End-to-End Deep Learning Reconstruction for CMS Experiment

(Google Summer of Code, 2021-Proposal)
Organization:
ML4SCI (E2E)

Sub-organizations:
Alabama
BITS Pilani Goa
Brown
CMU
KGI

NAME: Prayushi Mathur
EMAIL: prayushimathur@gmail.com , prayushi.mathur@research.iiit.ac.in

Index

Serial No.	Topic
1.	My Introduction (E2E evaluation Test links)
2.	Project Introduction
3.	Background Study
4.	Artificial Intelligence in CMS Experiment
5.	Artificial Intelligence Techniques proposed
6.	Detailed Timeline
7.	Deliverables

8.	References
9.	Working time and commitments
10.	Communicating with mentors

My Introduction

- **Personal Information and contact details**

Full Name	Prayushi Mathur
Email	prayushimathur@gmail.com , prayushi.mathur@research.iiit.ac.in
University and Course	- International Institute of Information and Technology, Hyderabad (IIIT-H) - MS by Research in Computer Science and Engineering
Location (Country, City)	India, Kota
Time Zone	Indian Standard Time (GMT+5:30) (IST)
Linkedin profile	https://www.linkedin.com/in/prayushi-mathur-59a470189/
Github profile	https://github.com/Prayushi9
Hangouts id	prayushimathur@gmail.com
CV and Resume	- https://drive.google.com/file/d/1DygDrObM4jE3X6a7S4AMrbqaauHTgmdr/view?usp=sharing

	- https://drive.google.com/file/d/1UOxdLearmqLs6ngerqw1neZVLzd06n7C/view?usp=sharing

- ## About Me

  I am Prayushi Mathur, a 2020 graduate from Nirma University, Ahmedabad, Gujarat. Currently, I am interning as a research associate under Dr. Syed Azeemuddin and Dr. Avinash Sharma at International Institute of Information Technology, Hyderabad. I would be joining as a student at IIIT-H for my masters (MS by Research in CSE) from Monsoon, 2021.

  I have worked across diverse domains like Computer Vision, Image Processing, Medical Imaging, Satellite Imagery, GANs from the past 3 years. I am confident about my skills in this field as I have worked on multiple types of data having different applications in this domain.

- ## Past Experience

- Currently, I am working as a Research Associate at **International Institute of Information Technology, Hyderabad (IIIT-H)** on the project 'Real-time superresolution on Jetson Nano'. I will be pursuing my masters degree (MS by Research in Computer Science and Engineering) from IIIT-H starting from Fall, 2021.
- I have completed my internship at **Government of India** for 9 months as an AI intern.
- I have worked on a project named 'Oceanographic element detection on SAR imagery' under **Indian Space Research Organisation (ISRO)** for 1 year.

- ## E2E Evaluation tests link:

  Task1: https://github.com/Prayushi9/Electron-photon_Classification
  Task2: https://github.com/Prayushi9/Quark-Gluon-classification

Task3: https://github.com/Prayushi9/E2E_Reconstruction

- **My Deep Learning projects link**

  Multi organ segmentation:
  https://github.com/Prayushi9/Multi-Organ-Segmentation
  U-net: https://github.com/Prayushi9/U-net
  V-net: https://github.com/Prayushi9/V-net
  WGAN-GP: https://github.com/Prayushi9/WGAN-GP
  AMC-SSDA: https://github.com/Prayushi9/AMC-SSDA
  YOLO-v2: https://github.com/Prayushi9/Yolo-v2
  YOLO-v3: https://github.com/Prayushi9/YOLO-v3

Project Introduction

End-to-End Deep Learning Reconstruction for CMS Experiment

The Compact Muon Solenoid (CMS) experiment is one of the largest scientific collaborations across the world at CERN in France and Switzerland. At the CERN laboratory , this experiment is a general purpose particle physics detector built on the Large Hadron Collider (LHC). This experiment has a large umbrella of physics programmes consisting of searching for extra dimensions, studying the standard model including the Higgs boson, identification of signal events from their corresponding backgrounds and the particles that are responsible for making the dark matter.

To understand the standard model (SM) of particle physics through the analysis of jets which are produced from Quantum Chromodynamics (QCD) and the decay of boosted heavy particles of resonances (e.g. Higgs boson or top quark), the study of jet structure at the CERN LHC has played a major role [1]. Jets arising from the boosted heavy resources, exotic or other beyond the standard model (BSM) can be majorly understood by the detailed study of the vastly more dominant QCD jet background. So, the discrimination and characterization of light quark versus gluon initiated jets that comprise of QCD jets has thoroughly examined. The difference of QCD color charges in both quark and gluon can be used to classify or distinguish them [3].

