Deep Neural Architecture Search
Speeding up scientific simulations with neural networks

Casper Tygesen Bang-Hansen (s183827) and Mathias Jensen (s176531)
Technical University of Denmark

Introduction

Running simulations of physical phenomena is at times a prohibitively expensive and therefore slow task. Speeding these up would accelerate scientific research. One method of speeding up simulations is using machine learning. Especially neural nets show promising results [1].
Here, a comparison between Residual Networks (ResNet) [3] and Neural Architecture Search (NAS) [2] is made to investigate which network is best for the task. In judging which network performs the best both training time, accuracy, network complexity and test time are compared between the two. To achieve comparable metrics the networks are both trained and tested on the MNIST [5] and CIFAR10 [6] data sets.

Key Points

- We compare important metrics of ResNet [3] and NAS [2] to investigate which is more appropriate for simulation speed-up.
- The data sets used to generate the metrics are MNIST [5] and CIFAR10 [6]
- The metrics used to compare the networks are the training time, test time, accuracy and network complexity

Data

Two data sets are used to compare the networks. The famous MNIST and CIFAR10 data sets. The MNIST data set consists of 60,000 training images and 10,000 test images of handwritten numbers from 0-9 in the resolution 28x28. The CIFAR10 data set is made up of 60,000 32x32 color images belonging to 10 different classes. The set is split into 50,000 training and 10,000 test images.
Examples of both data sets are given below.

