Prepared for submission as the report of PHAS0056 final project

Separating jet images using convolutional neural networks

Z. Huang

Department of Physics and Astronomy, University College London, Gower Street, London, WC1E 6BT, U.K.

E-mail: zian.huang.19@ucl.ac.uk

Abstract: A set of data containing images of W boson decay jets, top quark decay jets and background quantum chromodynamic radiation, was studied by performing binary classification. Three sets of the corresponding combination training results were obtained, with an accuracy of about $(80.0 \pm 0.5)\%$ in the jet taggers developed. The results were compared with well-established models used in modern jet physics experiments. However, due to the lack of details about the provided raw data set, no valid comparison could be done without ambiguity. Further extensions and improvements on the experiment were suggested.

Keywords: jet tagging, jet imaging, convolutional neural network

Contents

1 Introduction

Jets are the subjects of study in this paper. They are collimated sprays of hadrons produced from high energy particles in Large Hadron Collider (LHC). These sprays of particles are closely packed together and they can be understood as a single object named "fat jet". The origins of these sprays, as the sources of decay, are contained as information in the structure of the fat jet. Therefore, by studying the observed structure of it, hadronic decays of weak gauge bosons, top quarks and other standard model particles can be tagged and distinguished from the Quantum Chromodynamic (QCD) radiation in the shower evolution. Being able to classify individual jets efficiently from the significant large noise of background QCD is crucial in searching for new fundamental particles. In most of the case, these targeted new particles are more massive than their daughter standard model particles. The decay products will therefore have a large Lorentz boost, resulting in multiple sequential decays afterwards, forming a fat jet. Through the search of the sea of signals as the fat jet, the beyond standard model (BSM) physics may be extended with new insights and discoveries.

In this study, the two jets in focus were chosen to be the jets formed from hadronic decays of weak gauge boson and top quark. An algorithm model to separate W jets with the background was first trained with another model tagging top jets followed by. It was expected that there is a difference in the signal jets and the background, the Feynman diagram of W decay is shown in figure 1 [1]. Compared this to an example of background radiation due to gluon, as shown in figure 2 [1], their unique decay paths will result in a greater mass daughter particles in W and top decays. This deviation in the 4 vector of particle is one of the differences that can be extracted and used to separate them from the background. Finally an additional model performing binary classification on W jets and top jets was trained in order to separate these three classes when there is a mixture of them.

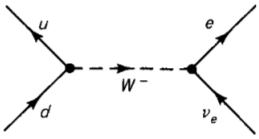

Figure 1. Feynman diagram showing an example of W decay, a down quark decays into a up quark and a W boson, which leads to further production of an electron and its neutrino

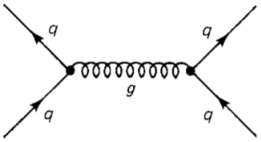

Figure 2. Feynman diagram showing an example of background QCD event, a high energy quark releases a gluon which leads to the production of a quark pair

This scenario is more realistic to be observed in a experimental LHC data. There were 3 training tasks aimed to be performed in total.

With their potential differences identified, the next step is to describe the observation in suitable data structure for further analysis. The most well-established method is to view the azimuthal angle against the rapidity plane with the calorimeter detector entries as a sparsely filled image [2] with pixel intensity being related to the deposited energy. To apply algorithm model on the image data set obtained, a more traditional approach is to use QCD-based method that takes in high level features such as the jet mass, into a Boosted Decision Trees (BDTs) learning algorithm. Although there is no well-established QCD model, these QCD methods with the use of BDTs are considered the standard in many LHC experiment and used to compared with new particle tagging approaches. One of the ways to improve is to reduce data discrimination. The traditional BDTs algorithm requires high lever features of the jets and could lead to "cherry picking" data, resulting in the model not being widely applicable to all experimental situations. An alternative method using low level information to classify the jet images could provide new insights to the search of new particles.

