Neutrino Interaction Event Classification Using Convolutional Neural Networks

Felix Potter
Department of Physics and Astronomy, University College London
London, UK

Abstract: This study presents an exploration of convolutional neural networks (CNN) for classifying images of simulated muon-neutrino interactions akin to Fermilab's NOvA experiment, a task of significant importance. Neutrinos, fundamental particles of the lepton family, interact rarely only through the weak force, presenting challenges for observation. Four models were developed and evaluated, with a multi-view CNN incorporating an inception module emerging as the most effective, achieving an accuracy of 80%. The model's performance varied with energy and interaction type, with higher energies and quasi-elastic interactions yielding higher accuracies. Additionally, machine learning models for predicting neutrino energies and classifying interaction modes were trained, achieving a mean absolute error of 6.6 GeV and an accuracy of 46.5%, respectively. These findings demonstrate the potential of CNNs in high energy particle physics research, and suggest broader applications in the field.

1. Introduction

Correctly characterising and identifying particle interaction data arising in experiments is a challenge frequently dealt with in high energy particle physics (HEP) [1,2]. In the past, methods such as triggering for gluon jets [3], classification of events [4] and reconstruction of objects like showers, tracks and clusters [5] have been tried. Here an attempt is made with convolutional neural networks to classify images of simulated neutrino interactions resembling those of Fermilab's NOvA experiment. More precisely, a machine learning model is built using CNNs with the goal of creating a binary classifier to correctly identify images of charged-current muon-neutrino events.

1.1 Neutrino Physics and Research

Neutrinos are electrically neutral particles of the lepton family with the smallest known mass of any of the fundamental particles. They come in three flavours, electron-neutrinos (ν_e), tau-neutrinos (ν_τ) and muon-neutrinos (ν_μ), and they are linked to three other fundamental particles, the electron (e^-), the tau (τ^-) and the muon (μ^-). Neutrino interactions take place through the weak interaction and are a very rare phenomenon to observe, occurring in neutral-current (NC) and charged-current (CC) events. NC events involve neutrinos scattering with the exchange of an electrically neutral particle. CC events involve neutrinos scattering with the exchange of an electrically charged particle, and the transformation of the neutrino into its corresponding pair-particle (e.g. ν_e turns into an e^-). CC events may be split up into three distinct classes of interactions: deep inelastic scattering (DIS), quasi-elastic (QE) and resonant (Res). In DIS interactions neutrinos scatter off nucleons (e.g. protons or neutrons). They tend to produce 'messier' images with numerous tracks and showers. QE interactions are clearer, often with a single track produced [7]. Res interactions are commonly comprised of a mixture of traits shared by the previous two interactions.

One of the major revolutions in the Standard Model (SM) of particle physics has been the discovery of the phenomenon of neutrino oscillations [8]. Neutrinos may change their flavour over time (e.g. a ν_e can over time come to be identified as a ν_μ) [9]. This phenomenon was not predicted by the SM and notably upset its prediction that neutrinos were massless. Presently, one of the major centres for research on neutrino oscillations has been Fermilab's 'long-baseline' NuMI Off-axis ν_e Appearance (NOvA) experiment. Among other goals, NOvA seeks to establish whether neutrino oscillations from ν_μ to ν_e are observable. NOvA possesses two detectors, a 14 metric-kiloton far detector situated in Minnesota and a 300 metric-ton near detector at Fermilab. They consist of 344,000 cells made of highly reflective plastic PVC containing liquid scintillator. Neutrinos which strike the atoms in the

scintillator release bursts of charged particles. When these particles slow down in the detector their energy is registered with wavelength-shifting fibers that connect to photo-detectors. This method is used due to the electrical neutrality of neutrinos, making them effectively invisible to many other detection methods. By analysing the features of light registered by the photo-detectors the flavour and energy of the detected neutrinos may be inferred.

NOvA produces data from two perpendicular perspectives, from the top and from the side (see Fig. 1). Using an appropriate coordinate system we may define them as the x-z and y-z planes respectively. This results in two images being produced for each event detected (shown in Fig. 2).

