Prepared for submission to PHAS0056

Classifying Jets at the LHC using Machine Learning Techniques

Abstract: Accurate jet tagging is of crucial importance at the LHC, where millions of collisions per second must be identified to study the properties of the particles and forces responsible for the observed jets. This paper presents a study of various neural network architectures including convolutional neural networks and multilayer perceptrons, used on jet calorimiter image representations as well as measured kinematic particle features to classify jets into being initiated by gluons, (light) quarks, W/Z bosons or top quarks. The most accurate model presented yields satisfactory performance (0.98 F1 score), offering promising LHC applications.

Contents

1 Introduction **2**

1.1 The Large Hadron Collider 2

1.2 Neural Networks 3

1.3 Dataset Description 5

2 Models Considered **7**

2.1 Standard CNN Model 7

2.2 CNN MLP Model 8

2.3 Multi-view CNN 8

3 Model Training **9**

4 Results **10**

4.1 CNN Performance 10

4.2 CNN + MVMLP 11

4.3 MVCNN Performance 12

4.4 MVMLP Performance 13

4.5 MVCNN + MLP Performance 13

4.6 Comparing Image Inputs and Feature Inputs 14

4.7 Which Classes of Events are Easiest to Separate? 14

4.8 How Does Efficiency Depend on the Features? 15

5 Conclusion **16**

A High Level Features **17**

1 Introduction

At the Large Hadron Collider (LHC), proton-proton collisions generate narrow cones of hadrons and other particles produced from the hadronization of light partons (quarks and gluons) known as jets [1]. At high energies, these jets can also be formed from the decay of heavy particles such as W and Z bosons and top quarks [2]. Jet tagging is the process of studying particle jets and identifying the particle (or particles) which initiated the jet [3]. Given the extreme amounts of data generated in these collisions and possibilities of new physics emerging from the study of high energy particle decays, machine learning techniques are becoming increasingly prevalent in streamlining the jet-tagging process [4]. This project attempts to apply different types of neural networks on image and kinematic representations of jets to accurately classify their origins as either gluons, (light) quarks, W bosons, Z boson or top quarks.

1.1 The Large Hadron Collider

At the LHC at CERN, 2 beams of protons bunches are accelerated in opposite directions in a synchrotron and collide together with a collision rate of 40 million collisions per second. Each beam has an energy of 6.5 Tev, giving a total centre of mass energy $\sqrt{s} = 13$ TeV [5]. These high energies lead to the creation of heavy, unstable particles, such as the Higgs (H) boson [6]. Only discovered in 2013, huge computing power was required to analyse the experimental data and make statistically significance inferences. Jet tagging was instrumental in resolving the background noise for Higgs decay, allowing physicists to isolate the events which most likely came from a Higgs decay.

Jet tagging reconstructs the events leading to the creation of an observed particles. Accurate classification of jets is essential to understanding the underlying physics behind each event. Fig. 1 illustrates how jet tagging works at the LHC. Gluon and quark jets are most abundantly formed at the LHC and make up the majority of the Quantum Chromodynamic (QCD) multijet background [7]. Jets from W/Z bosons and top quarks differ from the parton jets due their heavier masses, often decaying into 2 or more clusters, illustrated in Fig. 2

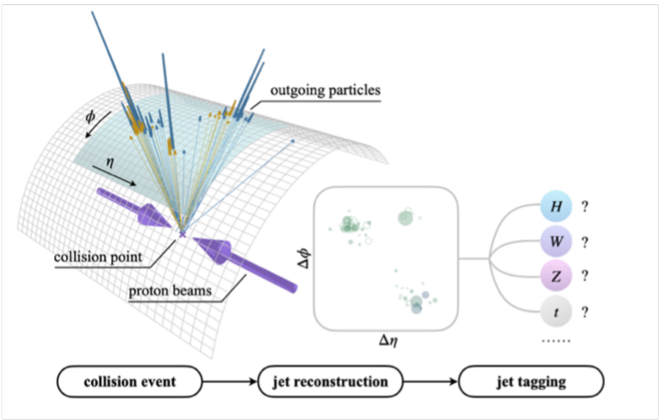


Figure 1: Jet tagging at the LHC. 2 proton beams collide head on, producing particles that decay into collimated sprays of particles. These particles are the ones detected by detectors, and jets are reconstructed from these, e.g. an image representation of azimuthal distance and psuedorapidity, and tagged to classify their origin, e.g. Higgs boson, W or Z boson or top quark [4].

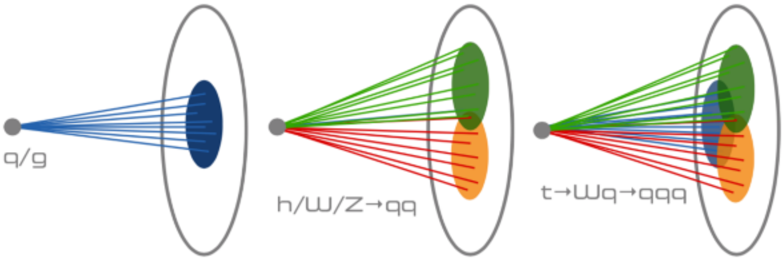


Figure 2: Pictorial representation of how the different type of particle jets look. The parton jets are collimated in a single cone shaped beam (left). W/Z jets are composed of 2 cluster (middle) and top quarks are made of 3 (right) [8].

1.2 Neural Networks

Neural Networks (NNs) (also known as Artificial Neural Networks (ANN)) are models in machine learning composed of artificial neurons which are connected together by edges, modelled after the structure of neurons and synapses in a brain. Artificial neurons take inputs (numbers) from the neurons feeding into them, process the information, then feed the outputs (also numbers) to the neurons they feed into. To enable the NN to develop complex representations and introduce non-linearity, an activation function is applied to the weighted sum of the outputs before being passed onto the next neuron [9]. Typical activation functions used are ReLu, Sigmoid and Softmax [10].

