

An investigation into the application of machine learning to classify sub-atomic particle jets in the Large Hadron Collider

E. Lemoine. SN: 21010005
University College London

January 2024

INDEX:

Contents

1.0 Introduction and Aims	1
1.1 Background and theory.	1
1.1.1 The application of machine learning	2
1.1.2 Convolutional Neural Network (CNN)	3
1.2.3 The LHC jets dataset	4
2. Methods Introduction	4
2.1 Task 1: Combined Image Classifier	5
2.1.1 Data Pre-processing	5
2.1.2 Model Architecture	6
2.1.3 Testing performance	6
2.1.4 Separability Performance Analysis	7
2.1.5 Separability Principal Component Analysis	8
2.1.6 Efficiency Relation to Jet Features	9
3. Extensions	11
3.1 Separating the HCAL and ECAL images	11
3.2 Combined Classifier	12
3.3 Jet Features Classifier	12
4. Discussion and Conclusion	13
5. References	14

1.0 Introduction and Aims

Particle physics experiments at the Large Hadron Collider (LHC) [1] involve the collision of high-energy particles. Among the phenomena observed in these collisions are jets. Jets are collections of hadronic particles associated with initial high-momentum particles. The study of this phenomena has been integral to numerous discoveries in particle physics over the past five decades, ranging from the observation of fundamental particles such as gluons to searches for rare decays involving particles like the Higgs boson [2].

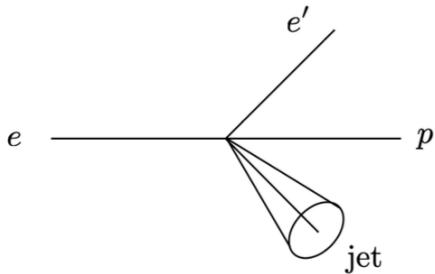


Figure 1: Jet produced via electron-proton scattering

Jets produced in particle collisions are typically detected through experimental setups involving tracking detectors, calorimeters, and muon detectors, among others. They generate vast amounts of data with each collision event, capturing information about the trajectories, energies, much more. These detectors provide detailed images of jets [3], representing how they appear within the detector. In this project, the aim is to develop a machine learning classifier capable of categorising these jets into 5 different classes based on the properties of the originating particles. The dataset used for this task, the HLS4ML LHC jet dataset, contains images of simulated jets along with associated jet features.

1.1 Background and theory.

The jets we aim to study are collated streams of particles that arise from the fragmentation and hadronization of high-energy quarks and gluons

produced in particle collisions. These collisions, which occur at facilities like the Large Hadron Collider (LHC), involve highly energetic protons or heavy ions accelerated to near-light speeds and then collided together. Upon collision, the immense energy involved can lead to the creation of quark-antiquark pairs or gluons, which subsequently undergo fragmentation and produce sprays of particles travelling in approximately the same direction. These sprays of particles form what is known as a jet as seen in figure 1.

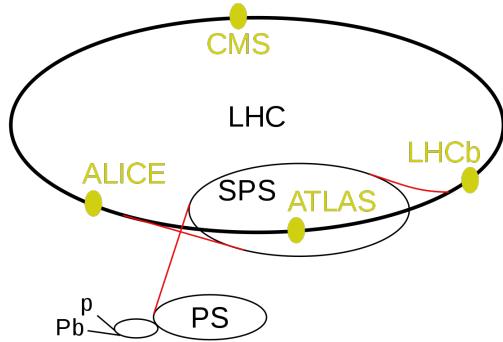


Figure 2: The Large Hadron Collider (LHC). ALICE: A Large Ion Collider Experiment, ATLAS: A Toroidal LHC Apparatus, CMS: Compact Muon Solenoid

Jets play a crucial role in the study of fundamental particles and forces. The LHC collides protons at extremely high energies, allowing physicists to probe the subatomic structure of matter with unprecedented precision. A diagram of the LHC can be seen in figure 2. When protons collide at such high energies, they produce a multitude of particles, including quarks and gluons. These particles undergo a process called hadronization, where they transform into a shower of stable hadrons, such as protons, neutrons, pions, and other particles. The resulting sprays of hadrons can form a jet, which are detected by the sophisticated instrumentation of the LHC's particle detectors.

The HCAL (Hadronic Calorimeter) is primarily designed to measure the energy of hadrons. Hadrons interact with matter primarily through the strong nuclear force. The HCAL typically consists of layers of dense materials, such as steel or brass, interspersed with sensitive detectors, such as plastic

scintillators or quartz fibers. When a high-energy hadron passes through the HCAL, it deposits its energy in this dense material, causing ionisation and the emission of secondary particles. These secondary particles then interact further with the sensitive detectors, producing signals that can be used to measure the energy of the original hadron. On the other hand, the ECAL (Electromagnetic Calorimeter) is specifically designed to measure the energy of electromagnetic particles, such as electrons and photons. Electromagnetic particles interact primarily through the electromagnetic force, which leads to processes like bremsstrahlung (radiation emitted when charged particles are accelerated) and pair production (the creation of an electron-positron pair from a high-energy photon). The ECAL typically consists of dense materials, such as lead or tungsten, followed by sensitive detectors, such as crystals or photomultiplier tubes (PMT). When an electromagnetic particle passes through the ECAL, it deposits its energy predominantly through electromagnetic interactions, causing the emission of photons or the creation of electron-positron pairs. These interactions, like in the HCAL, produce signals which can in turn be used to measure the energy of the original particle.

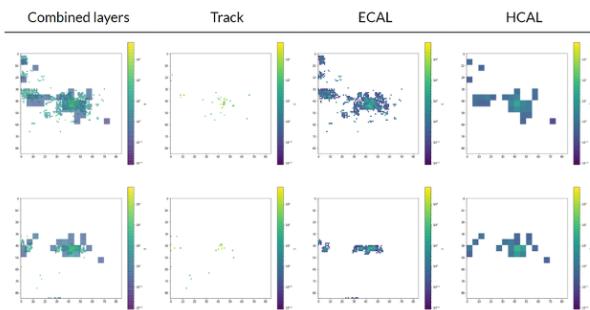


Figure 3: HCAL and ECAL images

1.1.1 The application of machine learning

A neural network consists of layers of nodes, often referred to as artificial neurons. These layers are structured into an input layer, one or more hidden layers, and an output layer. Within each layer, neurons are interconnected, and these connections

are weighted, determining their significance in the network's operations. Additionally, each layer incorporates a bias, typically manifesting as an additional neuron with no connections. This intricate architecture is inspired by the complex workings of the human brain. Through the intricate interplay of these layers and nodes, neural networks can learn and adapt, mimicking the cognitive processes observed in biological organisms. Consequently, the fields of 'machine learning' and 'artificial intelligence' have emerged, reflecting humanity's quest to replicate and understand the remarkable capabilities of the human mind.

