CSYE 7105

Team 13

Parallelizing Plant Health

A Comparative Study of Parallel Deep Learning for Plant Disease Classification

He Zhou
Weiyu Wang

Outline

Introduction	Methodology	Dataset Description	Result & Analysis	Conclusion
1	2	3	4	5
✓ Background ✓ Motivation ✓ Goals	✓ Roadmap ✓ Mechanism ✓ Tools	✓ Overview ✓ Splitting ✓ EDA	✓ Pre-processing ✓ Pre-training ✓ Hyperparameter tuning ✓ Final Evaluation	✓ How does parallelism perform in each phase?

1 Introduction

1.1 Background

How much economic loss do plant diseases cause each year?

1 Introduction

1.1 Background

$220 billion annually

according to the Food and Agriculture Organization of the **United Nations**

developing countries, small stakeholders…

1 Introduction

1.2 Motivation & goal

Motivation

- **Achieve timely disease identification**
- **Datasets are huge, Trainings are slow** - facing the challenge of the large image datasets

Goal

- **Accelerate** the training by leveraging the power of parallel techniques
- **Compare** the time efficiency of serial and parallel deep learning, analyze the potential benefits of parallel approaches.

2 Methodology

Roadmap

Step	Roadmap	Focuses	Tools	Parallelization Comparison
1	**Exploratory Data Analysis**	Number of classes	NumPy, Matplotlib	
2	**Data Preprocessing**	Mean, std calculation for dataset normalization	(1) Dask Scheduler (2) PyTorch	√
3	**Model Definition**	Resnet18	Torchvision	
4	**Model Pretraining**	(1) Train on an initial set of hyperparameters on small number of epochs (2) Compare time efficiency between serial and parallel techniques	PyTorch (1) Serial (2) Using multi-process (num_workers) (3) Using multi-thread (DataParallel)	√
5	**Hyperparameter Tuning**	Bayesian optimization	Optuna	√
6	**Final Training and Model Evaluation**	Use optimized hyperparameters on extended epochs accuracy, precision, recall, and F1 score, confusion matrix	PyTorch Scikit-learn, Seaborn	

3 Dataset Description

3.1 Overview

SAMIR BHATTARAI · UPDATED 5 YEARS AGO

860 | New Notebook | Download (3 GB)

New Plant Diseases Dataset

Image dataset containing different healthy and unhealthy crop leaves.

Data Card | Code (280) | Discussion (4)

About Dataset

This dataset is recreated using offline augmentation from the original dataset. The original dataset can be found on this github repo. This dataset consists of about 87K rgb images of healthy and diseased crop leaves which is categorized into 38 different classes. The total dataset is divided into 80/20 ratio of training and validation set preserving the directory structure. A new directory containing 33 test images is created later for prediction purpose.

Usability
7.50

License
Data files © Original Authors

Expected update frequency
Not specified

Tags
Biology | Image

Source:
https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset

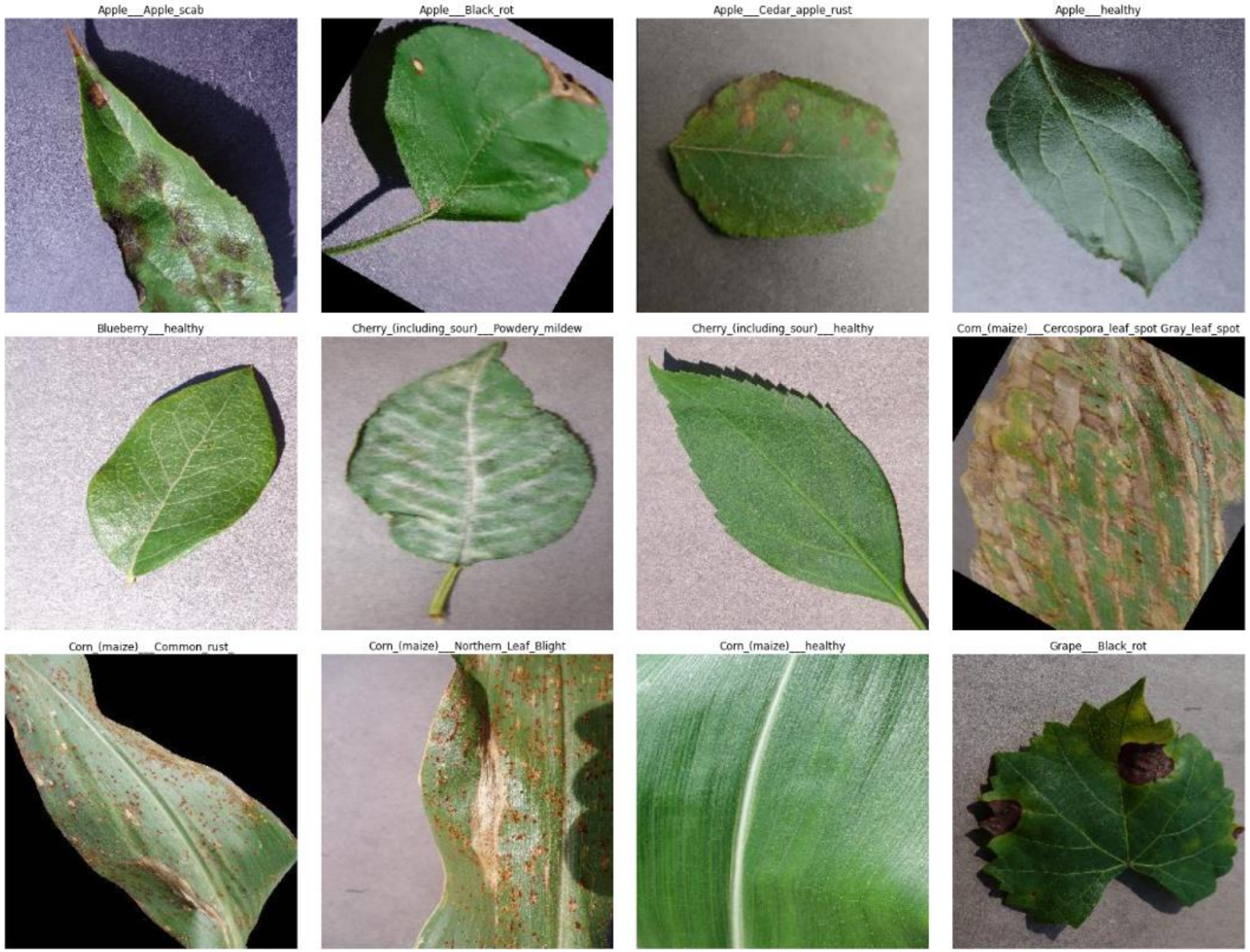
Total size:
1.43 GB, contains 87.9k images

Dataset after splitting:

Subset	Image counts	Percent
train	61494	70%
valid	17572	20%
test	8814	10%
Total	87880	100%

3 Dataset Description

3.2 Exploratory Data Analysis



3 Dataset Description

3.2 Exploratory Data Analysis

The dataset is organized into 38 distinct categories based on plant species and specific diseases. Among these categories, there are 14 distinct plant categories and 26 types of plant diseases.

