Friedrich-Alexander-Universität
Technische Fakultät

Research Project (10 ECTS): AutoML Techniques and Evolutionary Search for Efficient AI

Supervisors: **M. Sc. Muhammad Sabih**

Badar Alam

Friedrich-Alexander-Universität Erlangen-Nürnberg, Hardware-Software-Co-Design

March 20, 2024

1. Introduction
2. Evolutionary Pruning
3. Seed search approach
4. Questions

Introduction

Problem

Efficient deployment of NNs on the target hardware

Solution

Pruning
Neural Architecture Search
Quantization

Further Problems

Various research problems within Neural Architecture Search, pruning, etc.

Problems tackled in this Research Project

- How to utilize Evolutionary Algorithm with pruning?
- How to find seed in Neural Architecture Search?

Introduction

Problem

Efficient deployment of NNs on the target hardware

Solution

Pruning
Neural Architecture Search
Quantization

Further Problems

Various research problems within Neural Architecture Search, pruning, etc.

Problems tackled in this Research Project

- How to utilize Evolutionary Algorithm with pruning?
- How to find seed in Neural Architecture Search?

Introduction

Problem

Efficient deployment of NNs on the target hardware

Solution

Pruning
Neural Architecture Search
Quantization

Further Problems

Various research problems within Neural Architecture Search, pruning, etc.

Problems tackled in this Research Project

- How to utilize Evolutionary Algorithm with pruning?
- How to find seed in Neural Architecture Search?

Introduction

Problem

Efficient deployment of NNs on the target hardware

Solution

Pruning
Neural Architecture Search
Quantization

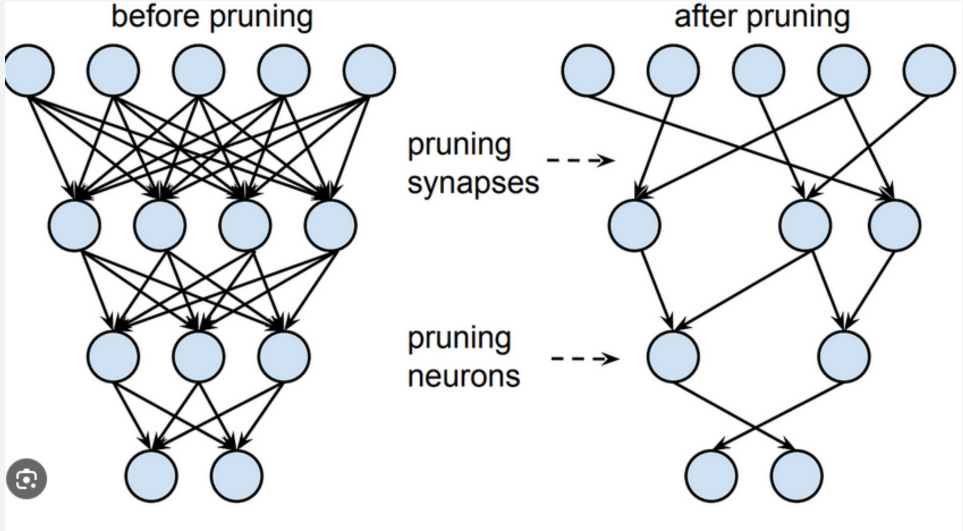
Further Problems

Various research problems within Neural Architecture Search, pruning, etc.

Problems tackled in this Research Project

- How to utilize Evolutionary Algorithm with pruning?
- How to find seed in Neural Architecture Search?

- Goal is to systematically prune the unimportant weights but also maintain high accuracy
- Options
  - Iterative Pruning
  - One-shot pruning
  - Evolutionary Algorithms-based pruning



Introduction

Neural Network Pruning

Iterative Pruning	One-shot Pruning	Evolutionary Pruning
	Pros	
Gradual control, adaptability	Quick reduction, simplicity	Diverse strategy exploration, adaptability
Potential for high accuracy	Simplicity	Non-intuitive discoveries
	Cons	
Increased computation	Potential accuracy loss	Computational overhead
Longer training	Limited fine-tuning	Parameter tuning complexity