In the recent year, there has been a revolutionary development in image recognition and machine vision, with the use of convolutional neural network (CNN). Since the data of detected jet can be stored as an image, the application of CNN in particle tagging has been a great interest in jet physics community. A CNN consists of two parts, feature decomposition and pattern recognition. The feature decomposition is achieved by convolutional layers, which can be further broken down into kernelling, activation, padding and pooling. A kernel is first applied on the image to perform linear transformation, with a nonlinear activation function followed by. A paddling stage is an optional addition to avoid edge discrimination. At last, a pooling processing is applied to reduce the size of the image, with popular choices being max pooling and average pooling. This step is required due to the heavy computational power CNN algorithms require. The pattern recognition part first

starts with a fatten function, turning 2-dimensional images into 1-dimensional vectors. It is followed by densely connected neural networks that applies linear transform with nonlinear activation function. At the last layer of densely connected network, the probability that the input image being the interested signal is computed, in the setting of binary classification.

Consider that there have been major break through of using CNN to build particle taggers in [2, 4], with performance similar or better than the state-of-the-art QCD "MotherOfTaggers" [4] approach. The accuracy will be assessed not only by the finalised testing accuracy but also the prediction distribution of the binary classes and a receiver operating characteristic graph with its area under the curve. In [4], the computing was performed on cloud platform Google Colab. It was considered as a medium light weight machine learning task so a total training time in the scale of a few minutes on Colab is considered a optimal network complexity in the assessment of performance.

The significance of this research is to train a classifier that can achieve similar accuracy to modern CNN taggers with input images of jets in a wider range of transverse momentum. This contributes the progress of establishing a general model and method classifying observed jets for all reserved energy values. This model can provide insights in searching and training a new standardised method that can be used to benchmark jet tagger performance.

2 Machine Learning Methodology

2.1 Jet images and pre-processed data

A major limitation to this study arose from the fact that the details of the raw data set were not provided. In the following session, hypothesis and suggestions on the nature of the given data set would be introduced.

All of the input images were downloaded from [5] in a Python Numpy savefile named

"20190920_pt600.0_1500.0_40bins_10k.npz".

The $10,000$ images contained in the file were 40×40 by pixel binary greyscale images, as suggested by the part "40bins_10k". The "20190920" part was proposed to be the date of measurement with high confidence. By considering the part "pt600.0_1500.0" in the filename and the fact that each pixel stores only one number, it was predicted that the detected particles have energy in the range $[600, 1500]$ GeV and the grey scale colour represents the sum of neutral and track p_T per pixel, as a single quantity p_T^{calo}, the colorimeter p_T per pixel.

With reference to state-of-the-art research paper on particle tagging in jets [2–4], a popular and efficient method to obtain training data set for network development is to use simulated data. In these paper, calorimeter images were produced using standard Monte Carlo simulations [6] and several Python programming language scientific tools were introduced. Pythia8 [7] was first used to simulate quarks, bosons and QCD samples with specified energy values. The sequential events followed up were passed to a detector simulation Delphes3 [8] with the calorimeter parameters defined. Finally these detector towers were then clustered using FastJets3 [9] to obtain simulated images with a circular

shape as the scope. These simulated images were prepared for further image processing to achieve a better testing performance efficiently.

Consider that the images all had resolution of 40×40 and they did not show any signs of circular scope, it was suggested that they had been processed after being generated. A strong link with the raw data images was observed in the model "DeepTop" [2] that a hypothesis was proposed the provided raw data set had gone through similar image processing stage as in [2]. With the assumption that the pre-processing had already been performed, the network performance using this set of images would be compared to the results using "DeepTop". The hypothesis was then revisited based on the results obtained.

It was noticed that a direct comparison between the trained results of DeepTop network and the final network on the given data set would no be very legitimate as DeepTop was developed with jet data in a much smaller transverse particle momentum range [2]. Therefore, only the testing accuracy would be compared with the final results, which was obtained from a data set of jets with much higher energy and energy range.