In a typical NN, signals go from an input layer all the way to an output layer, where between them there are multiple layers composed of interconnected neurons known as the hidden layers, where weights are applied to each input, and they are passed through activation functions. The architecture of the network; the number and size of hidden layers, types of activation functions, presence of backpropogation and feedback loops, determine the efficiency and accuracy of the network. In this study, accuracy is defined as the ability of the network to positively identify the origin of a jet based on features inputted into the network. Metrics such as accuracy, loss and F1 scores define how well models perform in their tasks [11]

Convolutional Neural Networks (CNNs) are a class of feed forward deep learning models, specifically designed to use a large number of network layers to learn and train from image data [12]. CNNs consists of convolutional layers, which identify features of the images, such as vertices, edges, and colours. In these convolutional layers, dot products are performed between a kernel and the layers input matrix [13]. The kernal size is smaller than the matrix representation, so the kernel will slide across the height and width of the image (striding), producing a feature map which retain important image features [14]. Fig. 3 shows how an image, convolutional filters and kernel look in a CNN. The outputs are passed through activation functions (e.g. ReLu) and fed into pooling layers which reduce the dimensionality of the input by clustering nearby neuron outputs together, to decrease computing power required, but preserve the important features of the image. Further layers may include dropout and batch normalisation to help prevent overfitting of data [15].

To classify an image into one of the available classes, fully connected dense layers connect the convolutional outputs to the output layer, where the image is classified. For multiclass classification, the Softmax activation function is the last activation function used. It normalises the output if the network to a probability distribution of the input belonging to each one of the available classes [16]. The networks presented in this paper all finish with a Softmax activation function.

CNN's suffer from requiring large computation resources and time to achieve acceptable

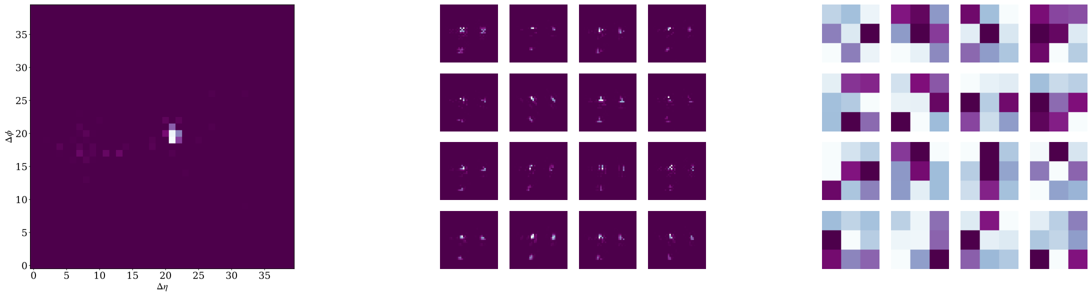


Figure 3: **Left**: An example jet image (cropped from 100 x 100 to 40 x 40 pixels), taken as an input in the CNN. **Middle**: The 16 feature maps after the image is passed through a convolutional layer with 16 3 x 3 filters. **Right**: The 16 3 x 3 convolutional filters.

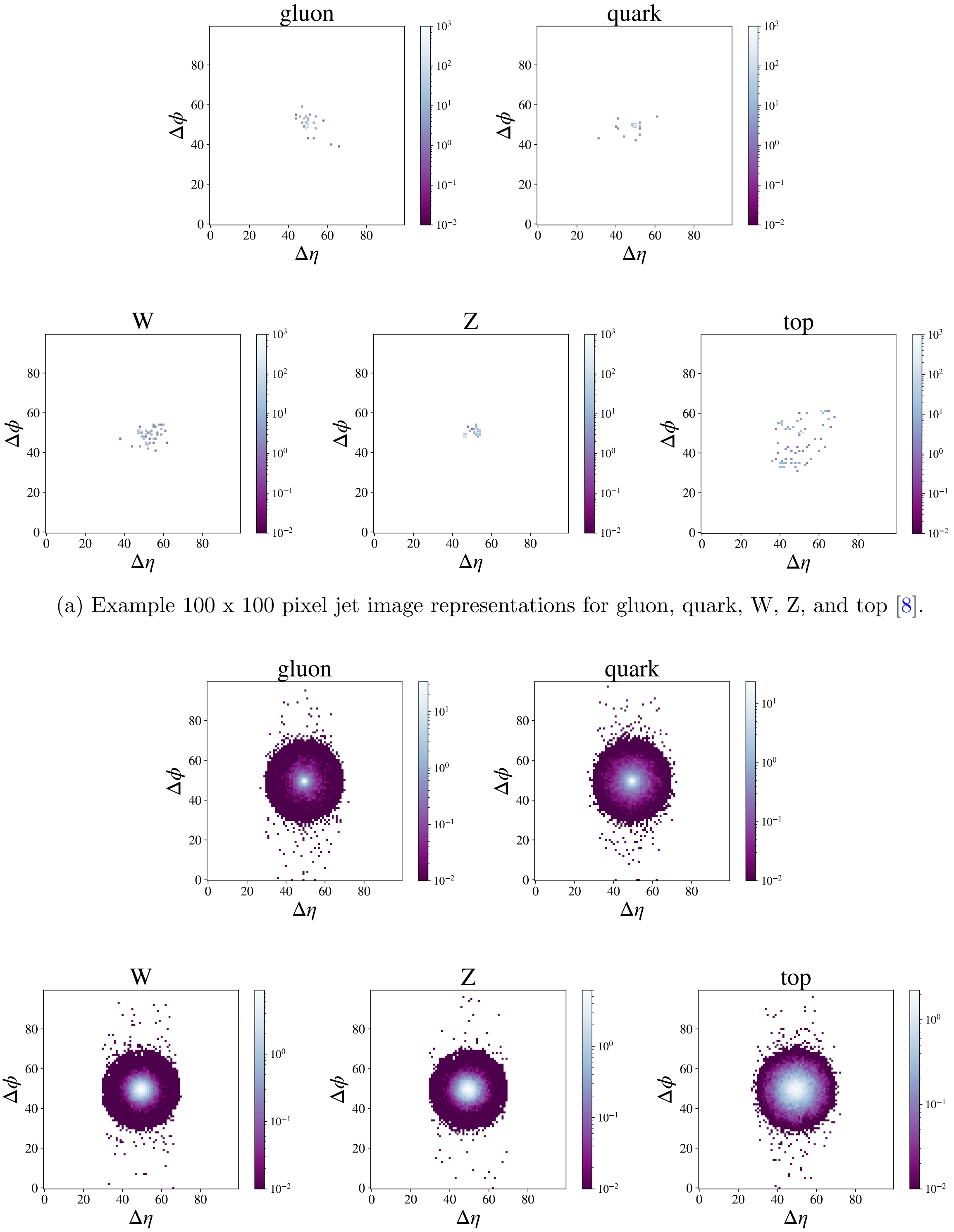
accuracy, given the calculations required on the inputs. In some classification tasks, CNN's will also struggle to differentiate between image classes when there are only slight differences in the image representation, requiring a trade-off between complexity and time efficiency. In addition to a standard CNN model, we propose a multi-input model which takes image representation as well as jet features to investigate the consequences on training accuracy and efficiency.