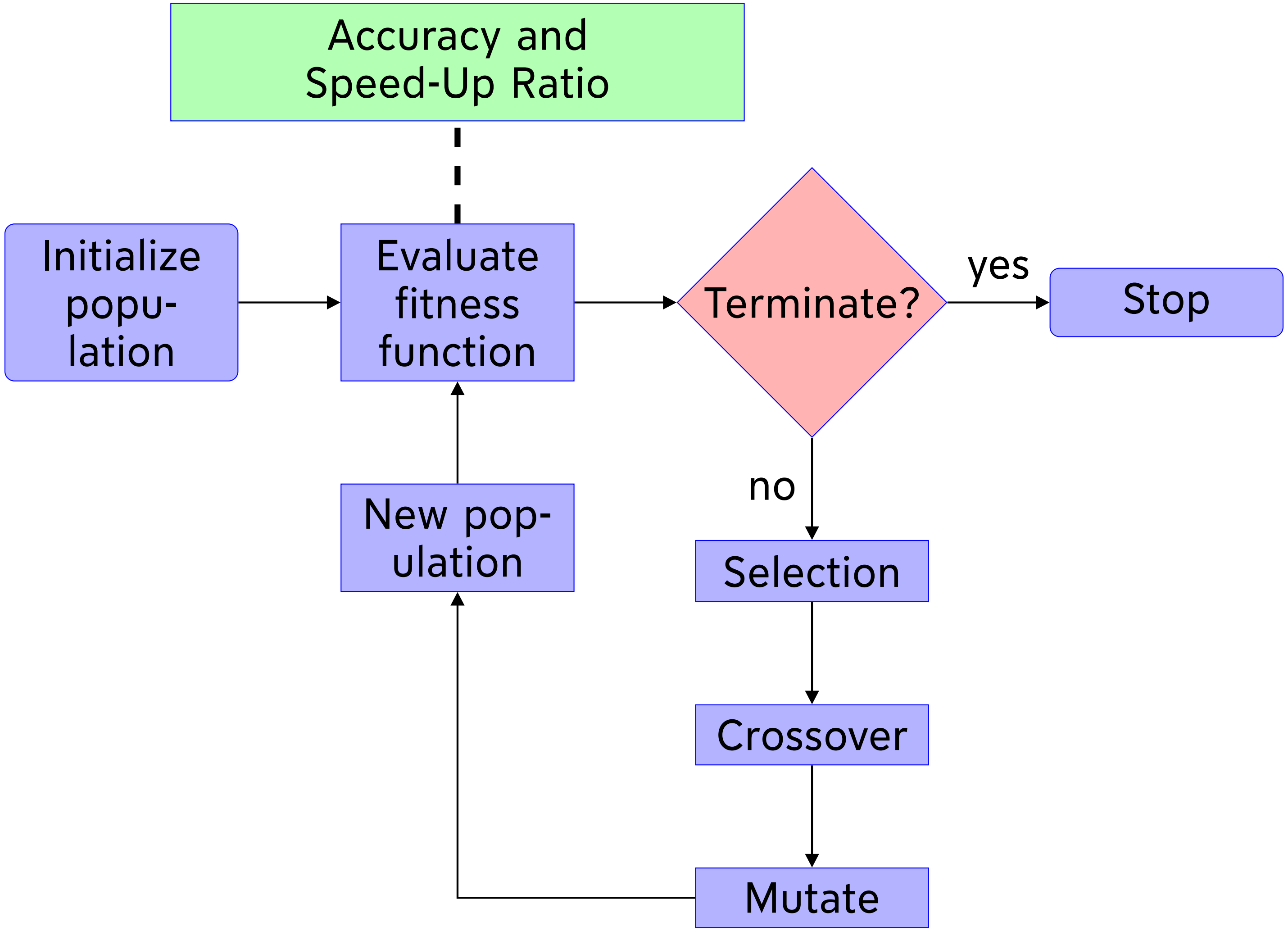
Table: Comparison of Pruning Techniques

Our choice

Our project was to investigate evolutionary Pruning .