2.2 CNN architecture and network training

To describe a CNN for binary classification, there are 3 categories of configuration, including network structural hyper-parameter, nonlinear transformation method and network training parameter.

The CNN structure hyper-parameters are presented in table 1. By referring to [10] the best practice of CNN architecture involves using multiple convolutional layers before pooling. This approach has be observed in many well-known models such as AlexNet [11], ResNet [12] and GoogLeNet [13]. This collection of multiple convolutional layers and a pooling layer was named as a block. The quantity "Block-Conv" in table 1 refers to the number of blocks in the network and "Layer-Conv" refers to the number of convolutional layers in a block. As mentioned in the introduction, a kernel, with a $N \times N$ dimension was applied in the linear transformation of images to extract graphical features. The quantity "Kernel-Conv" is taken as the kernel length, N. "Size-Conv" refers to the number of kernels and the times linear transformation was performed. These four hyper-parameters were used to describe the feature decomposition stage of the network. In the pattern recognition stage filled with densely connected neurons, the number of dense layers had the value of "Layer-Dense" with the number of neuron nodes in each layer being "Nodes-Dense".

In table 1, the DeepTop model is also presented. With the hypothesis made in the early paragraphs, the DeepTop network structure was used as a starting point to research for further improvement regarding the given data set. A first straight forward improvement was to add more nodes in the densely connected layers. It led to an increase in classification accuracy but raised the network complexity. Further adjustments were made at the convolutional stage. Some of the best performing CNN architectures, with simple structure that the training path is fixed, were studied, including AlexNet and LeNet-5 [14]. It was observed that a "bell-shape" convolutional stage, with the number of kernels, Size-Conv, being first increasing then decreasing along the depth of the network would lead to optimal testing results. This variation in kernel number was implemented in the final network. For the hyper-parameters being uncharged compared to the DeepTop architecture, both of

hyper-parameter	in DeepTop	final choice
Block-Conv	2	2
Layer-Conv	2	2
Kernel-Conv	4	4
Size-Conv	8, 8, 8, 8	128, 256, 128, 64
Layer-Dense	2	2
Nodes-Dense	64, 64	256, 256

Table 1. hyper-parameters of the CNN used in the DeepTop model and the improved model in the study

Block-Conv and Layer-Conv were kept to be 2, same as that in DeepTop. It was found to be the near optimal value at light-medium complexity. The kernel size was kept to be 4 as well to retain a large scanning area. It was the desired size capturing the detected signal peaks in the pixel area as features in the input images.

To decide the nonlinear transformation at each step, the activation function was chosen to be the ReLU function. The nonlinearity in ReLU funciton was added into the series of linear transformation in the network. The parameter in Keras layer "padding" was set to be "same", indicating a zero padding [15]. A softmax activation function was used in the last dense layer to achieve binary classification by returning the probability of the input image being identified as the signal jet. The smooth shape and the continuity characteristic of the sigmoid function allows it to achieve relatively high performance with low bias, when distinct deviation in the two classes was observed. Discrimination of the signal at the image edges was avoided, from the use of a relatively large kernel size.

Regarding the parameters used in model training, according to [4], training with smaller mini-batch size would lead to a better performance and efficiency. The batchsize in the final model was set to be 100, being 1/100 of the total size of raw data set. An arbitrarily large epoch of 50 was set with a early stop policy. Overfitting could be avoided and time was saved for more training instances to be obtained, leading to a more confident endpoint. The raw data set was shuffled and split into training and testing sets with the ratio 8 : 2 as a common practice. At the stage of model compiling, the Keras method "adam" [16] was chosen. In [4], it was found that both AdaDelta [17] and Adam resulted in similar optimal performance in jet tagger training. Followed the same improvement to the DeepTop model in [4], the loss function was chosen to be "categorical_crossentropy" rather than mean-square-error, though using the latter is considered more mathematically appropriate in first glance. Using categorical_crossentropy would avoid the slow down of learning when the prediction is close to 0 or 1 since the deviation in norm between steps would be relatively low.