There are various solutions proposed to address this issue such as order-invariant algorithms and artificial by physical motivated ordering schemes. The other approach which came into popular use is to exploit the spatial distribution of the particles in the

detector as a natural solution to the ordering problem. Deep learning can also be used to solve the problem using the most popular Convolutional Neural Network (CNN) by using the 'jet images' which have been created by pixelating the particle-level data into a grid that amounts to a coarse-grained histogram-like image of the underlying detector geometry.

Here, end-to-end deep learning techniques will be used to identify and reconstruct single particles, jets and event topologies of interest in collision events in the CMS experiment. Finally, this E2E code will be deployed and integrated with the CMSSW inference engine for use in reconstruction algorithms in offline and high-level trigger systems of the CMS experiment.

This E2E integrated code will be integrated within the CMSSW classes. Finally, the benchmark of end-to-end deep learning process inference will be on CPUs and GPUs for faster processing and inference.

Background Study

There are various applications on the E2E framework [2] [3]. A discriminator between electrons and photons was constructed for the very first application of the end-to-end application. ECAL energy images were used and those images were fed to a Resnet-15 [4] network obtaining an area under the curve of the Receiver Operating Characteristic (ROC AUC) score of 0.788. Also, a di-photon vs. di-electron discriminator was built similarly constructed detector images and a ResNet-23 network obtaining a ROC AUC of 0.997.

The E2E framework was also used to distinguish the Higgs boson to di-photon decay from non-resonant di-photon production and misidentified photons. This application used ECAL, HCAL and track images. One of the challenges in this particular application was the correlation of the discriminator output with the mass of the Higgs boson, which is an undesirable feature in physics analyses. The discriminator was decorrelated from the diphoton mass through the use of an additional loss penalty proportional to the learned mass correlation as measured by the Cramer-Von Mises metric [5]. The de-correlated discriminator yielded a ROC AUC score of 0.77 in distinguishing $H \rightarrow \gamma\gamma$ from non-resonant $\gamma\gamma$ production, 0.95 for $H \rightarrow \gamma\gamma$ vs. γ+jet (misidentified photon), and 0.81 for H vs. non-resonant $\gamma\gamma$ and γ+jet together.

Artificial Intelligence in CMS Experiment

1.) Anomaly Detection

the use of artificial neural networks for supervised and semi supervised problems related to the identification of anomalies in the data collected by the CMS muon detectors. We use deep neural networks to analyze LHC collision data, represented as images organized geographically. We train a classifier capable of detecting the known anomalous behaviors with unprecedented efficiency and explore the usage of convolutional autoencoders to extend anomaly detection capabilities to unforeseen failure modes. A generalization of this strategy could pave the way to the automation of the data quality assessment process for present and future high energy physics experiments.
(Reference: [11])

2.) Jet Classification

Machine learning has been applied to a wide range of jet classification problems, to identify jets from heavy (c, b, t) or light (u, d, s) quarks, gluons, and W, Z, and H bosons. Traditionally these classification problems have been grouped into flavor tagging, which discriminates between b, c, and light quarks, jet substructure tagging, which discriminates between jets from W, Z, t and H, and quark–gluon tagging.
(Reference: [12])

3.) Track-Reconstruction algorithms

Track-reconstruction algorithms are among the most central processing unit (CPU) and data intensive of all low-level reconstruction tasks. The initial stage of track reconstruction involves finding hits, or points where some charge is deposited on a sensing element. In the case of the pixel sensors that form the innermost layer of the detector, neighboring hits are clustered into pixel clusters, which then form track seeds. These seeds form a starting point for a Kalman filter, which extends the seeds into full tracks that extend to the calorimeters. The entire procedure can be viewed as a sequence of clustering algorithms, in which the zero-suppressed readout from O(108) channels provides O(104) hits, which are then clustered into O(103) tracks per event.
(Reference: [12])

4.) Jet Reconstruction

We present the jet tagging inputs in more detail in terms of the intrinsic data hierarchy and complexity. We present the currently recommended flavor tagger of the CMS experiment as well as the new proposal that is built on deep learning in the sense that

layers for feature engineering are added that use the intrinsic hierarchy of the data. Finally, we present results of the former standard CMS flavor tagger (CSVv2), the currently recommended flavor tagger (DeepCSV) and the generic new proposal (DeepJet). We also compare the generic tagging capabilities to other deep neural network architectures for quark vs. gluon classification
(Reference: [13])