1.3 Dataset Description

The dataset consists of high transverse momentum (p_t) jets originated from gluons, g, light quarks, q, W bosons, Z bosons and top quarks, t, produced from simulations of LHC proton-proton collisions with $\sqrt{s} = 13$ TeV. The data was created for the FastML/HLS4ML study and is available on Zenodo [17]. 3 representations of the jets are used in this study:

1. A 100 x 100-pixel image representation of the jet as a square of lengths $\phi =$ azimuthal distance and $\eta =$ psuedorapidity. These images are used to train a CNN to classify the jets into belonging to one of the 5 jet classes. Fig 4 represents how a jet may look for each class of particle, and how the means jet images compare. The images are produced from the highest 100 p_T constituents for each jet.

2. A list of high level features (HLFs), used in conjunction with the image representation in a multi-input network. The distributions of 6 HLFs for each of the 5 particle classes are shown in Appendix A.



(a) Example 100 x 100 pixel jet image representations for gluon, quark, W, Z, and top [8].

(b) Example average 100 x 100 pixel jet image representations for gluon, quark, W, Z, and top.

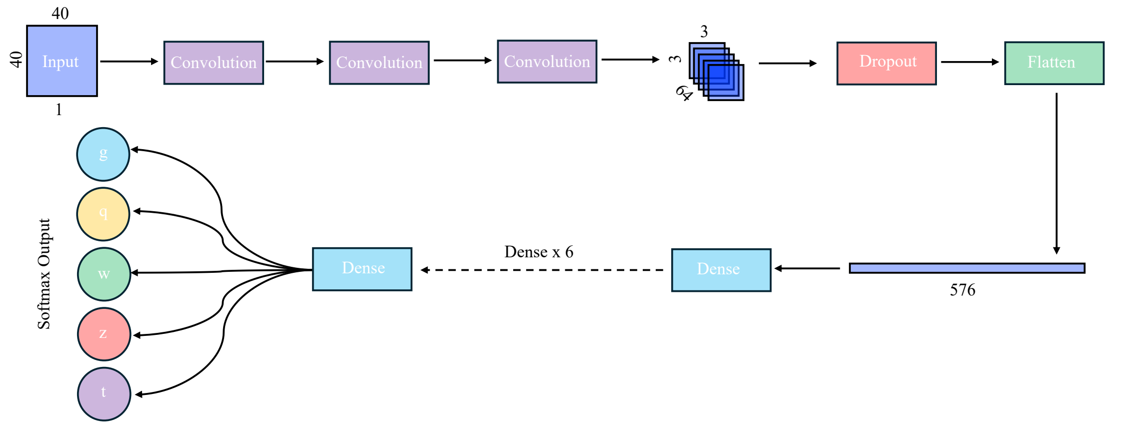
Figure 4: Example jet image representations. The pixel scale is the amount of p_T collected in each cell, in GeV, measured as the scalar sum of p_T of particles pointing into each cell.

3. 100 x 100-pixel image representations of the deposits on the Electromagnetic and Hadronic Calorimeters (ECAL and HCAL), used in combination with the jet image in a 3-input network.

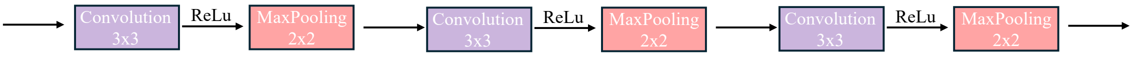
2 Models Considered

2.1 Standard CNN Model

The first model considered is a simple CNN trained on the jet image representations. The 100 x 100 pixel images were cropped to 40 x 40 pixels, centered arround the same position, to reduce the computational load on training and training time. It was found that this Dimensionality reduction process did not hinder the networks ability to classify the images. The model architecture is described in Fig. 5



(a) CNN model architecture used in this study taking only the jet image as inputs. 3 convolutional layers were deemed the optimal amount to attain an acceptable test accuracy while not introducing a large number of parameters or potential of overfitting. In the 8 dense layers, ReLu and sigmoid activation functions were used to introduce non-linearity and help the model learn abstractions in the images.



(b) Architecture of the convolutional layers, with number of neurons: 32, 64, 64.

Figure 5: CNN model architecture

As a 40 x 40 x 1 pixel image passes through the convolutional layers, its Dimensionality becomes (19,19,32), (8,8,64), then (3,3,64), before being flattened to a 1D array of size 576 and then fed into the 8 dense layers.

2.2 CNN MLP Model

A Multilayer Perceptron (MLP) is a type of feedfoward ANN consisting of fully connected neurons where each in a layer connects to all the neurons in the following layer with a certain weight [18]. While CNN use convolutional filters to identify spacial features in the pixels of images, MLPs are notable for being able to distinguish data that is not linearly separable [19], such as the HLF's described in 1.3. The HLF's are incorporated by concatenating a MLP network with the standard CNN described to investigate how the efficiency of the classifier depends on the HLF's. Fig. 6 illustrates the MLP branch of the CNN MLP network.

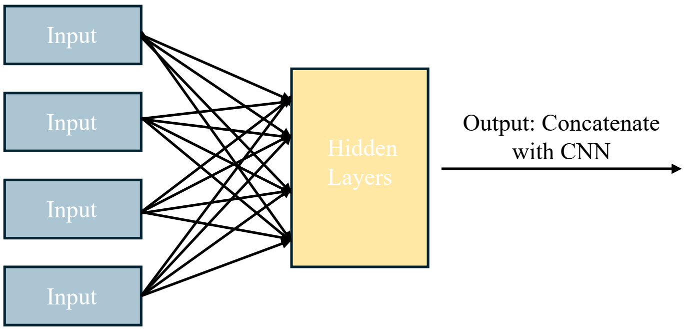


Figure 6: MLP network architecture. 4 inputs shown for illustration purposes. The hidden layers consist of 2 dense layers with ReLu activation functions. After concatenation with the CNN, further dense layers are applied.

2.3 Multi-view CNN

Multi-view CNN's (MVCNN) aim to combine different views or representations of an image to yield more efficient classifier models [20]. MVCNNs have become increasingly prevalent in the last few years, especially in 3D object recognition, where the CNN's state-of-the art performance in 2D image classification/recognition, allows for aggregation of many 2D images to classify 3D images. Examples of MVCNN works include 3D fingerprint matching [21], grain classification [22] and breast cancer screening [23], highlighting the wide applications of such networks. Taking the JetImage, HCALImage and ECALImage as the inputs, the architecture is similar to Fig. 5, where the 3 image inputs are convoluted 3 times then passed through a dropout and a flatten layer. These 3 inputs are then concatenated together before passing through the remaining dense layers, shown in Fig. 7.

