Project Proposal for Google Summer of Code 2023

Google Summer of Code

ML4SCI

CMS Experiment

Vision Transformers for End-to-End Particle Reconstruction

Mentors

→ Emanuele Usai (University of Alabama)
→ Ruchi Chudasama (University of Alabama)
→ Shravan Chaudhari (New York University)
→ Sergei Gleyzer (University of Alabama)
→ Eric Reinhardt (University of Alabama)
→ Samuel Campbell (University of Alabama)

Machine Learning for Science

TABLE OF CONTENTS:

CONTACT INFORMATION:

Name: Vishak K Bhat

University: Indian Institute of Technology (IIT), Dhanbad

Email: vishak.bhat5@gmail.com | 21je1047@iitism.ac.in

GitHub: vishak-github

Kaggle: vishak-kaggle (expert)

LinkedIn : vishak-linkedin

CV: Vishak-CV

Phone : (+91) 7760289129

Time zone: Indian Standard Time (UTC +05:30)

Location: Bengaluru, India.

PROJECT SYNOPSIS and PROBLEM DESCRIPTION:

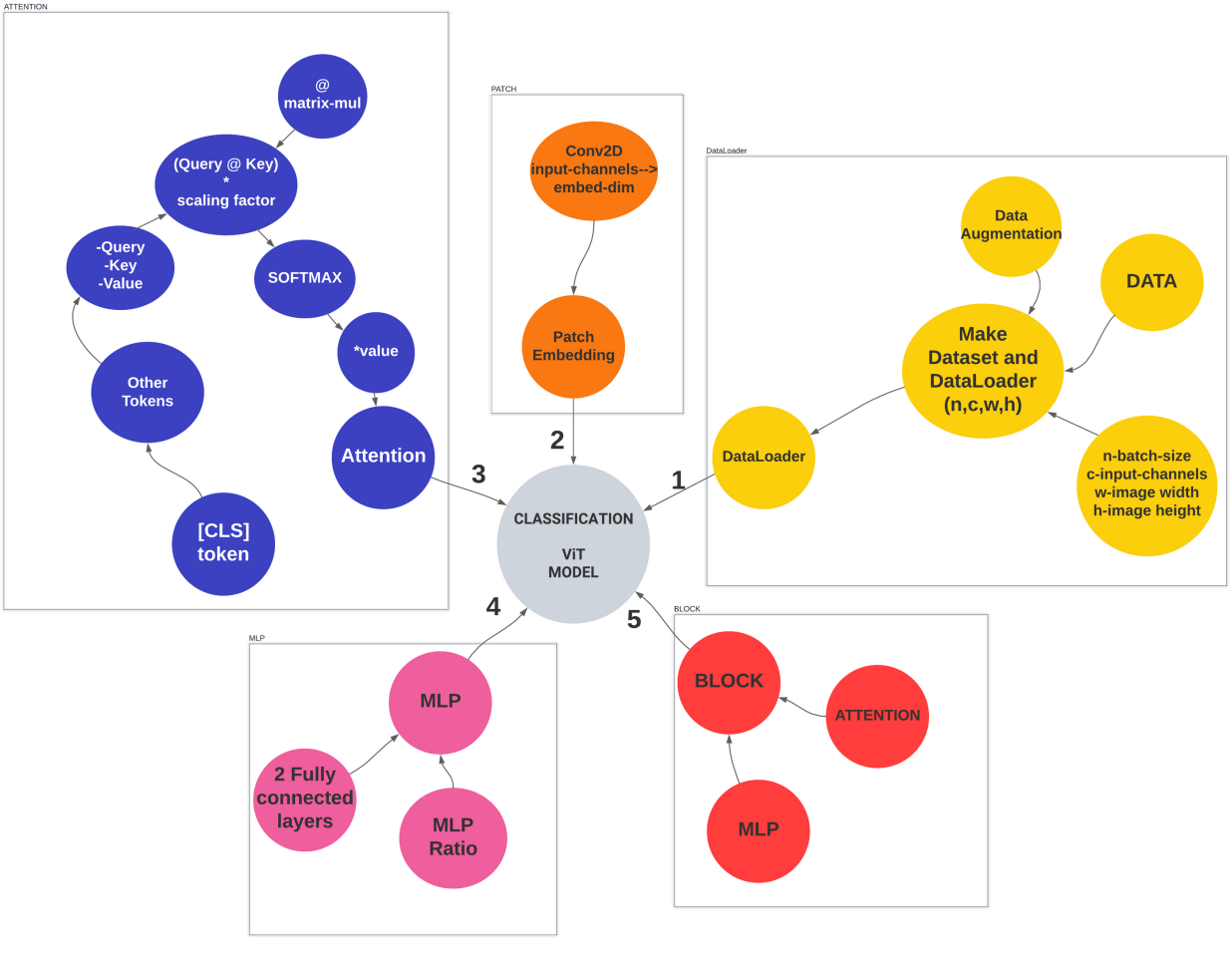
The **Large Hadron Collider (LHC)** at **CERN** is one of the largest and most complex scientific instruments ever built. Its primary goal is to study the fundamental nature of matter and energy by colliding particles at extremely high energies. One of the key challenges of **analyzing data** from the **LHC** is the large amount of information produced by its detectors.