5.) Event Classification

In any real physics decay, energy and momentum conservation impose physical constraints on the allowed kinematics of the decaying particles. In this section, we therefore attempt to classify realistic H → γγ vs. γγ vs. γ + jet decays. The end-to-end (E2E) event classification results are divided by pseudorapidity (see Section 2), with the results for the central (central+forward) category shown in Figure 2 (Figure 4). The ECAL-only classifier is labeled EB (ECAL) and the Tracks+ECAL+HCAL classifier in the ECAL-centric geometry is labeled CMS-B (CMS-I). For the central+forward region, we also include the results of the HCAL-centric classifier (CMS-II). In each category, we plot the signal vs. combined background ROC (1-vs-Rest), as well as the signal vs. single background ROC component (1-vs-1). For context, we also include the results of the (mass de-correlated) 4-momentum only classifier (4-mom).
(Reference: [14])

Artificial Intelligence Techniques proposed

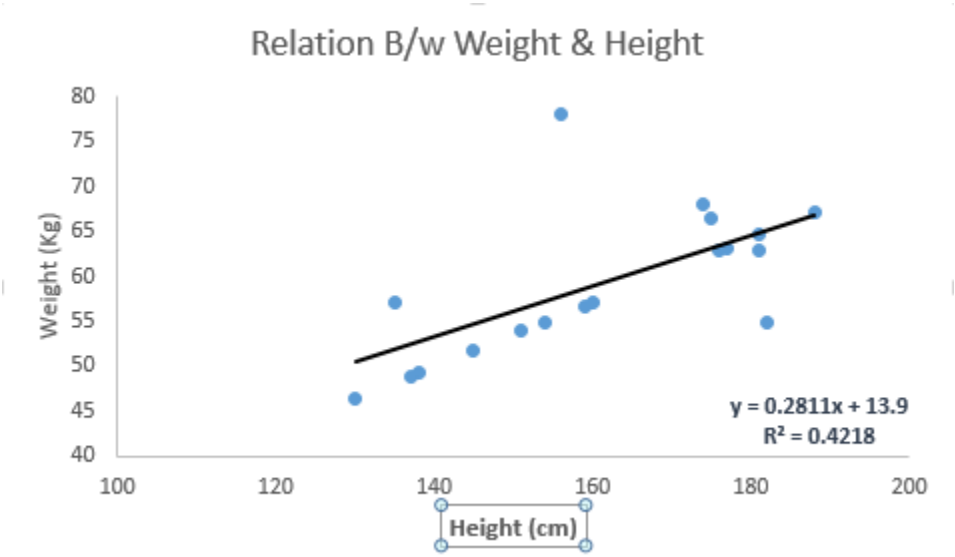
The advent of Artificial intelligence has given a key to unlock some new doors of physics research in the CMS experiment. Deep Learning can be used to classify, reconstruct, distinguish or predict the particle characteristics in this experiment. This can be done using various deep learning models. There are various computer vision tasks such as object detection, object recognition, object classification, object segmentation, super-resolution, etc. which can be performed using deep learning techniques. The algorithms/models are as follows:

1.) Basic Machine Learning algorithms

The following are the list of commonly used machine learning algorithms which can be applied to various datasets:

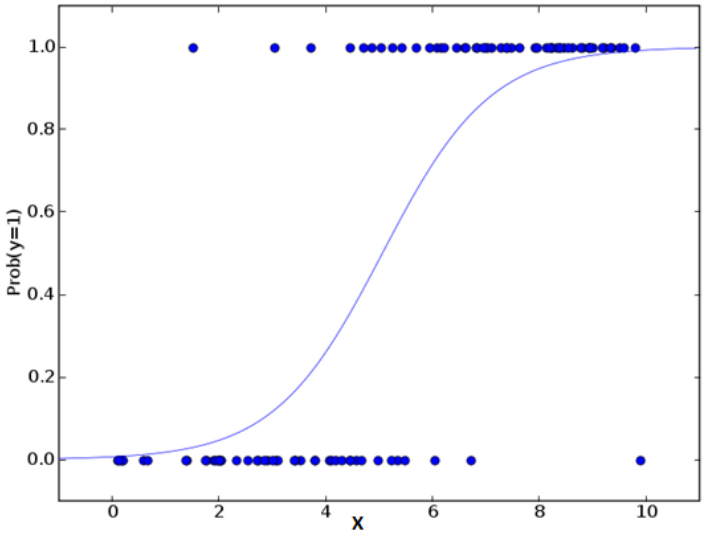
(Reference: https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/)

1. **Linear Regression**



It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a *X + b.