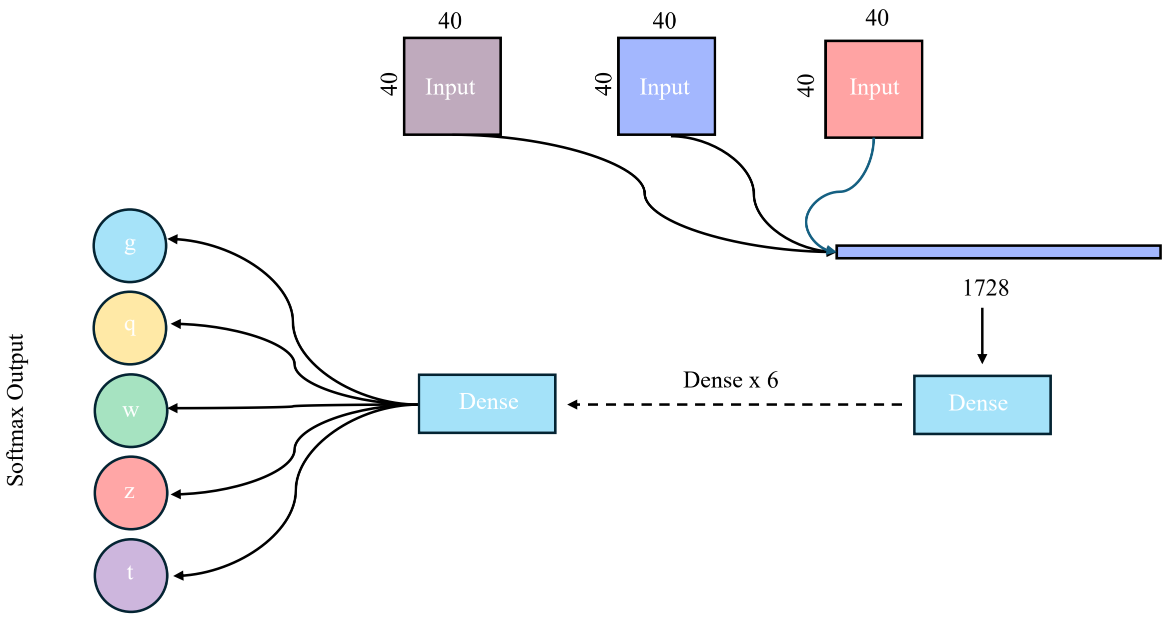


Figure 7: MVCNN model architecture. Each branch is individually fed through 3 convolutional filters then concatenated. The resulting 1D array has size 3 times greater than the 1D array from the CNN model, owing to 3 times as much pixel input.

We will also consider a classifier that takes only the HLF's as inputs, a multi-view MLP, as well as a combined classifier that takes in the 3 image representations and the HLF's, a multi-view CNN + MLP.

3 Model Training

8 Files were used in the training and testing of the models, consisting of 80,000 images split 56,000 for training, 12,000 for validation and 12,000 for testing. Each model was trained for 50 epochs and used the default batch size of 32 and standard learning rate for the Adam optimiser of 0.001. As previously mentioned, the images were cropped to 40 x 40 pixels. The HLF's we normalised using sklearn StandardScalar feature to speed up learning and allow for better weight initialisation in the propagation process. It was found that there was a mismatch in sample number of ECAL images to HCAL and jet images, there we many blank ECAL images, possibly due to particles not leaving a significant electromagnetic signature. The counter the mismatch, ECAL images were randomly duplicated to resize the data to match the other image array shapes.

Early stopping was initially implemented with a patience of 5 epochs, but later removed to allows all models to have the same number of training epochs for better comparison. It was found that 50 epochs did not significantly increase the likelihood of overfitting, with validation loss continuing to decrease, and where overfitting did occur, it was taken into consideration.

4 Results

The 5 models trained were a CNN taking only the jet image input, a combined CNN and MVMLP taking the jet image input and 16 HLFs, a MVCNN taking the 3 image representations, a MVMLP taking only the 16 HLFs and finally a combined MVCNN and MVMLP taking the 3 image representations and 16 HLFS. Table 1 summarises the classification results of each model.

Model	**Metric**				
	No. of Params.	Epoch time[s]	Loss	Accuracy	Avg. F1 score
CNN	**270,581**	19	0.643	0.767	0.73
CNN + MVMLP	663,941	25	0.075	0.967	0.95
MVCNN	401,285	43	0.170	0.938	0.90
MVMLP	460,741	**7**	0.141	0.935	0.89
MVCNN + MVMLP	1,136,133	56	**0.047**	**0.980**	**0.98**

Table 1: Metrics used to gauge accuracy and efficiency of the 5 models tested, with bold indicating the optimal ones.

The MVCNN + MLP, the most complex model with the highest number of trainable parameters and taking the longest to compile, was the most accurate, correctly classifying the jets 98% of the time. The simplest model, the CNN, was the least accurate but had the least number of trainable parameters and the second least accurate, the MVMLP took the shortest time to train. This highlights the trade-off required between accuracy and computing load with machine learning techniques.

4.1 CNN Performance

A models performance is most widely characterised by it loss and accuracy. The loss (or cost) quantifies how the models predictions vary from true values. Validation data/labels correspond to a set of input and labels that haven't been trained by the model, providing an unbiased view of how the model performs on unseen data [24]. A model with increasing/decreasing training accuracy/loss at higher epochs but stagnating validation accuracy/loss is a possible indication that the model has begun overfitting. For a good model, training accuracy and loss should steadily increase/decrease.

For multiclass classification, confusion matrices are also a good visualisation of the model's ability at differentiating between the classes [25]. For 5 particle types, a 5x5 matrix is plotted. The main diagonals represent true positives, and the value is the true positive rate for that class. The off diagonal row elements represent the proportion of the row class being misidentified as the corresponding column class.