14 Plant Categories: Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato.

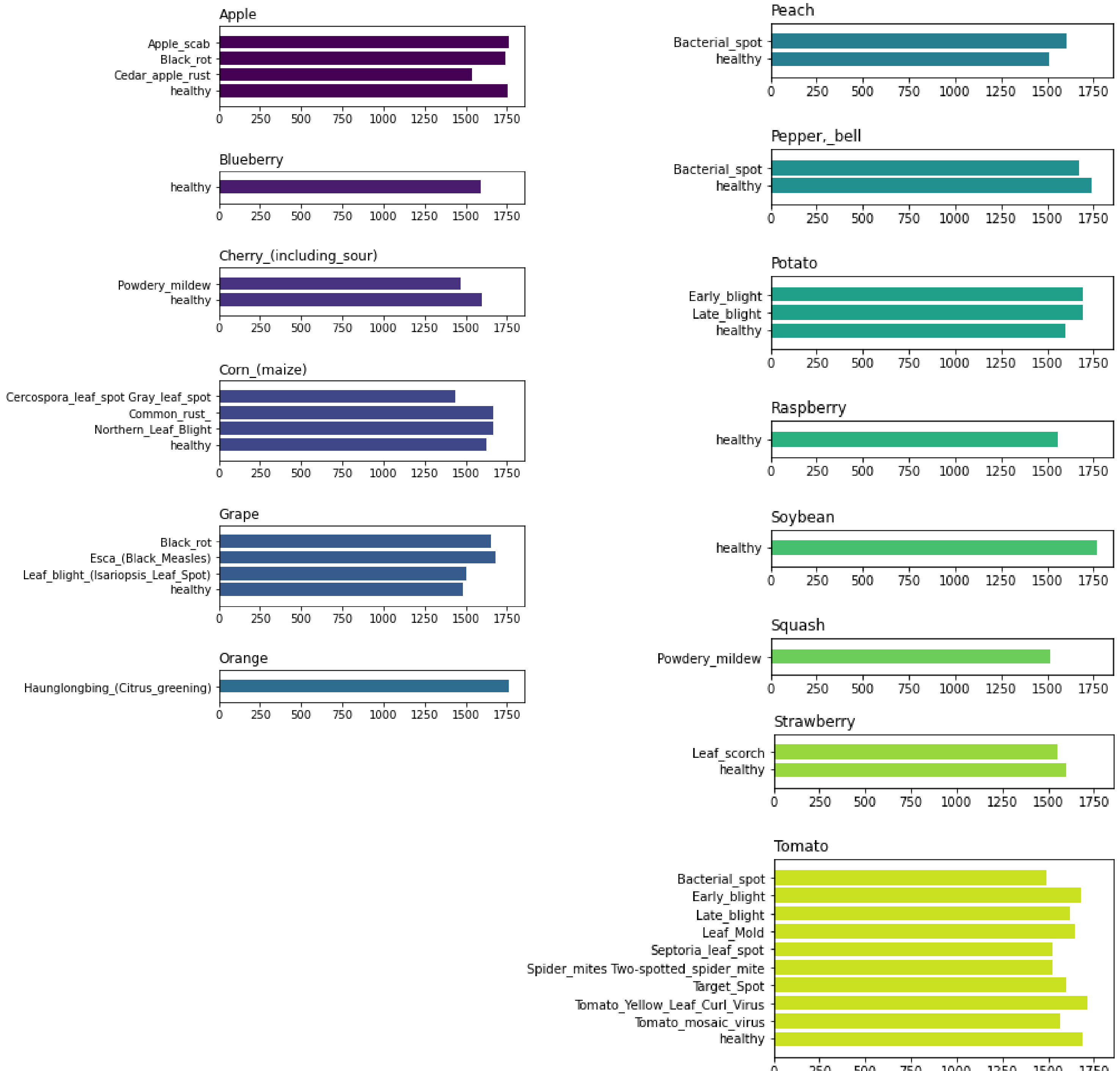
	File Name	Image Count
0	Apple___Apple_scab	1764
1	Apple___Black_rot	1738
2	Apple___Cedar_apple_rust	1540
3	Apple___healthy	1757
4	Blueberry___healthy	1589
5	Cherry_(including_sour)___Powdery_mildew	1472
6	Cherry_(including_sour)___healthy	1597
7	Corn_(maize)___Cercospora_leaf_spot Gray_leaf_...	1436
8	Corn_(maize)___Common_rust_	1668
9	Corn_(maize)___Northern_Leaf_Blight	1669
10	Corn_(maize)___healthy	1626
11	Grape___Black_rot	1652
12	Grape___Esca_(Black_Measles)	1680
13	Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1506
14	Grape___healthy	1480
15	Orange___Haunglongbing_(Citrus_greening)	1758
16	Peach___Bacterial_spot	1608
17	Peach___healthy	1512
18	Pepper,_bell___Bacterial_spot	1673
19	Pepper,_bell___healthy	1739
20	Potato___Early_blight	1696
21	Potato___Late_blight	1696
22	Potato___healthy	1596
23	Raspberry___healthy	1558
24	Soybean___healthy	1769
25	Squash___Powdery_mildew	1519
26	Strawberry___Leaf_scorch	1552
27	Strawberry___healthy	1596
28	Tomato___Bacterial_spot	1489
29	Tomato___Early_blight	1680
30	Tomato___Late_blight	1619
31	Tomato___Leaf_Mold	1646
32	Tomato___Septoria_leaf_spot	1526
33	Tomato___Spider_mites Two-spotted_spider_mite	1523
34	Tomato___Target_Spot	1598
35	Tomato___Tomato_Yellow_Leaf_Curl_Virus	1715
36	Tomato___Tomato_mosaic_virus	1566
37	Tomato___healthy	1685

3 Dataset Description

3.2 Exploratory Data Analysis

The dataset is organized into 38 distinct categories based on plant species and specific diseases. Among these categories, there are 14 distinct plant categories and 26 types of plant diseases.

14 Plant Categories: Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato.



4 Result & Analysis

4.1 Data Preprocessing

Data Cleaning: Identify or delete dirty image data and control image data pixel consistency.

Import all the crucial libraries:
import os
import dask
import dask.array as da from skimage.io
import imread import numpy as np

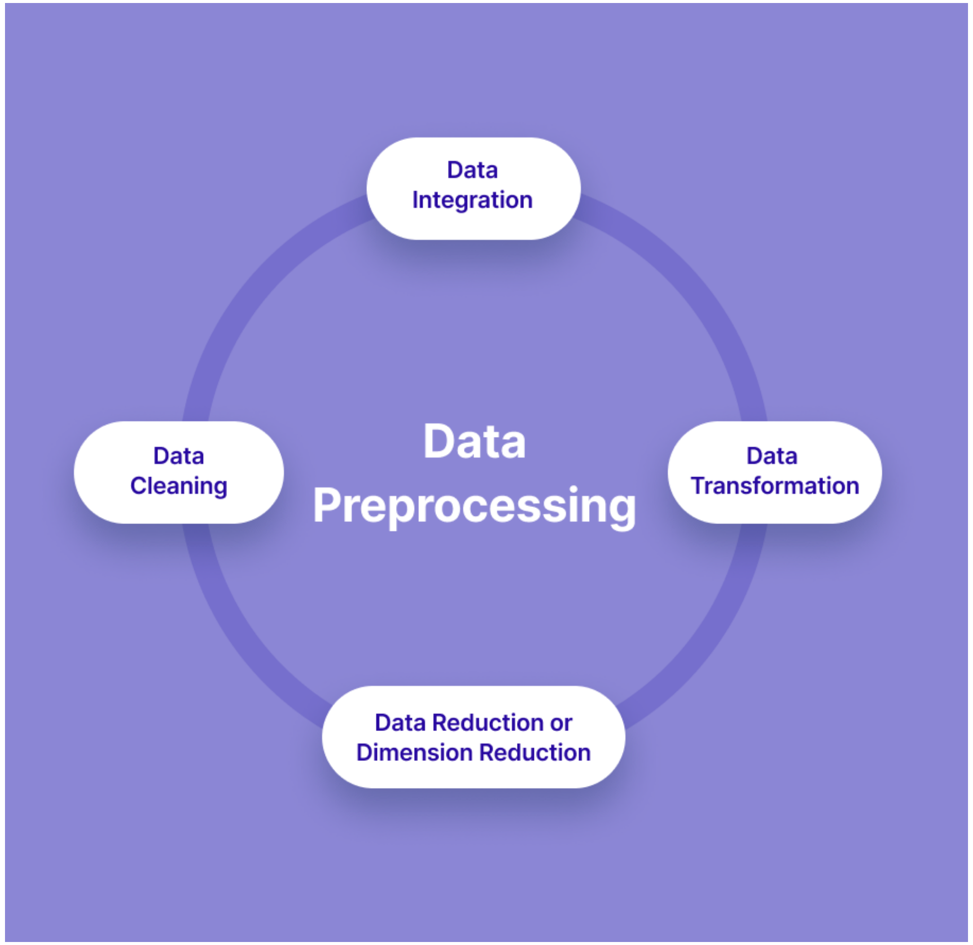
Data Transformation:
Read the image and convert it to an array of values.
Convert an image to a PyTorch tensor using transforms.Compose.

Data Normalization: Calculate mean and Standard Deviation

Classic ImageNet statistics:

mean = [0.485, 0.456, 0.406]

std = [0.229, 0.224, 0.225]