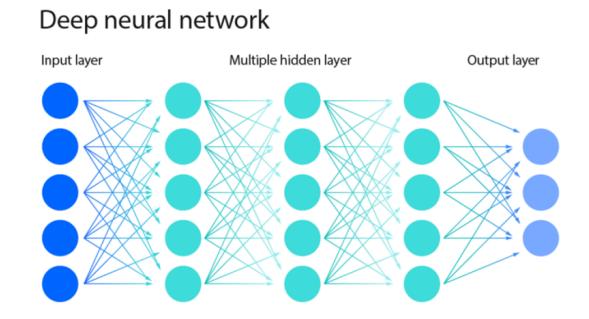


Figure 4: Shows a neural network.

The weighting equation governs the strength of connections between neurons. Each connection between neurons is associated with a weight that determines its impact on the network's output to the following neuron. Mathematically, the weighted sum of inputs to a neuron is calculated using the following equation:

$$z = \left(\sum_{i=1}^n x_i \times w_i \right) + b \quad (1)$$

$$\text{output} = f(z) \quad (2)$$

Figure 1: Equation: (1) The weighting equation and (2) the activation function

In equation 1, w_i represents the weight of the connection between the i th input neuron and the current neuron, x_i denotes the value of the i th input neuron, n is the total number of input neurons, and

b represents the bias associated with the neuron [4].

The activation function (equation 2) f takes the weighted sum z as input and produces the output of the neuron. Common activation functions include sigmoid, tanh, ReLU (Rectified Linear Unit), and softmax. Each of these functions serves a different purpose depending on the network architecture and the nature of the task.

The activation function introduces non-linearity into the network, enabling it to approximate complex functions and capture intricate patterns which are hidden within the data. This non-linearity is crucial for the network to learn and generalise effectively. Linear transformations alone would restrict both the rate and capacity to which a neural network can learn.

Loss minimisation is a very important concept in training neural networks. It involves adjusting the model's parameters, such as weights and biases, to minimise the difference between the predicted outputs and the actual targets. The measure of this difference is quantified using a loss function, also commonly known as a cost function.

The choice of loss function depends on the nature of the task and the type of data being used, in regression tasks, mean squared error (MSE) is commonly used shown in equation 3.

$$\text{LOSS} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (3)$$

Where:

- LOSS: mean squared error
- n : number of data points
- Y_i : observed values
- \hat{Y}_i : predicted values

Regardless of the specific function chosen, the goal remains the same: to minimise the discrepancy between prediction and target values. To achieve this, optimisation algorithms, such as stochastic gradient descent (SGD) and its variants, are used. These algorithms iteratively update the model parameters in the direction that reduces the

loss most rapidly. The gradient of the loss/cost function with respect to each parameter indicates the direction of steepest descent, guiding the optimisation process. Figure 5 shows a possible loss gradient, with multiple local minima and a global minimum. Although SGD algorithms may not always find the global minimum, efficiently finding a local minimum will often suffice.

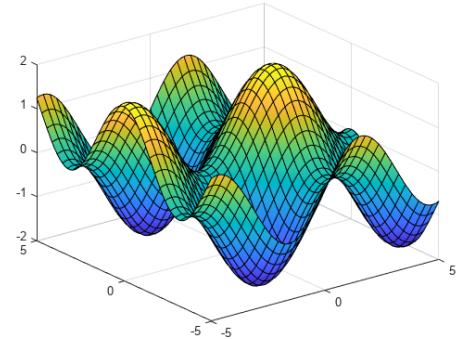


figure 5: Loss against model parameters, showing local minima and maxima

During training, the model is presented with training examples, and its predictions are compared against the actual targets for validation. The resulting error is then back propagated through the network, and the parameters are adjusted accordingly to reduce the loss.

By minimising this loss function, neural networks learn to better capture the underlying patterns in the data and generalise to unseen examples, thus 'training' the neural network.

It is important to strike a fine balance between model complexity and generality. Overly complex models may overfit the training data, leading the model to perform worse on test data it has not yet seen. On the other hand, an oversimplified model may fail to capture important patterns within the data, again leading to worse performance.

1.1.2 Convolutional Neural Network (CNN)

Convolutional neural networks are a type of regularised feed forward neural network, that are op-

timised specifically for image processing [5]. Feature extraction enables the identification of intricate patterns by analysing local segments of data, such as images, within individual layers. The primary components of CNN architecture include convolution, pooling, fully connected layers, and a mix of activation functions as shown in figure 6.

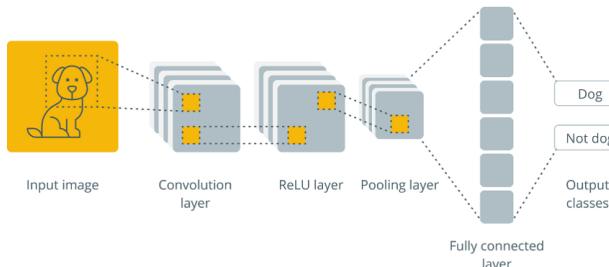


Figure 6: A convolutional neural network

In the convolution layer, a matrix filter calculates the dot product with segments of the input data, across the entire input, generating a feature map that discerns patterns within the image. This architecture's key advantage lies in its spatially independent feature detection, this is achieved by applying filters across all sections of the image rather than the entire dataset. Pooling layers serve to reduce the dimensionality of the feature matrix, also reducing the risk of overfitting by consolidating neighboring pixels into a more compact feature map, as illustrated in Figure 6. Preceding the final fully connected (dense) layers in CNNs, a flatten layer collapses the data into a single dimension. The flattened data is then processed by fully connected layers, which combine the insights from convolutional and pooling layers to generate predictions usually into what's called a 'one-hot' matrix, a one dimensional matrix summarising the prediction outcome.