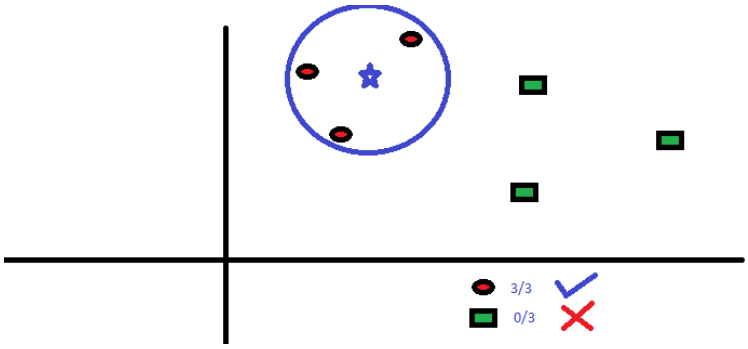
2. **Logistic Regression**



It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of

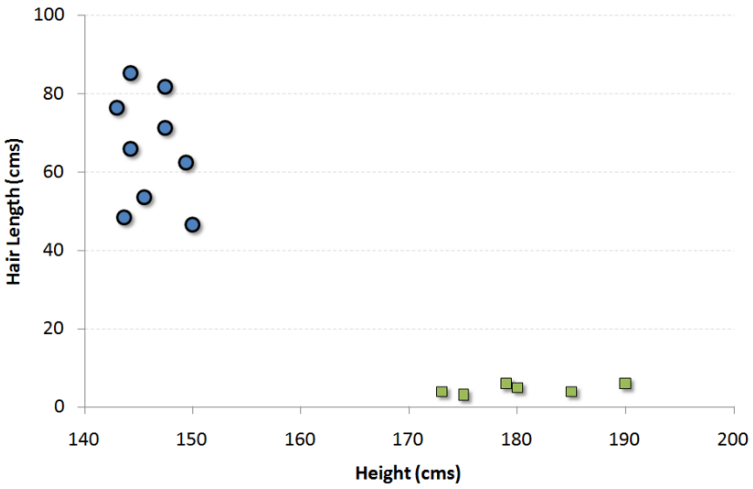
independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as logit regression. Since it predicts the probability, its output values lies between 0 and 1 (as expected).

3. **K-Nearest Neighbours (KNN)**



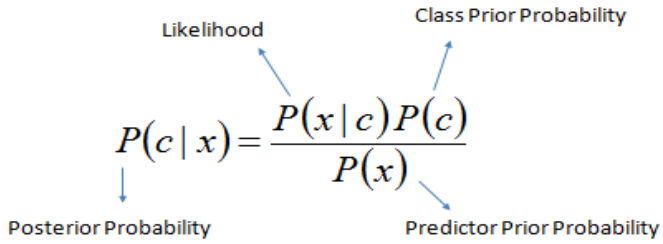
It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function.

4. **Support Vector Machine (SVM)**



It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate.

5. **Naive Bayes**



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

It is a classification technique based on Bayes' theorem with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

6. **Dimensionality Reduction algorithms**
   The Dimensionality reduction algorithm helps us along with various other algorithms like Decision Tree, Random Forest, PCA, Factor Analysis, Identify based on correlation matrix, missing value ratio and others.

2.) Convolutional Neural Networks

As per the requirement of the project, we will pick the type of architecture we want to proceed with. The CNN architecture can be made from scratch or an existing architecture can be used.

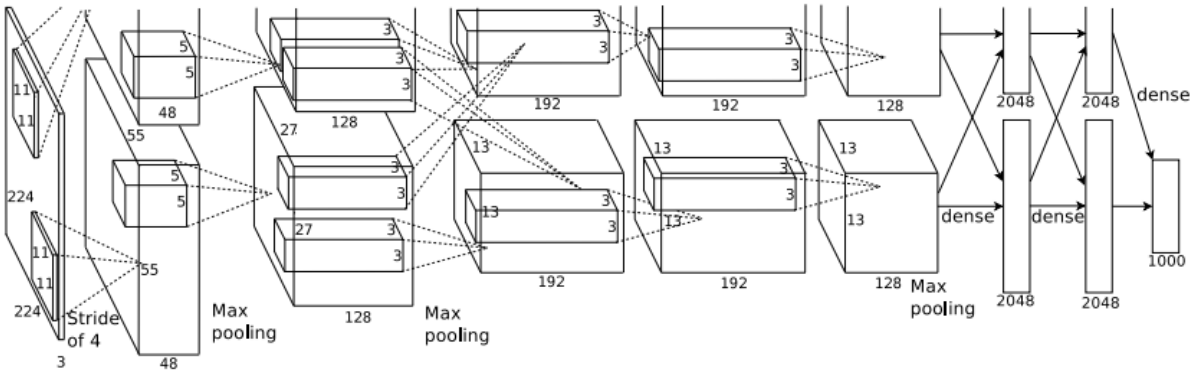
Languages	Python, C++
Libraries and DL frameworks	Keras, tensorflow, pytorch, opencv, tensorflow-serving, scikit-learn
Platforms to be used	Google colaboratory, Jupyter notebook
Hardware	CPU, GPU

