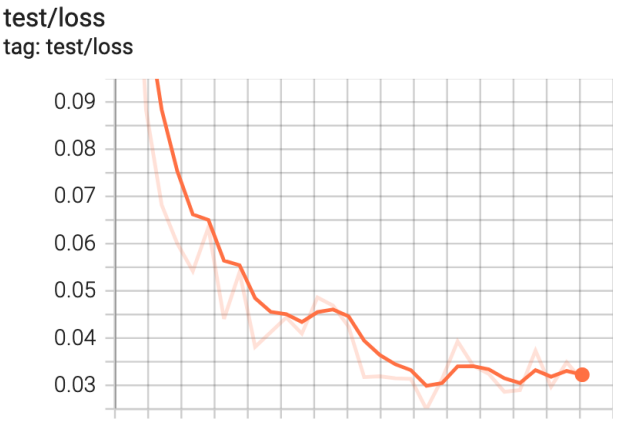
In all experiments early stopping at 30 epochs will be applied to avoid overfitting since accuracy improvements after that point are marginal in most cases. Additionally, the same configuration in terms of batch size, optimizer, loss function, etc. will be used everywhere to ensure that the results are comparable afterwards.

## 1. Spatial Transformer Networks (baseline)

### Experiment details

- **Epochs**: 30
- **Batch size**: 64
- **Optimizer**: SGD
- **Learning rate**: 0.01
- **Momentum**: 0.5
- **Loss function**: Negative log-likelihood
- **Accuracy**: 98.98%




### Discussion

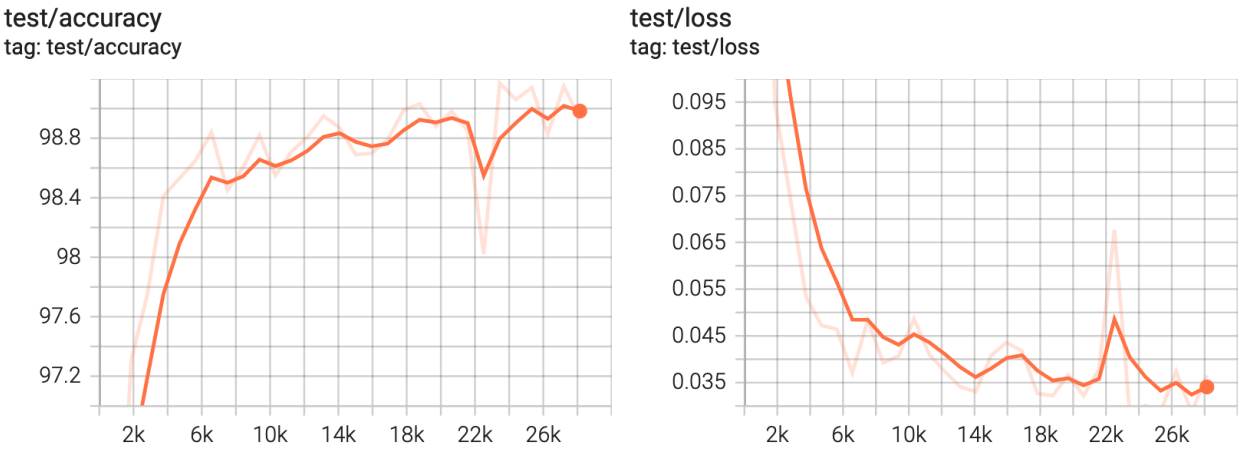
STNs were designed to address the spatial invariance of traditional CNN architectures. In the context of image classification spatial invariance refers to the invariance of the model towards spatial transformations such as transformation, rotation and scaling. Therefore, invariance is the model's ability to identify features from images even when they are transformed, roasted or scaled. Spatial Transformers can be integrated with CNNs to improve tasks such as image classification.

The training of the model quickly converged to a very high accuracy of over 98.9% from epoch 20. With an even earlier stop we could avoid the slight drop of accuracy in the last 10 epochs.