In particular, the **Compact Muon Solenoid (CMS) detector** records a vast array of data from particle collisions, including energy deposits, particle tracks, and other information. CMS acts as a giant, high-speed camera, taking 3D “photographs” of particle collisions from all directions up to 40 million times each second. The CMS collects a few **tens of Peta-Bytes** of data each year.

In order to extract meaningful physics information from this data, advanced data analysis techniques such as deep learning, Convolutional Neural Networks(CNN) and Vision Transforms(ViT) are developed.

The **aim** of this project is to explore the potential of **Transformers** in computer vision applications . Transformers have shown to **outperform** traditional **Convolutional Neural Networks** (CNNs) through **attention** mechanisms. The project will compare the effectiveness of Transformers and CNNs in image classification, contributing to advancements in the field. It also aims to achieve **the benchmark on GPU**.

PROJECT APPROACH and IMPLEMENTATION:



MODEL ARCHITECTURE

Fig1: The above mind map shows the approach that I have implemented while training my ViT(Vision Transformer) model (**NOTE:** The numbers denote the step number)

1. Data Preprocessing:

- I will apply **feature scaling** to the data before training the machine learning model. This involves **rescaling** the data to ensure that all features have a similar range and distribution, which can prevent some features from dominating others.

- I will be using **data augmentation** techniques. This will involve applying various transformations on the original data to prevent overfitting. The data is then fed to the data loader so that it can be further trained. **(fig1: step 1)**

2. Model training:

I will be implementing these layers:

- **Patch embedding:** Splits the input image into **patches** and embed them into an **embedding dimension**. This is achieved by passing the input tensor through a **Conv2d** layer with a specified number of filters. The output tensor is then flattened and transposed to obtain a sequence of patches with shape (batch_size, no_patches, embed_dim), which is used as input to the next layer in the model. **(fig1: step 2)**

- **Attention:** It is a mechanism that allows a model to focus on the most **relevant parts** of the input when making predictions. I will be implementing the following attention mechanisms: **(fig1: step 3)**

  → **Multi-head self-attention:** Attends **different parts** of the input at the same time. This is achieved by splitting the input into multiple "heads" and computing the attention separately for each head.

  → **Local and global self-attention:** Used to capture both local and global relationships between tokens. Local attention focuses on **neighboring tokens**, while global attention looks at all tokens in the **sequence**.

  → **Cross-attention:** Attends to a **different input**, such as an image, in addition to the sequence of tokens. This can be useful for capturing spatial relationships between different parts of the image.