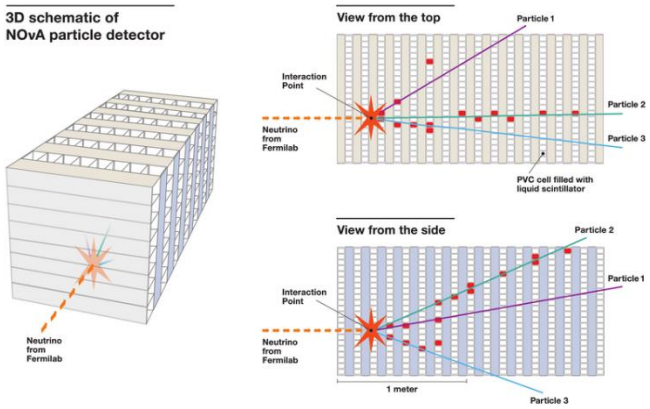

Fig .1 NOvA detector schematic showing Fermilab's neutrino beam interactions with side and top views. Taken from Ref [18].

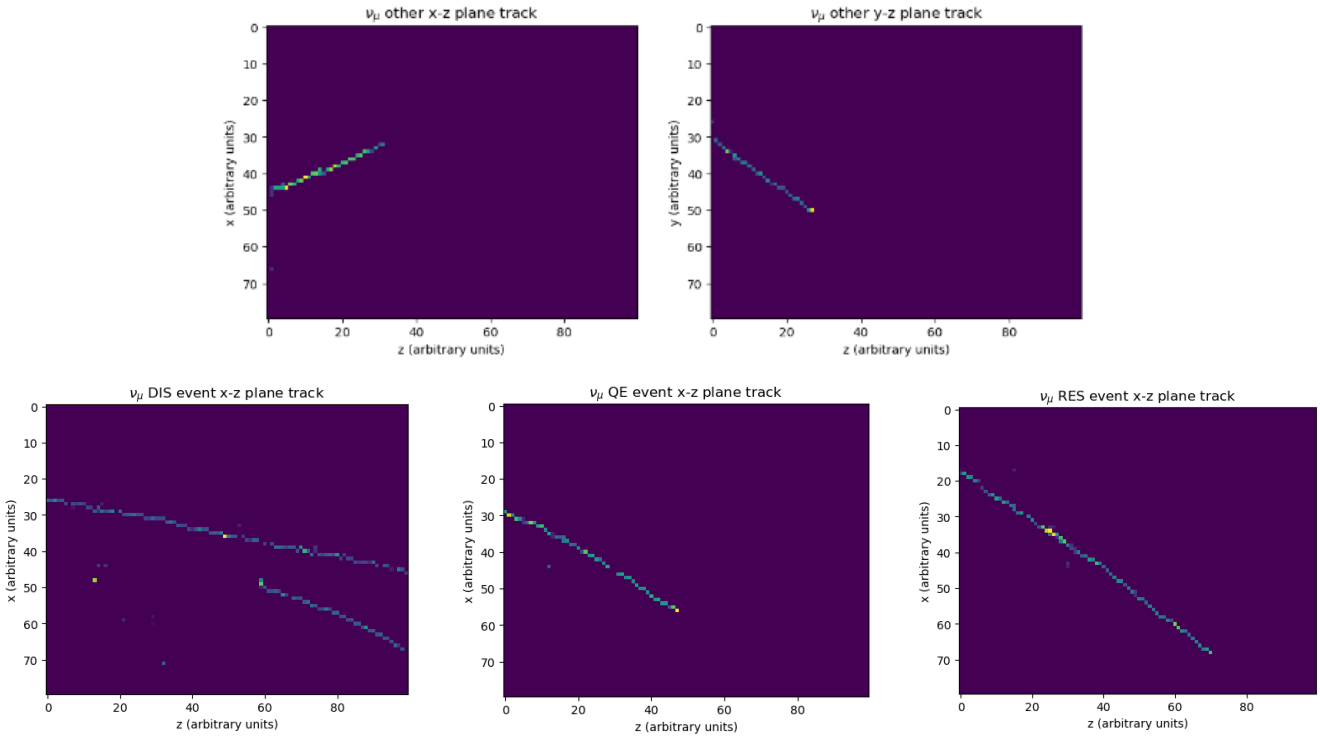

Fig .2 A collection of NOvA simulated neutrino interaction images. Above are plotted the x-z and y-z views for a non DIS/Res/QE interaction. Below are the x-z views of a DIS, QE and Res event.