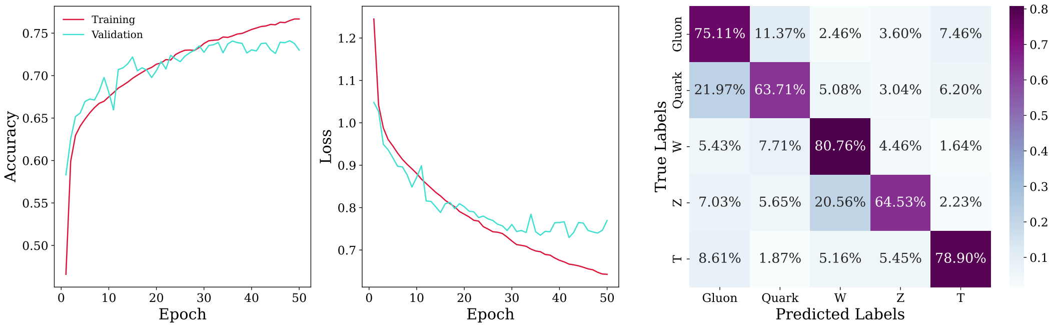


Figure 8: CNN model.**Left:** Accuracy against epoch. **Middle:** Loss against epoch. **Right:** Confusion matrix with percentage probabilities, darker indicates a greater true positive rate and more accurate class classification. The CNN model begins overfitting after approximately 30 epochs, as indicated by the plateau in validation.

The CNN achieves satisfactory accuracy of 77%. From the confusion matrix, many gluons and quarks are misidentified as eachother and many Z bosons are misidentified as W bosons.

4.2 CNN + MVMLP

The 16 HLFs where p_T, η, jet mass, multiplicity, $z \log z$ (a splitting function), $\tau_3 b_1$, $d_2 b_2$, $d_2 a_1 b_2$, $n_2 b_2$, $n_2 b_1$, $tau_{32} b_1$, $c_1 b_1$, $c_2 b_1$, trimmed mass, mMDT mass (modified MassDrop Tagger [26]) and sdb2 mass (soft drop mass with $\beta = 2$ [27]). Combining these with the single jet image representation yields results as shown in 9.

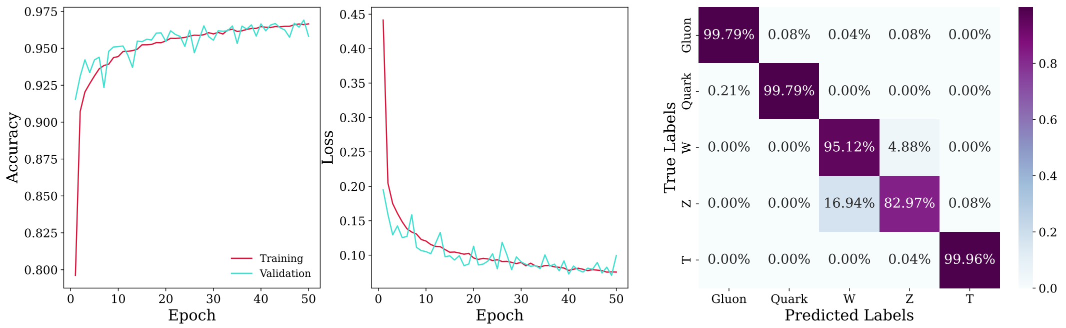


Figure 9: CNN + MVMLP model.**Left:** Accuracy against epoch. **Middle:** Loss against epoch. **Right:** Confusion matrix with percentage probabilities.

4.3 MVCNN Performance

The performance of the MVCNN taking 3 image inputs is shown in Fig 10.

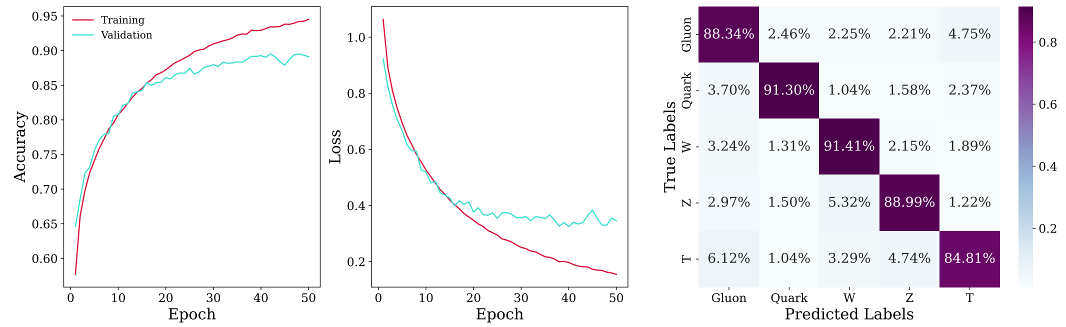


Figure 10: MVCNN model.**Left:** Accuracy against epoch. **Middle:** Loss against epoch. **Right:** Confusion matrix with percentage probabilities.

The MVCNN achieves a higher accuracy than the CNN, 94% and a much lower loss, 0.17. Like the CNN, overfitting does begin occuring at 20 epochs, earlier than the CNN possibly due to the 3 image inputs which signficantly increase the models complexity and no. of parameters (Table 1), allowing to model to more quickly learn image features, but also makes it more susceptible to memorising the training data rather than generalising from it.

From the confusion matrix, there are a lot less misidentifications than the standard CNN. The additional ECAL and HCAL images provide useful representations for the model which helps it do differentiate between images and classify more accurately.

4.4 MVMLP Performance

Taking only the 16 HLFs inputs and no image representations led to the MVMLP taking the shortest time to train, but still achieved satisfactory accuracy of 94%. Fig. 11 shows the results.

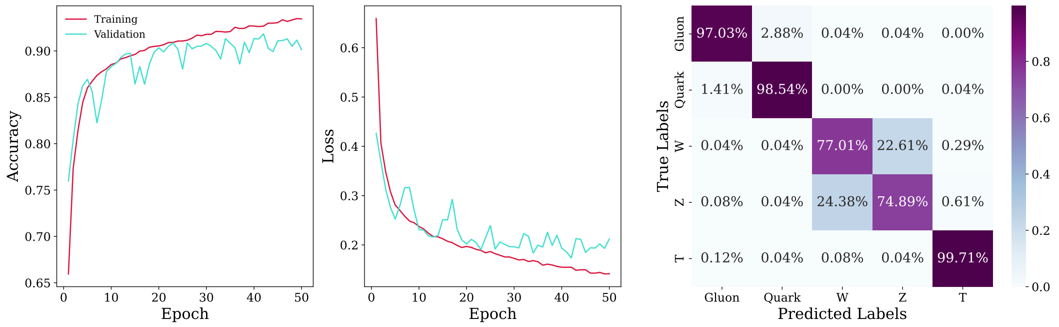


Figure 11: MVMLP model**Left:** Accuracy against epoch. **Middle:** Loss against epoch. **Right:** Confusion matrix with percentage probabilities.

