PHOENIX 13

DASHBOARD LECTURE GAMIFY ASSIGNMENT STUDENTS INISGHTS

Learners'Ed

Open Innovation in Education

Intel oneAPI (oneDNN & OpenMP)

Rajat Kulshrestha (Mentor)
Saumya Srivastava (Team Leader)
Sujal Kulshrestha

oneAPI <HACK>ATHON

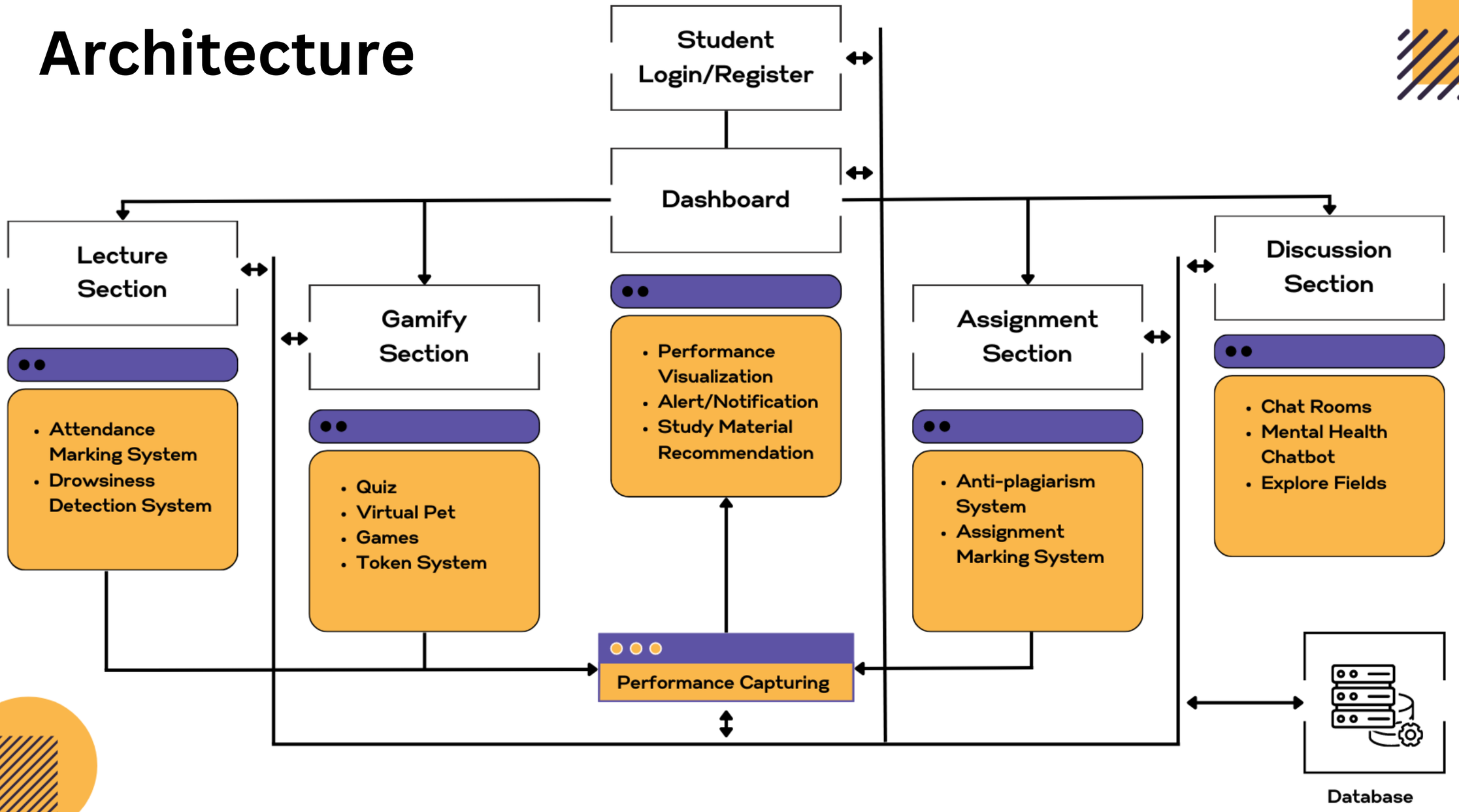
Problem Statement

Quality education is the path to transcendence. However during the recent difficult times, one of the fields that suffered a significant setback was education. There was a loss of peer learning and interactions, healthy competition, usage of unfair means for personal gains, and mental health challenges. In order to prevent only such situations in the wake of a similar event and indeed bring ease to the world of online education we introduce our application **Learners'Ed**