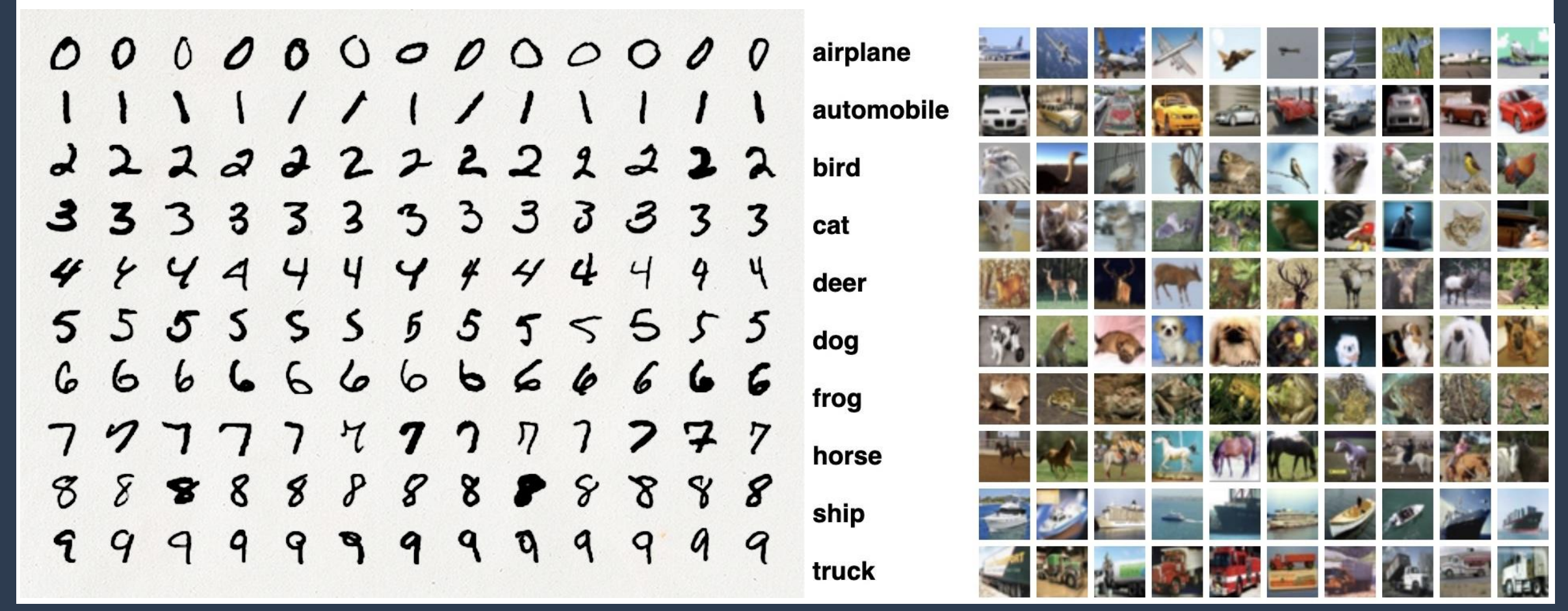
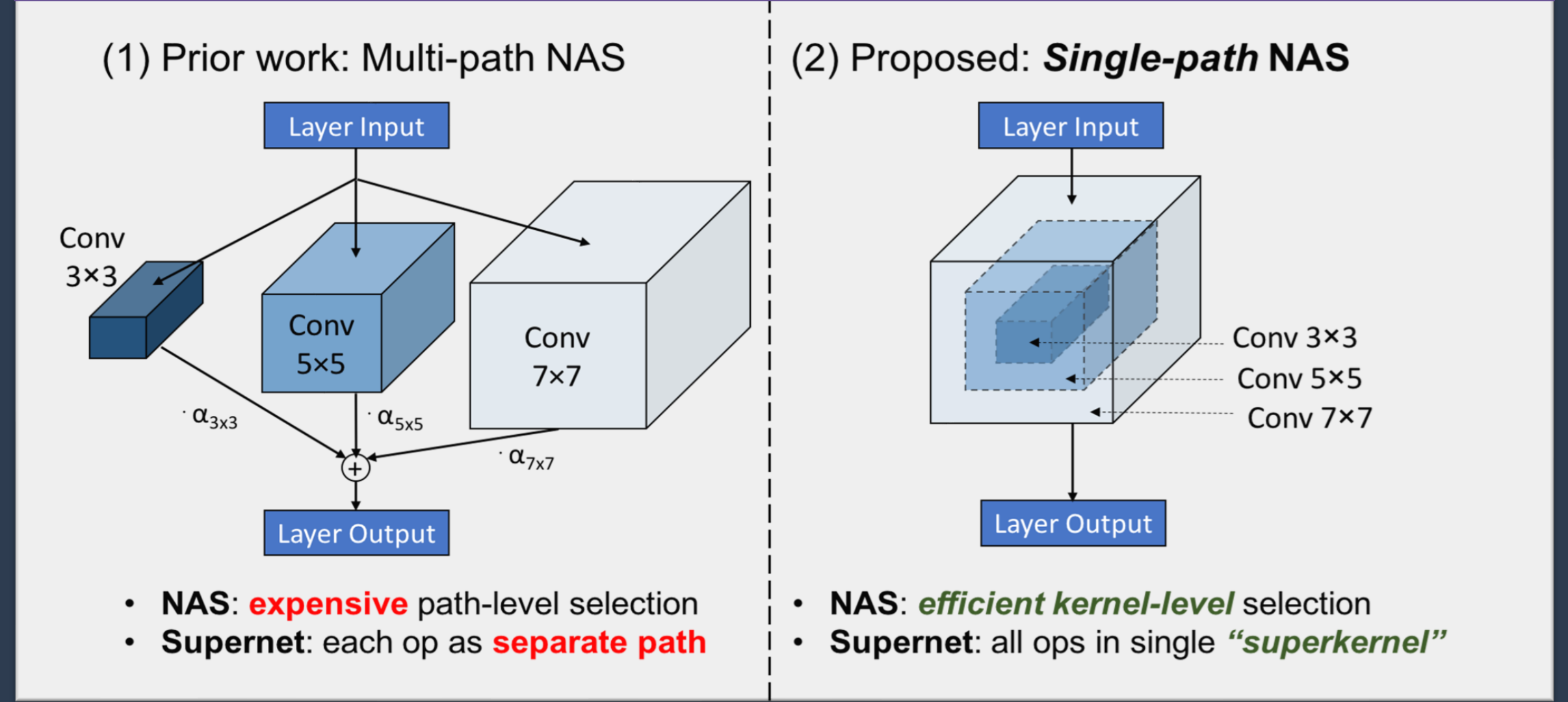