There is less overfitting with this model than with the CNN models. The model is highly accurate at classifying gluons, quarks and top, but struggles with W and Z, misclasssifying them as eachother for almost 25% of W and Z jets.

4.5 MVCNN + MLP Performance

The most accurate model, with the most parameters and longest training time, it is no surprise incorporating 3 image representations and 16 HLFs yields 98 % classification accuracy, summarised in 12.

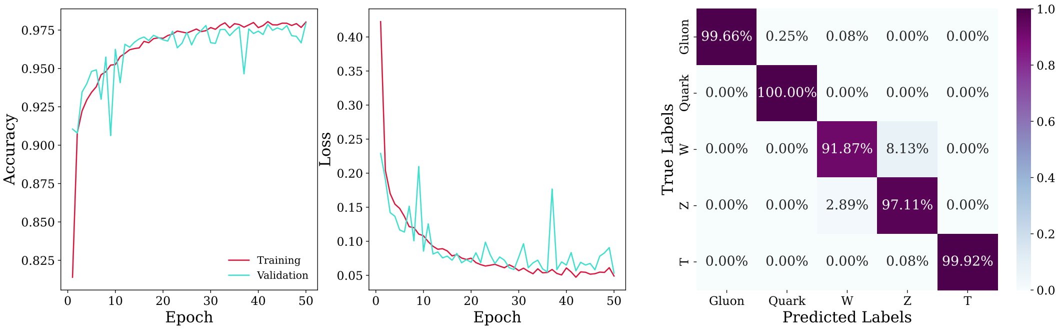


Figure 12: MVCNN + MVMLP model.**Left:** Accuracy against epoch. **Middle:** Loss against epoch. **Right:** Confusion matrix with percentage probabilities.

Gluons and top jets are correctly classified $> 99.6\%$ of the time, and all quark jets were correctly identified. Once again, it is the misclassification of W and Z jets as eachother that hold the model back, but never the less the model still achieves the highest accuracy and F1 score and lowest loss.

4.6 Comparing Image Inputs and Feature Inputs

Comparing the model that only takes the 3 image inputs (MVCNN) to the model that only takes 16 HLF inputs (MVMLP), both achieve a similar accuracy, 93%, similar loss, $0.1 \geq$ loss ≤ 0.2 and similar number of parameters, about 400,000. Where they differ is the training time, 7 seconds per epoch for the MVMLP, 8 times as fast as the MVCNN. The reason for this maybe due to the complexity in the tasks performed by the models. Convolutional layers in the MVCNN require significant amounts of computation, while the dense layers in the MVMLP perform relatively simple matrix calculations on already flattened 1D data.

4.7 Which Classes of Events are Easiest to Separate?

The Receiver Operator Characteristic (ROC) curve is a plot of the True Positive Rate against the False Positive Rate. Fig. 13 shows the ROC curve for each of the 5 classes of events.

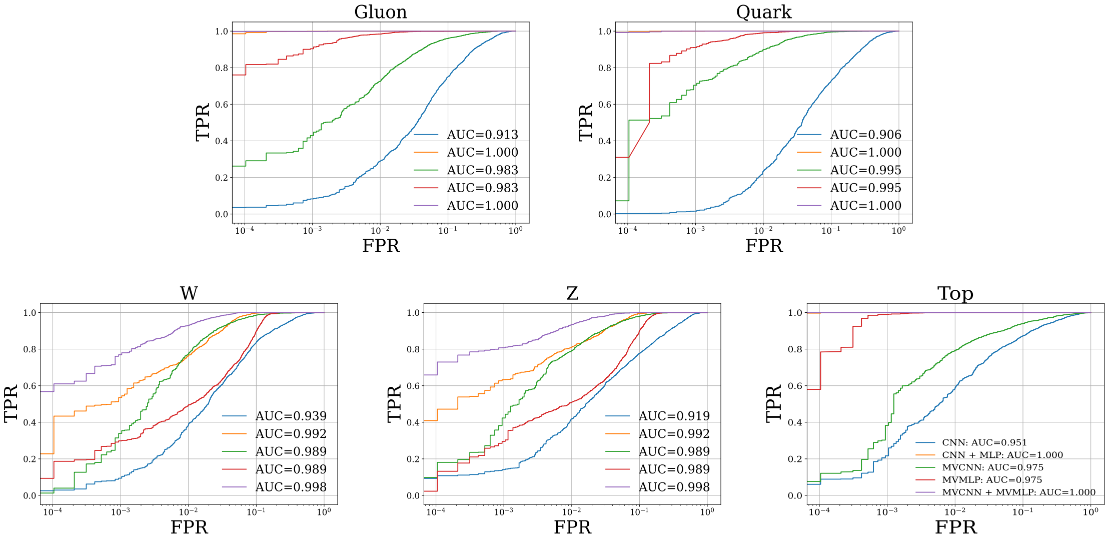


Figure 13: ROC curves with area under curve (AUC) displayed for each type of event, for each of the 5 models.

The shape of the curves and AUC give a means of comparison between the classes of events and how easy they are separated. A class which is very easily classified will have a curve which rapidly approaches TPR = 1, such as gluon, quark and top for the CNN + MLP and MVCNN + MVMLP models. By the shape of the curves for W and Z and their smaller AUC values, these events are the hardest to accurately classify. The class most easily classified is top.

4.8 How Does Efficiency Depend on the Features?

Focusing on 5 features, p_T, η, jet mass, multiplicity, $z \log z$ yields the results summarised in Table 2 and Fig. 14.

Feature	**Metric**			
	Epoch time[s]	Loss	Accuracy	Avg. F1 score
Image	19	0.643	0.767	0.73
p_T	19	0.547	0.799	0.75
η	20	0.665	0.760	0.71
Mass	20	0.291	0.885	0.86
Multiplicity	20	0.319	0.880	0.82
$z \log z$	19	0.390	0.857	0.78

Table 2: Metrics from model trained on jet images + only of HLF, with the CNN model (only jet image) in the first row included for comparison.