## 2. Spatial Transformer Networks with CoordConv

### Experiment details

- **Epochs**: 30
- **Batch size**: 64
- **Optimizer**: SGD
- **Learning rate**: 0.01
- **Momentum**: 0.5
- **Loss function**: Negative log-likelihood
- **Accuracy**: 98.93%




### Discussion

The replacement of convolutional layers with CoordConv didn't result in any notable improvement in performance. The main reason behind this is that the MNIST dataset contains images that are already cropped, zoomed and centered leaving very little work for the CoordConv layers since the additional coordinate channels cannot offer extra info to help improve the features extracted from an image.

## 3. Group Equivariant Convolutional Nets

### Experiment details

- **Epochs**: 30
- **Batch size**: 64
- **Optimizer**: SGD
- **Learning rate**: 0.01
- **Momentum**: 0.5
- **Loss function**: Negative log-likelihood
- **Accuracy**: 99.42%


### Discussion

G-CNNs improve on classic CNNs by exploiting the underlying translated, reflective and rotational symmetries of the inputs. In this way, G-CNNs reduce the spatial invariance of CNNs by learning powerful representations based on symmetry patterns.

After running the experiment for 30 epochs the accuracy reached 99.42% which is quite close to the SOTA. This is an expected result since the MNIST dataset contains images of digits that have in many cases symmetrical patterns. Once the model learns the existing from the existing symmetries it can easily identify variations of the same digits during the evaluation phase.

## 4. Vision Transformers

### Experiment details

- **Epochs**: 30
- **Batch size**: 64
- **Optimizer**: SGD
- **Learning rate**: 0.01
- **Momentum**: 0.5
- **Loss function**: Negative log-likelihood
- **Accuracy**: 98.7%


### Discussion

Transformers have become the de facto solution for NLP tasks but are now becoming an emerging topic in computer vision and have recently shown very promising results in areas such as image classification. Vision Transformers were the first pure transformer implementation to achieve similar or even surpass the results of CNNs in common image classification tasks. It is worth mentioning that ViT-H/14 is currently the leader in SOTA image classification with the CIFAR-10 dataset with 99.5% accuracy [6].

After 30 epochs accuracy reached 98.7%, a result that is marginally below the baseline. Again here the MNIST images might not be ideal to take advantage of the attention mechanism of the transformer.

## Conclusion

The experiment that was presented here demonstrated the process of researching new methods for a computer vision task by starting from a baseline solution and trying to improve upon that using alternative approaches from the SOTA.

MNIST is not the ideal dataset to test more novel techniques that address the spatial invariability of CNNs since it contains images with very little spatial variation. The experiments showed that using a SOTA CNN model such as G-CNNs can easily outperform the other solutions and achieve SOTA accuracy.

Transformers were not originally designed for image classification tasks, but they seem to be gaining ground fast as advancements in ViT already outperform CNNs in specific datasets.

One more observation obtained from the experiment was that the STNs gave significantly more robust models with much smaller training times when compared to the G-CNN and the Vit. This can be an important factor to take into consideration when deciding to go for a less complex model that doesn't have a huge accuracy tradeoff.

## Notes

TensorboardX [5] was used for tracking the metrics of the experiments so that the results can be easily recovered and compared. The tensorboardX logs and the test loss/accuracy graphs for each experiment are stored in their respective directory under the "tensorboard_logs" directory in the repo. The user can simply run "tensorboard --logdir tensorboard_logs/<experiment_dir>" from a terminal and then access all the metrics from a browser by going to "localhost:6006".

## References

[1] http://yann.lecun.com/exdb/mnist/

[2] https://arxiv.org/abs/1506.02025

[3] https://arxiv.org/abs/1602.07576

[4] https://arxiv.org/abs/2010.11929

[5] https://github.com/lanpa/tensorboardX

[6] https://paperswithcode.com/sota/image-classification-on-cifar-10