

Transfer Learning and Deep Neural Network Acceleration for Image Classification

Yeqing Huang, Weihua Huang, Arik Horodniceanu, Bowen Zhang, Houjian Yu
{yehuang, w1huang, ahorodni, boz004, h9yu} @ucsd.edu

Motivation and Goal

- Training a network from scratch is time consuming, complex, and ineffective [1].
- Inference runtime for image data depends on a different GPU devices, and fast devices are expensive.
- Transfer the pre-trained neural network model to new models for classification, and compare the results.
- Use TensorRT [2] as the software inference accelerator to optimize and speed up the prediction step, then compare the runtime results to the original GPU runtime.

Dataset and Features

- Monkey Dataset [3] (Demo): 10 species and 1,300 images.
- Stanford Car Dataset [4]: 196 car brands and makes, 16,185 images.



Fig. 1. Monkey dataset



Fig. 2. Car dataset

Models and Methods

Phase 1: Choose models and train

- Use VGG, Resnet18, and Resnet 152 with pretrained weights for transfer learning.
- Fine-tune or freeze and train datasets using VGG and ResNet models.
- Save the weights, and network models for optimization.

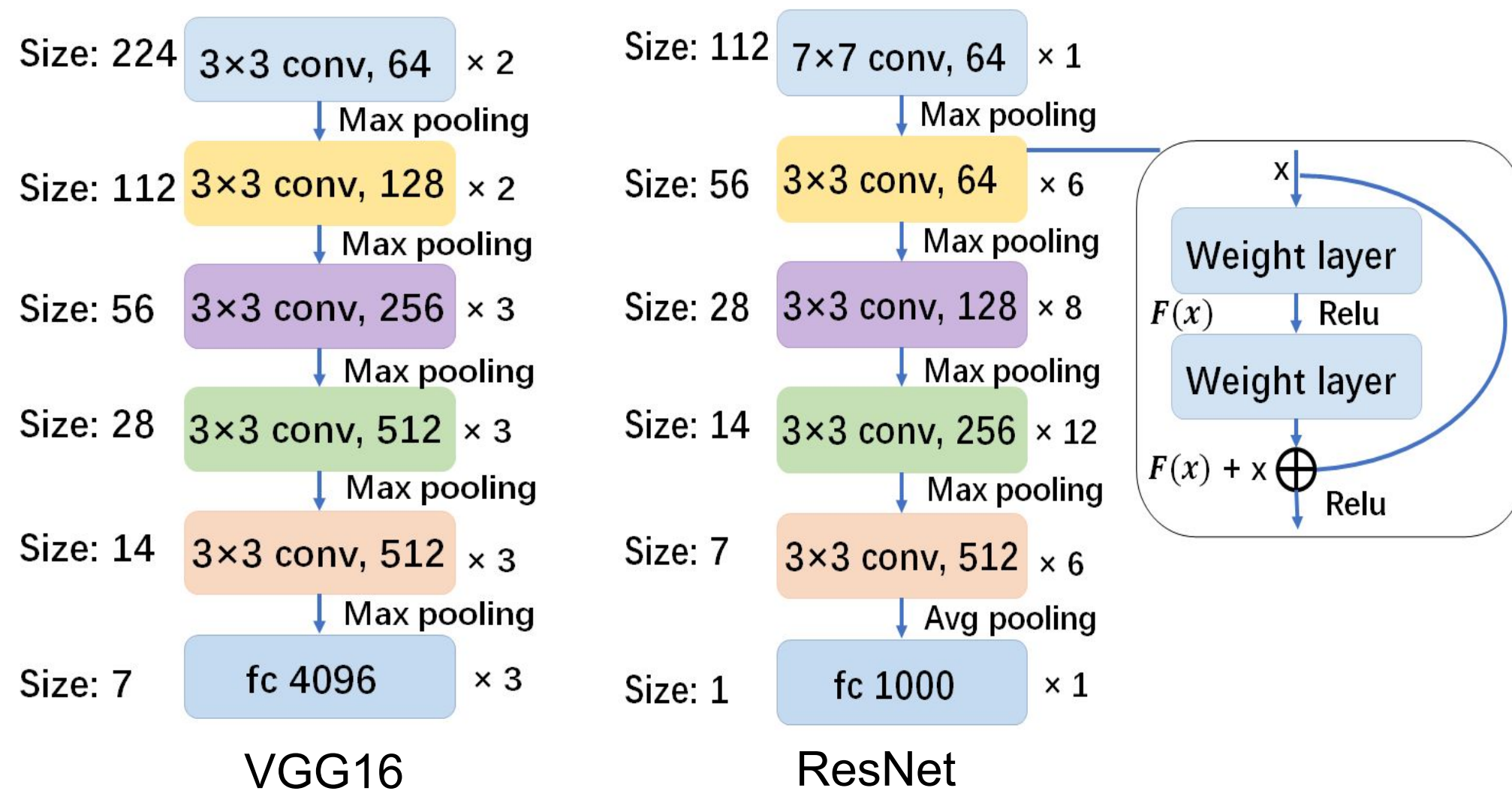


Fig. 3. VGG16 and ResNet models

Phase 2: Optimize models for deployment

- Using TensorRT for inference accelerating.
- Load and parse trained neural network models.
- TensorRT optimizes the models to produce a deployment-ready runtime inference engine.