1.2.3 The LHC jets dataset

The LHC dataset contains images and associated metadata of 10,000 independent jet events within the LHC. Associated with each of these jets is a

label found within the metadata, identifying it as one of the 5 jet classes. These jet labels correspond to the origin of the jet, i.e. by which particle the jet was initiated. Jets initiated by each class typically exhibit certain characteristics in their energy distributions and substructure.

The dataset contains three types of images:

- HCAL images which are images taken inside the hadronic calorimeter.
- ECAL images taken inside the electromagnetic calorimeter.
- Combined images, which collate both the HCAL and ECAL images.

As we will see it is often useful to separate the two types of image in the pursuit of separating the various jet classes, and more generally gaining insight into how to teach a neural network to separate these classes.

The dataset also contains 54 Jet features (not including their respective labels) within the metadata. These features will allow us to quantify differences between our jets and are extremely important when it comes to jet separability. For example, Transverse Momentum (j_{pt}) and Transverse Momentum Fraction (j_{ptfrac}): these 2 features are measures of the jet's momentum perpendicular to the beam axis. They provide information about the jet's energy and are fundamental in characterising its properties and therefore its origins.

Another important feature is Pseudorapidity (j_η), which is a measure of the angle between the jet's momentum and the beam axis. It provides information about the jet's direction in the detector. Throughout this investigation, a relative importance will be attempted to be assigned to each of these features to gain further insight into which considerations are conducive to a higher jet separability within a machine learning context.

2. Methods Introduction

The primary objective of this project is to build a machine learning classifier capable of separat-

ing the different classes of jet events. The separability, a measure of the ease with which each class is separated and the accuracy with which our classifier performs this task will also be investigated. Through various methods of statistical analysis, the link between the various jet features and the efficiency of our classifier will be examined. An attempt will be made to try and understand the degree to which the different jet feature variables have an effect on our classifier. This project will also include some extensions, centred around investigating the use of alternative data with which to separate the jets including separated HCAL and ECAL images, and numerical data associated with the jet features by themselves.

2.1 Task 1: Combined Image Classifier

In this initial task, a classifier is built to separate the jets into five classes based on the particle by which they were initiated, using a CNN network to be trained on only the combined images of the jets.

2.1.1 Data Pre-processing

When formatting training data for a CNN, it is essential to prepare the data in a suitable format. One common approach is to use one-hot encoding for the labels, especially in classification tasks. One-hot encoding represents categorical data as binary vectors, where each class is represented by a vector with all zeros except for one at the index corresponding to the class label. The labels existing in the metadata are stored in a binary format within the metadata, thus a one hot classifier function was built. This classifier assigns a number between 0 and 4 to each jet, creating a label array, which was then converted to vector format within a 2D nested array.

The distribution of each jet classes can be seen below, since there are not the same number of jets in each class, the results later will have to be normalised, although the resultant bias from this de-

viation is small it is better to normalise the images later than to exclude data from the classifier, as to do so would be to the detriment of the classifier's performance.

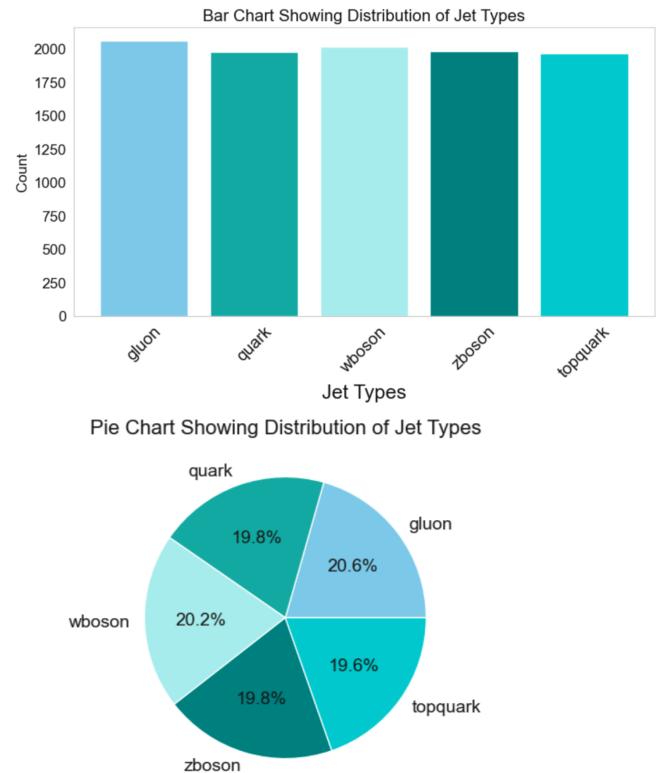


Figure 7 (top) and 8 (bottom): Bar chart and pie chart showing distribution of jet classes

The images were firstly processed using contrast normalisation, which refactored the pixel values to range from zero to one. This was done for several reasons, firstly this helped to improve convergence, ensuring that the input data is on a similar scale which is more suited to optimisation algorithms such as the stochastic gradient descent. Normalisation also helped with better generalisation and improved interpretability of the data.

The images were split into training validation and testing data in a ratio of 70:15:15 respectively. This particular ratio was chosen after trial and error led to higher training and validation accuracy scores, and a more rapid reduction in training and validation loss. It is also important to have as large

a dataset as possible. The validation set was used to improve the generalisation ability of the neural network improving its performance on unseen data.

2.1.2 Model Architecture

When designing a machine learning image classifier for the task of classifying five distinct types of events based solely on combined jet image data, several key considerations influenced the design of the model architecture.

The input to the model consists of 100x100-pixel images, passed through a Convolutional Neural Network (CNN) architecture. Initially, a Conv2D layer with 15 filters of size 5x5 and ReLU activation is applied, preserving spatial dimensions. Subsequently, an AveragePooling2D layer with a 5x5 filter size is applied to reduce dimensionality while retaining important features. Following this, another Conv2D layer with 15 filters of size 5x5 and ReLU activation is applied, maintaining spatial dimensions. A dropout layer with a dropout rate of 0.2 is incorporated to prevent overfitting by randomly deactivating neurons during training. The data is then flattened to be fed into a fully connected Dense layer with 65 units and ReLU activation, facilitating the extraction of high-level features. Finally, a Dense layer with 5 units and softmax activation is employed to produce class probabilities. The model is then compiled using the Adam optimiser.