  → **Multi-scale attention:** Captures relationships between tokens at different **levels of granularity**, allowing a model to capture both fine-grained and coarse-grained features.

  → **Pyramid structure:** Splits the input into multiple levels, each of which captures features at a different scale. This allows a model to capture both local and global relationships between tokens, as well as features at different levels of abstraction.

- **MLP(Multi Layer Perceptron):** I will use this after the attention layer to extract more useful features from the output of the previous layer. This basically contains multiple fully connected layers. **(fig1: step 4)**

- **Transformer Block:** Consists of attention and MLP layers. **(fig1: step 5)**

Using the above layers I will train the following models: (Reference section has the links to research papers of them)

→ **DeiT** (Dense Encoder Innovations for Vision Transformers): uses a **ResNet-**like architecture with self-**attention** layers and **distillation** during training, and data augmentation techniques for robustness.

→ **TNT** (Token-Transformers for Image Recognition at Scale): combines convolutions and transformers, uses **patch merging** and **multi-head self-attention** to handle large image sizes.

→ **CaiT** (Cross-Attention Image Transformer): improves cross-attention in vision transformers with a **cross-**attention mechanism that allows for interactions between tokens from different **spatial** locations.

→ **SETR** (Semi-supervised Transformers for Image Recognition): designed for semi-supervised learning and uses a **hybrid** architecture with **multi-scale feature maps** and a set prediction head.

→ **PVT** (Pyramid Vision Transformer): uses a **pyramid-style** architecture with spatial pyramid pooling and transformer layers to improve performance on small objects in images.

- After training these models I will make an ensemble model to finally make the classification.

- At the final stage, the model will undergo testing and benchmarking using a **GPU** (Graphics Processing Unit) to measure its performance metrics, such as accuracy, speed, and memory usage.

PROJECT DELIVERABLES:

- Implement the transformer models such as **DeiT**, **TNT**, **CaiT**, **SETR**, **PVT.**
- **Compare** these models and tune the hyperparameters to maximize the accuracy.
- Perform testing and **benchmarking** on GPU.
- Make **ensembles** of the created models.
- Apply the model made on the **similar type** of problems.
- **Summarizing** the entire work and making proper documentation.

EVALUATION TASKS:

I had completed the evaluation tasks mentioned in the stipulated deadline. The Jupyter Notebook has been commented adequately.

As mentioned, I have made a model to classify electrons and photons and got the minimum ROC AUC of 80%. This can be seen from the graph of ROC(TASK 1). Then I have made a model to classify the particles - Quarks and Gluons.(TASK 2). Lastly I have made a ViT(Vision Transformer) model which gave the ROC AUC score close to the CNN model. Due to computational limitations the transformer based model had slightly less score than that of the CNN model. All the models were trained in GPU.

The GitHub link of the evaluation task - **GitHub** .

PROJECT DETAILED TIMELINE:

PHASE/WEEK	DATE	WORK DESCRIPTION
COMMUNITY BONDING		
COMMUNITY BONDING	**May 4- May 12**	**→ Familiarizing with the Community and Organization Standards** **→ Discussion of problems and final goals**
	May 13- May 21	**→ Define the project's outcomes more clearly** **→ Validate them with mentors.**
	May 22- May 28	**→ Break project goals into smaller, trackable issues for better analysis of progress and milestones.**

		→ Complete initial setup of working environment
PHASE 1- May 29 - July 10		
Week 1	**May 29 - June 4**	**→ Revisiting the research paper of few more Transformer based Models especially these:-** **DeiT, TNT, CaiT, SETR, PVT (Implemented ViT(Vision Transformer) model in the test). (Already read once during the pre GSoC'23 application period)**
Week 2	**June 5 - June 11**	**→ Data Preprocessing: Augmentation , Normalization.** **→ Testing and bench marking on GPU.**
Week 3	**June 12 - June 18**	**→ Tune the Hyper parameters of the ViT model already made.** **→ Plot the graphs of ROC AUC to see the accuracy and make a report.**
Week 4	**June 19- June 25**	**→ Pytorch implementation of DeiT(Dense Encoder Innovations for Vision Transformers), TNT(Token-Transformers for Image Recognition at Scale), CaiT(Cross-Attention Image Transformer)** **→ Testing and bench marking on GPU**
Week 5	**June 26 - July 2**	**→ Implementing SETR(Semi-supervised Transformers for Image Recognition) PVT(Pyramid Vision Transformer)** **→ Testing and bench marking on GPU**