1.2 Neural Networks and Convolutions

A neural network (NN) is an artificially constructed network comprised of interconnected nodes (regularly called neurons) loosely inspired on the neurological wiring of the brain. Linked to the connections are a set of weights (numerical values), and for each neuron there exists a bias (another number) [10]. The weights can act to supress or augment the strength of connections such that often times in a network certain neural connections become more important than others. Researches have often used the multi-layer perceptron (MLP) for HEP. Its architecture is comprised of an input layer, followed by intermediate 'hidden' layers and an output layer, with all of them sequentially connected. For a given network layer with n neurons – each with its own weight (w_i) and input (x_i)— and a bias b, its output (y) is given by:

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{1}$$

Where $f(x)$ is a non-linear activation function. In training, the MLP is fed an input, and its final output is compared to a 'label' which in some way corresponds to the input. The degree to which the network's output diverges from the known label is quantitatively calculated with a loss function, and its weights and biases are subsequently optimised through backpropagation using the gradient of the loss function. Though fit for some tasks, its comprehensive interlinking has rendered the MLP susceptible to overfitting, which makes it often incapable at identifying 'core', general features of a given dataset such that it often can't cope with small idiosyncrasies in datasets.

Convolutional neural networks (CNN) are a subset of deep learning architectures that are principally used for computer vision. Much of the same operations (e.g. weights, biases) used in regular NNs are still used in CNNs, however they differ in that they are more fitted for pattern recognition problems using images. They allow for image-specific features to be baked within the architecture, all the while significantly decreasing the number of parameters being used when compared to regular NNs [11]. They use sets of filters (with dimensionality ordinarily smaller than that of the image being analysed) to produce feature maps via convolutions with the given input. Like other architectures, these filters 'learn' via backpropagation, and non-linear functions (e.g. ReLU) are used on the outputs of convolutional layers which these filters produce. Pooling layers are also used, which downsize the input's dimensionality, extracting its critical details [12]. Typically, in a CNN, a combination of convolutional and pooling layers are used, which get stacked together (see Fig. 3).

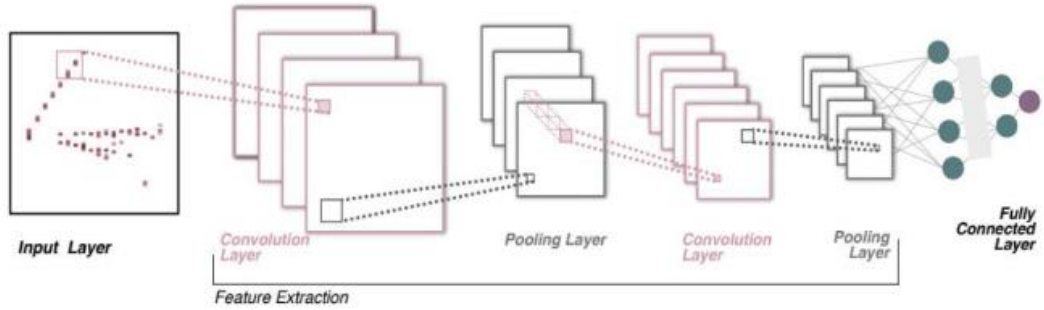

Fig. 3 Example of CNN with multiple stacked layers. Taken from Ref [17].