The decision to incorporate multiple convolution layers was a natural one, as they enable the model to effectively capture spatial hierarchies of features within the jet image data. Following this, fully connected layers were introduced to tackle the complexity and variability of the images, allowing the model to perform high-level feature learning and classification based on the representations learned from the convolutional layers. To address issues such as overfitting and to reduce dimensionality, pooling layers were added. Dropout layers were also integrated to enhance generalisation by preventing co-adaptation of feature detectors. In determining the activation functions

and regularisation techniques, inspiration was drawn from a similar study separating particle jets [6], which dealt with a similar multiclassification problem. The adoption of these techniques proved to be effective in handling image data of this nature. Activation functions such as ReLU followed by softmax were selected for their simplicity and efficacy in introducing non-linearity to the model. This aids in faster convergence during training and empowers the model to capture intricate relationships and gradients. The choice of the Adam optimiser was driven by its efficiency and effectiveness in optimising neural networks. Its ability to adapt learning rates for each parameter individually made it well-suited for the task at hand.

2.1.3 Testing performance

The model was trained and tested, the accuracy and loss data from training and validation are shown below in figure 9 and 10. The training parameters consisted of using the default learning rate of 0.001 and a batch size of 35, trained over 35 epochs.

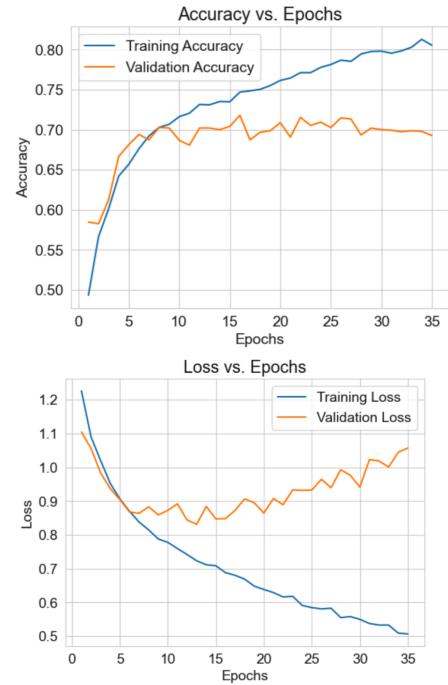


Figure 9 (top): Accuracy vs Epoch, Figure 10 (bottom): Loss vs Epoch

2.1.4 Separability Performance Analysis

In the context of our aims within this task, performance is defined as the ability of our neural network to separate the jet images into one of 5 classes accurately.

To investigate the separation ability and the efficiency of the neural network a confidence matrix was constructed, each tile represents the probability (confidence) assigned to each instance, normalised by the actual frequency of positive outcomes. This allows for a visual interpretation of the predicted probabilities align with the results from test data, and therefore an indication of which classes are easiest to separate.

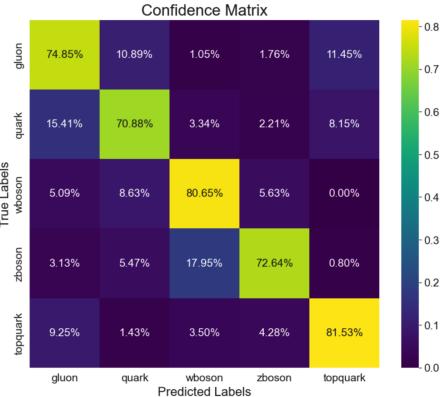


Figure 11: Confidence Matrix

Additionally, a confusion matrix was plotted, which provides the counts of true positives and true negatives, false positives, and false negatives for each class, this was once again normalised.

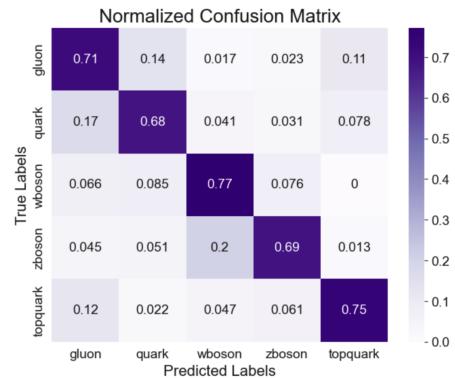


Figure 12: Confusion Matrix

Based on the results from both the confidence matrix and the confusion matrix, several conclusions can be drawn regarding the ability of the model to separate jet classes. The top quark seems to be the easiest to separate, with high accuracy and low instances of false positives (wrong guesses when the class is a top quark). This is followed by W-boson and then the z-boson. While the model performs well at separating gluons from the other jet classes, with the second highest recall rate, this is accompanied by a high number of false positives, namely quarks. This means that the model often mistakes features which belong to quarks as belonging to gluons, leading to relative low separability performance when it comes to identifying a quark jet.

The receiver operating curve (ROC) is a good metric of the model's ability to discriminate a given jet class from the rest. Since the ROC is designed to be used in binary classification problems, figure 13 shows 5 lines, one for each jet class. This metric is defined in terms of the true positive rate (TPR) and false positive rate (FPR).

$$\text{TPR} = \frac{\text{True Positives}}{\text{Total Positives}} \quad (4)$$

$$\text{FPR} = \frac{\text{False Positives}}{\text{Total Negatives}} \quad (5)$$

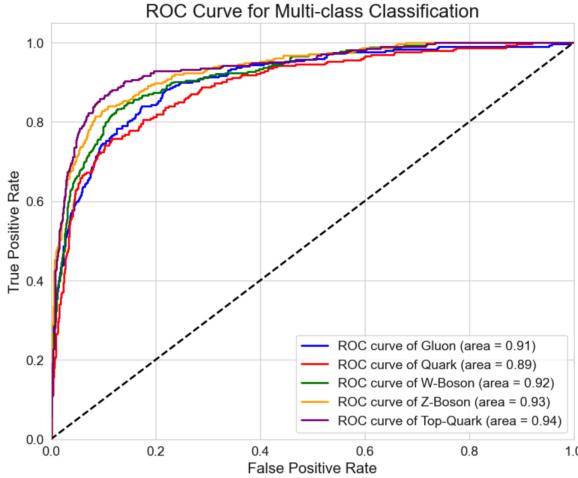


figure 13: ROC curve

```
47/47 [=====] - 2s 35ms/step
Micro-average AUC: 0.9161768888888888
Macro-average AUC: 0.9164131962204813
```

Table 1 : Average AUC

The ROC curve is a graph of the TPR against the FPR, as the decision threshold is varied. The diagonal line, for reference shows a curve resulting from a random classifier which has an area under the curve (AUC) of 0.5. Generally, the closer the curve is to the top left hand corner of the plot the more effective of a classifier it is, with the perfect classifier resulting in an AUC of 1.