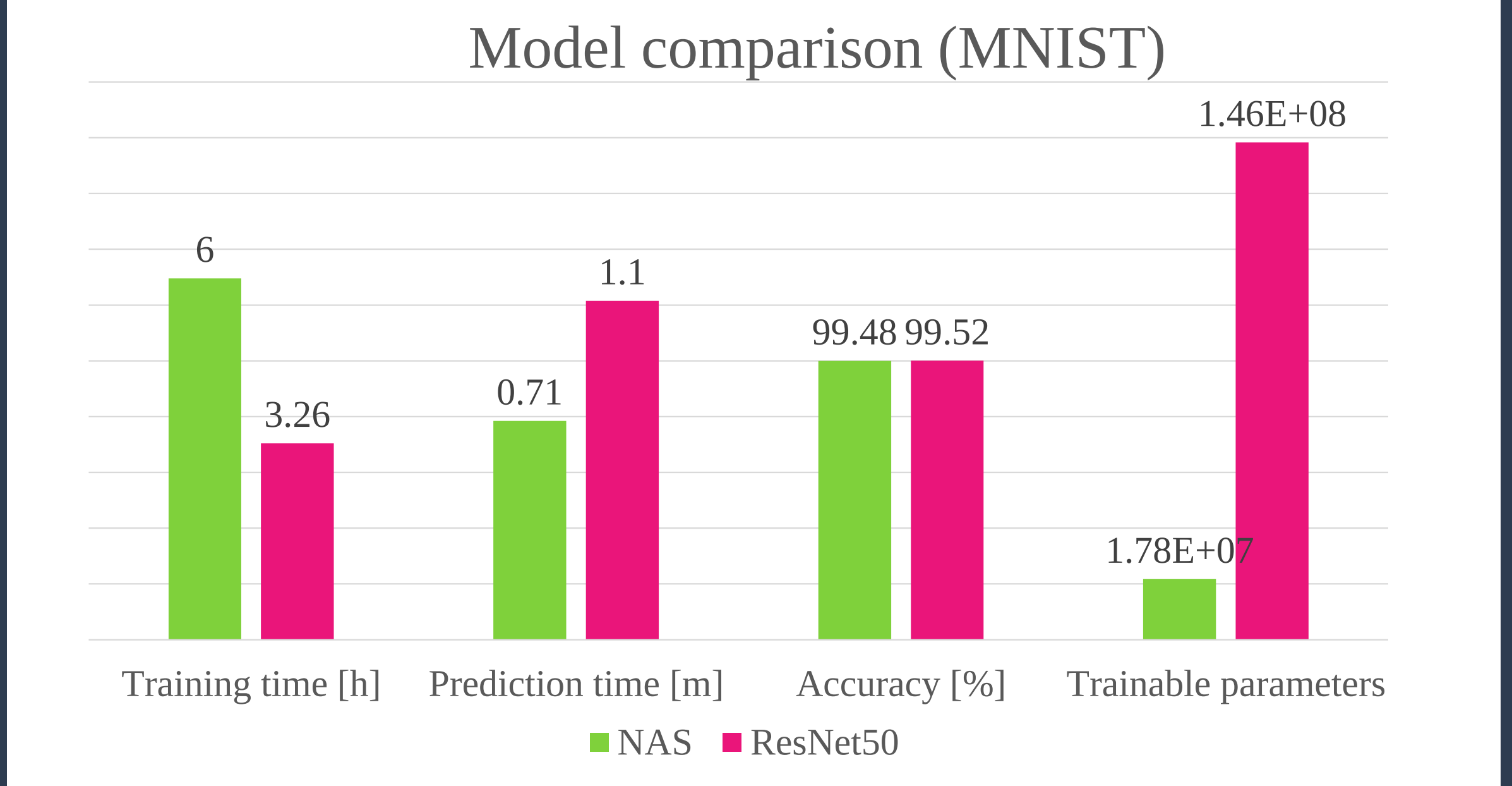
Figure A

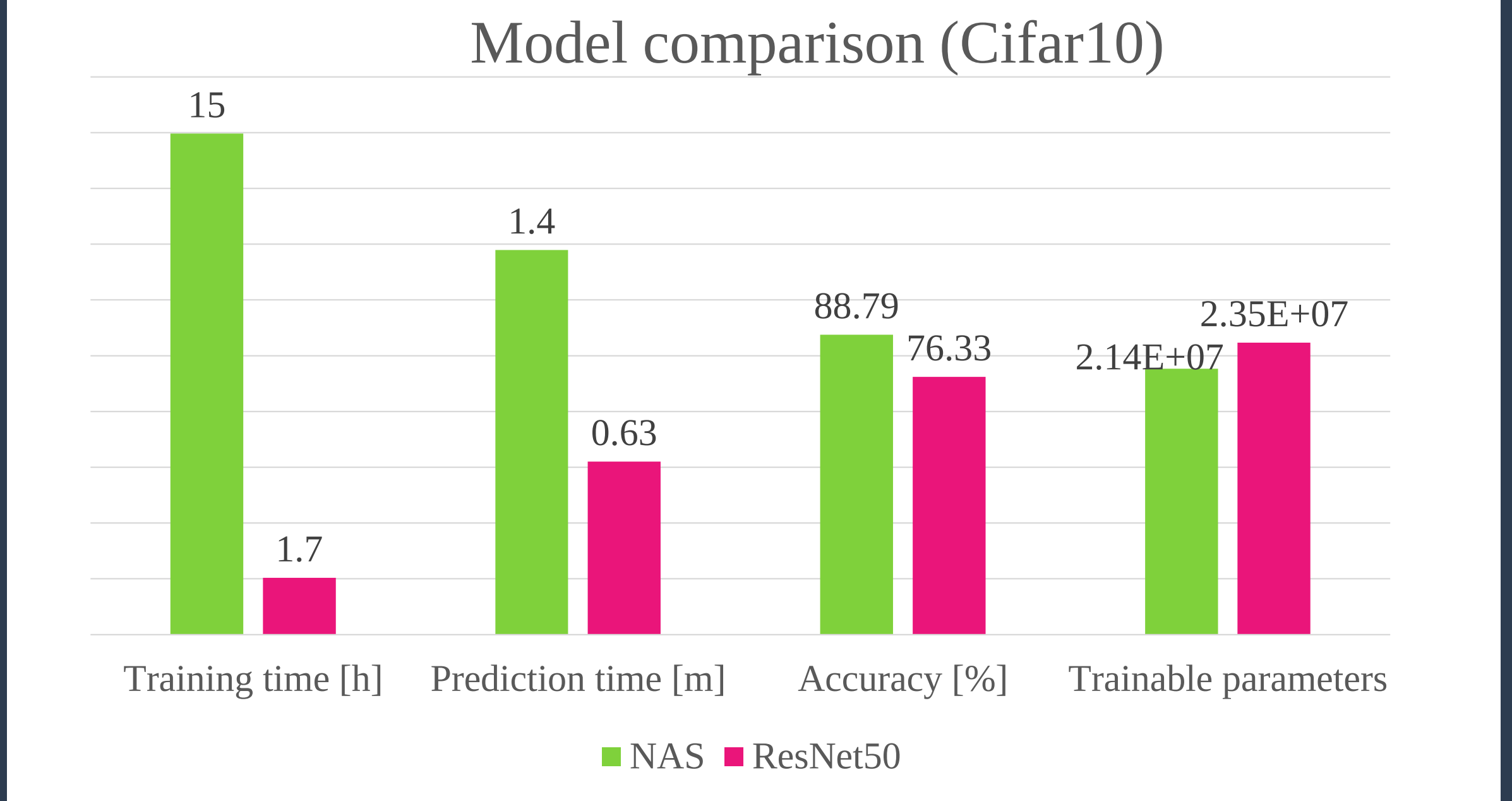

(1) Prior work: Multi-path NAS
Layer Input
Conv 3×3
Conv 5×5
Conv 7×7
α3x3
α5x5
α7x7
Layer Output
NAS: expensive path-level selection
Supernet: each op as separate path
(2) Proposed: Single-path NAS
Layer Input
Conv 3×3
Conv 5×5
Conv 7×7
Layer Output
NAS: efficient kernel-level selection
Supernet: all ops in single "superkernel"