Components	Description
Layers	Convolutional (2D and 3D), flatten, dense, input, embedding, lamda, maxpool, average pool, LSTM, GRU, bidirectional,

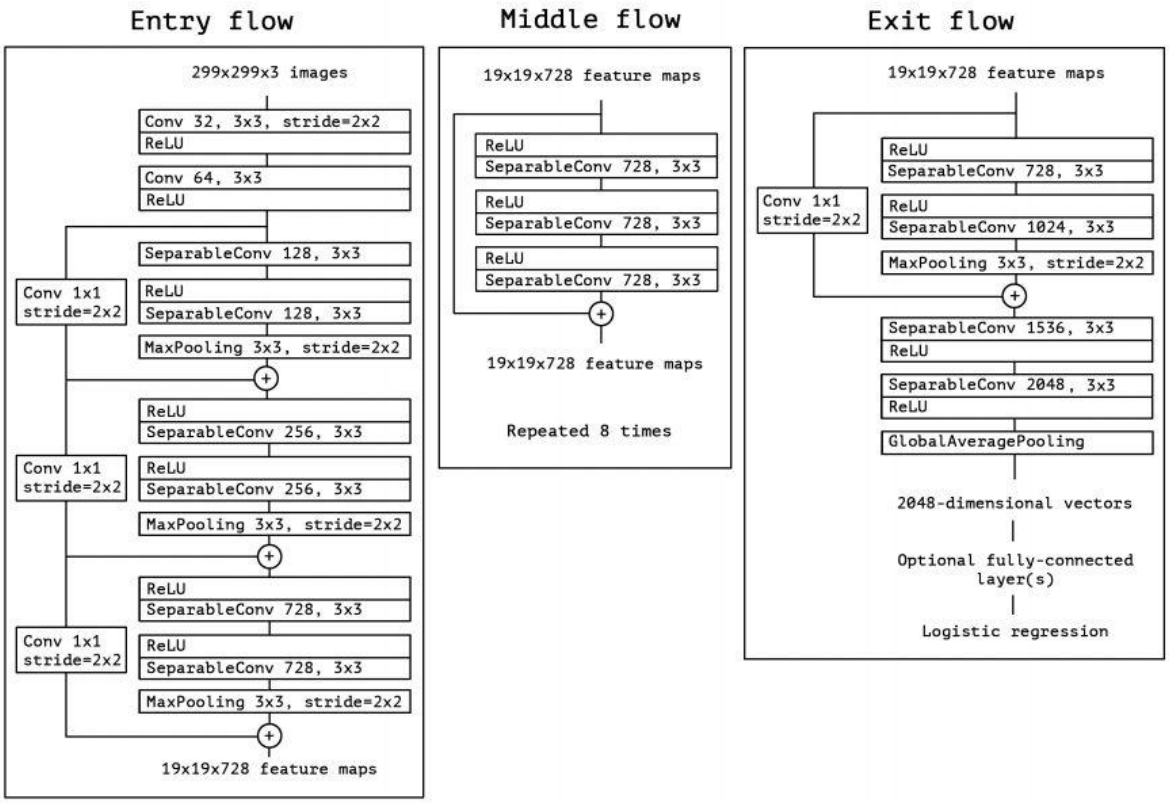
	batch normalization, etc.
Activation functions	ReLU, linear, sigmoid, softmax, tanh, elu, selu, exponential, softplus, softsign
Regularization	Dropout, L1, L2
Loss	Binary crossentropy, categorical crossentropy, mean squared error, mean absolute error, accuracy, cosine similarity
Optimizer	SGD, RMSprop, adam, adadelta, adagrad, adamax, nadam
Metrics	Accuracy, precision, recall, F1 score, ROC AUC

The below mentioned CNN architectures which are the state-of-the-art deep learning models:

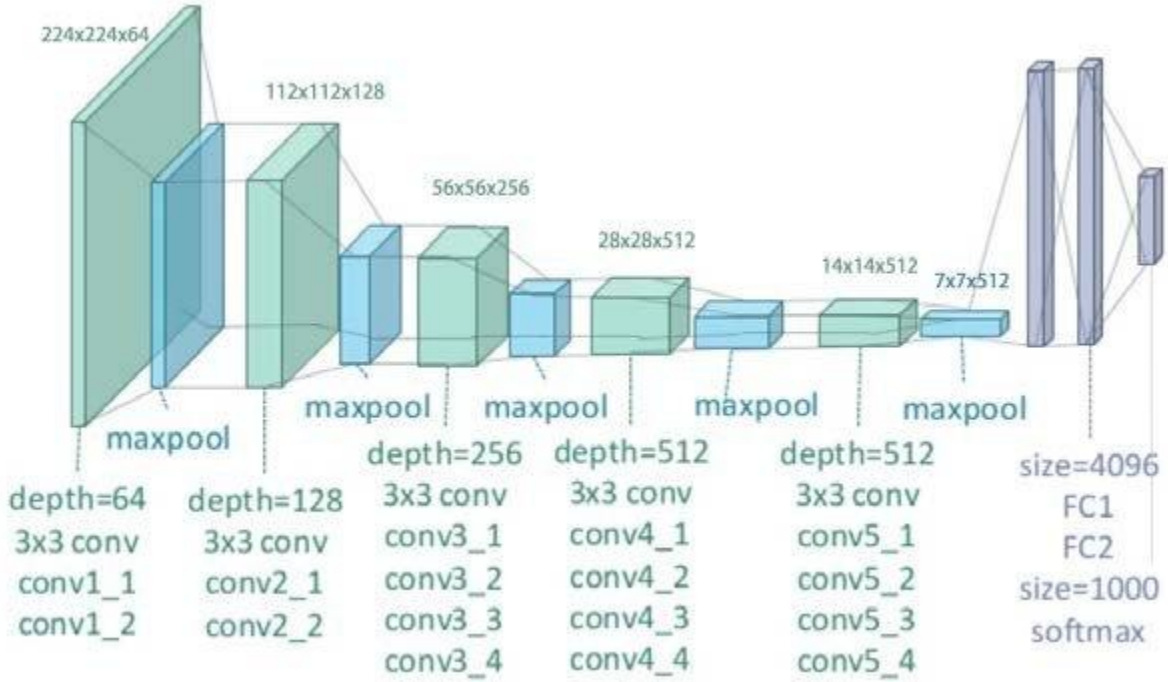
1. **Alexnet** [6]



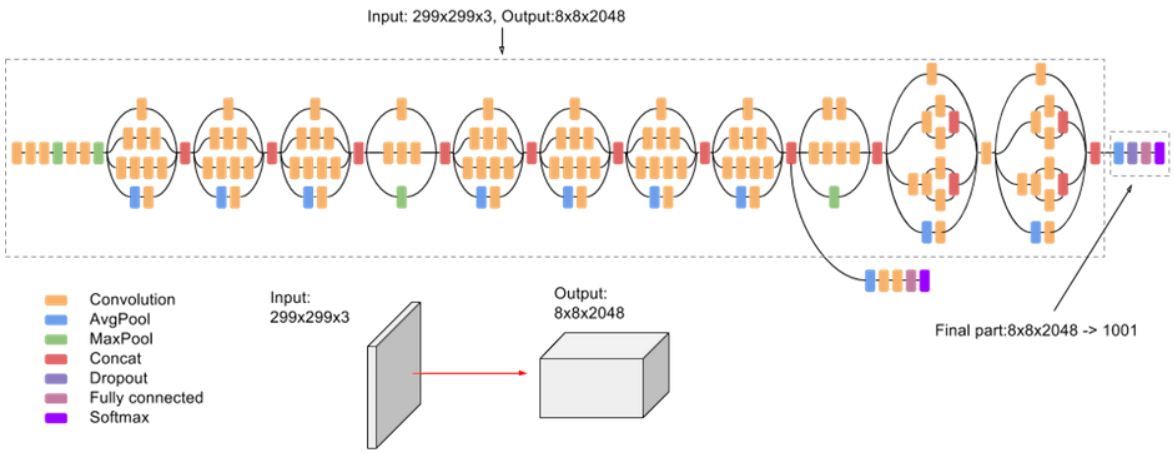
2. **Xception** [7]



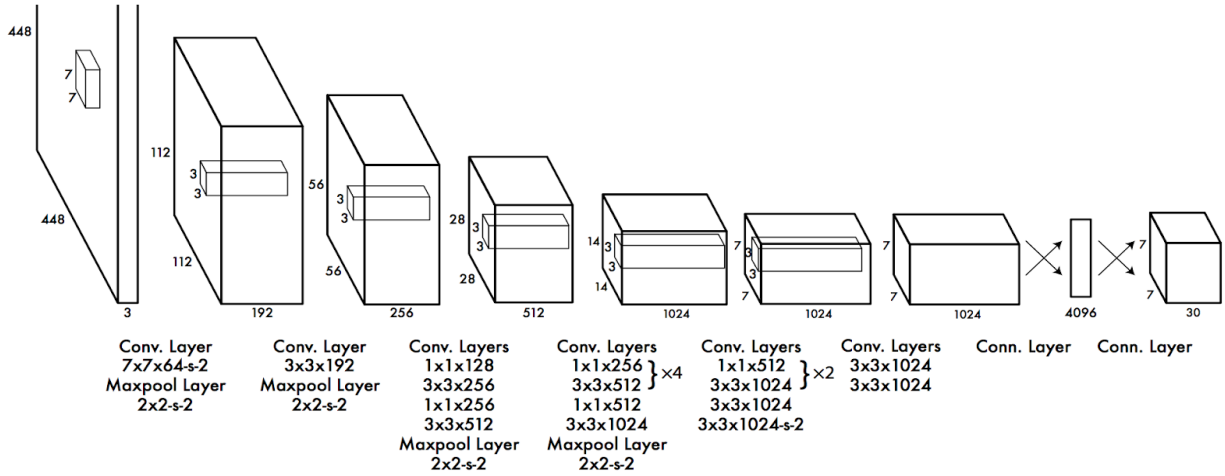
3. **VGG-19** [8]