Regularisation measures are often implemented in the form of batch normalisation [13], dropout layers [14] and L2 regularisation, with an aim to avoid overfitting to a particular dataset, and allow the network to learn features which are common to most instances of a particular class of images. Though CNNs have found great success, a number of its shortcomings have been raised such as its difficulty at effectively grasping global and local features at the same time [15], and its restricted capacity at representing distant dependencies in sequentially arranged datasets [16].

2. Binary Classifier for Muon-Neutrino Interactions

2.1 Model Architectures

Four models were tried in total, each with the purpose of correctly distinguishing CC ν_μ interactions from non CC ν_μ interactions. The first model consisted of a regular CNN model, with 2 convolutional layers (using ReLU activation), each followed by a pooling layer and a dropout layer. 32 3x3 filters were used in each convolutional layer. Dropout values were 0.3, and pooling was implemented with Max pooling 2D layers with sizes of 2x2. The loss function used was the binary cross-entropy loss function, commonly used for binary classification tasks. The output layer had 1 neuron using sigmoid activation. The model architecture is shown in Fig 4.

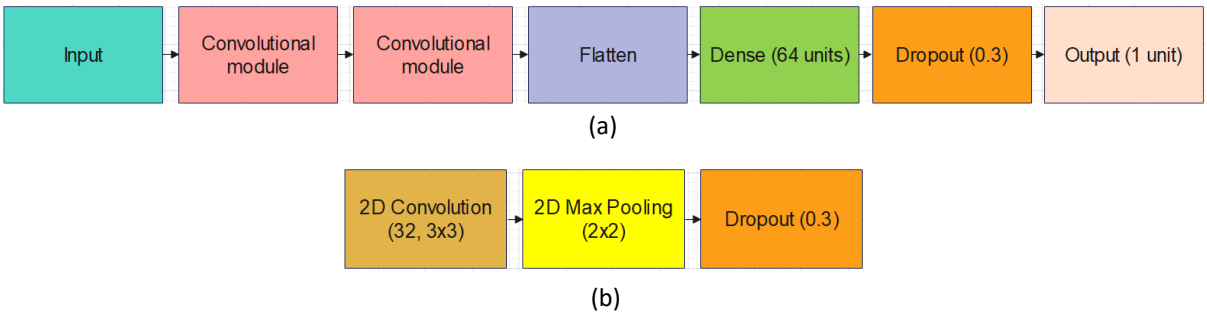

Fig. 4 First model architecture, shown in layers. Overall architecture (a) and convolutional module architecture (b).

The second model was virtually identical to the first one except for the use of L2 regularisation (values of 0.01) on the dense and convolutional layers for the purpose of reducing the level of overfitting.

The third model was a multi-view CNN (MVCNN). It makes use of one or more separate branches, each taking its own input. Each branch has a number of layers like a regular CNN, but later they get concatenated, where their joint output is subject to further layers. MVCNNs have the benefit over regular CNNs of making use of more than one viewpoint of a given object, allowing the network to pick out features that persist through different perspectives or lighting configurations, which may enhance the network's performance. For further discussion on MVCNNs see Ref [19].

As input, the model took in the x-z and y-z plane images separately in each branch. Both branches were comprised of 2 sets of a 2D convolutional layer followed by a 2D pooling layer. The layers were concatenated and put through a further convolutional layer, followed by a pooling layer. Lastly, a flattening layer was used, followed by two dense layers and an output layer. As before, L2 regularisation (with a value of 0.01) was used on 3 of the convolutional layers and the final dense layer. See Fig. 5 for the detailed architecture.