Based on the binary classification of W decay and background QCD event, the final CNN model is shown in figure 3. This CNN network, which optimized W tagging, was used for the tasks of top tagging and W-Top classification. The same final network design was used to perform different tasks for potential insights being observed from this consistence.

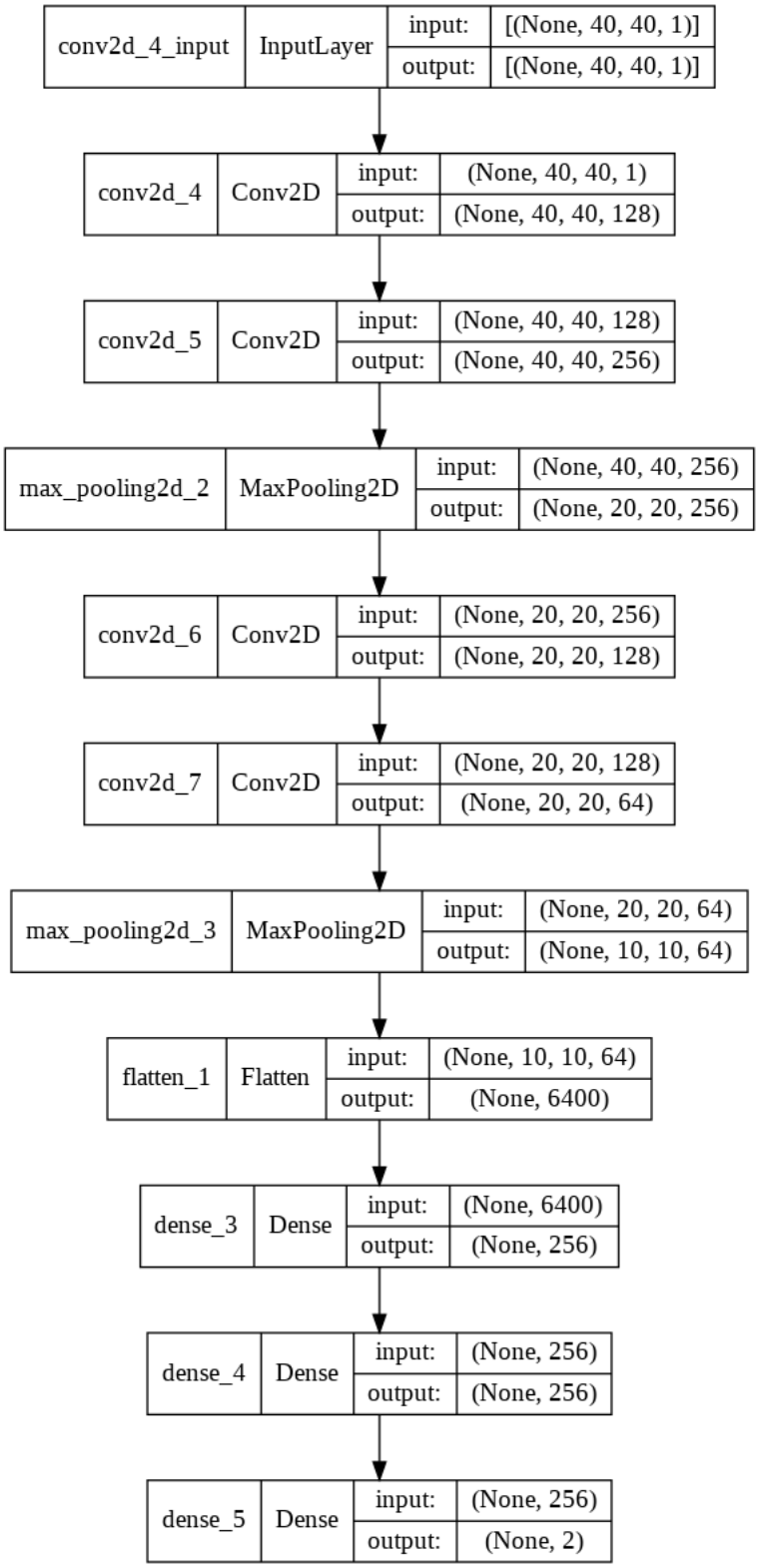

Figure 3. network structure of the CNN optimal for W tagging in jet shower

training model	validation loss	validation accuracy
W-QCD	0.4772	0.8005
Top-QCD	0.4404	0.8033
W-Top	0.4163	0.8255