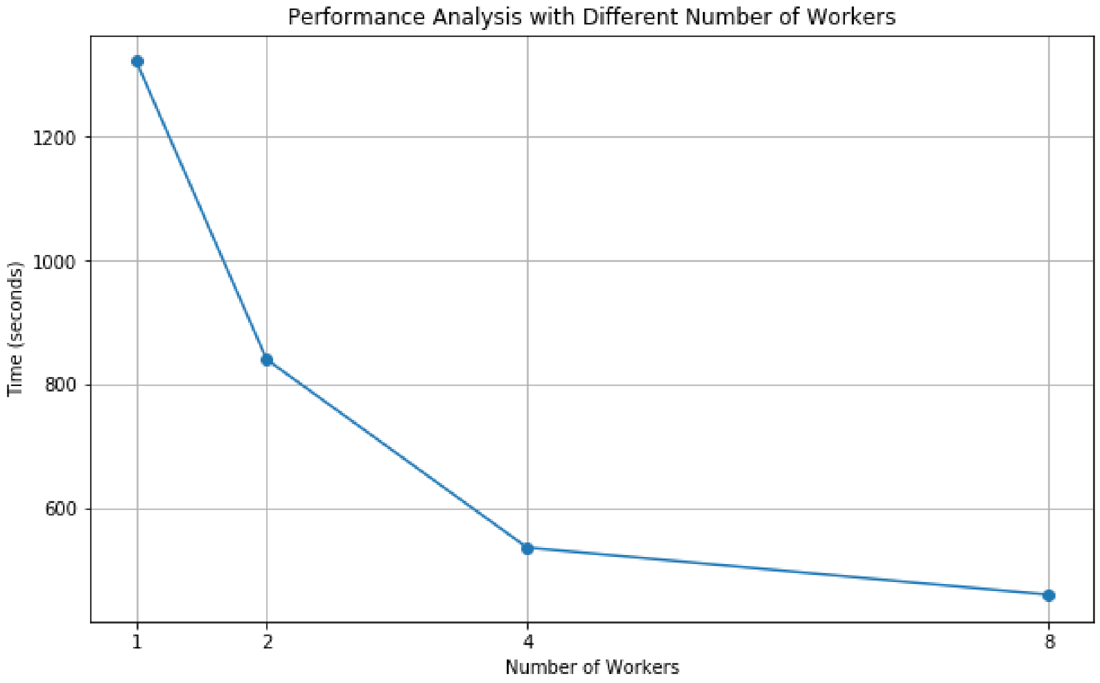


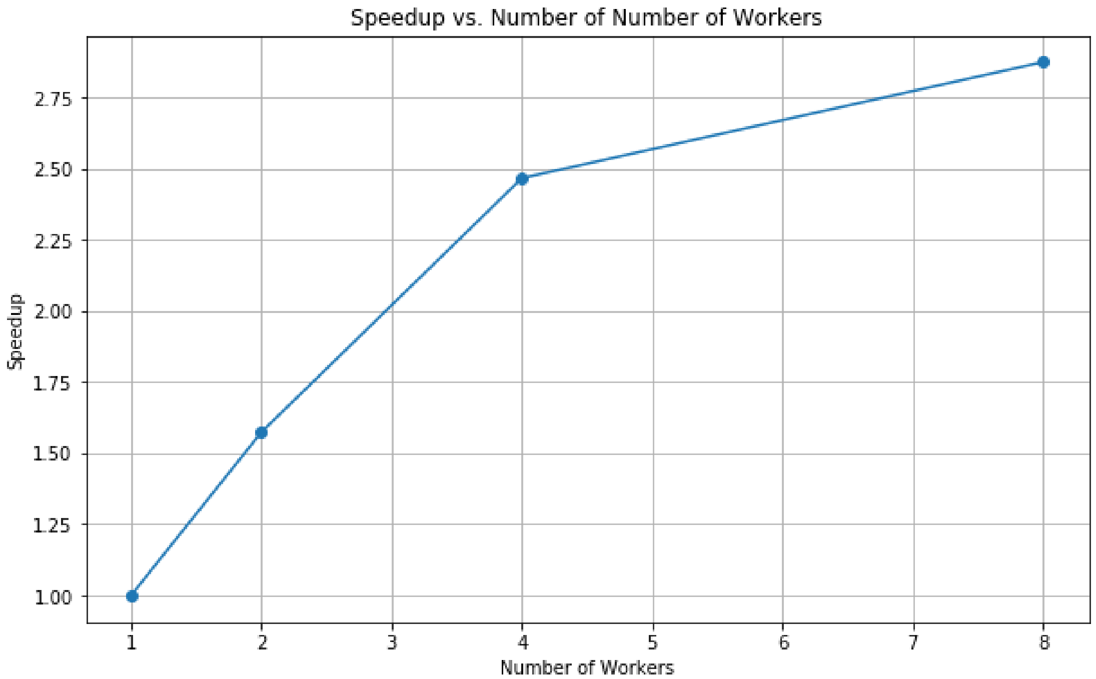
4 Result & Analysis

4.1 Data Preprocessing

Using Dask

Workers	Time(s)	Speed-up
1	1321.3	1
2	840.0	1.57
4	526.0	2.47
8	459.9	2.87

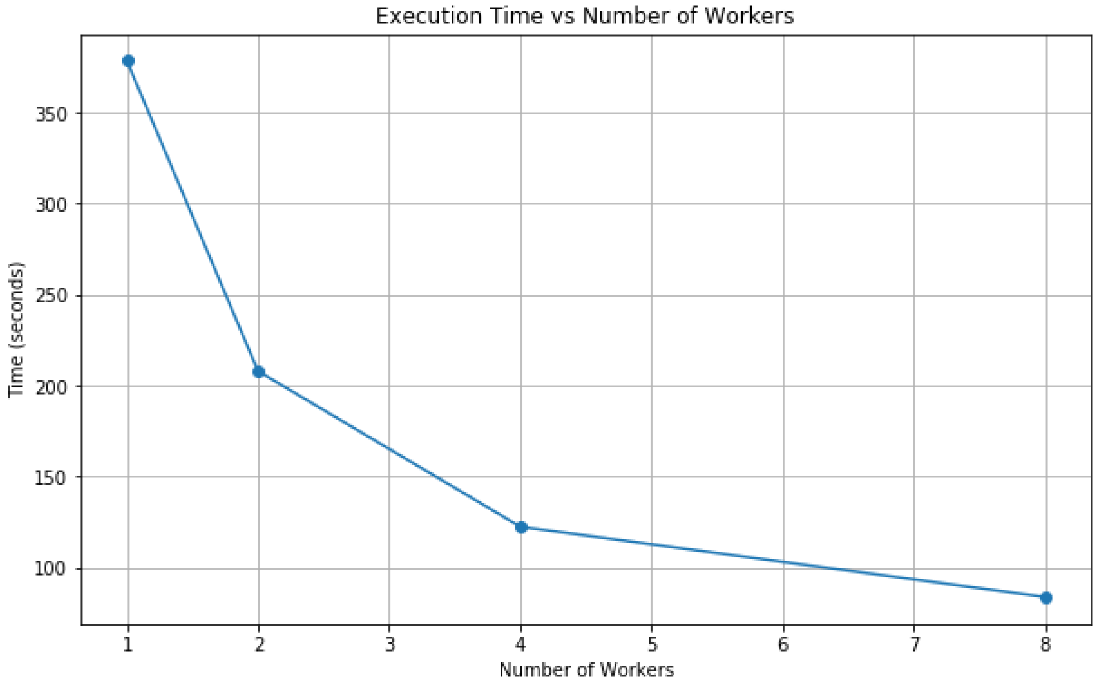


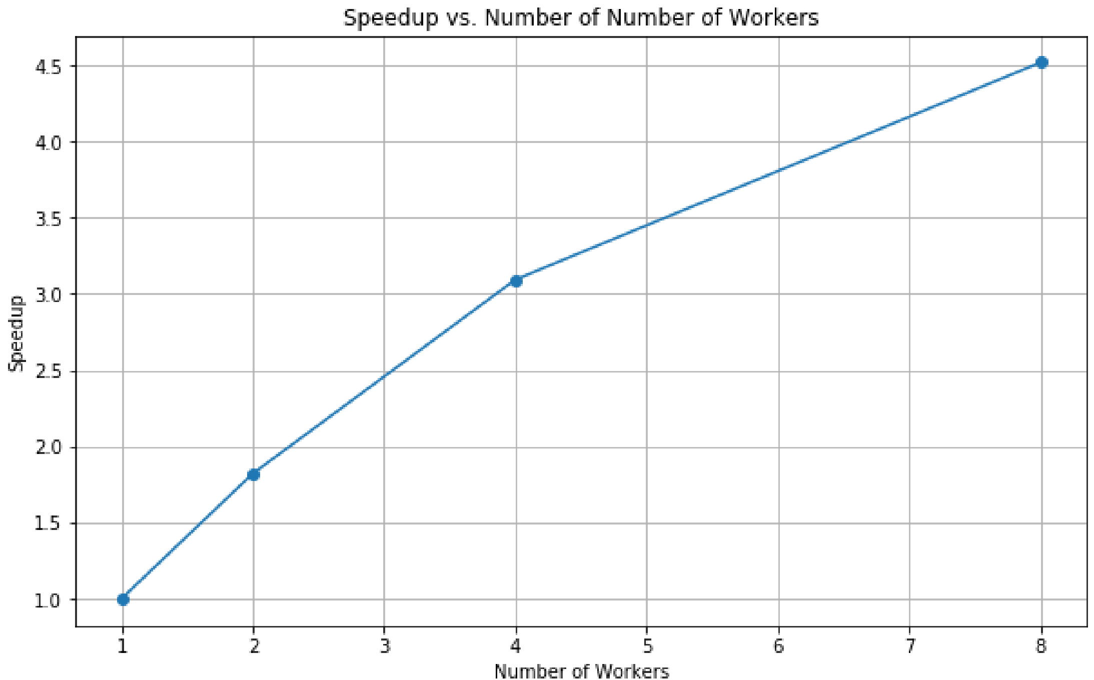


4 Result & Analysis

4.1 Data Preprocessing Using PyTorch

Workers	Time(s)	Speed-up
1	378.3	1
2	207.8	1.82
4	122.3	3.09
8	83.7	4.52





4 Result & Analysis

4.2 Pre-training

Serial Training

Hardware

CPU
Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz
Architecture: X86_64
Count: 4 **(1 processes)**

GPU
Tesla T4
Count: 1

```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:3B:00.0 Off |                    0 |
| N/A   52C    P0              29W /  70W |      2MiB / 15360MiB |      6%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

4 Result & Analysis

4.2 Pre-training

Serial Training

Progress

```
Running on 1 GPU with 1 process---------------------------
Epoch [1/5]-------------------------------------------------------------------------
| Train Loss: 2.6422 | Train Acc: 0.3406 | Val Loss: 2.2021, Val Acc: 0.4074
| Elapsed Time: 445.8777 s | Max GPU Memory Alloc: 1960.8457 MB
Epoch [2/5]-------------------------------------------------------------------------
| Train Loss: 2.0209 | Train Acc: 0.4719 | Val Loss: 1.7971, Val Acc: 0.5216
| Elapsed Time: 424.5686 s | Max GPU Memory Alloc: 1960.8472 MB
Epoch [3/5]-------------------------------------------------------------------------
| Train Loss: 1.8274 | Train Acc: 0.5141 | Val Loss: 1.6481, Val Acc: 0.5436
| Elapsed Time: 425.9290 s | Max GPU Memory Alloc: 1960.8481 MB
Epoch [4/5]-------------------------------------------------------------------------
| Train Loss: 1.2209 | Train Acc: 0.6451 | Val Loss: 1.2316, Val Acc: 0.6412
| Elapsed Time: 428.5931 s | Max GPU Memory Alloc: 1960.8491 MB
Epoch [5/5]-------------------------------------------------------------------------
| Train Loss: 1.2085 | Train Acc: 0.6483 | Val Loss: 1.2196, Val Acc: 0.6451
| Elapsed Time: 427.7931 s | Max GPU Memory Alloc: 1960.8501 MB