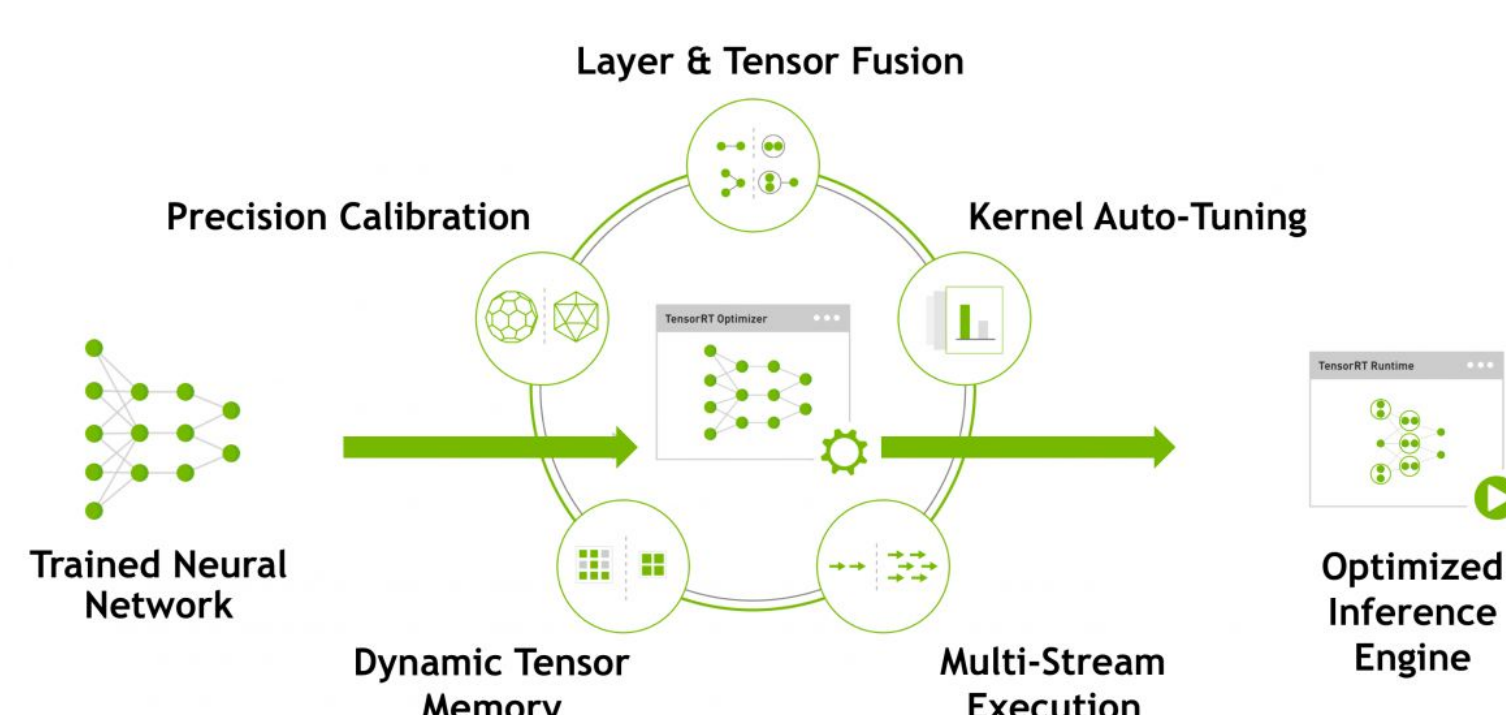


Fig. 3. TensorRT process[2]

Phase 3: Deploy the model

- Load the optimized inference engine.
- Run inference using the same input to compare the accuracy and