Table 2. results of the 3 model fitting tasks, obtained by passing test sets into the `model.evaluate()` Keras method

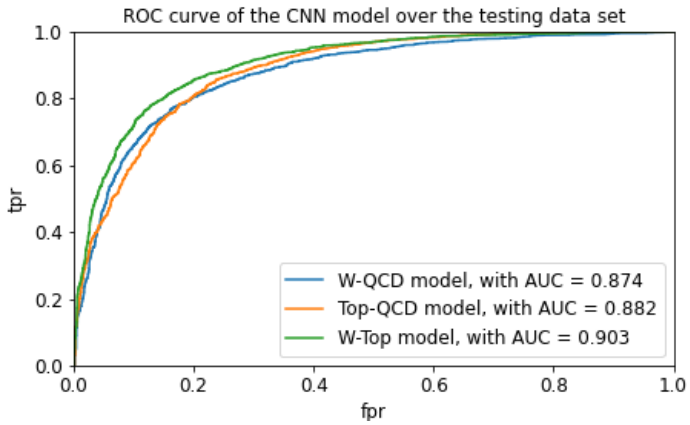

Figure 4. receiver operating characteristic (ROC) curve of the 3 model fitting tasks studying the network performance on different jet pairs

3 Performance Tests Results

With a total training time of approximately 15 minutes for the 3 fitting tasks on GPU boosted Google Colab Pro, the evaluation results on corresponding test sets are presented in table 2. The receiver operating characteristic (ROC) curve and the rejection-efficiency curve, showing all of the 3 tasks, are presented in figure 4 and 5.

Results showing the evolution of both training and testing loss value and accuracy in the W-QCD model are presented in figure 6 and 7. Overfitting started to be observed in all of the three models after the 15th epoch, with training ended at around the 28th epoch. Therefore, only the W-QCD model history is presented to avoid repeated communication. More detail results could be referred to the source codes and plotted graphs on public accessible Colab platform [18].

For the distribution histogram of prediction probability, 3 graphs are given for each of the testing results, shown in figure 8, 9 and 10. The set of plots based on training accuracy was also generated. Same shape as the testing result distribution was shown in these graphs. However, they contained less significant information as slight overfitting was observed, showing identical shape but with less noise. Similar to the other two history plots, these training distributions could be found in [18], being public with the Python source codes.

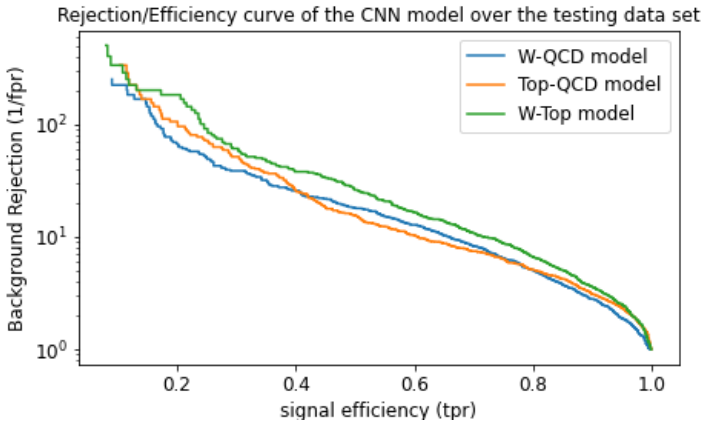

Figure 5. rejection-efficiency curve curve of the 3 model fitting tasks studying the network performance on different jet pairs