4. **InceptionV3** [9]



5. **YOLOv3** [10]



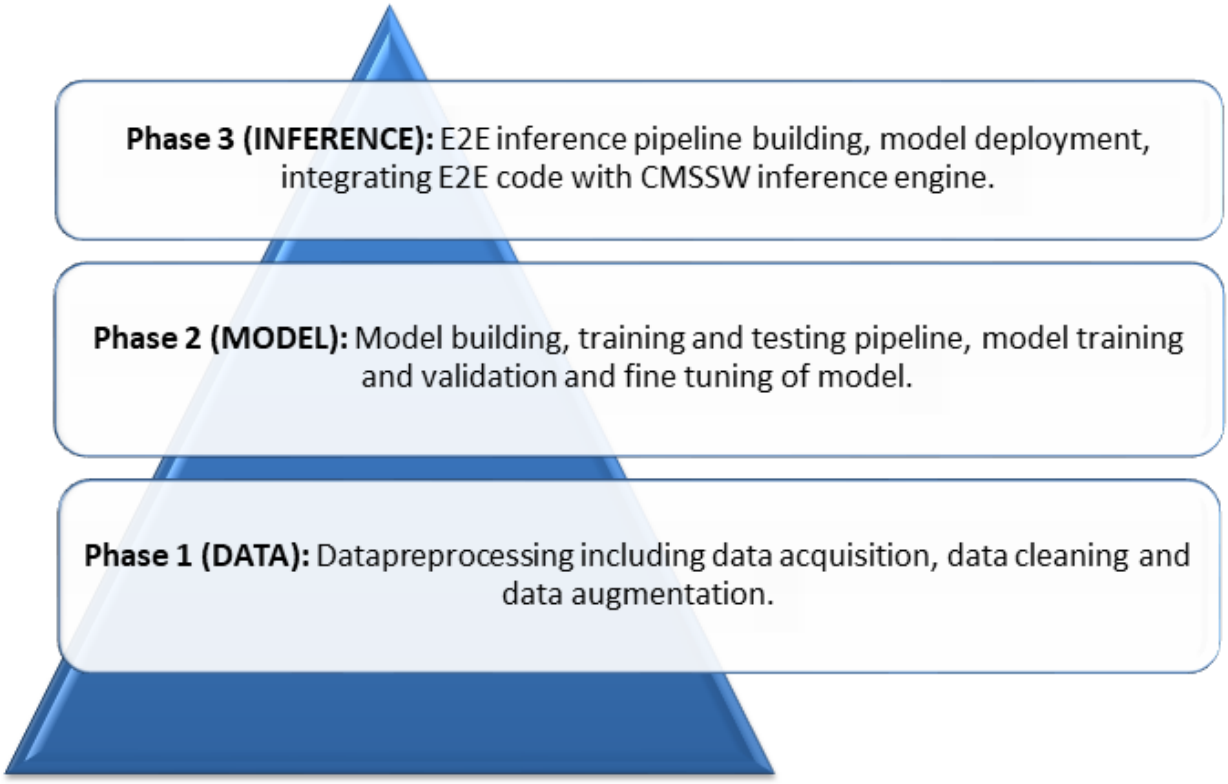
3.) Autoencoders and GANs

Autoencoders (AE) are neural networks that aims to copy their inputs to their outputs.They work by compressing the input into a latent space representation, and then reconstructing the output from this representation. It has two major components known as encoder and decoder.