The problems that we intend to solve

- **Integrated Platform**
- **Distant Learning (from remote institute/coaching)**
- **Awareness among the students**
- **Usage of unfair means on online platforms**
- **Active participation in online lectures**
- **Enriches learning through fun gesture-based quizzes**
- **Creating a healthy competitive environment through educational games**
- **Taking care of the mental health of the students**

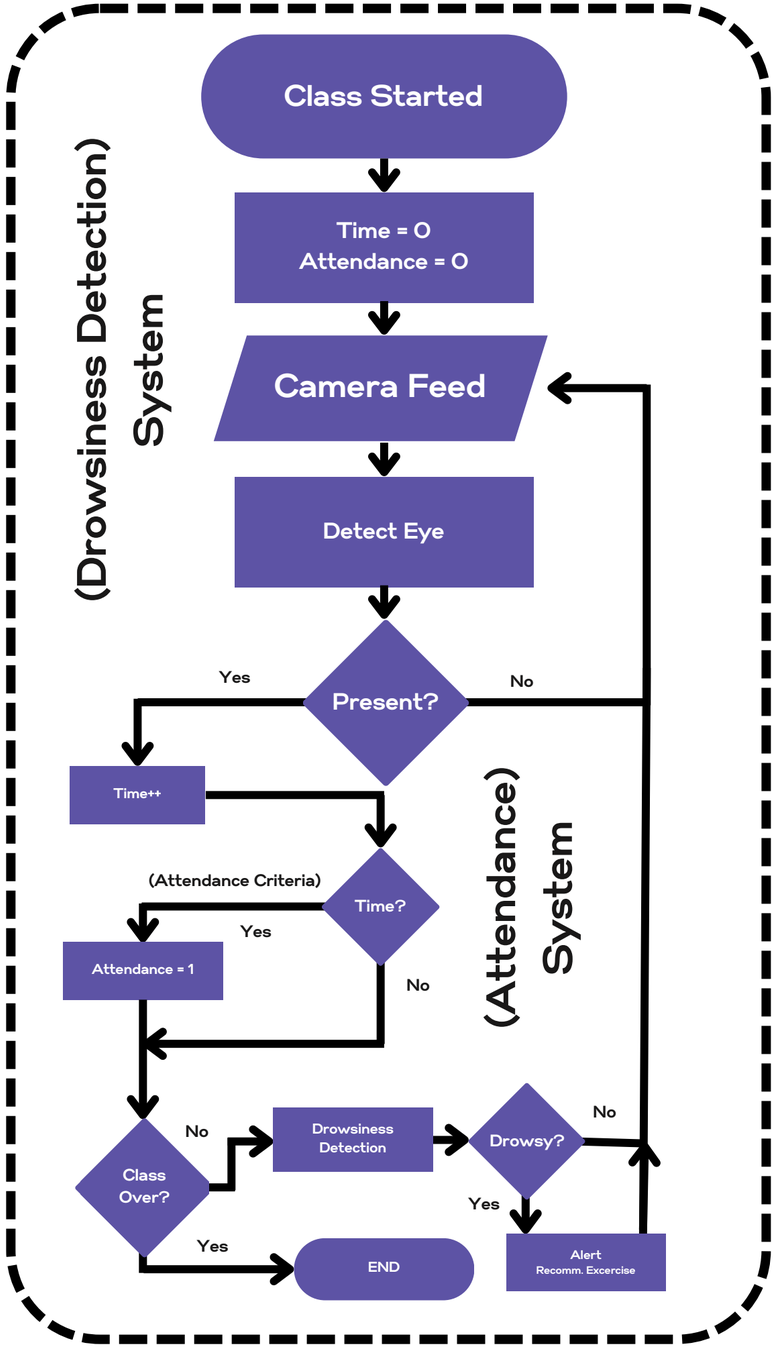
Architecture