Our approach

Evolutionary Algorithm based Pruning



Our approach

Evolutionary Algorithm based Pruning

Objectives → speedup and accuracy

fitness function → Accuracy and speedup from semi-structured pruned model

reducing search space → Grouping of layers and sparsity step size

Our approach

Evolutionary Algorithm based Pruning

Objectives → speedup and accuracy

fitness function → Accuracy and speedup from semi-structured pruned model

reducing search space → Grouping of layers and sparsity step size

Objectives → speedup and accuracy

fitness function → Accuracy and speedup from semi-structured pruned model

reducing search space → Grouping of layers and sparsity step size

Evaluation

- Dataset: Cifar10
- Model: Resnet50
- Search-time: 5 Hours
- Accuracy of searched model:0.9113
- speedup of searched model: 2.226818
- target hardware:FPGA (Estimated)

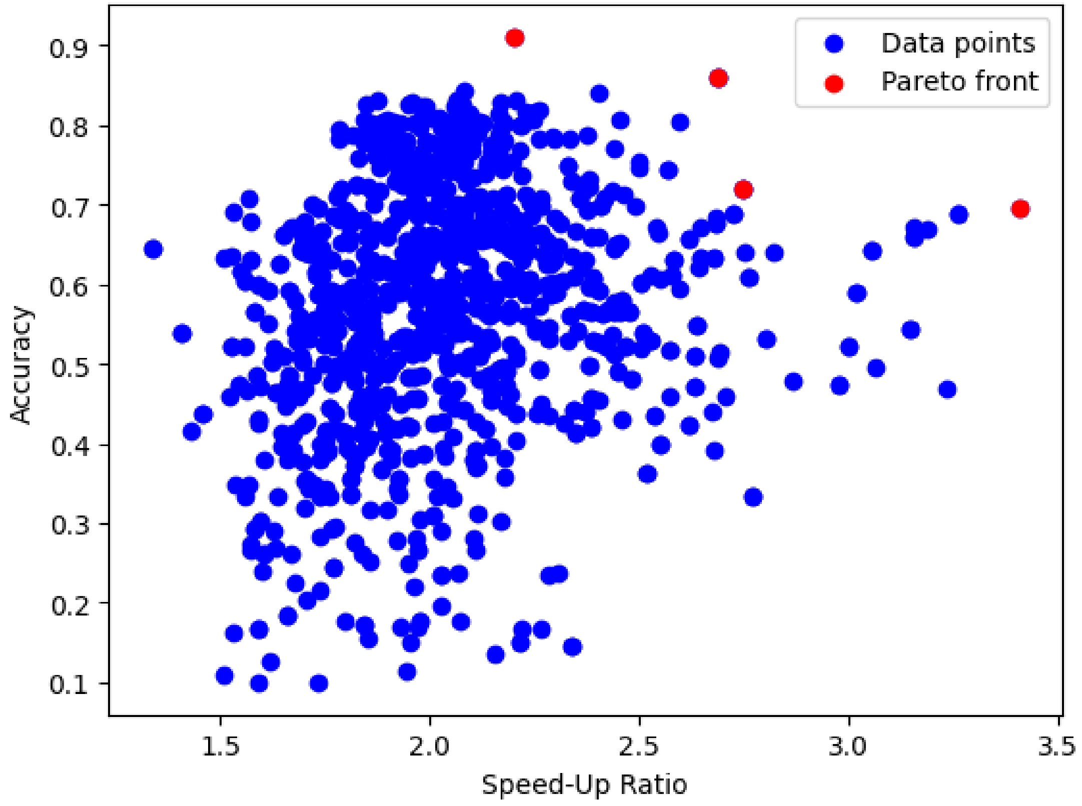


Figure: Fig.4 front-sparcity-selection

Our approach

Seed search approach

Problem → Most NAS approaches assume initial topology or cell

Solution → We propose to search a seed model from HW-NAS dataset

Our approach
Seed search approach

Problem

Most NAS approaches assume initial topology or cell

Solution

We propose to search a seed model from HW-NAS dataset

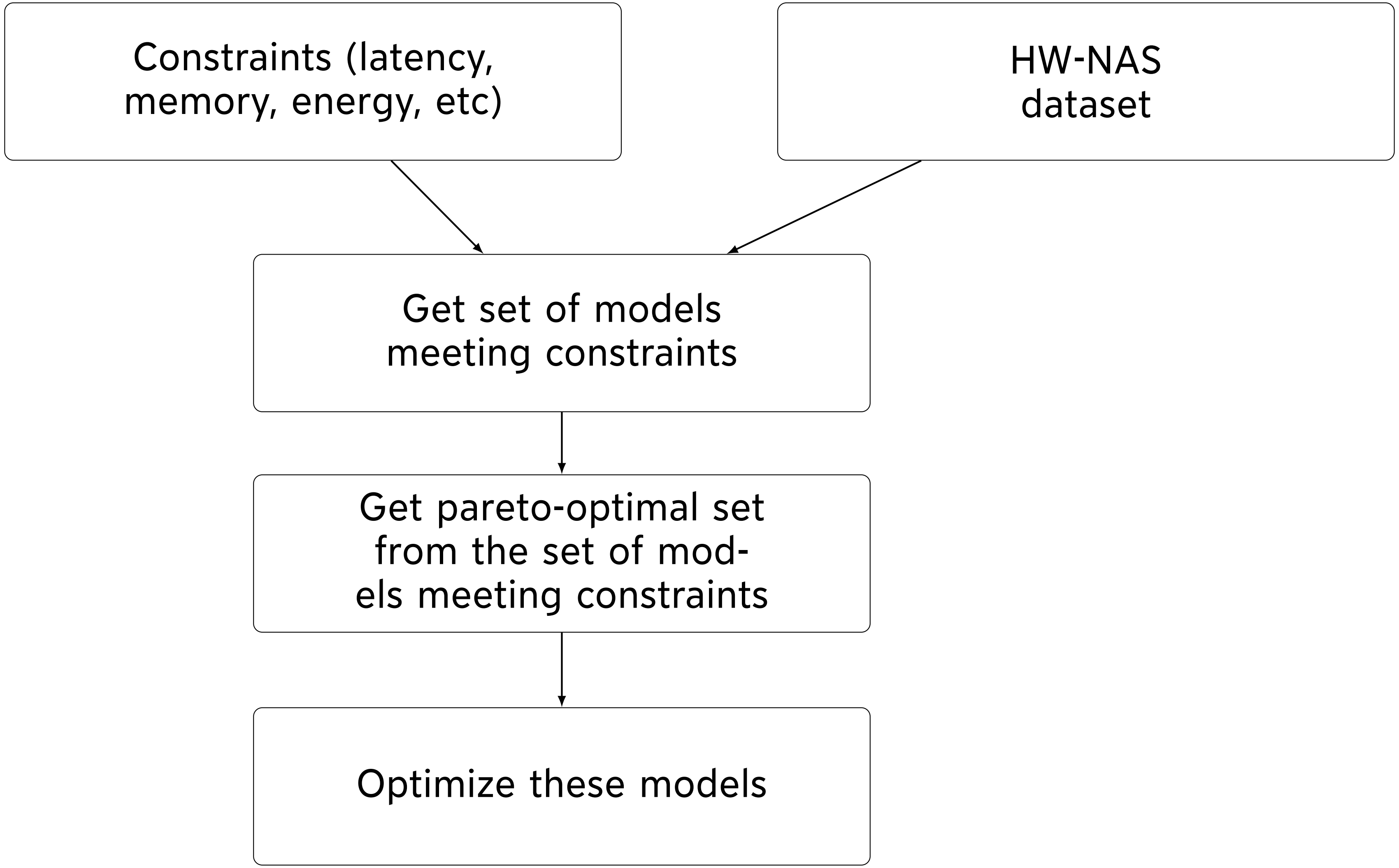
HW-NAS

- HW-NAS benchmark is a dataset consisting total 46k neural network architecture trained on CIFAR10,CIFAR100 and imageNet.
- For an unknown image related dataset, we choose points based on similary with CIFAR10, CIFAR100 or imageNet, in terms of complexity