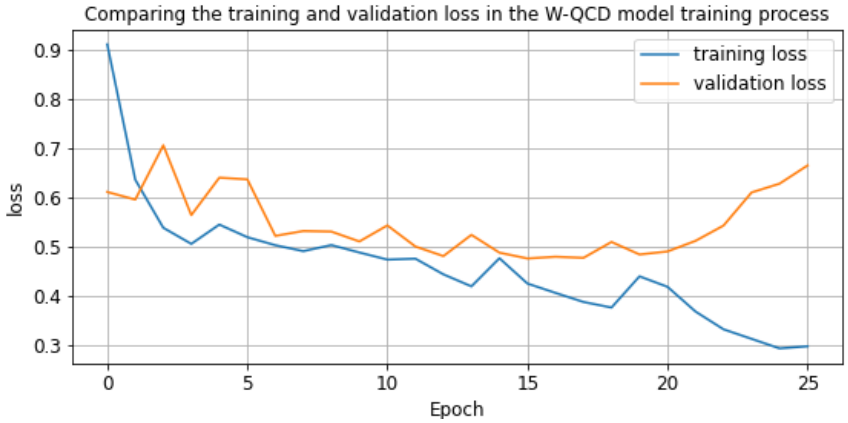

Figure 6. The evolution of the loss value along the training process

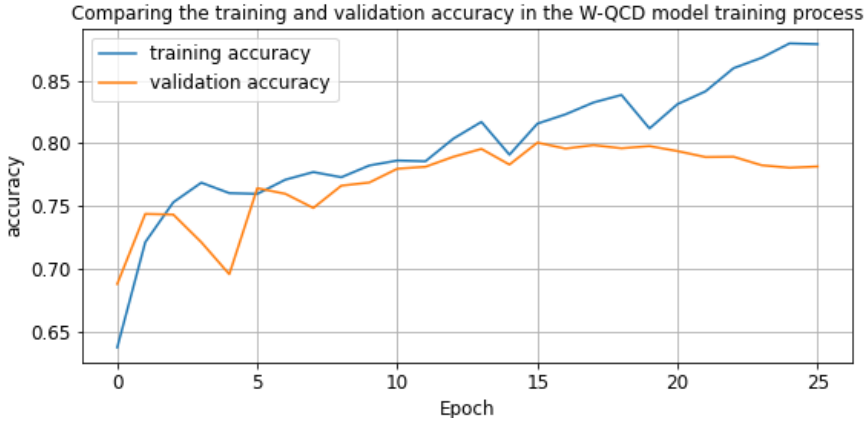

Figure 7. The evolution of the accuracy along the training process

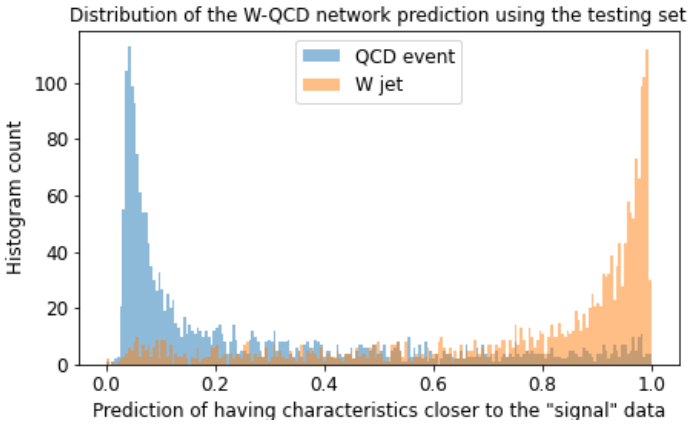

Figure 8. The distribution of the probability that jet image being identified as the class of focus in the stage of testing the W-QCD model