total_time: 2153.4934573173523
best_valid_acc: 0.6450603232415206
```

4 Result & Analysis

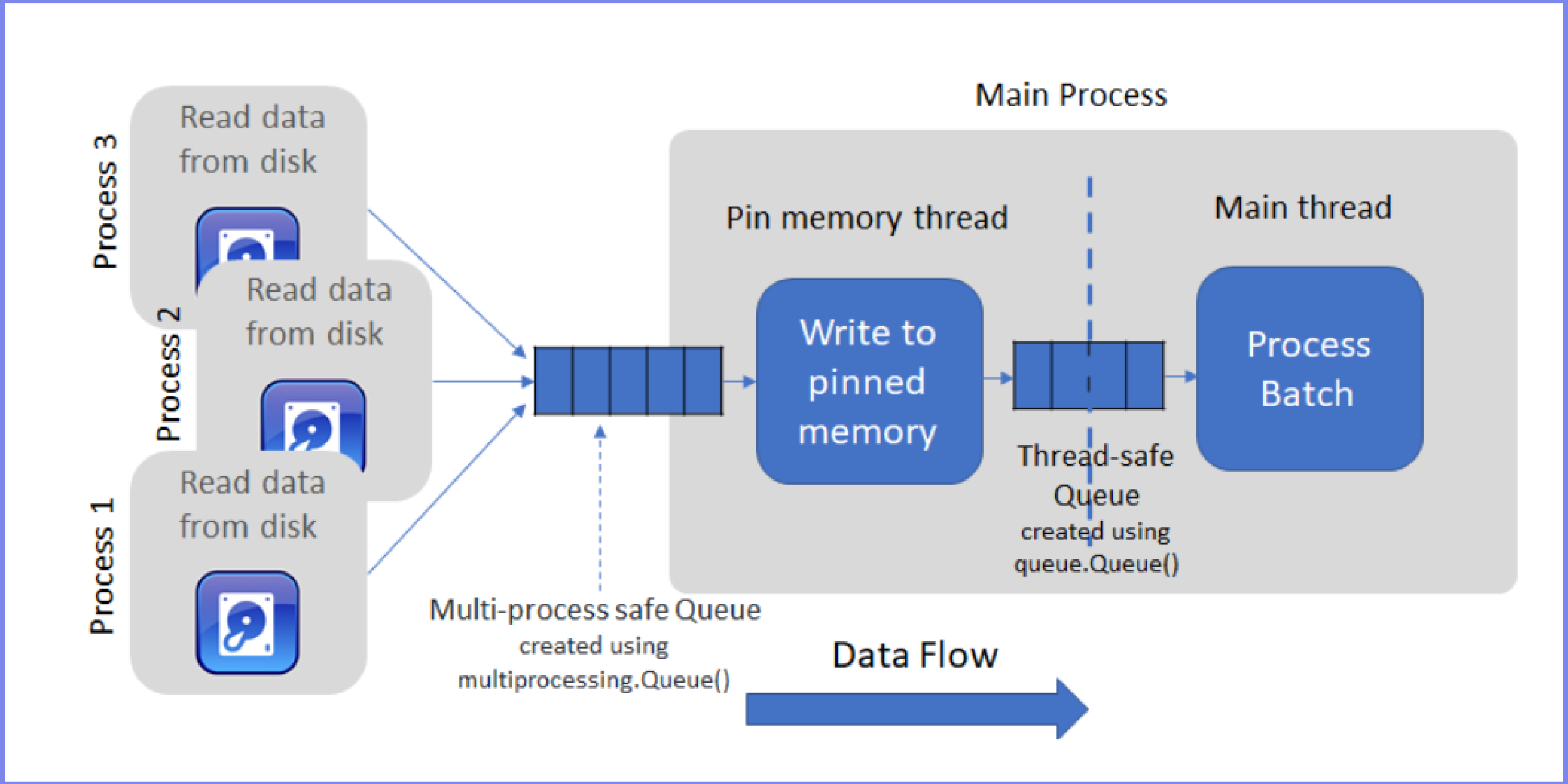
4.2 Pre-training Using Multi-process (workers)

Parallel Mechanism

Multi workers in DataLoader

It uses multiple processes to load and process batch data from the disk into memory and puts it into a queue.

The main training process then consumes batches from this queue, reducing the overall time taken for each training iteration.



4 Result & Analysis

4.2 Pre-training

Using Multi-process (workers)

Hardware

CPU
Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz
Architecture: X86_64
Count: 4 **(2, 4, 8 processes)**

GPU
Tesla T4
Count: 1

```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:3B:00.0 Off |                    0 |
| N/A   52C    P0              29W /  70W |      2MiB / 15360MiB |      6%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

4 Result & Analysis

4.2 Pre-training

Using Multi-process (workers)

2

processes

Progress

```
Running on 1 GPU with 2 processes---------------------------
Epoch [1/5]------------------------------------------------------------------------
| Train Loss: 2.6422 | Train Acc: 0.3406 | Val Loss: 2.2021, Val Acc: 0.4074
| Elapsed Time: 248.3875 s | Max GPU Memory Alloc: 1960.8457 MB
Epoch [2/5]------------------------------------------------------------------------
| Train Loss: 2.0209 | Train Acc: 0.4719 | Val Loss: 1.7971, Val Acc: 0.5216
| Elapsed Time: 227.4754 s | Max GPU Memory Alloc: 1960.8472 MB
Epoch [3/5]------------------------------------------------------------------------
| Train Loss: 1.8274 | Train Acc: 0.5141 | Val Loss: 1.6481, Val Acc: 0.5436
| Elapsed Time: 231.8289 s | Max GPU Memory Alloc: 1960.8481 MB
Epoch [4/5]------------------------------------------------------------------------
| Train Loss: 1.2209 | Train Acc: 0.6451 | Val Loss: 1.2316, Val Acc: 0.6412
| Elapsed Time: 227.1396 s | Max GPU Memory Alloc: 1960.8491 MB
Epoch [5/5]------------------------------------------------------------------------
| Train Loss: 1.2085 | Train Acc: 0.6483 | Val Loss: 1.2196, Val Acc: 0.6451
| Elapsed Time: 234.1333 s | Max GPU Memory Alloc: 1960.8501 MB

total_time: 1169.8230679035187
best_valid_acc: 0.6450603232415206
```

4 Result & Analysis

4.2 Pre-training Using Multi-process (workers)

4
processes

Progress

```
Running on 1 GPU with 4 processes----------------------------
Epoch [1/5]-----------------------------------------------------------------------
| Train Loss: 2.6422 | Train Acc: 0.3406 | Val Loss: 2.2021, Val Acc: 0.4074
| Elapsed Time: 223.4081 s | Max GPU Memory Alloc: 1960.8457 MB
Epoch [2/5]-----------------------------------------------------------------------
| Train Loss: 2.0209 | Train Acc: 0.4719 | Val Loss: 1.7971, Val Acc: 0.5216
| Elapsed Time: 222.7155 s | Max GPU Memory Alloc: 1960.8472 MB
Epoch [3/5]-----------------------------------------------------------------------
| Train Loss: 1.8274 | Train Acc: 0.5141 | Val Loss: 1.6481, Val Acc: 0.5436
| Elapsed Time: 223.5448 s | Max GPU Memory Alloc: 1960.8481 MB
Epoch [4/5]-----------------------------------------------------------------------
| Train Loss: 1.2209 | Train Acc: 0.6451 | Val Loss: 1.2316, Val Acc: 0.6412
| Elapsed Time: 220.5944 s | Max GPU Memory Alloc: 1960.8491 MB
Epoch [5/5]-----------------------------------------------------------------------
| Train Loss: 1.2085 | Train Acc: 0.6483 | Val Loss: 1.2196, Val Acc: 0.6451
| Elapsed Time: 223.2077 s | Max GPU Memory Alloc: 1960.8501 MB

total_time: 1114.2051734924316
best_valid_acc: 0.6450603232415206
```

4 Result & Analysis

4.2 Pre-training

Using Multi-process (workers)

8

processes

over-subscription

Progress

```
Running on 1 GPU with 8 processes---------------------------
Epoch [1/5]------------------------------------------------------------------------
| Train Loss: 2.6422 | Train Acc: 0.3406 | Val Loss: 2.2021, Val Acc: 0.4074
| Elapsed Time: 224.6506 s | Max GPU Memory Alloc: 1960.8457 MB
Epoch [2/5]------------------------------------------------------------------------
| Train Loss: 2.0209 | Train Acc: 0.4719 | Val Loss: 1.7971, Val Acc: 0.5216
| Elapsed Time: 222.2073 s | Max GPU Memory Alloc: 1960.8472 MB
Epoch [3/5]------------------------------------------------------------------------
| Train Loss: 1.8274 | Train Acc: 0.5141 | Val Loss: 1.6481, Val Acc: 0.5436
| Elapsed Time: 222.1483 s | Max GPU Memory Alloc: 1960.8481 MB
Epoch [4/5]------------------------------------------------------------------------
| Train Loss: 1.2209 | Train Acc: 0.6451 | Val Loss: 1.2316, Val Acc: 0.6412
| Elapsed Time: 222.4511 s | Max GPU Memory Alloc: 1960.8491 MB
Epoch [5/5]------------------------------------------------------------------------
| Train Loss: 1.2085 | Train Acc: 0.6483 | Val Loss: 1.2196, Val Acc: 0.6451
| Elapsed Time: 221.2881 s | Max GPU Memory Alloc: 1960.8501 MB

total_time: 1113.5023527145386
best_valid_acc: 0.6450603232415206
```

Warning

```
local/lib/python3.9/site-packages/torch/utils/data/dataloader.py:557: UserWarning: This DataLoader will cre
ate 8 worker processes in total. Our suggested max number of worker in current system is 4, which is smaller than what this
DataLoader is going to create. Please be aware that excessive worker creation might get DataLoader running slow or even fre
eze, lower the worker number to avoid potential slowness/freeze if necessary.
```