Learners'Ed

Lectures

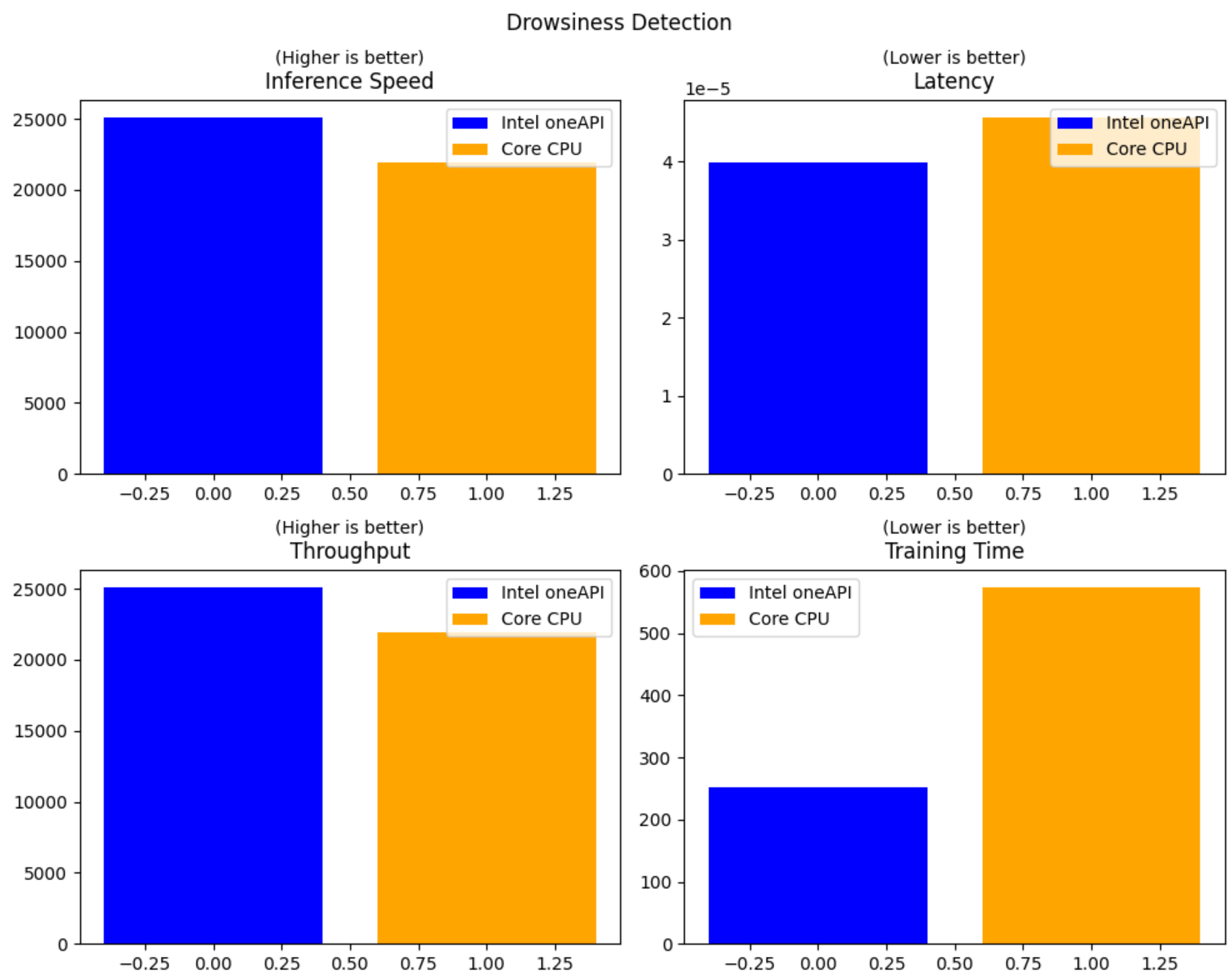
		Watch Time	Lecture Time		Status
Lecture 1	Lorem ipsum dolor, sit amet consectetur adipisicing elit.	29:38	1:32:12	Watch	✗
Lecture 2	Lorem ipsum dolor, sit amet consectetur adipisicing elit.	29:38	1:32:12	Watch	✗
Lecture 3	Lorem ipsum dolor, sit amet consectetur adipisicing elit.	29:38	1:32:12	Watch	✗



Drowsiness Detection System

Tracks students' active listening time during uploaded lectures to determine attendance.