runtime to the original models.

Results and Discussion

Monkey Dataset (Demo)

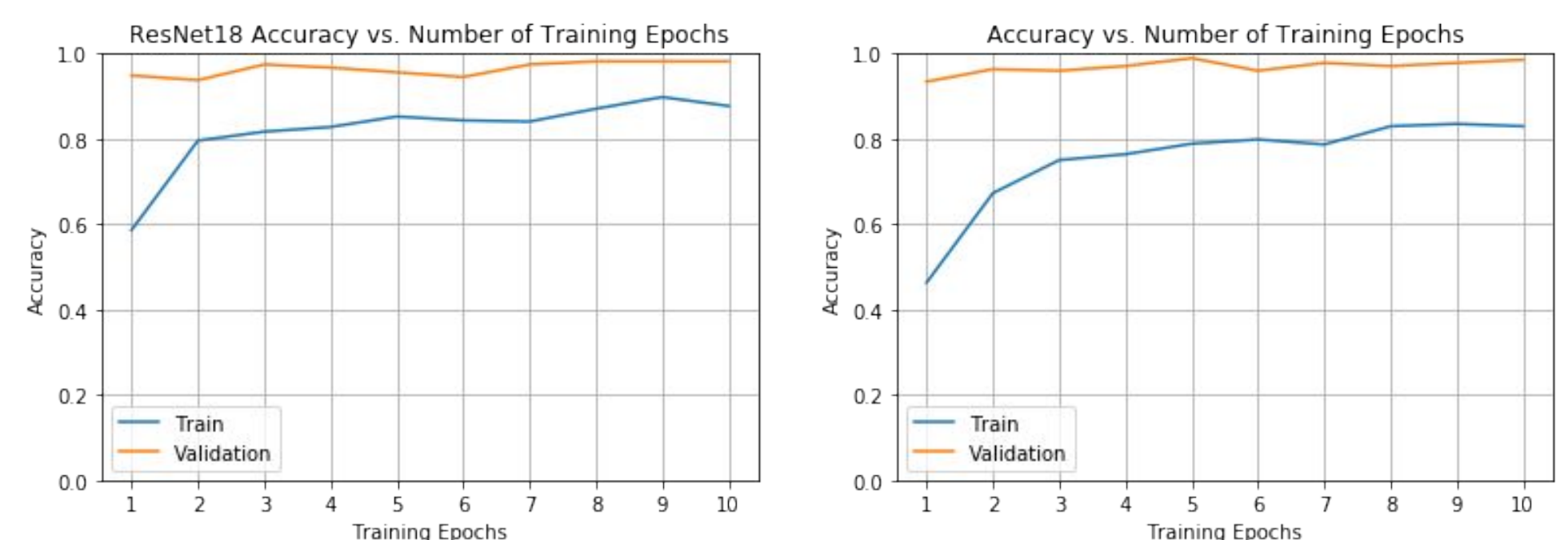


Fig. 4. Accuracy vs. epochs (fine tune) Fig. 5. Accuracy vs. epochs (frozen)