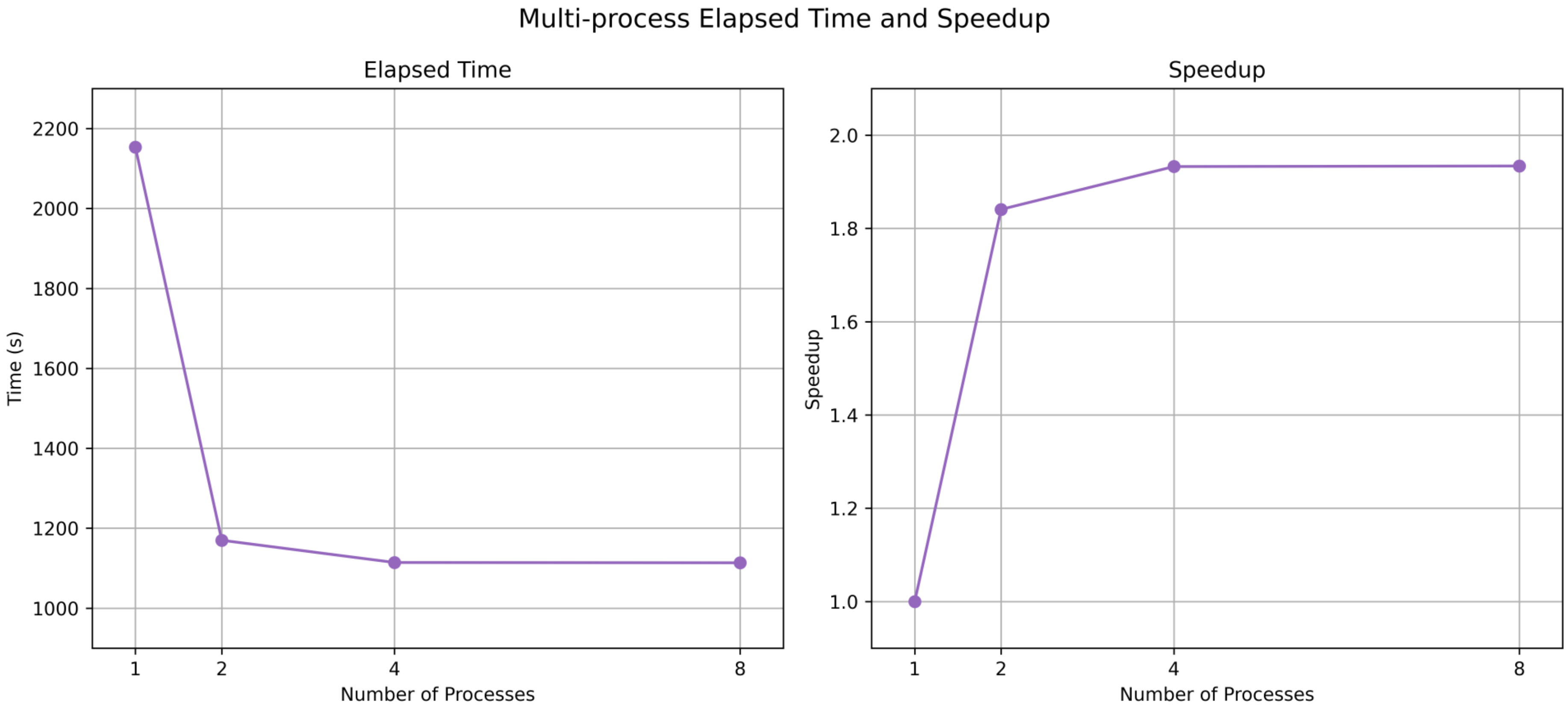
4 Result & Analysis

4.2 Pre-training

Using Multi-process

Count of Process	Elapsed time (s)	Speedup
Serial (1 process)	2153.4935	1
2 processes	1169.8231	1.8409
4 processes	1114.2052	1.9328
8 processes	1113.5024	1.9340

Comparison



4 Result & Analysis

4.2 Pre-training

Using Multi-thread (DataParallel)

Parallel Mechanism

Distribute the input data and the model across multiple GPUs → Compute the forward and backward passes on each GPU → Aggregate the results to update the model parameters

Before we dive in, let's clarify why, despite the added complexity, you would consider using `DistributedDataParallel` over `DataParallel`:

- First, `DataParallel` is single-process, multi-thread, and only works on a single machine, while `DistributedDataParallel` is multi-process and works for both single- and multi- machine training. `DataParallel` is usually slower than `DistributedDataParallel` even on a single machine due to GIL contention across threads, per-

4 Result & Analysis

4.2 Pre-training

Using Multi-thread (DataParallel)

Hardware

CPU
Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz
Architecture: X86_64
Count: 4

GPU
Tesla T4
Count: 2, 4(expected)

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.161.03   Driver Version: 470.161.03   CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   46C    P8    10W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:00:05.0 Off |                    0 |
| N/A   43C    P8    10W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

4 Result & Analysis

4.2 Pre-training

Using Multi-thread (DataParallel)

GPU
2

Progress

```
Running DataParralel on 2 GPUs----------------------------------------
Epoch [1/5]-----------------------------------------------------------
| Train Loss: 2.7000 | Train Acc: 0.3273 | Val Loss: 2.3335, Val Acc: 0.3904
| Elapsed Time: 335.0122 s | Max GPU Memory Alloc: 1079.4219 MB
Epoch [2/5]-----------------------------------------------------------
| Train Loss: 2.0822 | Train Acc: 0.4565 | Val Loss: 1.8347, Val Acc: 0.5254
| Elapsed Time: 321.0952 s | Max GPU Memory Alloc: 1079.4233 MB
Epoch [3/5]-----------------------------------------------------------
| Train Loss: 1.9114 | Train Acc: 0.4969 | Val Loss: 1.7686, Val Acc: 0.5411
| Elapsed Time: 319.5789 s | Max GPU Memory Alloc: 1079.4243 MB
Epoch [4/5]-----------------------------------------------------------
| Train Loss: 1.2854 | Train Acc: 0.6260 | Val Loss: 1.2389, Val Acc: 0.6403
| Elapsed Time: 324.5508 s | Max GPU Memory Alloc: 1079.4253 MB
Epoch [5/5]-----------------------------------------------------------
| Train Loss: 1.2701 | Train Acc: 0.6299 | Val Loss: 1.2242, Val Acc: 0.6469
| Elapsed Time: 329.6849 s | Max GPU Memory Alloc: 1079.4263 MB

total_time: 1630.4430594444275
best_valid_acc: 0.6469383109492375
```

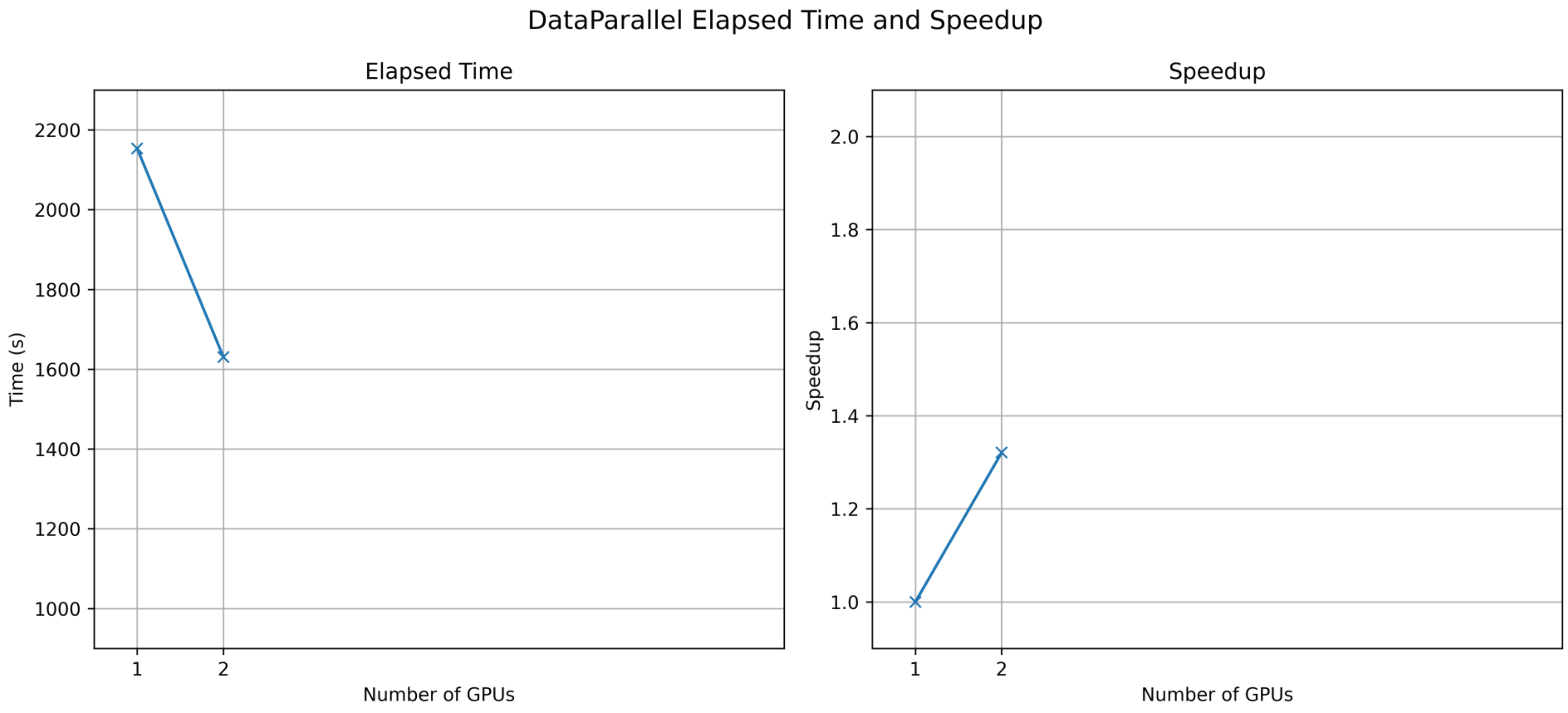
4 Result & Analysis

4.2 Pre-training

Using Multi-thread (DataParallel)

Count of GPU	Elapsed time (s)	Speedup
Serial (1 GPU)	2153.4935	1
2 GPU	1630.4431	1.3208
4 GPU	-	-