Week 6	**July 3 - July 10**	**→ Make all the code(notebook) written in presentable format for the mid evaluation** **→ Making the summary and report of all the models**
PHASE 1 Evaluation: July 10 - July 14		
PHASE 2: July 14 - August 21		
Week 7	**July 14 - July 20**	**→ Compare between all the models created** **→ Tune the hyperparameters of all the models created in the previous weeks**
Week 8	**July 20 - July 27**	**→ Make ensemble of all the models made to improve the results** **→ Training on the above model**
Week 9	**July 28 - Aug 3**	**→ Extending the model to other datasets and similar problem** **→ Cleaning up the code and adding comments so that its presentable**
Week 10	**Aug 3 - Aug 10**	**→ Summarizing the entire work done, tabulating and plotting the results obtained.** **→ Completing the documentation, integrating all code.**

Week 11	**Aug 10 - Aug 21**	**→ Buffer week and further enhancements**
Final Evaluation: Aug 21 - Aug 28		

About Me and Motivation for GSOC:

Overview:

I am a second year undergraduate pursuing Integrated MTech in Mathematics and Computing from the Indian Institute of Technology, Dhanbad(IIT ISM Dhanbad), India.

I am a typical geek who loves programming and enjoys problem-solving and making side projects as a part of hobby coding. Along with my friends, I manage a university-level open-source community, Cyber Labs, where we regularly participate in discussion of research papers related to computer vision and take part in machine learning competitions.

The pride in the feeling that my code will cause an impact in the lives of millions of people who will use it is unparalleled. Moreover, it allows me to grow as an individual and learn how to work in a team with such a big community. I have also been active in introducing people to the world of open source and also getting them involved with various open-source projects and communities.

SKILLS:

C++(STL, Armadillo, mlpack, Boost), Python(NumPy, Pandas, Matplotlib, Scikit-learn),MATLAB, Bash, SQL, Pytorch, TensorFlow(elementary proficiency).

EXPERIENCE:

- Completed the below **courses** from Coursera:
  - Convolutional Neural Network by Stanford, Coursera.
  - Improving Deep Neural Network by Stanford, Coursera.
  - Machine Learning by Stanford, Coursera.
  - Neural Network and Deep Learning by Stanford, Coursera.

Under **CyberLabs** made few projects like:

- "Emotional fool"- Which basically detects the emotions of the human using open CV. Further it displays the emoji of the reaction detected.
- "Dementia Classifier"- Given the image of the CDT(Clock Drawing Test), this model classifies the severity of the disease ranging from 0-5.

Kaggle: participated in many Kaggle competitions and currently **Expert tier** in Kaggle.

Represented college in the inter IIT tech fest for the machine learning problem statement.

REFERENCES:

1. ML4SCI Project Ideas
2. Vision Transformers for End-to-End Particle Reconstruction for the CMS Experiment
3. Evaluation test
4. End-to-End Physics Event Classification with CMS Open Data
5. End-to-End Jet Classification of Quarks and Gluons with the CMS Open Data
6. GSoC 2023 Projects Related To CMS
7. CMS Experiment
8. Large Hadron Collider
9. CMS experiments
10. An Image Is Worth 16x16 Words
11. DeiT paper
12. TNT paper
13. CaiT paper
14. SETR paper
15. PVT paper
16. ResNet paper
17. Ensemble

THANK YOU