The first model we have developed is the Drowsiness Detection Model, which is utilized in the Lecture Section. By leveraging Intel oneDNN and OpenMP optimization, we have significantly accelerated the training process of this model. This optimization framework has not only expedited the training phase but has also improved the overall inference speed during real-time drowsiness detection. As a result, students' attendance can be accurately determined based on their active listening time during lectures.



OpenMP Params used

inter: 6

intra: 6

KMP_BLOCKTIME: 1

Test_Set: 25

Benchmarks Rates

Inference Time Rate: 1.14

Latency Rate: 0.87

Throughput Rate: 1.14

Training Time Rate: 0.43

With oneAPI

Inference Speed: 25085.55

Latency: 3.99E-05

Throughput: 24285.65

Training Time: 251.31s

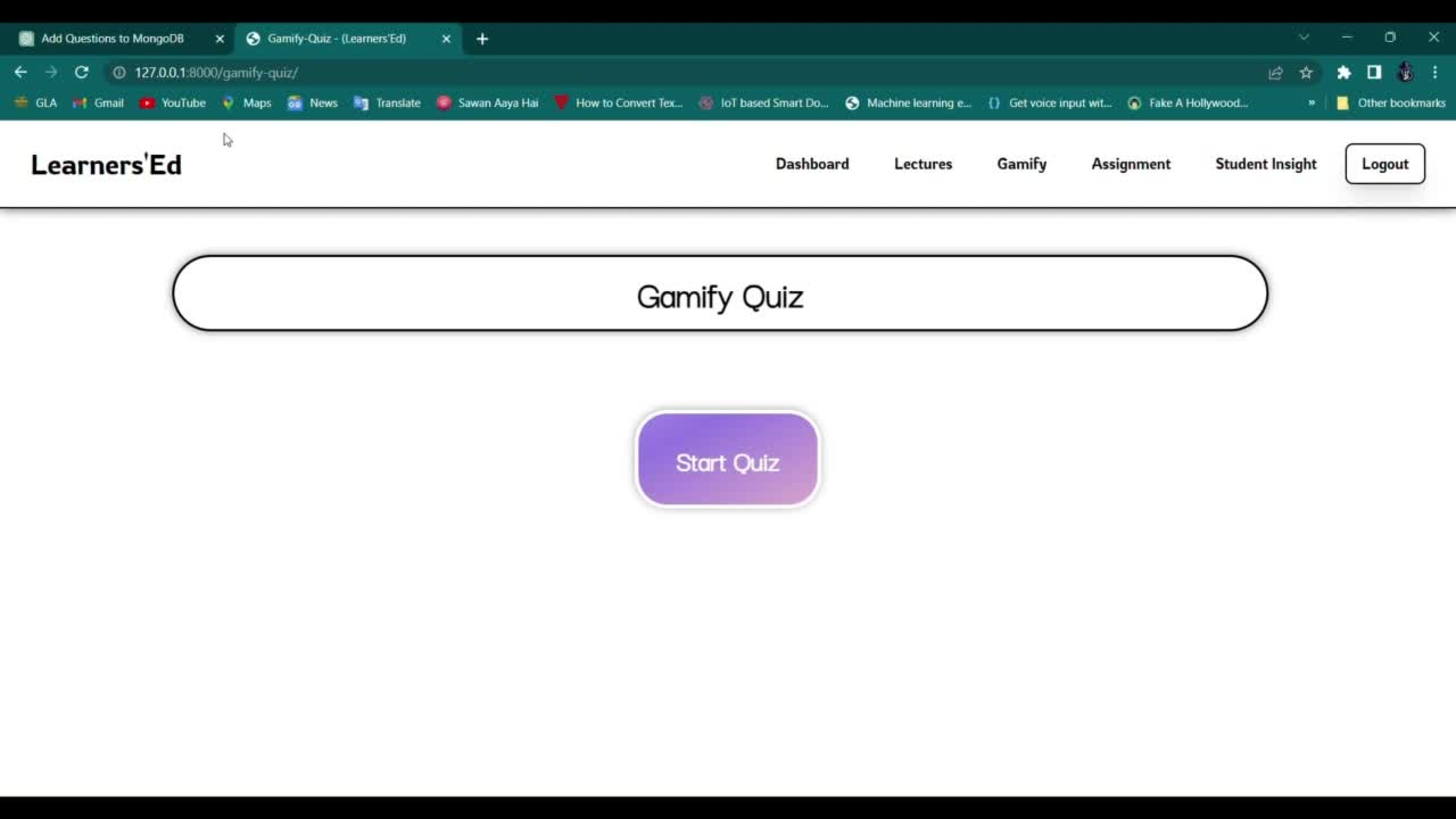
Without oneAPI

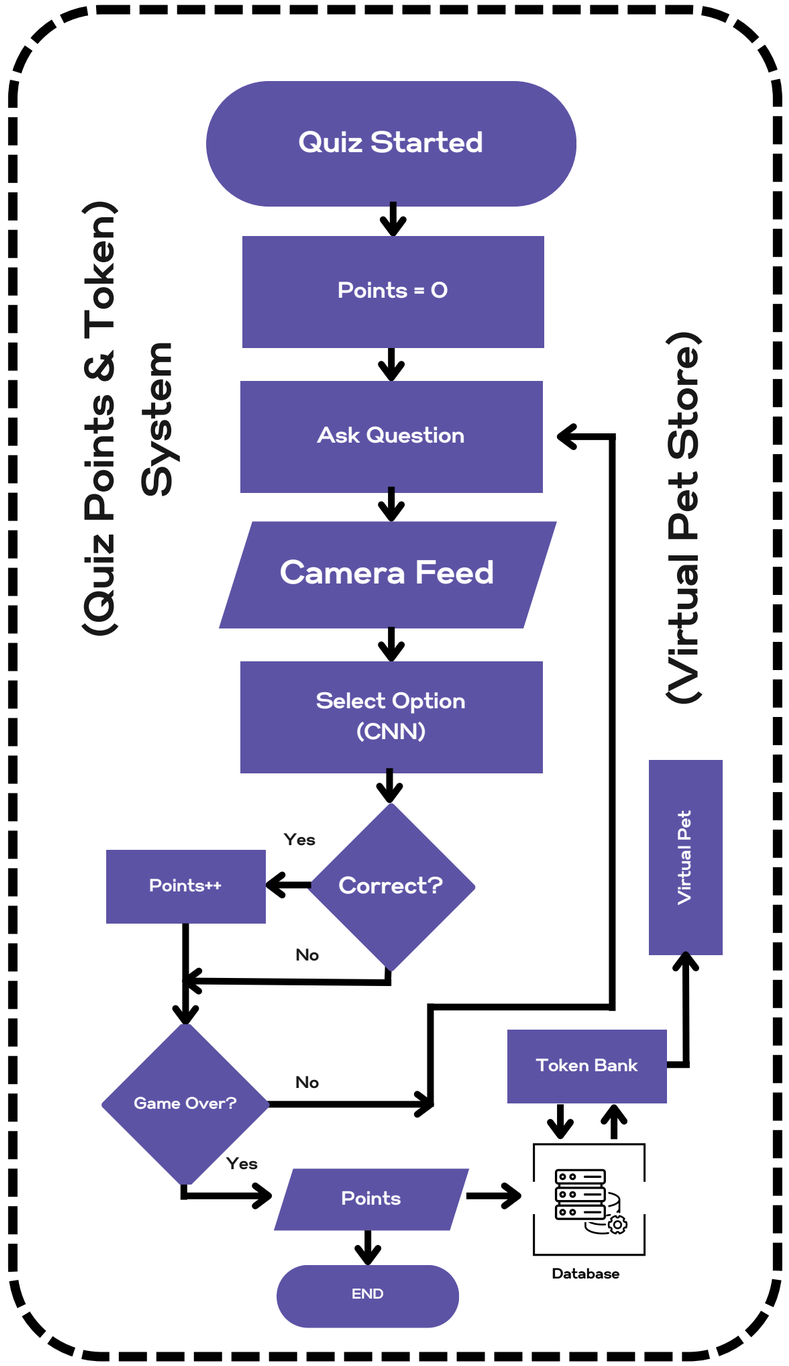
Inference Speed: 21927.56

Latency: 4.56E-05

Throughput: 21927.56

Training Time: 573.34s

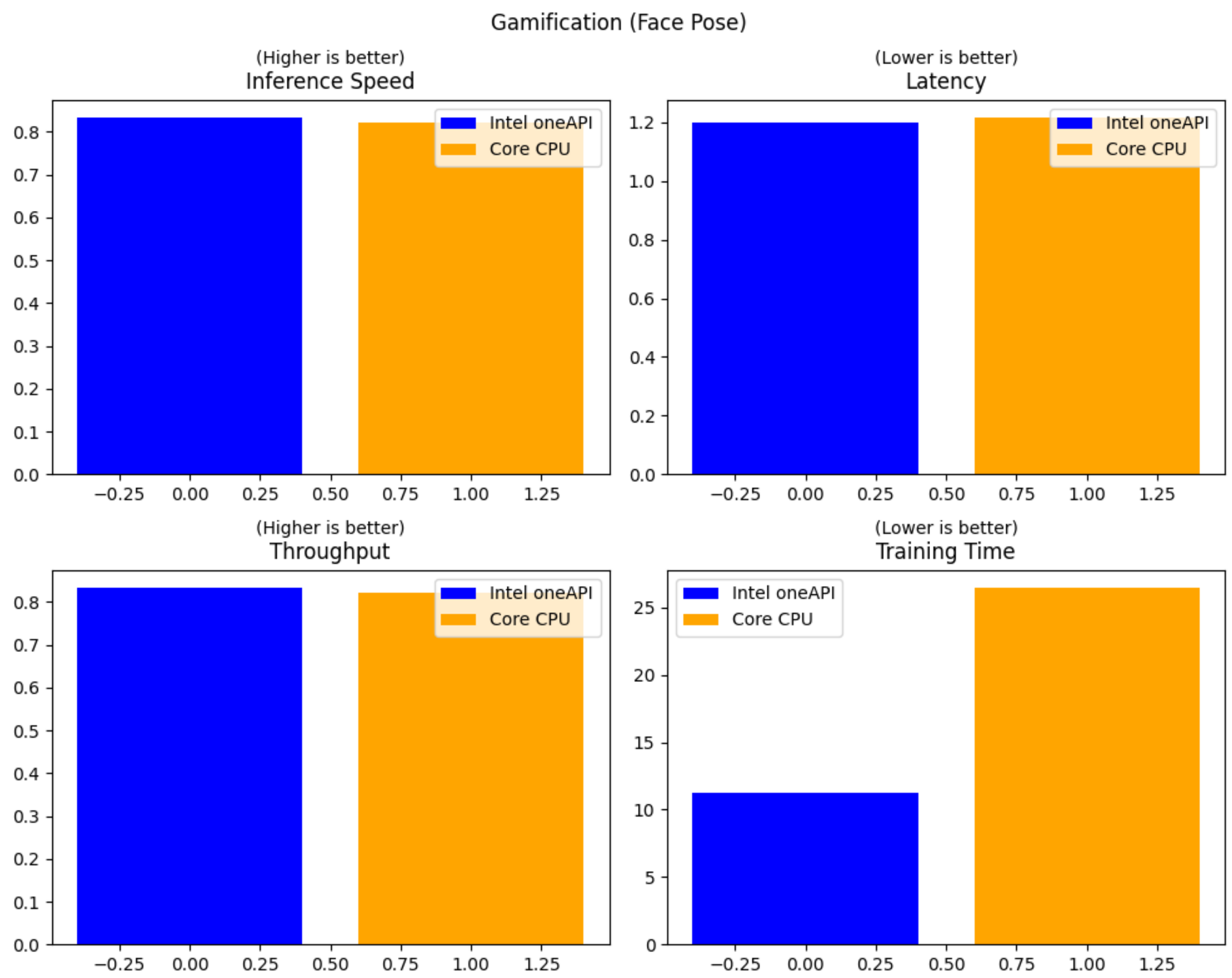

Add Questions to MongoDB
Gamify-Quiz - (Learners'Ed)
127.0.0.1:8000/gamify-quiz/
GLA
Gmail
YouTube
Maps
News
Translate
Sawan Aaya Hai
How to Convert Tex...
IoT based Smart Do...
Machine learning e...
Get voice input wit...
Fake A Hollywood...
Other bookmarks
Learners'Ed
Dashboard
Lectures
Gamify
Assignment
Student Insight
Logout
Gamify Quiz
Start Quiz



Face Pose Detection

Gamify Section: Offers interactive learning experiences, including a quiz system that uses facial movements and gestures for answering questions. Students can also choose and name virtual pets, earn Learners'Ed coins, and improve pet ranks by participating in quizzes.