Comparison



4 Result & Analysis

4.2 Pre-training

Comparison Summary

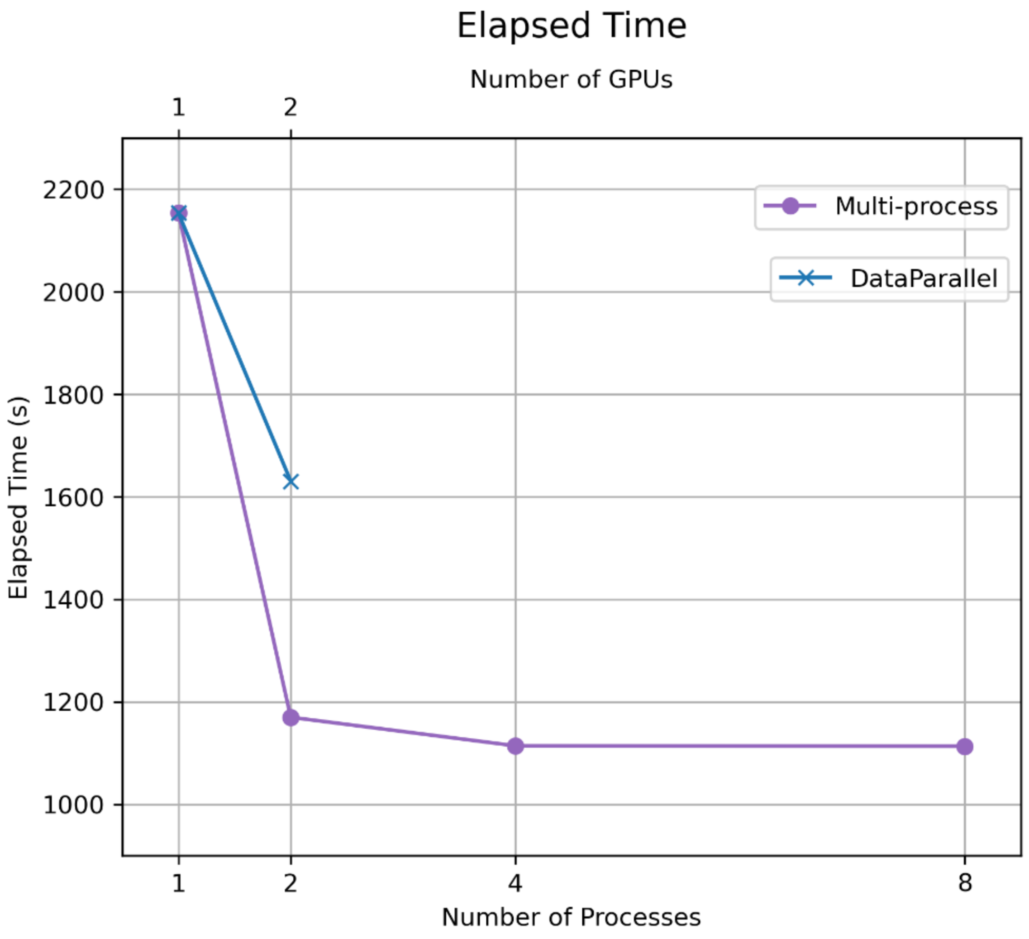
Training	Resources	Elapsed Time (s)	Speedup
Serial	Main process	2153.4935	1
Multi-process (workers)	2 processes	1169.8231	1.8409
	4 processes	1114.2052	1.9328
	8 processes	1113.5024	1.9340
Multi-thread (DataParallel)	2 GPUs	1630.4431	1.3208
	4 GPUs	-	-

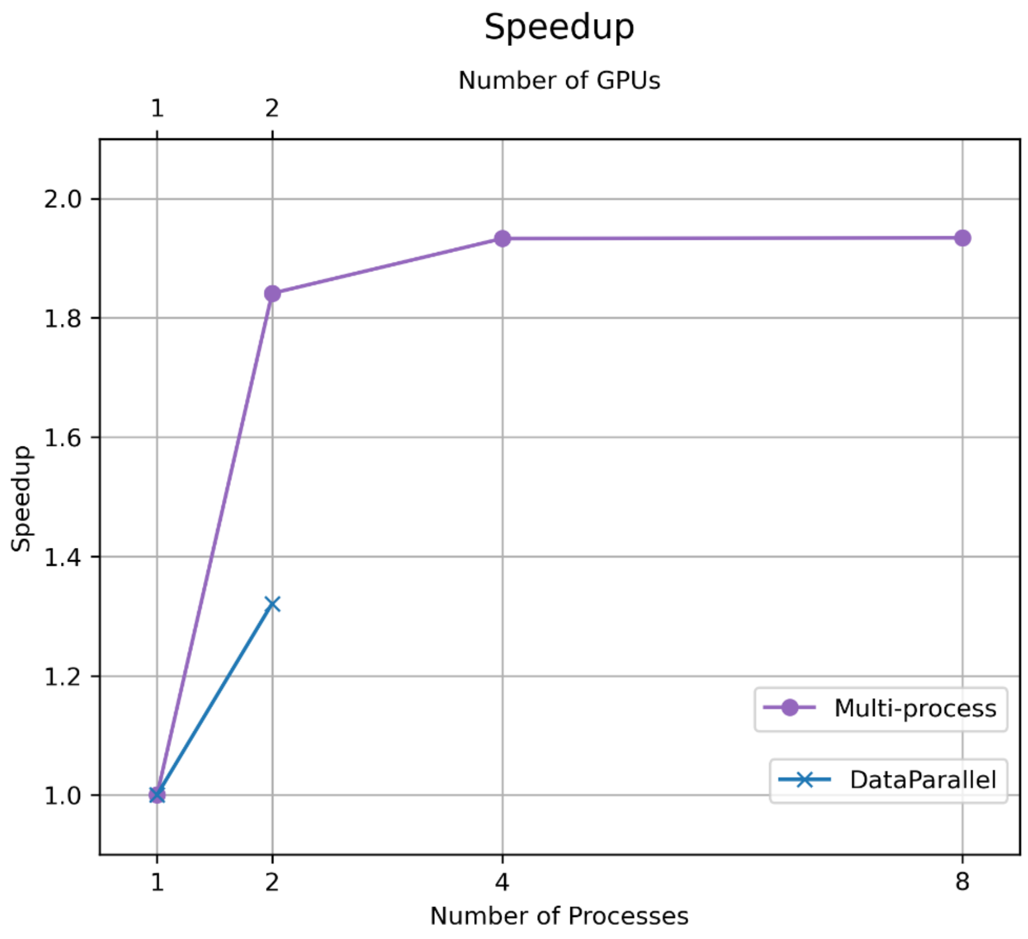
DistributedDataParallel

4 Result & Analysis

4.2 Pre-training

Comparison Summary





4 Result & Analysis

4.3 Hyperparameter Tuning

Parallel Mechanism

Optuna study
Apply parallelism using
Joblib threading

```python
import joblib
from joblib import delayed
from joblib import Parallel
```

```python
with Parallel(n_jobs=n_jobs, prefer="threads") as parallel:
    if not isinstance(
        parallel._backend, joblib.parallel.ThreadingBackend
    ) and isinstance(self._storage, storages.InMemoryStorage):
        msg = (
            "The default storage cannot be shared by multiple processes. "
            "Please use an RDB (RDBStorage) when you use joblib for "
            "multi-processing. The usage of RDBStorage can be found in "
            "https://optuna.readthedocs.io/en/stable/tutorial/rdb.html."
```

Hyperparameters we tuned:

- learning rate
- step size
- weight decay
- momentum

4 Result & Analysis

4.3 Hyperparameter Tuning

Hardware

CPU
AMD EPYC 7543 32-Core Processor
Architecture: X86_64
Count: 4

GPU
A100-SXM4-80GB
Count: 1

```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA A100-SXM4-80GB          Off | 00000000:81:00.0 Off |                    0 |
| N/A   39C    P0              59W / 500W |      4MiB / 81920MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

4 Result & Analysis

4.3 Hyperparameter Tuning

thread 1

```
Trial 0 starts-----------------------------------
[I 2023-12-13 02:49:02,477] Trial 0 finished
Trial 1 starts-----------------------------------
[I 2023-12-13 02:52:21,330] Trial 1 finished
Trial 2 starts-----------------------------------
[I 2023-12-13 02:55:39,138] Trial 2 finished
Trial 3 starts-----------------------------------
[I 2023-12-13 02:59:02,808] Trial 3 finished
Trial 4 starts-----------------------------------
[I 2023-12-13 03:02:25,692] Trial 4 finished
Trial 5 starts-----------------------------------
[I 2023-12-13 03:05:50,770] Trial 5 finished
Trial 6 starts-----------------------------------
[I 2023-12-13 03:09:11,952] Trial 6 finished
Trial 7 starts-----------------------------------
[I 2023-12-13 03:12:33,509] Trial 7 finished
Elapsed time: 1633.419298171997 s
```

threads 2

```
Trial 9 starts-----------------------------------
Trial 8 starts-----------------------------------
[I 2023-12-13 03:16:45,775] Trial 9 finished
[I 2023-12-13 03:16:45,776] Trial 8 finished
Trial 10 starts----------------------------------
Trial 11 starts----------------------------------
[I 2023-12-13 03:20:42,194] Trial 11 finished
[I 2023-12-13 03:20:42,197] Trial 10 finished
Trial 12 starts----------------------------------
Trial 13 starts----------------------------------
[I 2023-12-13 03:24:35,779] Trial 12 finished
[I 2023-12-13 03:24:35,780] Trial 13 finished
Trial 14 starts----------------------------------
Trial 15 starts----------------------------------
[I 2023-12-13 03:28:37,332] Trial 15 finished
[I 2023-12-13 03:28:37,334] Trial 14 finished
Elapsed time: 954.1966986656189 s
```

4 Result & Analysis

4.3 Hyperparameter Tuning

threads 4