We can see that the AUC scores from the ROC curve align well with the separability rankings drawn from both the confusion and confidence matrices, with the Top-quark being the easiest to separate and the Quark jet being the most difficult to identify correctly. It is worth noticing that this is not a binary classification problem and therefore an AUC of 0.5 is not the (random) baseline for performance but rather 0.2 due to there being 5 classes. This means that an average AUC of 0.91 indicates that this classifier is much better than random at sorting the jets.

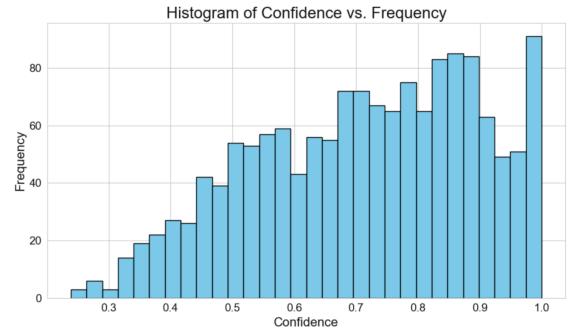


Figure 14: Confidence distribution

A histogram was plotted showing the distribution of confidence in guesses made by the model. The graph seems to suggest that there is a subset of jet images which are easily separable, leading to an output from the neural network which takes the form of a high confidence output. The vast majority of the rest of the jets fall within a less certain confidence range of approximately $0.5 < x < 0.85$. So this seems to indicate that there are certain properties within these jet images, visible to the human eye or not, which seem to divide jets into groups those with high and low implied separability. The next subsection of our investigation seeks to identify what these certain properties are, and how they affect the efficiency of our classifier. Identifying properties which lead to a high separability would allow extensions to be made to the classifier and a higher performance achieved.

2.1.5 Separability Principal Component Analysis

In order to understand how the jet variables effect the efficiency of our classifier it is worth considering the use of PCA (Principal Component Analysis). PCA is a fundamental technique used in data analysis, and more generally it plays a crucial role in understanding complex relations in large datasets.

PCA was chosen with the aim of gaining insights into the underlying structure and relationships within the high-dimensional feature space which is the jet variables dataset. By reducing the dimensionality of the data, PCA helps uncover the

most significant patterns and trends while retaining as much variance as possible.

PCA transforms the original variables into a new set of uncorrelated variables, these are called principal components, and they are ordered by the amount of variance they explain.

These principal components represent orthogonal directions in the feature space, allowing for a more concise representation of the data while preserving essential information about its variability and structure. Through PCA, more insight can be gained into how the jet variables ultimately allow us to separate the jets, and the degree to which certain variables can do this, if they even can. PCA can help uncover the key factors influencing the efficiency of the classifier.

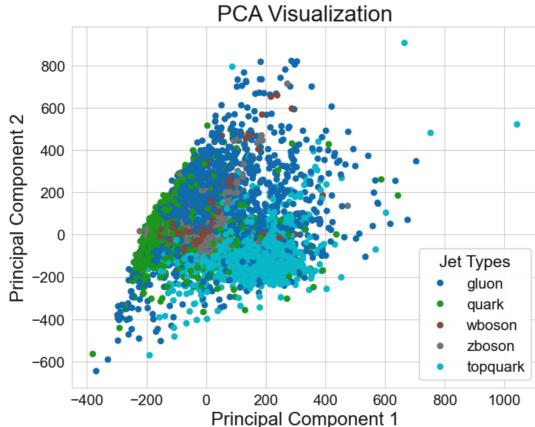


Figure 15: PCA chart

By randomly conducting PCA on certain groups of features it is easy to see underlying patterns in the data. Figure 16 shows the jets plotted along 2 principal components with low separability, the data is clustered together seemingly without any pattern or underlying relation.

Figure 17 shows the same jets plotted along 2 different principle components. As one can see the principle components in figure 17 are a more effective at separating the jets into their different classes, with patterns emerging through jet discrimination

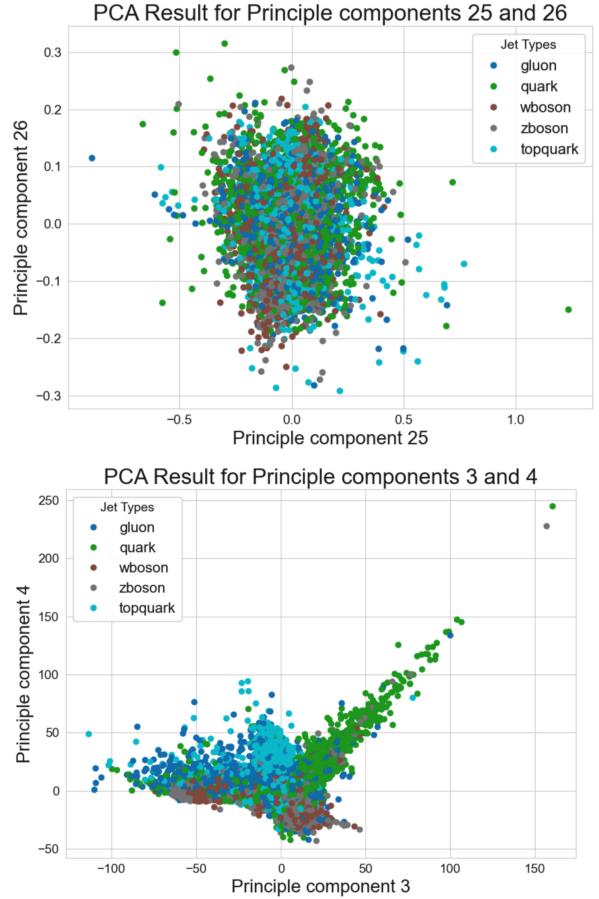


Figure 16 (top): PCA with low separability , Figure 17 (bottom): PCA with higher seperability.