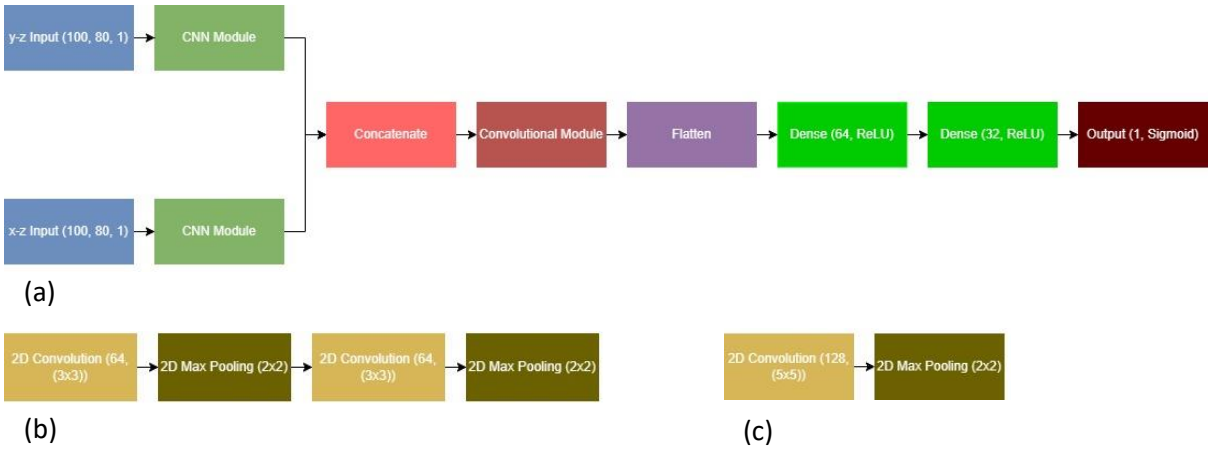

Fig. 5 Third model architecture. Total architecture (a), CNN module (b) and convolutional module (c). L2 regularisation values and the number of convolutional filters were optimised via a manual grid search.

The fourth model was similar to the third, but made use of an inception module. The inception module was designed to deepen networks without the downside of resulting in an overly (computationally) expensive process [16]. Multiple convolution (with varying filter dimensions e.g. 1x1, 3x3 and 5x5) and pooling layers are able to be stacked in parallel to operate on the output of prior layers. This makes the network more versatile, capable of extracting global and local features simultaneously [20]. Like the MVCNN the outputs of the inception module's filters get concatenated to serve as input for the next layer. Inception modules have been used in past projects like Inception-ResNet [21] and Inception-v3 [20]. For the fourth model's inception module, 3 convolutional branches and 1 pooling branch were used, with their respective concatenation serving as the output

of the module. Dropout layers were also added to supress overfitting. The full model architecture is shown in Fig. 6.

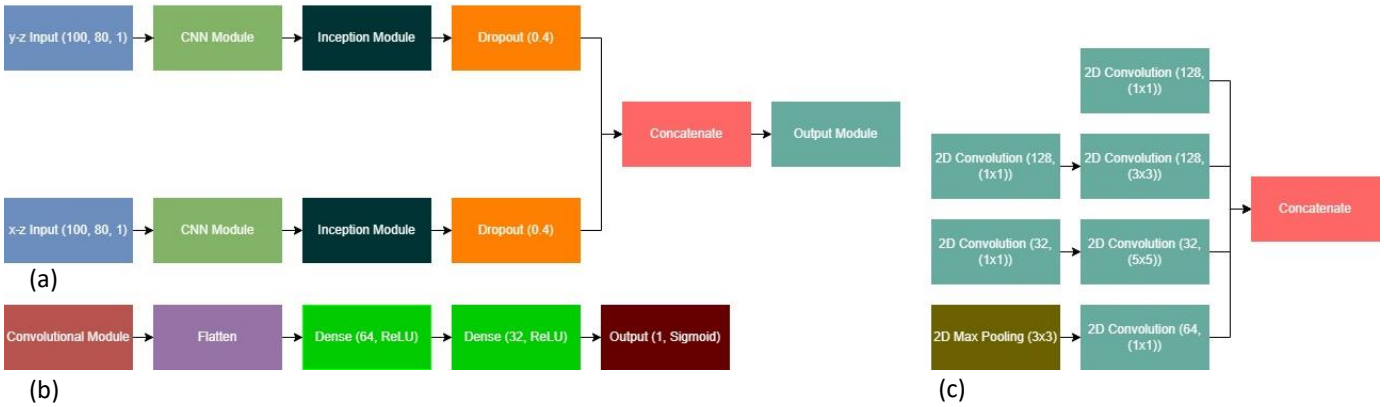

Fig. 6 Fourth model architecture. Total architecture (a), output module (b) and inception module (c). The convolutional module is the same as the third model's.