Table 1. Performance of transfer learning

	Finetune	Freeze and Train
Evaluation Loss	0.0536	0.0809
Accuray (%)	98.16	98.53

Stanford Car Dataset

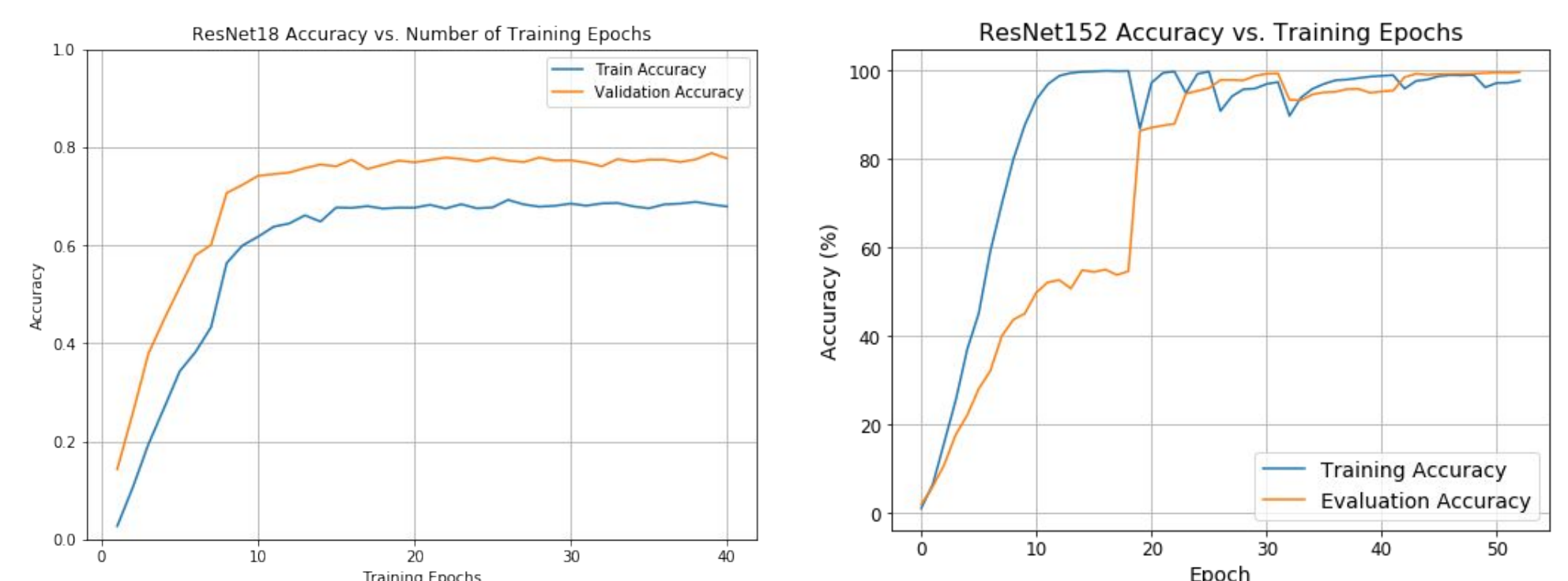


Fig. 6. Accuracy vs. epochs (fine tune) Fig. 7. Accuracy vs. epochs (frozen)

Table 2. How data augmentation improved ResNet152 accuracy

	w/ Data Augmentation	wo/ Data Augmentation
Top-1 Accuracy (%)	82.20%	92.55%
Top-5 Accuracy (%)	94.45%	98.14%