The variance in separability seems to suggest that different jet feature variables, and more importantly different combinations of jet features carry different levels of importance or weight when it comes to the separability of the jets. An attempt will now be made to understand which features bare most importance when it comes to separability, and their effect on the efficiency of our image classifier will be analysed.

2.1.6 Efficiency Relation to Jet Features

To try and gauge how the efficiency (performance) of the model depends on the various jet features. The variance from the average value of each feature was calculated, under the assumption that

separability was linked to a deviation from this feature average.

The top 5 percent of variance was separated from the rest and then the TPR (true positive rate or recall) was calculated for these points. The TPR or recall scores are then summed together for each feature as a measure of implied ‘importance’. This is graphed in figure 18.

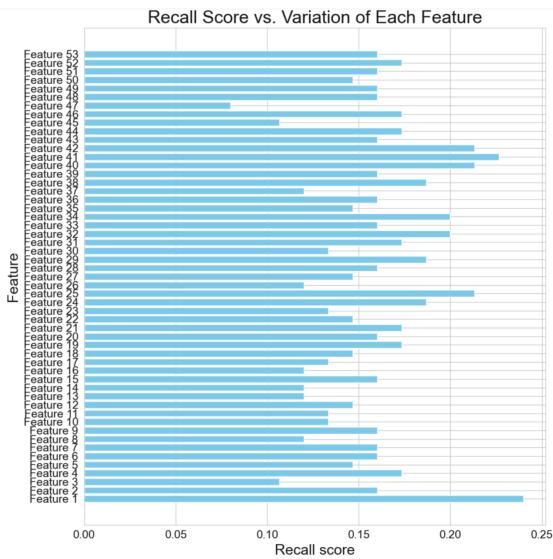


Figure 18: Recall score for each feature

Although this analysis provides some insight into the variance in feature importance, it does not give a direct insight into how the variance in jet features affect how the classifier makes its guess.

To try and gain more insight into the efficiency of the classifier the same process was repeated, this time with the confidence scores of each jet. The deviations where then plotted against the confidence of the guess for the deviation of that particular jet variable. The distribution of a random feature is plotted against confidence in figure 19.

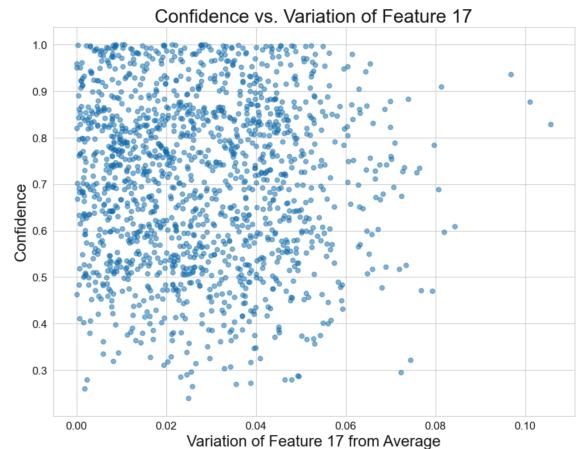


Figure 19: Confidence vs a feature variation

Although in figure 19; most of the points were grouped around the average, this graph, and similar graphs for other features seemed to suggest that at higher values of variance there was a correlation between jet feature variation and confidence output. The same graph was plotted for all 54 features, this time taking only the top 15, 10 and 5 % of variance values for each feature, a trend line was added.

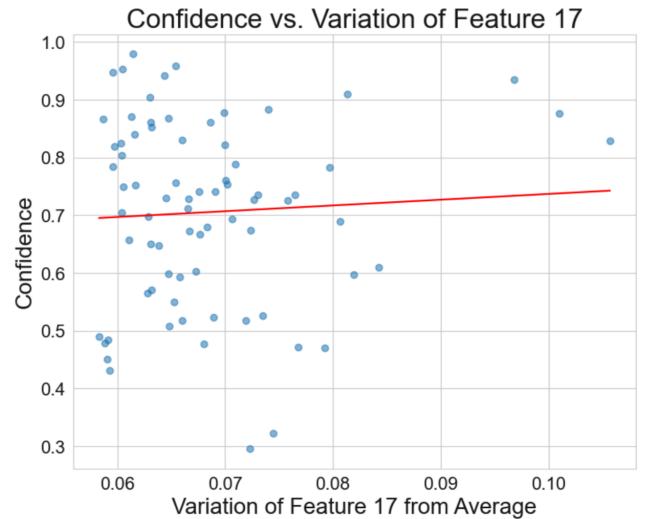


Figure 20: Distribution of top 10% of variance values against confidence for feature 17

The gradient of this trend line therefore acts as a quantitative measure of the relation between the weighted confidence of guesses and the variance for that specific feature. The positive gradient suggests that deviations or fluctuations in the value of

this feature lead to a higher confidence within our classifier. Therefore, it is reasonable to say that the value of the gradient can be assumed to be a normalised measure of the implied importance of that feature. The value of this gradient line gives us an indication of how the model has made guess, and how this the numerical value of that guess is tied to weighted by each feature. This ‘implied importance’ which is nothing more than the gradient of the fitted line in figure 20 can be seen for each feature in figure 21. The model was then run again, and the same calculation was made for each feature, as graphed in figure 22. As can be seen, the same jet features where highlighted as having a high implied importance, namely jet feature 15,16 and 37 seem to be of particular interest to the classifier.