Devices	Edge GPU	Raspi 4	Edge TPU	Pixel 3	ASIC-Eyeriss	FPGA
Collected Metrics	Latency (ms) Energy (mJ)	Latency (ms)	Latency (ms)	Latency (ms)	Latency (ms) Energy (mJ)	Latency (ms) Energy (mJ)
Collecting Method	Measured	Measured	Measured	Measured	Estimated	Estimated
Runtime Envi-ronment	TensorRT	TensorFlow Lite	Edge TPU Runtime	TensorFlow Lite	AccelerEyes	Vivado HLS
Customizing Hardware?	×	×	×	×	✓	✓
Category	Commercial Edge Devices			ASIC		FPGA

Our approach

Seed Search



Seed Search Approach

Application

- We used the PathMNIST part of MedMNIST dataset .
- It contain 9 classes, so it resembles with CIFAR-10 in number of classes.
- Queried a network based on these constraint and then employed Optuna for hyperparameter optimization.
- State-of-the-art manually designed model obtained around 90% test accuracy with 11.7 M parameter.
- We obtain a model from the HW-NAS bench using the approach above and obtained 90% accuracy with 1.1 M parameters .

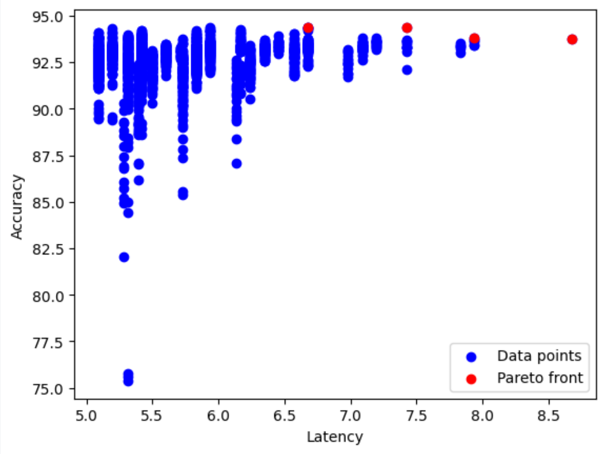


Figure: Fig.5 Pareto front based on constraints

Seed Search Approach

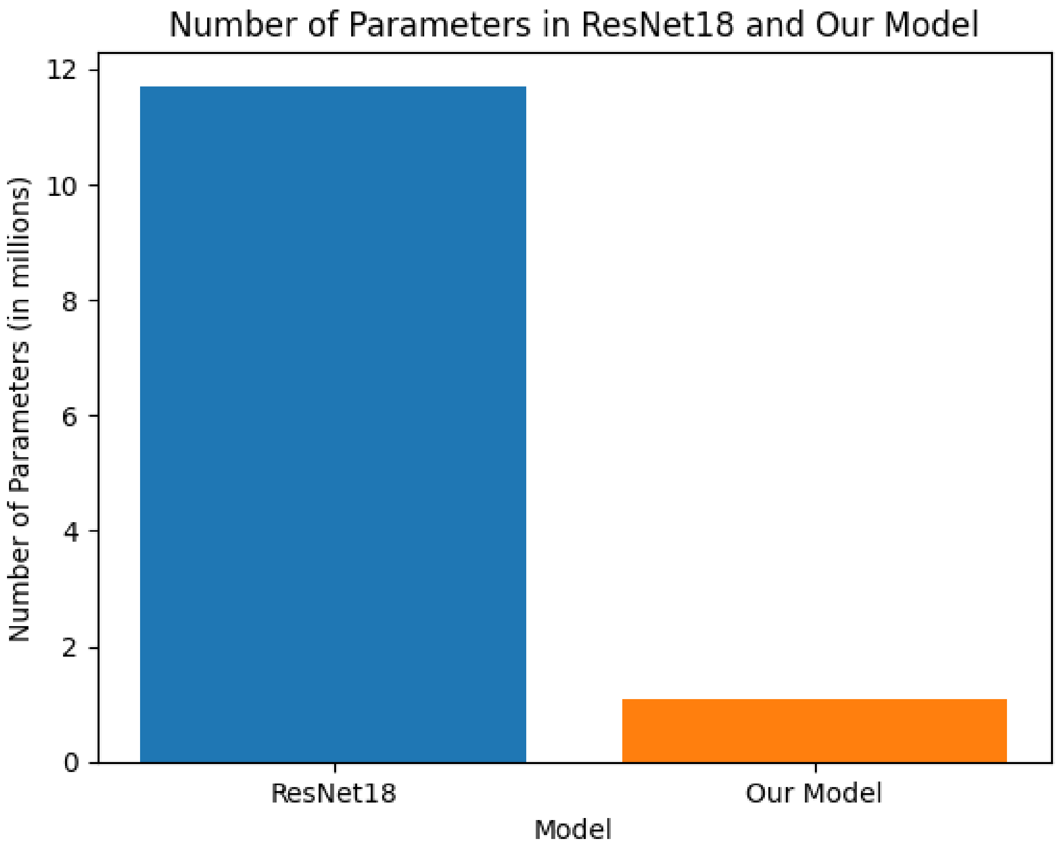


Figure: Fig.6 Parameter Comparison

Seed Search Approach

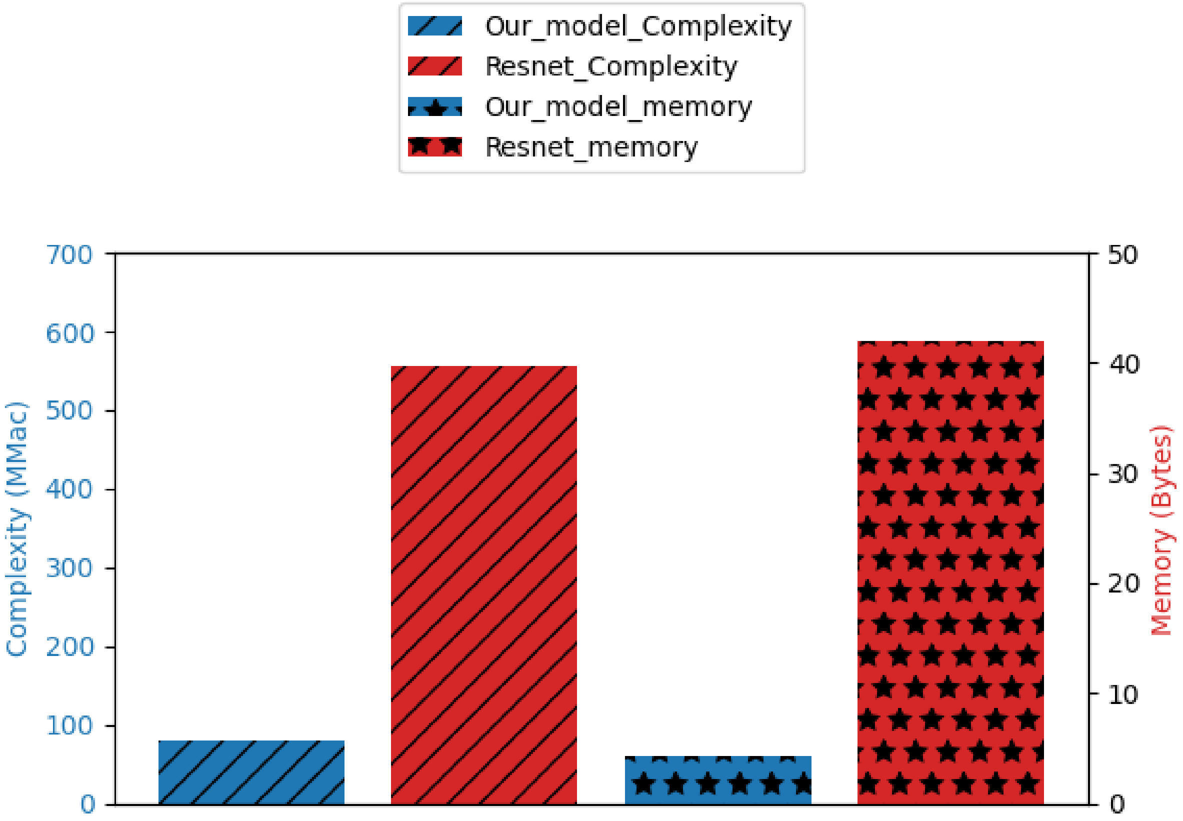


Figure: Fig.5 Memory consumption and Latency