NAS Model

The architecture is built by constructing the same kind of blocks and putting them together sequentially. These blocks are green in Figure B. Each block in the network takes the same image size (n x n) (in the figure reduced to *n*) and outputs a different dimensionality, which the next block in the chain takes as an input. In the figure s_i is the number of channels in layer *i*. *d* refers to the dimensionality.
Each block in the main chain is built by a sub-chain, which is constructed using MBConv blocks. The sub-chain can be seen in yellow in Figure (B).
The MBConv blocks are constructed by a Searchable Depthwise Convolution surrounded by 1x1 convolutions. The residual is added to output of this chain of operations. The MBConv is the blue blocks in Figure (B). The Searchable Depthwise Conv is a superkernel, which is a way of finding the optimal kernel size, expansion ratio or skip-op by training. The different kernel sizes are constructed by creating a subset of the maximum kernel size (See Figure A). Thus, a threshold for each subset can be trained to achieving the optimal kernel size.
We have constructed our model by using 7-blocks in the main chain, and 2-blocks in each sub-chain. Furthermore, the maximum kernel size in the superkernel is 9x9 and the maximum expansion ratio is 9.
Equation (1) shows the iterative approach to finding the optimal kernel size in each layer.

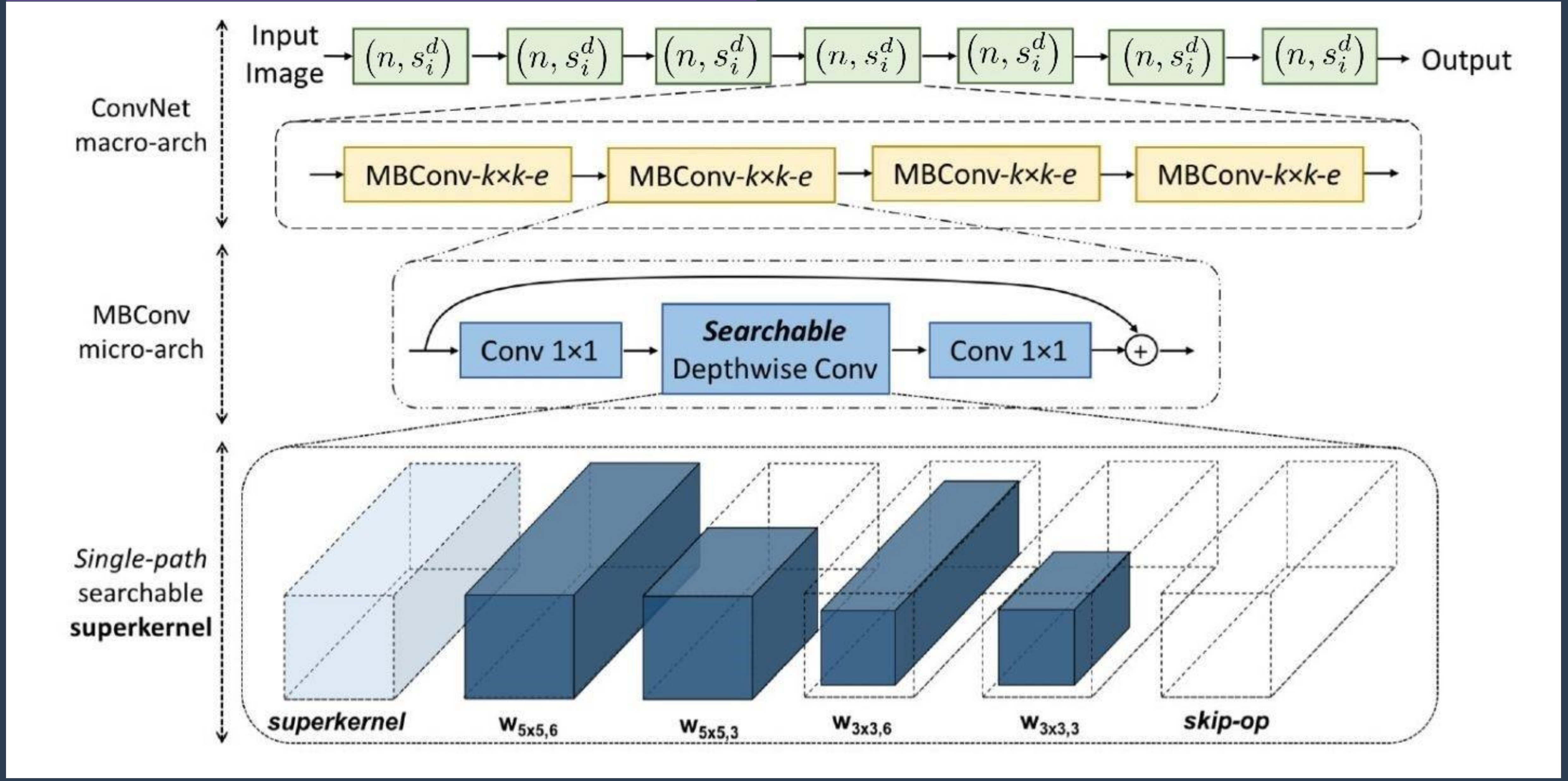
Tables and Equations (Figure C & D)