The second model we have incorporated is the Face Pose Estimation Model, which plays a crucial role in the Gamify Quiz section. Through the implementation of Intel oneDNN and OpenMP optimization techniques, we have achieved remarkable improvements in training efficiency, throughput, and inference speed. These optimizations have enabled us to accurately estimate facial movements and gestures during the quiz, providing an engaging and interactive learning experience for students.



OpenMP Params used

inter: 2

intra: 6

KMP_BLOCKTIME: 0

Test_Set: 25

Benchmarks Rates

Inference Time Rate: 1.013

Latency Rate: 0.98

Throughput Rate: 1.01

Training Time Rate: 0.42

With oneAPI

Inference Speed: 0.83

Latency: 1.19

Throughput: 0.83

Training Time: 11.26s

Without oneAPI

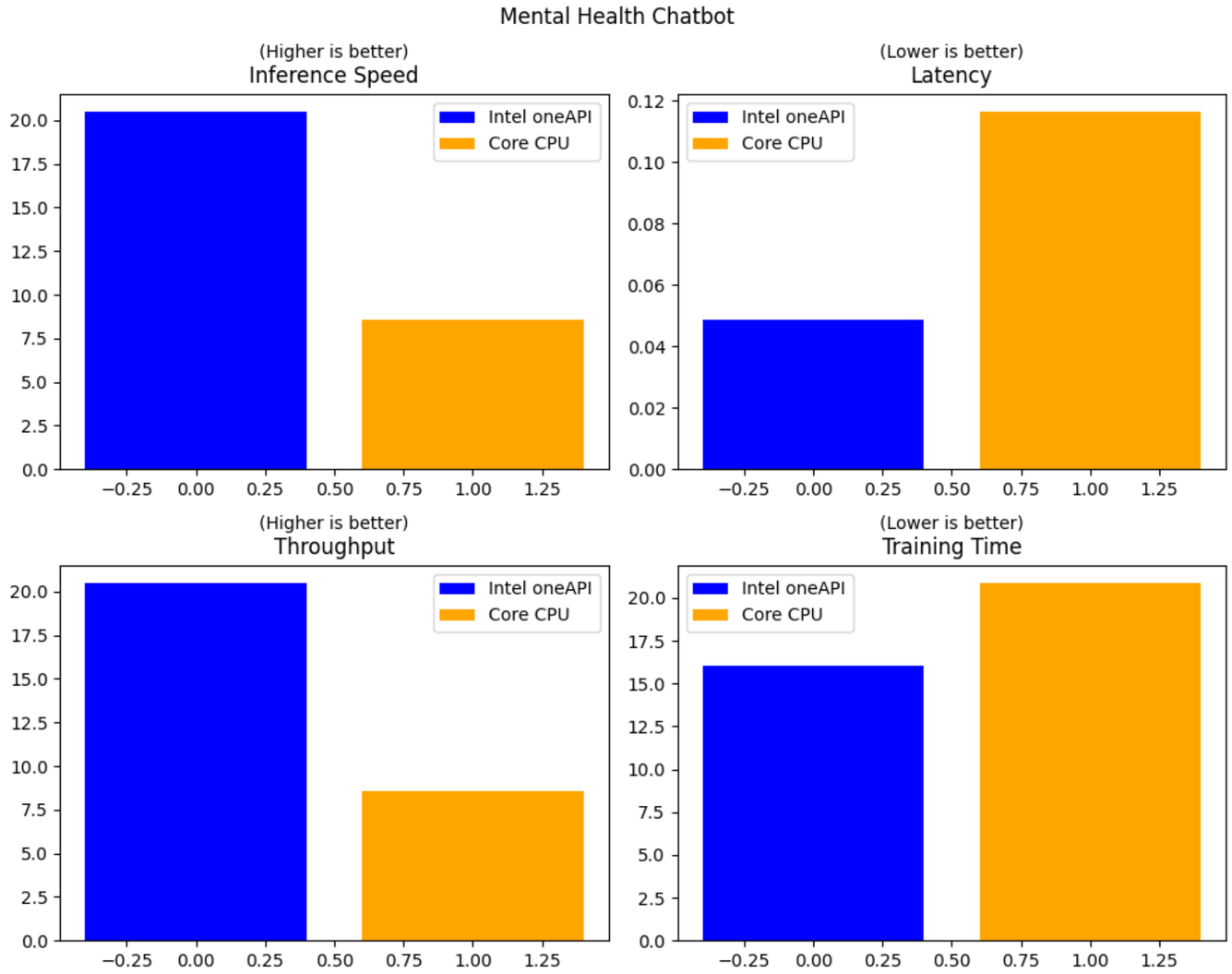
Inference Speed: 0.82

Latency: 3.99E-05

Throughput: 0.82

Training Time: 26.46s

Mental Health Chatbot



OpenMP Params used
inter: 2
intra: 6
KMP_BLOCKTIME: 1
Test_Set: 20

Benchmarks Rates
Inference Time Rate: 2.38
Latency Rate: 0.41
Throughput Rate: 2.38
Training Time Rate: 0.77

With oneAPI
Inference Speed: 20.48
Latency: 0.04
Throughput: 20.44
Training Time: 16.07s

Without oneAPI
Inference Speed: 8.59
Latency: 0.11
Throughput: 8.59
Training Time: 20.86s

Lastly, we have developed a sophisticated Mental Health Chatbot that leverages **Intel oneDNN and OpenMP optimization**. This optimization framework has significantly enhanced the performance of our chatbot, resulting in faster response times, improved throughput, and reduced latency. By employing Intel's optimization tools, such as **scikit-learn-intelex**, we have been able to train the chatbot model efficiently and deliver prompt and insightful responses to students seeking support.

Core components of oneAPI AI Toolkit

List of oneAPI AI Analytics Toolkits & its libraries used

Intel oneAPI Base Toolkit (General Compute)
Intel® oneAPI Data Analytics Library
Intel® oneAPI Deep Neural Networks Library
Intel® Distribution for Python
Intel® oneAPI Math Kernel Library

Intel® AI Analytics Toolkit
(End-to-End AI and Machine Learning Acceleration)
Intel® Distribution for Python with highly optimized scikit-learn
Intel® Optimization for TensorFlow
Intel® Optimization of Modin

Base Technology Stack
Web Plugins JQuery - Web Application (Frontend)
Tailwind CSS - (Style)
Django - Web Application (Backend)