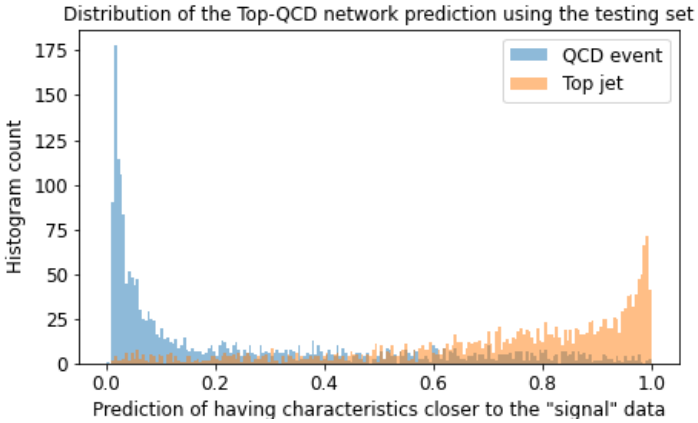

Figure 9. The distribution of the probability that jet image being identified as the class of focus in the stage of testing the Top-QCD model

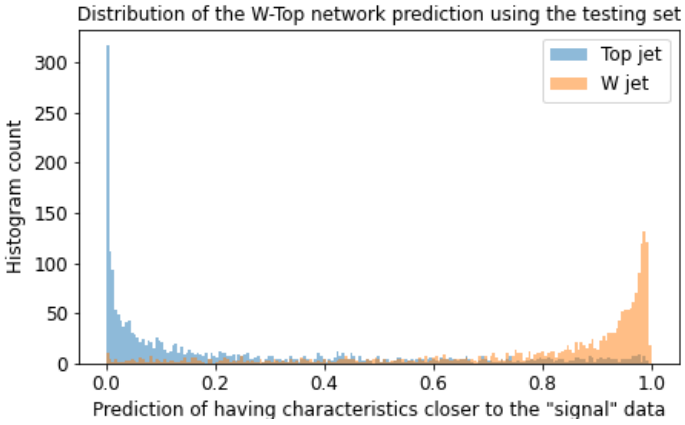

Figure 10. The distribution of the probability that jet image being identified as the class of focus in the stage of testing the W-Top model

4 Discussions

There were 2 methods of comparison on these results obtained. The performance of the W tagger and the top tagger could be compared to existing well-established CNN and BDTs model trained in a narrower range of particle momentum. Comparison between these 3 sets of results could also be conducted to draw new insights on jet image feature.

In [4], the DeepTop model is cited to have an accuracy of 85.5% with $AUG = 0.930$, trained with particle having momenta in a range of $[350, 450]$ GeV. This was compared to the Top-QCD model results in table 2. With a wider range of $[600, 1500]$ GeV, the accuracy drops to 80.33% with $AUG = 0.882$. Potential ambiguity could arise from the fact that the top tagger was trained using the network structure optimised for W tagger. However, since the network engineering principle of CNN remains the same and they belonged to the same big class as jet images, it was considered that reaching an top tagging accuracy higher than 80.33% under the same complexity constrain would require a much longer time to reach and develop. Moreover, it was dependent on the hypothesis that the raw data underwent the same image processing stage as in DeepTop. Therefore, no valid conclusion could be carried out due to not enough information given on the raw jet images. Repeating the CNN development on a new set of image data with known origins and simulation parameters would be the most efficient way to improve this study.

Since it was difficult to compare with external sources of model performance, the observation in the 3 sets of results would be discussed. The performance in the W-Top model was concluded to be the best with high confidence, due to its largest area under the curve in both of the ROC and rejection-efficiency curves. This high performance could also be supported by the 2 sharp peaks in figure 10, indicating a high confidence in tagging with low probability of true negative or false positive. Both of the W tagger and top tagger performed similarly well. Same as the argument stated in the previous paragraph, it was considered that the network being optimised for a specific tagger had neglectable impact on this similarity. Solely based on this data intensive approach, new insight could be suggested that W jets and Top jets both have unique graphical features that are more distinct to each other, compared to that of background QCD radiation. These unique features resulted in the observed deviation among models trained. If there is a higher confidence distinguishing a W jet from a Top jet than from a background QCD event, recognising the background QCD event becomes the limiting stage to achieve high performance jet tagging method. Since background QCD event is always present in observation, multi-class categorisation could be an extension to this study, using 3 labels to classify W jets, Top jets and QCD radiation. A model with multiple class jet tagging using relatively few computing resources could greatly improve the rate of searching new particles in LHC experiment.