$$\mathbf{w}_k(u) = \begin{cases} \mathbf{w}_{n_k} + \mathbb{1}\left(\|\mathbf{w}_{n_{u+1}\setminus n_u}\|_2^2 > t_{k=u}\right) f(u+1) & u \leq N \\ 1 & \text{Otherwise} \end{cases} \tag{1}$$

Figure B



Results

The goal of this project is to compare ResNet with our NAS model. The results between the two architectures can be found on Figure C & D. The results show the comparison between NAS and ResNet50 for two datasets. The first data-set is MNIST, where the results show the accuracy for the two models are similar, but the training time of the NAS is much larger than the training time of ResNet50. Furthermore, the trainable parameters for the ResNet50 is much higher than NAS. The results for the MNIST dataset can be found on Figure C.
The second data-set is Cifar10, the results for Cifar10 can be found on Figure D. The results show the accuracy for NAS is much higher than for ResNet50 (88.79%) compared to 76.33%), but the training and prediction time is also much longer for NAS compared to ResNet50. Furthermore, the trainable parameters are roughly the same, thus the complexity of the architectures are similar.
Training and predictions for both architectures and datasets have been done on a NVIDIA Tesla V100 32 GB.
The prediction time is the time the different architectures have used for making predictions on the test-set.

The different kernel sizes of the superkernel and expansion ratios have been investigated. The results show the superkernel almost always have the full size of the initial kernel size (when using a 9x9), but a few times have the size 7x7. Furthermore, the expansion ratios have only been observed to have the full initial size. The reason for both is the datasets are too simple or the architecture is too small. When testing on more complicated datasets such as Cifar100, the kernel sizes varied more and the expansion ratios where not always full. A larger architecture has not been studied, since the training time would have been too long for the length of this project.

Conclusion

The results achieved in this project show that our NAS architecture outperforms ResNet50 on more complex data-sets. On simpler data-sets ResNet50 performs as well as NAS in a fraction of the training time. Our NAS architecture is more adaptable to more complex problems than ResNet, but the training and prediction time is also much longer for the NAS architecture. Thus, there is a trade-off between adaptability and training time when comparing the two architectures.

References

[1] M. F. Kasim et al., Up to two billion times acceleration of scientific simulations with deep neural architecture search, 2020
[2] D. Stamoulis et al., Single-Path NAS: Device-Aware Efficient ConvNet Design, 2019
[3] K. He et al., Deep Residual Learning for Image Recognition, 2015
[4] D. Stamoulis et al., Designing Hardware-Efficient ConvNets in less than 4 hours, 2019
[5] http://yann.lecun.com/exdb/mnist/
[6] https://www.cs.toronto.edu/-kriz/cifar.html