```
Trial 17 starts--------------------------------------
Trial 16 starts--------------------------------------
Trial 18 starts--------------------------------------
Trial 19 starts--------------------------------------
[I 2023-12-13 03:33:46,613] Trial 16 finished
[I 2023-12-13 03:33:46,616] Trial 18 finished
[I 2023-12-13 03:33:46,617] Trial 19 finished
[I 2023-12-13 03:33:46,617] Trial 17 finished
Trial 20 starts--------------------------------------
Trial 21 starts--------------------------------------
Trial 22 starts--------------------------------------
Trial 23 starts--------------------------------------
[I 2023-12-13 03:38:36,570] Trial 23 finished
[I 2023-12-13 03:38:36,570] Trial 21 finished
[I 2023-12-13 03:38:36,571] Trial 22 finished
[I 2023-12-13 03:38:36,572] Trial 20 finished
Elapsed time: 591.6575014591217 s
```

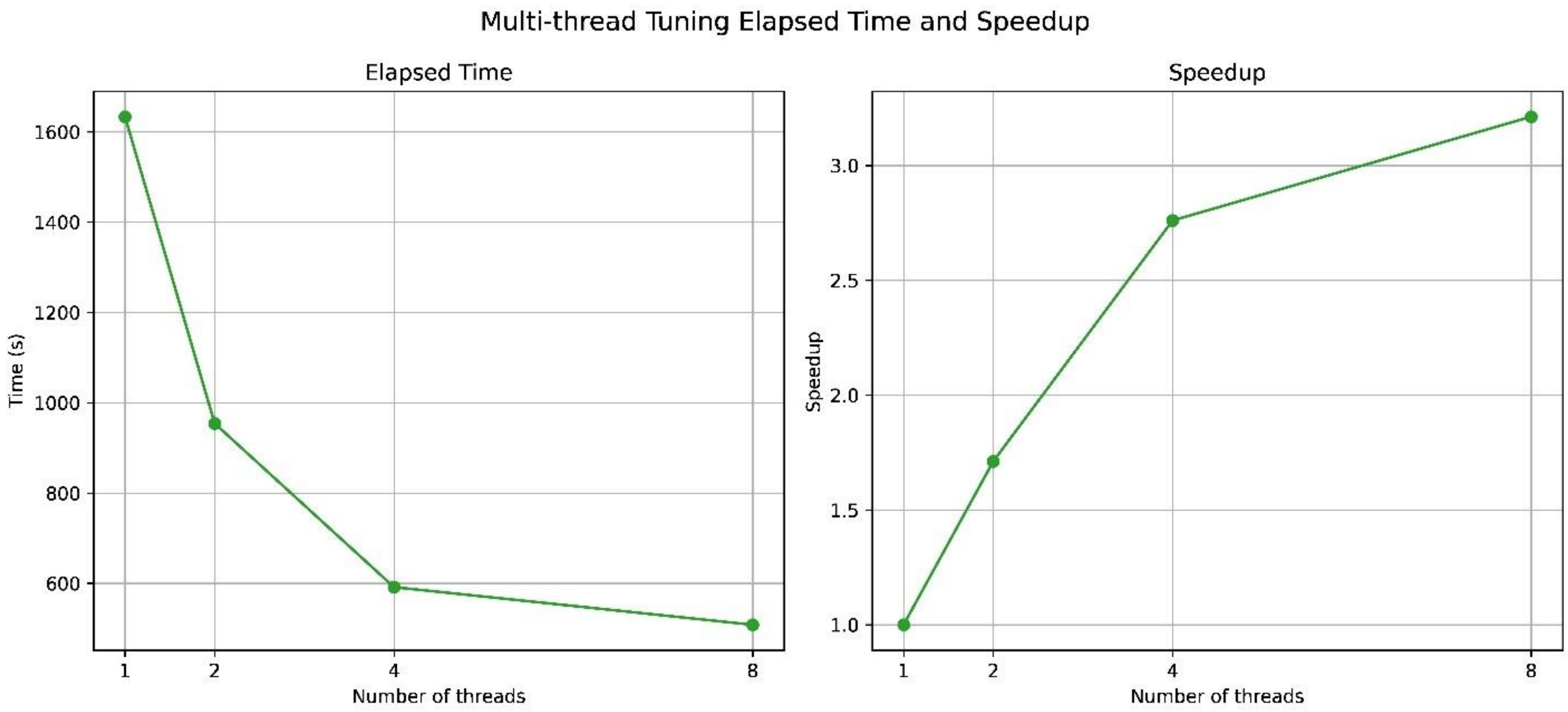
threads 8