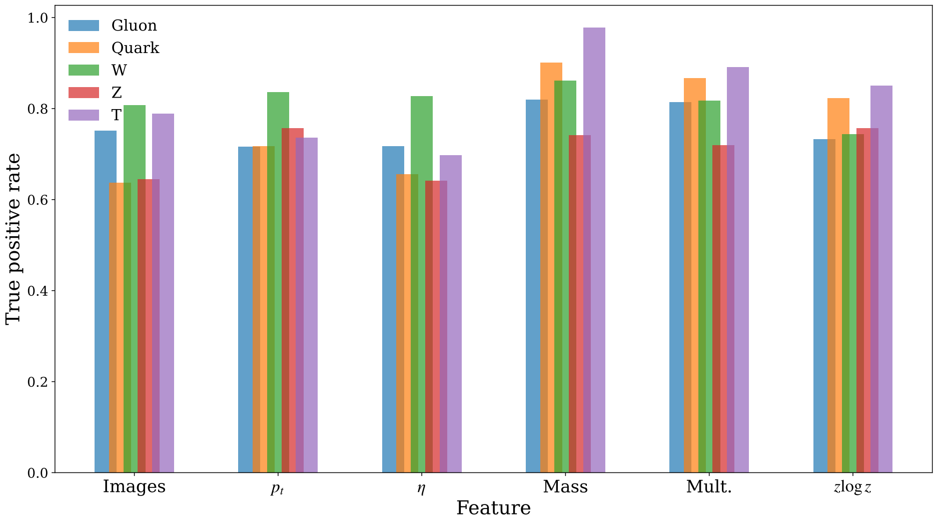


Figure 14: True positive rate for each class for models incorporating the jet image + 1 HLF (CNN model included for comparison).

Jet mass is the feature that provides the greatest assistance to the model in accurate classification. Unsurprising given how mass is distributed, with there being significant differences between the 5 particle types.

5 Conclusion

5 models were tested on the LHC data, using image and feature representations in a variety of combinations to attempt to classify jets into belonging to quarks, gluons, W, Z or top. Satisfactory performance (Accuracy > 0.90) was achieved on the 4 models which took more than a single type of input. The most accurate model was the MVCNN + MVMLP combination which took 3 image representations and 16 HLFs, achieving an accuracy of 90% and F1 score of 0.98, but also requiring the most computing resources and time.

The classification process could be improved through the use of an Interaction Network (IN), such as the JEDI jet identification algorithm proposed by Morena et al [8]. Improved accuracy in jet tagging gives means for distinguishing between ordinary jets from the hadronization of light partons (gluons, quarks) and hadronic decays of high energy particle (W, Z top). Prospects of physics beyond the Standard Model lay with these high energy particles, and the LHC is at the forefront of experimental particle physics, so these jet tagging algorithms and machine learning models have vast practical applications.

A High Level Features

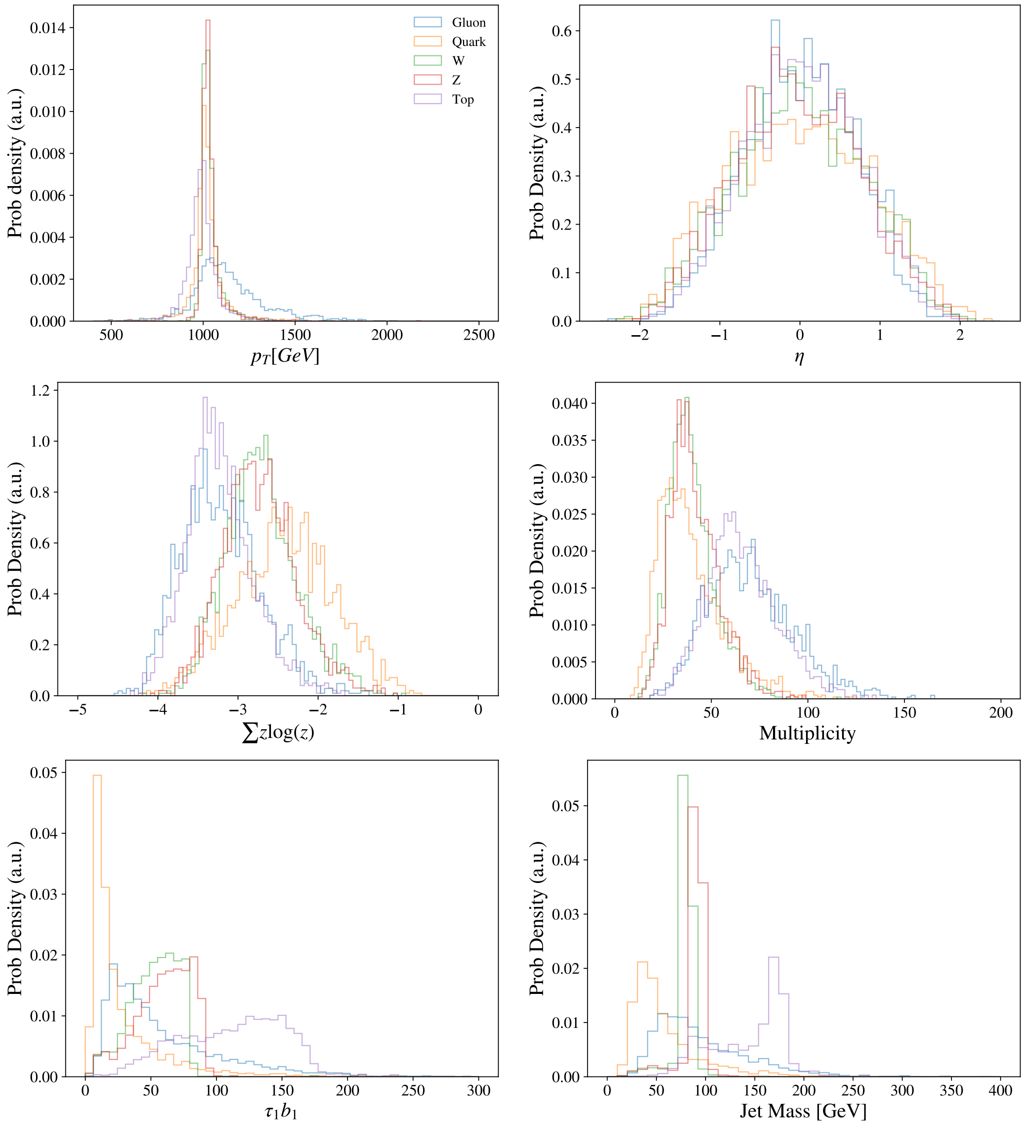


Figure 15: Distributions of some of the HLFs used in network training, further described in [28].