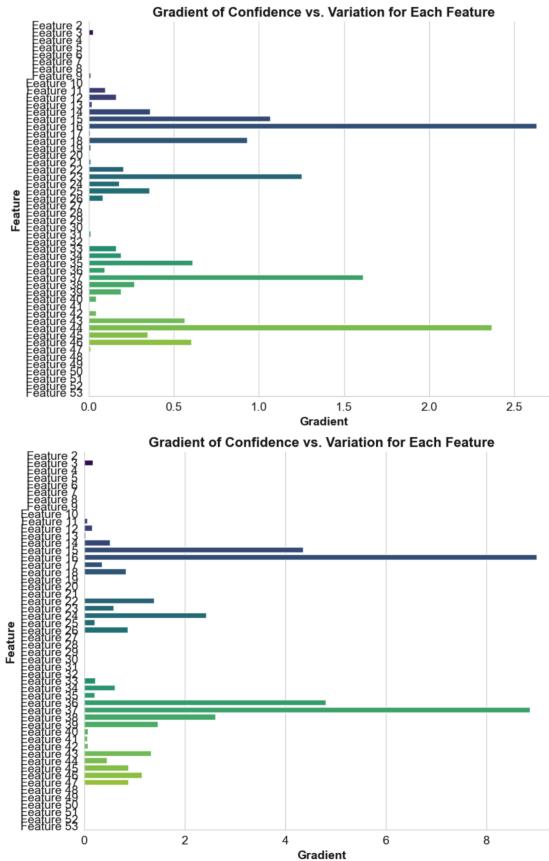


Figure 21 (top): Set 1 of Relative importance for each feature. Figure 22 (bottom): Set 2 of Relative importance for each feature

These results were suggest a much more signif-

icant deviation in implied importance than those derived from the TPR scores in figure 18. Note that there were several instances where an increase in the variance of certain features led to a decrease in the confidence of a guess and that these values of negative gradient were not included in figures 21 and 22 as they were small and dismissed as noise. Building a machine learning classifier that is trained on these features alone is a separate task that is covered in the extensions.

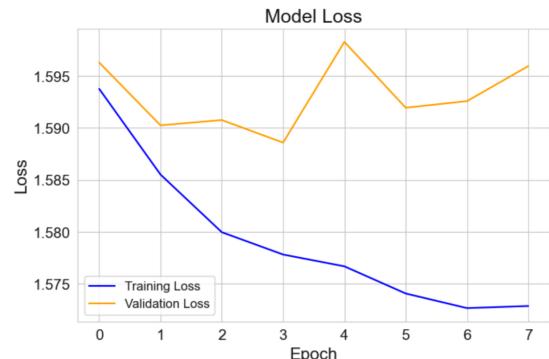
3. Extensions

The extension of this project involves training a neural network on alternative data. This included training the same neural network separately on the separated HCAL and ECAL images which were taken inside the hadronic and electromagnetic calorimeters respectively. A separate model will also be built to test the efficiency of a classifier which is trained only on the numerical data associated with the jet features, and isn’t trained on any of the image data.

3.1 Separating the HCAL and ECAL images

A model with identical architecture to the one trained in 2.1.2 was firstly trained on the ECAL images with a training and testing split ratio of 75:25.

The loss and accuracy results after training and testing the model can be seen below in figure 23 and 24.



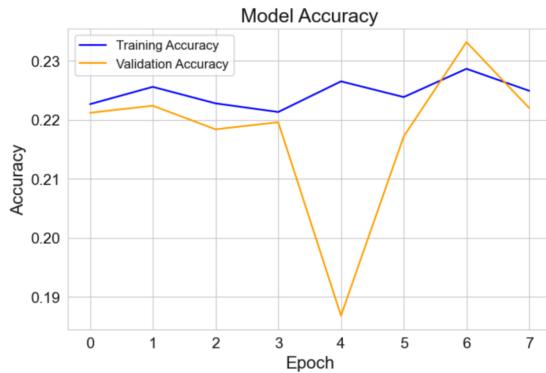


Figure 23 and 24: Loss and accuracy for HCAL trained model

As we can see this classifier performed very poorly only achieving a maximum test accuracy of 23% which is only very slightly better than a random classifier which we would expect to have an accuracy of 20%.

Next, the same model was retrained on images from the HCAL dataset. The corresponding accuracy and loss results are shown below in figure 25 and 26.

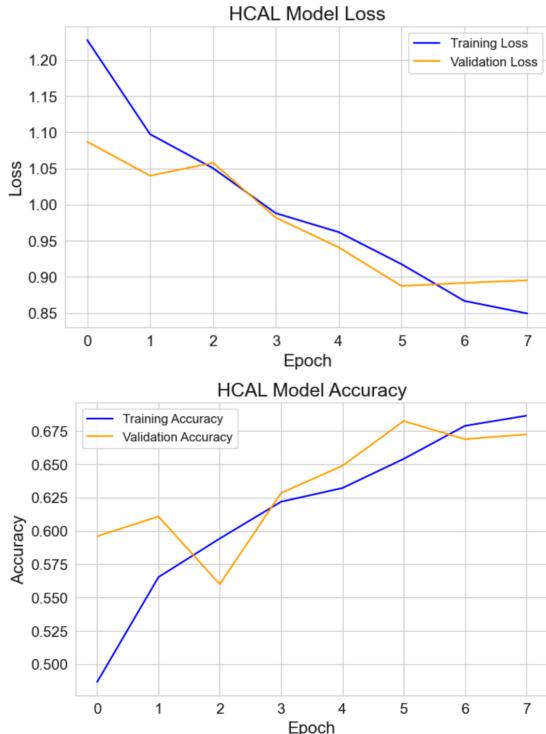


Figure 25 and 26: Loss and accuracy for HCAL trained model

As can be seen from comparing the models trained independently on the HCAL and ECAL images, the classifier which has been trained on the HCAL images outperforms the other, matching the efficiency of our classifier discussed in 2.1.4.

This is most likely due to the fact that hadronic jets are far more likely to create detectable activity within the much denser layers found within the HCAL and far less likely to create electromagnetic interactions within the ECAL.

3.2 Combined Classifier

For this section a combined classifier was built which was trained on the ECAL images, the HCAL images and the original combined images from 2.1.3. The 3 datasets were merged together and shuffled and trained on the same model as in 2.1.2 as a basis of comparison. The results from this classifier were not different from the original results or the results from the HCAL classifier, apart from the fact that this model took significantly longer to run.

3.3 Jet Features Classifier

In 2.1.5 the relation between the various jet features and the behaviour of the classifier was explored. Now a separate model will be trained on the data from the jet features metadata. This is an array with 54 values, not including the labels. It is important to remove the last 5 elements of this jet feature array from each jet as otherwise data-leakage would be encountered as the model would train itself to learn from the jet labels rather than adjusting its weights efficiently. Such a problem manifests itself as perfect accuracy and loss scores, with the model suffering when it comes to real world application.

MLPs (Multi-Layer Perceptrons) were chosen for this task due to their versatility and effectiveness in handling various types of data. MLPs are capable of effectively learning complex non-linear relationships in tabular data, making them suitable for tasks such as classification and regression. Their architecture, which consists of multiple layers of neurons, again with non-linear activation

functions, allows them to capture intricate patterns and features from the input data.