```
Trial 25 starts--------------------------------------
Trial 26 starts--------------------------------------
Trial 24 starts--------------------------------------
Trial 27 starts--------------------------------------
Trial 28 starts--------------------------------------
Trial 29 starts--------------------------------------
Trial 30 starts--------------------------------------
Trial 31 starts--------------------------------------
[I 2023-12-13 03:47:09,112] Trial 28 finished
[I 2023-12-13 03:47:09,113] Trial 31 finished
[I 2023-12-13 03:47:09,115] Trial 30 finished
[I 2023-12-13 03:47:09,116] Trial 29 finished
[I 2023-12-13 03:47:09,743] Trial 25 finished
[I 2023-12-13 03:47:12,015] Trial 27 finished
[I 2023-12-13 03:47:12,171] Trial 24 finished
[I 2023-12-13 03:47:12,826] Trial 26 finished
Elapsed time: 508.36329221725464 s
```

4 Result & Analysis

4.3 Hyperparameter Tuning

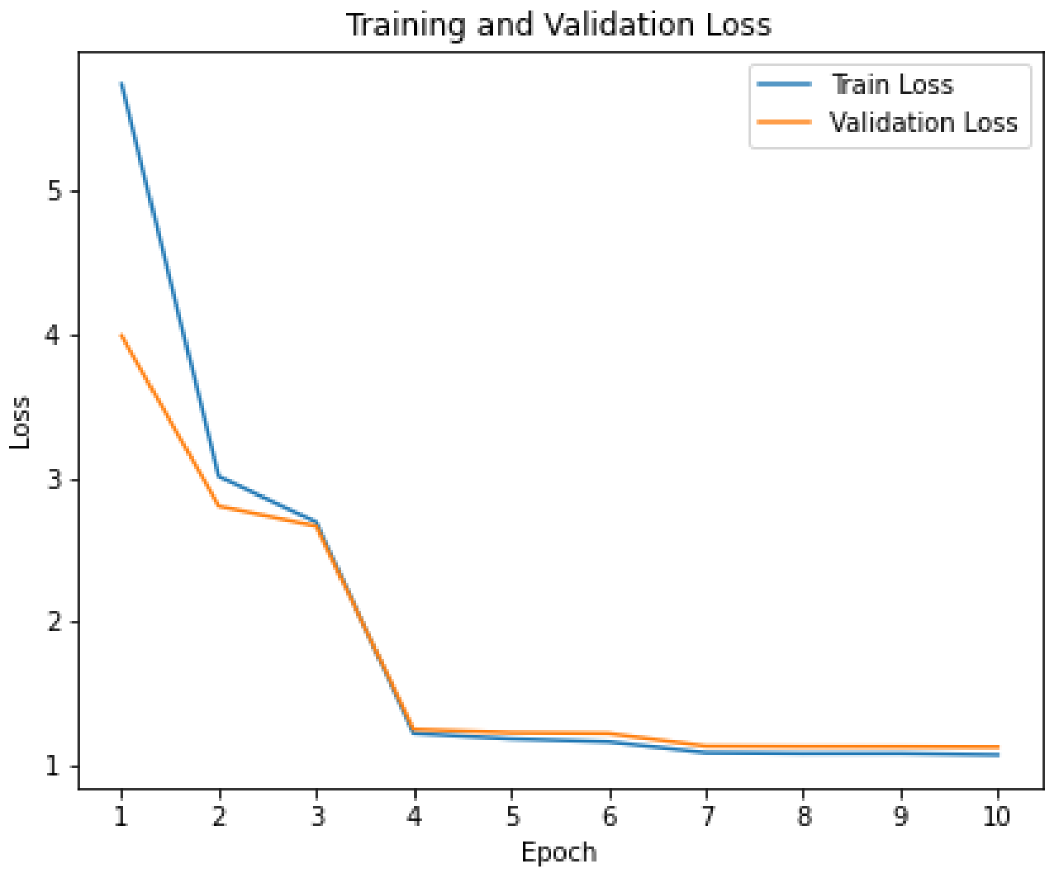
Count of Threads	Total Elapsed time (s)	speedup
Serial (1 thread)	1633.4193	1
2 threads	954.1967	1.7118
4 threads	591.6575	2.7608
8 threads	508.3633	3.2131

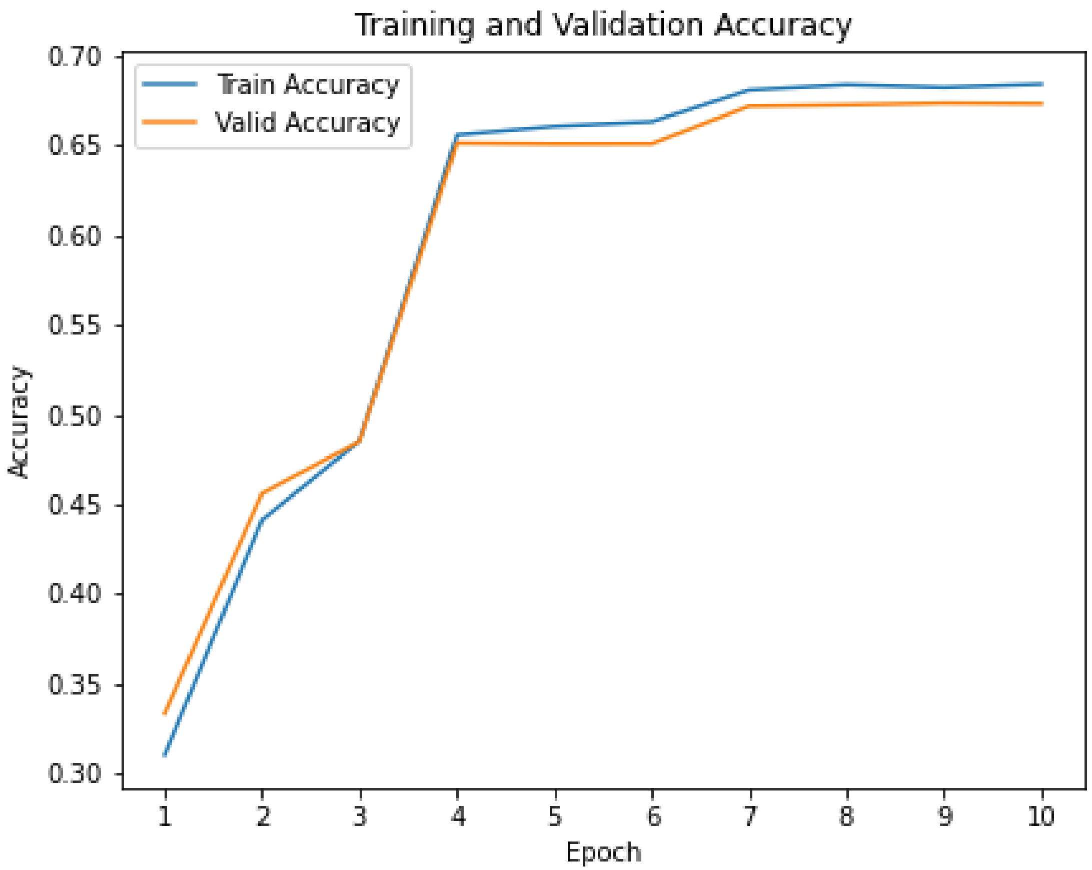


4 Result & Analysis

4.4 Final Training & Model Evaluation

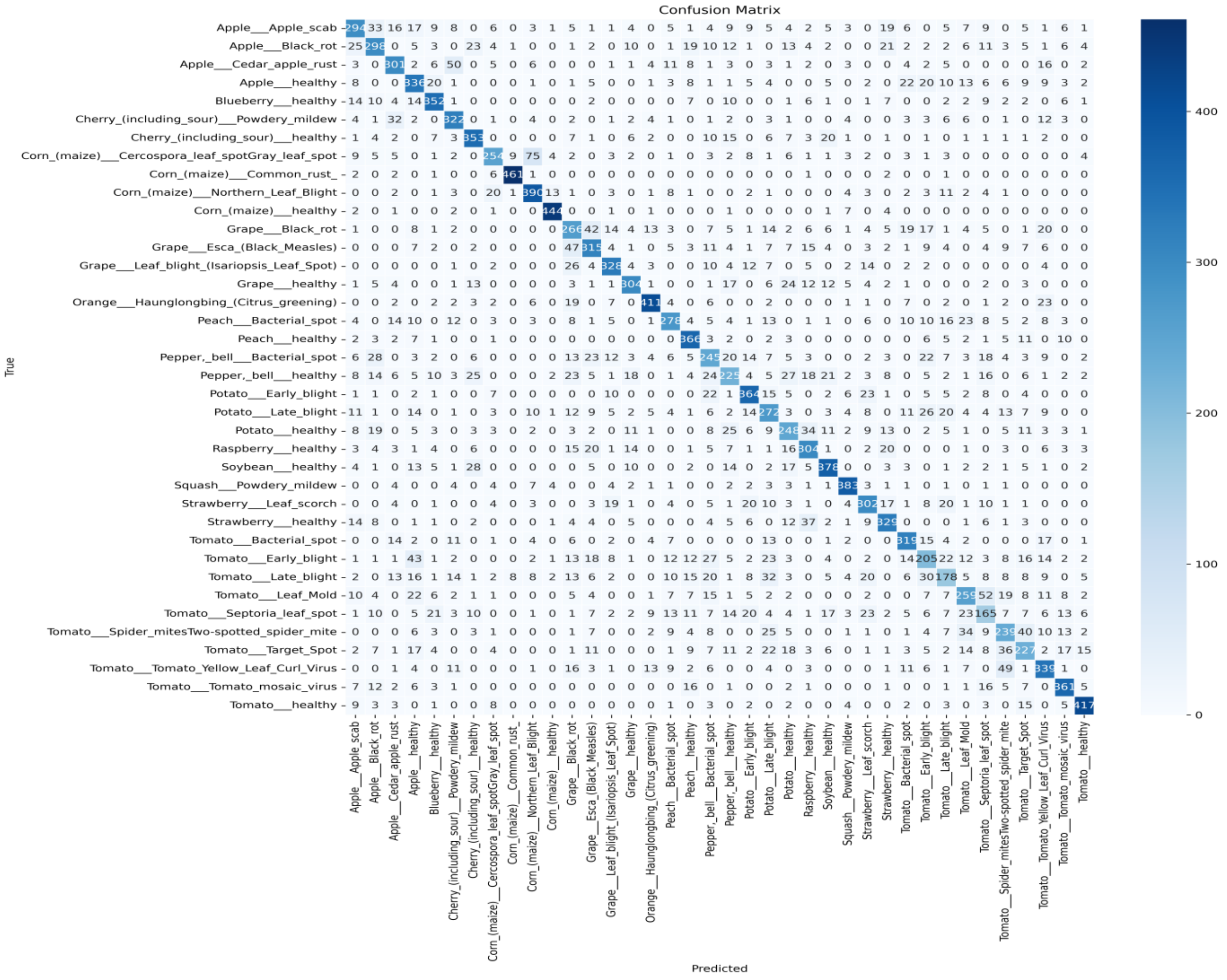
Training acc: 68.43%
Validation acc: 67.36%





4 Result & Analysis

4.4 Final Training & Model Evaluation



	precision	recall	f1-score	support
Apple___Apple_scab	0.64	0.58	0.61	504
Apple___Black_rot	0.63	0.60	0.62	497
Apple___Cedar_apple_rust	0.69	0.68	0.68	440
Apple___healthy	0.59	0.67	0.63	502
Blueberry___healthy	0.74	0.78	0.76	454
Cherry_(including_sour)___Powdery_mildew	0.70	0.76	0.73	421
Cherry_(including_sour)___healthy	0.74	0.77	0.76	456
Corn_(maize)___Cercospora_leaf_spotGray_leaf_spot	0.74	0.62	0.67	410
Corn_(maize)___Common_rust_	0.96	0.97	0.96	477
Corn_(maize)___Northern_Leaf_Blight	0.74	0.82	0.78	477
Corn_(maize)___healthy	0.94	0.95	0.95	465
Grape___Black_rot	0.52	0.56	0.54	472
Grape___Esca_(Black_Measles)	0.63	0.66	0.64	480
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	0.75	0.76	0.76	430
Grape___healthy	0.75	0.72	0.73	423
Orange___Haunglongbing_(Citrus_greening)	0.85	0.82	0.83	503
Peach___Bacterial_spot	0.69	0.61	0.64	459
Peach___healthy	0.72	0.85	0.78	432
Pepper,_bell___Bacterial_spot	0.51	0.51	0.51	478
Pepper,_bell___healthy	0.53	0.45	0.49	497
Potato___Early_blight	0.71	0.75	0.73	485
Potato___Late_blight	0.52	0.56	0.54	485
Potato___healthy	0.55	0.54	0.55	456
Raspberry___healthy	0.65	0.68	0.67	445
Soybean___healthy	0.74	0.75	0.74	505
Squash___Powdery_mildew	0.85	0.88	0.87	434
Strawberry___Leaf_scorch	0.67	0.68	0.68	444
Strawberry___healthy	0.71	0.72	0.72	456
Tomato___Bacterial_spot	0.69	0.75	0.72	425
Tomato___Early_blight	0.49	0.43	0.45	480
Tomato___Late_blight	0.49	0.38	0.43	463
Tomato___Leaf_Mold	0.59	0.55	0.57	470
Tomato___Septoria_leaf_spot	0.42	0.38	0.40	436
Tomato___Spider_mitesTwo-spotted_spider_mite	0.55	0.55	0.55	435
Tomato___Target_Spot	0.55	0.50	0.52	457
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.63	0.69	0.66	490
Tomato___Tomato_mosaic_virus	0.78	0.81	0.79	448
Tomato___healthy	0.87	0.87	0.87	481

Test acc: 67.33%

5 Conclusion

In data preprocessing

- Both frameworks effectively harness parallel processing to expedite computations and enhance overall efficiency.
- Compared to Dask, PyTorch exhibits a more substantial time reduction compared to Dask during data preprocessing tasks.

In model training

- Utilizing multiprocessing yields speedup improvements, but diminishing returns occur when processes exceed CPU core count.
- Increased GPU count with DataParallel accelerates training times, though multiprocessing outperforms in terms of performance.

In hyperparameters tuning

- Optuna demonstrates notable speedup improvements in hyperparameter tuning with an increasing number of threads.

References

- Gula, L. T. (2023, February 6). Researchers helping protect crops from pests. National Institute of Food and Agriculture. https://www.nifa.usda.gov/about-nifa/blogs/researchers-helping-protect-crops-pests
- Ingle, A. (2020, December 15). Plant disease classification - resnet-99.2%. Kaggle. https://www.kaggle.com/code/atharvaingle/plant-disease-classification-resnet-99-2
- Brownlee, J. (2019, July 5). How to manually scale image pixel data for deep learning. MachineLearningMastery. https://machinelearningmastery.com/how-to-manually-scale-image-pixel-data-for-deep-learning/
- Schmitt, M. (n.d.). Understanding dask architecture: Client, scheduler, workers. Datarevenue. https://www.datarevenue.com/en-blog/understanding-dask-architecture-client-scheduler-workers.
- Modi, R. (2022, February 25). ResNet - understand and implement from scratch. Medium. https://medium.com/analytics-vidhya/resnet-understand-and-implement-from-scratch-d0eb9725e0db
- Datasets & dataloaders. Datasets & DataLoaders - PyTorch Tutorials 2.1.1+cu121 documentation. (n.d.). https://pytorch.org/tutorials/beginner/basics/data_tutorial.html
- Zhu, J. (n.d.). Getting started with distributed data parallel. Getting Started with Distributed Data Parallel - PyTorch Tutorials 2.2.0+cu121 documentation. https://pytorch.org/tutorials/intermediate/ddp_tutorial.html