References

[1] Andrew J. Larkoski, Ian Moult, and Benjamin Nachman. Jet substructure at the large hadron collider: A review of recent advances in theory and machine learning. *Physics Reports*, 841:1–63, January 2020. ISSN 0370-1573. doi: 10.1016/j.physrep.2019.11.001. URL http://dx.doi.org/10.1016/j.physrep.2019.11.001.

[2] Leandro G. Almeida, Seung J. Lee, Gilad Perez, George Sterman, Ilmo Sung, and Joseph Virzi. Substructure of high pt jets at the lhc. *Physical Review D*, 79(7), April 2009. ISSN 1550-2368. doi: 10.1103/physrevd.79.074017. URL http://dx.doi.org/10.1103/PhysRevD.79.074017.

[3] Josh Cogan, Michael Kagan, Emanuel Strauss, and Ariel Schwarztman. Jet-images: computer vision inspired techniques for jet tagging. *Journal of High Energy Physics*, 2015(2), February 2015. ISSN 1029-8479. doi: 10.1007/jhep02(2015)118. URL http://dx.doi.org/10.1007/JHEP02(2015)118.

[4] Huilin Qu, Congqiao Li, and Sitian Qian. Particle transformer for jet tagging, 2024.

[5] Cian O'Luanaigh. First successful beam at record energy of 6.5 tev [accessed march 2024]. URL https://home.cern/news/news/accelerators/first-successful-beam-record-energy-65-tev.

[6] ATLAS Collaboration. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, September 2012. ISSN 0370-2693. doi: 10.1016/j.physletb.2012.08.020. URL http://dx.doi.org/10.1016/j.physletb.2012.08.020.

[7] Simon Badger, Benedikt Biedermann, Peter Uwer, and Valery Yundin. Numerical evaluation of virtual corrections to multi-jet production in massless qcd. *Computer Physics Communications*, 184(8):1981–1998, 2013. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2013.03.018. URL https://www.sciencedirect.com/science/article/pii/S001046551300115X.

[8] Eric A. Moreno, Olmo Cerri, Javier M. Duarte, Harvey B. Newman, Thong Q. Nguyen, Avikar Periwal, Maurizio Pierini, Aidana Serikova, Maria Spiropulu, and Jean-Roch Vlimant. Jedi-net: a jet identification algorithm based on interaction networks. *The European Physical Journal C*, 80(1), January 2020. ISSN 1434-6052. doi: 10.1140/epjc/s10052-020-7608-4. URL http://dx.doi.org/10.1140/epjc/s10052-020-7608-4.

[9] Siddharth Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural

networks. *International Journal of Engineering Applied Sciences and Technology*, 04: 310–316, 05 2020. doi: 10.33564/IJEAST.2020.v04i12.054.

[10] Johannes Lederer. Activation functions in artificial neural networks: A systematic overview, 2021.

[11] Zachary Chase Lipton, Charles Elkan, and Balakrishnan Narayanaswamy. Thresholding classifiers to maximize f1 score, 2014.

[12] Chandradeep Bhatt, Indrajeet Kumar, · Vijayakumar, · Kamred, Kamred Singh, and Abhishek Kumar. The state of the art of deep learning models in medical science and their challenges. *Multimedia Systems*, 27, 08 2021. doi: 10.1007/s00530-020-00694-1.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[14] Mayank Mishra. Convolutional neural networks, explained [accessed march 2024]. URL https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[16] Thomas Kurbiel. Derivative of the softmax function and the categorical cross-entropy loss [accessed march 2024]. URL https://towardsdatascience.com/derivative-of-the-softmax-function-and-the-categorical-cross-entropy-loss-ffceefc081d1.

[17] Maurizio Pierini, Javier Mauricio Duarte, Nhan Tran, and Marat Freytsis. Hls4ml lhc jet dataset (100 particles), January 2020. URL https://doi.org/10.5281/zenodo.3602254.

[18] Marius-Constantin Popescu, Valentina Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8, 07 2009.

[19] George V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989. URL https://api.semanticscholar.org/CorpusID:3958369.

[20] Marco Seeland and Patrick Mäder. Multi-view classification with convolutional neural networks. *PLOS ONE*, 16(1):1–17, 01 2021. doi: 10.1371/journal.pone.0245230. URL https://doi.org/10.1371/journal.pone.0245230.

[21] Chenhao Lin and Ajay Kumar. Contactless and partial 3d fingerprint recognition using multi-view deep representation. *Pattern Recognition*, 83:314–327, 2018. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2018.05.004. URL https://www.sciencedirect.com/science/article/pii/S0031320318301687.

[22] Przemysław Dolata, Mariusz Mrzygłód, and Jacek Reiner. Double-stream convolutional neural networks for machine vision inspection of natural products. *Applied Artificial Intelligence*, 31:1–17, 02 2018. doi: 10.1080/08839514.2018.1428491.

[23] Krzysztof J. Geras, Stacey Wolfson, Yiqiu Shen, Nan Wu, S. Gene Kim, Eric Kim, Laura Heacock, Ujas Parikh, Linda Moy, and Kyunghyun Cho. High-resolution breast cancer screening with multi-view deep convolutional neural networks, 2018.

[24] Maren Mahsereci, Lukas Balles, Christoph Lassner, and Philipp Hennig. Early stopping without a validation set, 2017.

[25] Mohammadreza Heydarian, Thomas E. Doyle, and Reza Samavi. Mlcm: Multi-label confusion matrix. *IEEE Access*, 10:19083–19095, 2022. doi: 10.1109/ACCESS.2022.3151048.

[26] Simone Marzani, Lais Schunk, and Gregory Soyez. A study of jet mass distributions with grooming. *Journal of High Energy Physics*, 2017(7), July 2017. ISSN 1029-8479. doi: 10.1007/jhep07(2017)132. URL http://dx.doi.org/10.1007/JHEP07(2017)132.

[27] Andrew J. Larkoski, Simone Marzani, Gregory Soyez, and Jesse Thaler. Soft drop. *Journal of High Energy Physics*, 2014(5), May 2014. ISSN 1029-8479. doi: 10.1007/jhep05(2014)146. URL http://dx.doi.org/10.1007/JHEP05(2014)146.

[28] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, R. Rivera, N. Tran, and Z. Wu. Fast inference of deep neural networks in fpgas for particle physics. *Journal of Instrumentation*, 13(07):P07027, jul 2018. doi: 10.1088/1748-0221/13/07/P07027. URL https://dx.doi.org/10.1088/1748-0221/13/07/P07027.