Generative Adversarial Networks (GANs) can also be used for the classification task. They are advanced deep learning models with highly optimized training capabilities (minimax game). It has two major components known as Generator and Discriminator. Some of the famous GANs are:

1. **DCGAN**
2. **CGAN**
3. **WGAN/WGAN-GP**

Project Flow



Detailed Timeline

DURATION	TASKS
May 17 - June 7 (Community Bonding)	• Connecting with mentors and gathering more information about the project. • Learning the physics aspect of the project and linking it with the deep learning models. Also, studying the previous work of the organization in the project related domain and doing the literature study. • Discussing more ideas with mentors and defining the clear goals.
June 7 - July 16	**Coding Phase-1**

June 7 - June 13	• Studying about the dataset required for this task. • Accessing the dataset and cleaning it as per the requirement. • Preprocessing the data for better results.
June 14 - June 20	• Doing the literature survey about the deep learning models. • Finalising the base model to be built for the required task. • Studying the existing implementation of the model and doing hands-on.
June 21 - June 27	• Building the deep learning model for the task. • Building an end-to-end training and testing pipeline for the model.
June 28 - July 4	• Start training and validating the model with the prepared dataset. • Understanding the hyperparametric variations (combinations of hyperparameters) in the model.
July 5 - July 11	• Fine-tuning the model after doing the analysis to increase the accuracy. • Correcting the pipeline and the preprocessing part if required.
July 12 - July 16	• Testing the final model and obtaining the results. • Tracing the code and optimizing it for better results.
July 12 - July 16	**Evaluation Phase-1**
July 17 - Aug 16	**Coding Phase-2**
July 17 - July 23	• Testing the final model on the CMSSW engine. • Tracing the errors in the process and fixing it.

July 24 - July 30	• Integrating the final code with the CMSSW classes.
July 31 - Aug 6	• Testing and benchmarking the E2E implementation on GPUs. • Deploying the whole implementation on the CMSSW engine for the inference task.
Aug 7 - Aug 15	• Preparing the documentation. • Fixing the bugs and optimizing the code and documentation accordingly.
Aug 16 - Aug 23	**Students submit code and evaluation phase-2** • Discussing the documentation and final implementation with the mentors. • Submitting the code and documentation.

Deliverables

- The required deep learning model building.
- An end-to-end training and testing pipeline building of the model.
- Fine-tuning the model according to the requirement and type of data
- Integrating the whole E2E code in the CMSSW classes.
- Testing on CMSSW engine with CPUs and GPUs to optimize the performance.

Working Time and commitments

I will be able to spend 45-55 hours per week for the GSoC project. My masters programme will start from 1st august, so will be able to devote 30-40 hours per week from August 1, 2021 to August 23, 2021. I do not have any other commitments apart from this GSoC project.

Communicating with mentors

- Weekly updates and progress regarding the implementation of the project can be given via email. Weekly meetings via skype calls/hangouts can be done for regular interaction.
- For minor doubts and authorization issues (eg. dataset access), whatsapp chats will be useful.
- I am fine with any mode of communication and my preferred language for communication is English.

References

[1] M. Andrews , J. Alison , S. An , B. Burkle , S. Gleyzer , M. Narain , M. Paulini , B. Poczos , E. Usai, “End-to-End Jet Classification of Quarks and Gluons with the CMS Open Data”, arXiv:1902.08276 [hep-ex], 2019

[2] M. Andrews, J. Alison , S. An, P. Bryant , B. Burkle , S. Gleyzer , M. Narain , M. Paulini , B. Poczos , and E. Usai, “End-to-end particle and event identification at the Large Hadron Collider with CMS Open Data”, arXiv:1910.07029v1 [hep-ex], 2019

[3] M. Andrews, M. Paulini, S. Gleyzer, and B. Poczos, “End-to-End Physics Event Classification with the CMS Open Data: Applying Image-based Deep Learning on Detector Data to Directly Classify Collision Events at the LHC”, arXiv e-prints (Jul, 2018) arXiv:1807.11916, arXiv:1807.11916

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”, in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016 [10], pp. 770–778. doi:10.1109/CVPR.2016.90.

[5] A. Rogozhnikova et al., “New approaches for boosting to uniformity”, JINST 10 (2015).

[6] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks” , NIPS, 2012

[7] François Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions” , CVPR, 2017

[8] Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, ICLR, 2015

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe and Jonathon Shlens, “Rethinking the Inception Architecture for Computer Vision”, CVPR, 2016

[10] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection ”, CVPR, 2016

[11] Adrian Alan Pol, Gianluca Cerminara, Cecile Germain, Maurizio Pierini, Agrima Seth, “Detector Monitoring with Artificial Neural Networks at the CMS Experiment at the CERN Large Hadron Collider”, Springer Nature Switzerland AG 2019

[12] Dan Guest, Kyle Cranmer and Daniel Whiteson, “Deep Learning and Its Application to LHC Physics ”, arXiv:1806.11484v1 [hep-ex], 2018

[13] Markus Stoye, “Deep learning in jet reconstruction at CMS”, ACAT, 2017

[14] Michael Andrews, Manfred Paulini, Sergei Gleyzer and Barnabas Poczos, “Exploring End-to-end Deep Learning Applications for Event Classification at CMS”, CHEP, 2018