5 Conclusions

The best performing W jet tagger and Top jet tagger were trained to have and an accuracy of $(80.0 \pm 0.5)\%$. However, due to the lack of details about the provided raw data set, no valid comparison could be done with existing state-of-the-art CNN models such as DeepTop, or traditional BDTs QCD models such as MotherOfTaggers. Although potential insights regarding the hidden graphical features in W jets and Top jets were suggested, further research shall be conducted to obtain more supporting evidence and theoretical prediction.

References

[1] D. Griffiths, *Introduction to Elementary Particles,* WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim Germany (2004), pg. 60,66.

[2] AutG. Kasieczka, T. Plehn, M. Russell, and T. Schellhor, *Deep-learning Top Taggers or The End of QCD?, JHEP* **05** (2017) 006, arXiv:1701.08784 [hep-ph].

[3] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, *Jet-images – deep learning edition, JHEP* **07** (2016) 069, arXiv:1511.05190 [hep-ph].

[4] S. Macaluso and D. Shih, *Pulling Out All the Tops with Computer Vision and Deep Learning, JHEP* **10** (2018) 121, arXiv:1803.00107 [hep-ph].

[5] M. Campanelli, Y. Zhu, *Index of /undergrad/0056/other/projects/jetimage/.* https://www.hep.ucl.ac.uk/undergrad/0056/other/projects/jetimage/ (accessed Decomber 15, 2021).

[6] ATLAS collaboration, *Identification of boosted, hadronically-decaying W and Z bosons in* $\sqrt{3} = 13$ *TeV TeV Monte Carlo Simulations for ATLAS, ATL-PHYS-PUB-2015-033* (2015).

[7] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel et al., *An Introduction to PYTHIA 8.2,* arXiv:1410.3012 [hep-ph].

[8] J. de Favereau et al., *A modular framework for - 21 - fast simulation of a generic collider experiment, JHEP* **1402** (2014) 057, arXiv:1307.6346 [hep-ph].

[9] M. Cacciari, G. P. Salam, *Dispelling the N3 myth for the kt jet-finder, Phys. Lett. B* **641** (2006) 057, arXiv:hep-ph/0512210; M. Cacciari, G. P. Salam, G. Soyez, *FastJet User Manual, Eur. Phys. J. C* **72** (2012) 1896, arXiv:1111.6097 [hep-ph]. http://fastjet.fr

[10] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow,* O'Reilly, Sebastopol Canada (2019), pg. 460-477.

[11] G. E. H. A. Krizhevsky, I. Sutskever, *ImageNet Classification with Deep Convolutional Neural Networks,* https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[12] K. He, X. Zhang, S. Ren, J. Sun, *Deep residual learning for image recognition,* O'Reilly, Sebastopol Canada (2019), CoRR abs/1512.03385 (2015) , arXiv:1512.03385.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, *Going deeper with convolutions,* CoRR abs/1409.4842 (2014) , arXiv:1409.4842.

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *Gradient-based learning applied to document recognition, Proceedings of the IEEE* (1998).

[15] F. Chollet, https://github.com/fchollet/keras (2015).

[16] D. P. Kingma, J. Ba, *Adam: A method for stochastic optimization, CoRR* **abs/1412.6980** (2014), arXiv:1412.6980.

[17] M. D. Zeiler, *ADADELTA: an adaptive learning rate method, CoRR* **abs/1212.5701** (2012), arXiv:1212.5701.

[18] Z. Huang, *PHAS0056_finalProj - Google Drive.* https://drive.google.com/drive/folders/17w4MT4VB_KXlOWj7UtmlqwxO2LHXSq7w/ (accessed January 9, 2022).