Javascript - Validation & Client-Side Scripting
MongoDB & Sqlite3 - DBMS
Matplotlib & Seaborn - Data Visualization
Google Charts, Charts.js and/or any other 3rd Party - Data Visualizer
TensorFlow & scikit-learn (scipy) - DL and ML Model
OpenCV - Computer Vision

Result Summary

- By utilizing Intel oneDNN, our three models benefit from accelerated training, improved throughput, enhanced inference speed, and reduced latency. The library's optimized computations and parallelization techniques optimize the performance of our models, enabling faster and more efficient processing.
- With OpenMP, we can leverage multi-threading capabilities to distribute computations among multiple cores or processors, maximizing performance and speeding up training and inference processes for our models. This parallelization significantly improves the overall efficiency and scalability of our models.
- Modin for Pandas is a powerful library that enhances the performance of Pandas, a popular data manipulation and analysis tool. By integrating Modin, we can scale Pandas operations across multiple processors or nodes, enabling faster data preprocessing and manipulation.
- The test_train_split function from scikit-learn (sklearnex) is particularly helpful in our models. It enables us to split our dataset into training and testing subsets, facilitating proper model evaluation and validation. This function ensures that our models are trained on a representative subset of data and are subsequently tested on an independent portion, helping us gauge their performance accurately.
- In addition, the patch functionality of scikit-learn (sklearnex) allows us to apply specific fixes or modifications to the library, enhancing its compatibility with our models and ensuring seamless integration.
- These combined capabilities of scikit-learn (sklearnex) and the test_train_split function enable us to effectively train and evaluate our models, leading to more accurate predictions and reliable performance.
- Faster inference speed is crucial in deployment, as it ensures real-time responsiveness and enables scalability in high-concurrency scenarios. It also contributes to cost efficiency by optimizing resource utilization. The Intel i5 11th gen processor plays a pivotal role in driving the efficiency and speed of our models in our web-based education platform.

GitHub Link:
https://github.com/SaumyaSrivastava13/Phoenix_13

Any Questions?

THANK YOU