Inference Speed Comparison

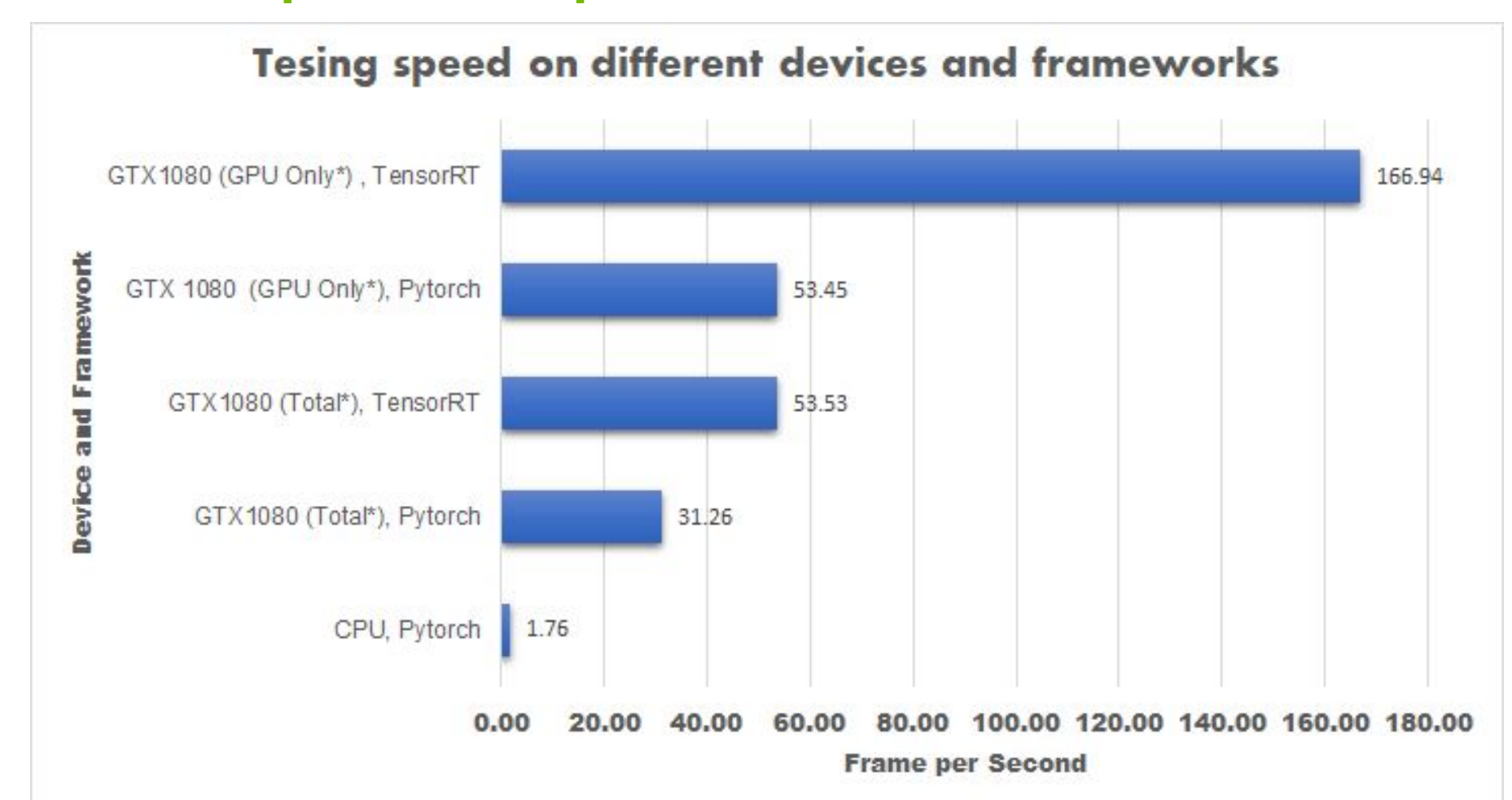


Fig. 8. Inference speed performance (FPS)

* GPU only: the GPU processing speed for prediction. Total: total processing speed including loading image, preprocessing, and post processing using CPU.

- Successfully reach a high accuracy which is the state of are for Cars Dataset, and the total speed after optimization is 30 times faster than CPU and 1.7 times faster than the GPU without optimization. GPU processing speed is 3 times faster than the one without optimization.

Future Work

- Compare different NN models in training
- Modify network structure to overcome overfitting and further improve testing accuracy (state of the art top-1 accuracy 95%)
- Try freeze less layers to see if similar result can be achieved in less training time
- Apply network compression to save storage and further improve training and testing speed

References

- [1] DAWNBench <https://dawn.cs.stanford.edu/benchmark>
- [2] Abstract <https://docs.nvidia.com/deeplearning/sdk/tensorrt-developer-guide/index.html>
- [3] 10 Monkey Species Mario - <https://www.kaggle.com/slothkong/10-monkey-species>
- [4] Cars Dataset https://ai.stanford.edu/~jkruse/cars/car_dataset.html