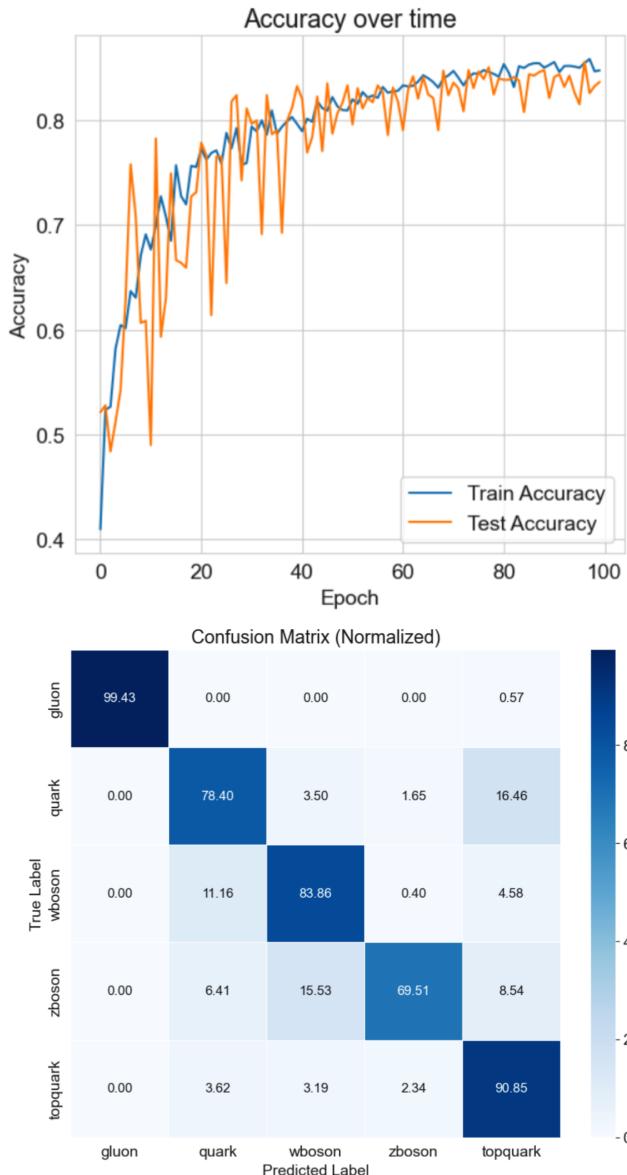


Figure 27: (top) Accuracy vs Epoch for the jet features classifier. Figure 28: (bottom) Confusion matrix for the jet features classifier

As can be seen from figure 27 and 28 the jet classifier based on only the numerical data outperforms all the other image classifiers previously covered in this investigation. This classifier easily averaged a test accuracy of 85%. This is most likely due to the fact that jet classification often comes

down to identifying characteristics within energy distributions, something which a neural network does more successfully through numerical data rather than an image. Furthermore the jet features dataset has higher dimensionality than than the images do. In general neural networks perform much better when dealing with numerical tabular data. It can also be seen that the individual separability of the jet classes differs across the confusion matrix when compared to our image model. The jet features model seems to be able separate gluons with an accuracy of 99.43%. Interestingly enough both the image and jet feature classifiers perform similarly when it comes to the task of separating the Z-boson.

4. Discussion and Conclusion

To conclude, the aims of task 1 and 2 where fulfilled. A machine learning classifier was successfully built which was able to separate the 5 classes with an average accuracy of 73%. An ROC curve was plotted with the 5 jets having a high TPR, low FPR and an average AUC of 0.91 which is quite satisfactory considering a relatively small dataset of only 10,000 jets was used both for training and testing. Further analysis of confidence and recall rates across the 5 classes when cross referenced with the ROC curve gave a strong indication of separability across the 5 classes. This allowed us to conclude which classes where most easily separated as well as which classes caused our model some confusion.

Driven by varying separability found in principle component analysis, further analysis into how the efficiency of our model relates to the jet features led to examining the relation between confidence and recall scores and the fluctuation of various jet feature values. A range of correlations was found between jet features and the confidence with which our model made its guesses. This give us an idea of which variables are linked with the behaviour of the classifier.

To conclude this investigation comprised of an investigation into the factors surrounding the behaviour of the jet classifier, rather than only an attempt to maximise its performance. Although

a future investigation could continue from the insights gained from this one with the soul focus of achieving a much higher separation accuracy, the focus of this project was to investigate how neural networks separate the jets, and what factors are at play.

In a future investigation several things could be done differently to allow for a more rigorous investigation. Firstly more data could be used, 10,000 jets is a relatively small amount and there is much more data available. Secondly the link between the jet features highlighted in 2.1.5 could have been further investigated within the extension 3.3 by comparing performance on the highlighted features compared to random features. Additionally different network architectures could have been investigated with the aim of measuring their effect on the different types of images as well as the separability of each jet class.

5. References

- [1] Evans, L.R. (2009). The Large Hadron Collider. EPFL Press.
- [2] Djouadi, A., Kalinowski, J. and Spira, M. (1998). HDECAY: a program for Higgs boson decays in the Standard Model and its supersymmetric extension. Computer Physics Communications, [online] 108(1), pp.56–74
- [3] Kogler, R., Nachman, B.P., Schmidt, A., Asquith, L., Winkels, E., Campanelli, M., Delitzsch, C.M., Harris, P., Hinzmann, A., Kar, D., Mclean, C., Pilot, J., Takahashi, Y., Tran, N.V., Vernieri, C. and Vos, M. (2019). Jet substructure at the Large Hadron Collider. Reviews of Modern Physics, 91(4)
- [4] Weisburd, S. (1987). Neural Nets Catch the ABCs of DNA. Science News, 132(5), p.76
- [5] Nebauer, C. (1998). Evaluation of convolutional neural networks for visual recognition. IEEE Transactions on Neural Networks, 9(4), pp.685–696.
- [6] Lee, K., Mulligan, J., Mateusz Płoskoń, Ringer, F. and Yuan, F. (2023). Machine learning-based jet and event classification at the Electron-Ion Collider with applications to hadron structure and spin physics. Journal of High Energy Physics, 2023(3).