2.2 Dataset Manipulation and Model Training

Originally, the dataset was composed of 83,737 pairs of x-z and y-z plane images, and was imbalanced with an 88%-12% split of CC ν_μ events to non-CC ν_μ events. 80% of the images were chosen for training and 20% for testing. The uneven split of events was dealt with by withdrawing the surplus of CC ν_μ events, resulting in a training dataset with 15,446 pairs of images and a testing dataset with 4,154 pairs. Batch normalisation was applied instead of individually normalising each image as this left the underlying pixel trends in the dataset mostly unchanged; this is shown in Fig. 7.

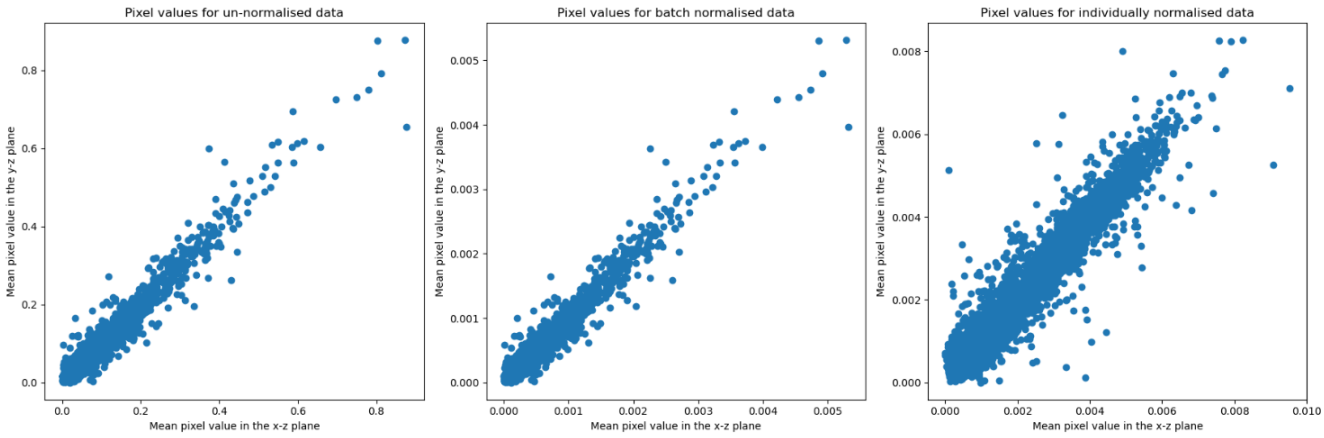

Fig. 7 Different normalisation techniques. As can be seen, Individually normalising the data deforms the overall shape compared to the unnormalized data. This does not occur with batch normalisation.

For training, the training-validation split was chosen to be 80%-20%. All models were compiled using the Adam optimiser and the binary cross-entropy loss, with accuracy as the metric. Epochs and batch size details on models trained are shown in **Table 1**.

Table 1: Model Training Details

Model	Epochs	Batch Size	Final Validation Accuracy	Final Validation Loss	Number of Parameters
Regular CNN	15	100	0.8304	0.4091	857,857
CNN with L2 Regularisation	20	80	0.7356	0.6225	857,857
MVCNN	15	64	0.7812	0.5082	1,486,465
MVCNN with inception module	15	100	0.7942	0.4788	2,800,257

L2 regularisation was found to diminish overfitting at the cost of a smaller final validation accuracy. To compensate for this it was found that increasing the number of parameters lead to better validation accuracies. Though the smaller first model (Regular CNN) achieved the highest accuracy, it succumbed to overfitting as can be seen in Fig. 8. The fourth model (MVCNN with inception module) however, achieved the best accuracy and loss values while avoiding overfitting as its validation accuracy didn't plateau relative to the training accuracy in later epochs. Because of this, it was the model chosen for further investigations.

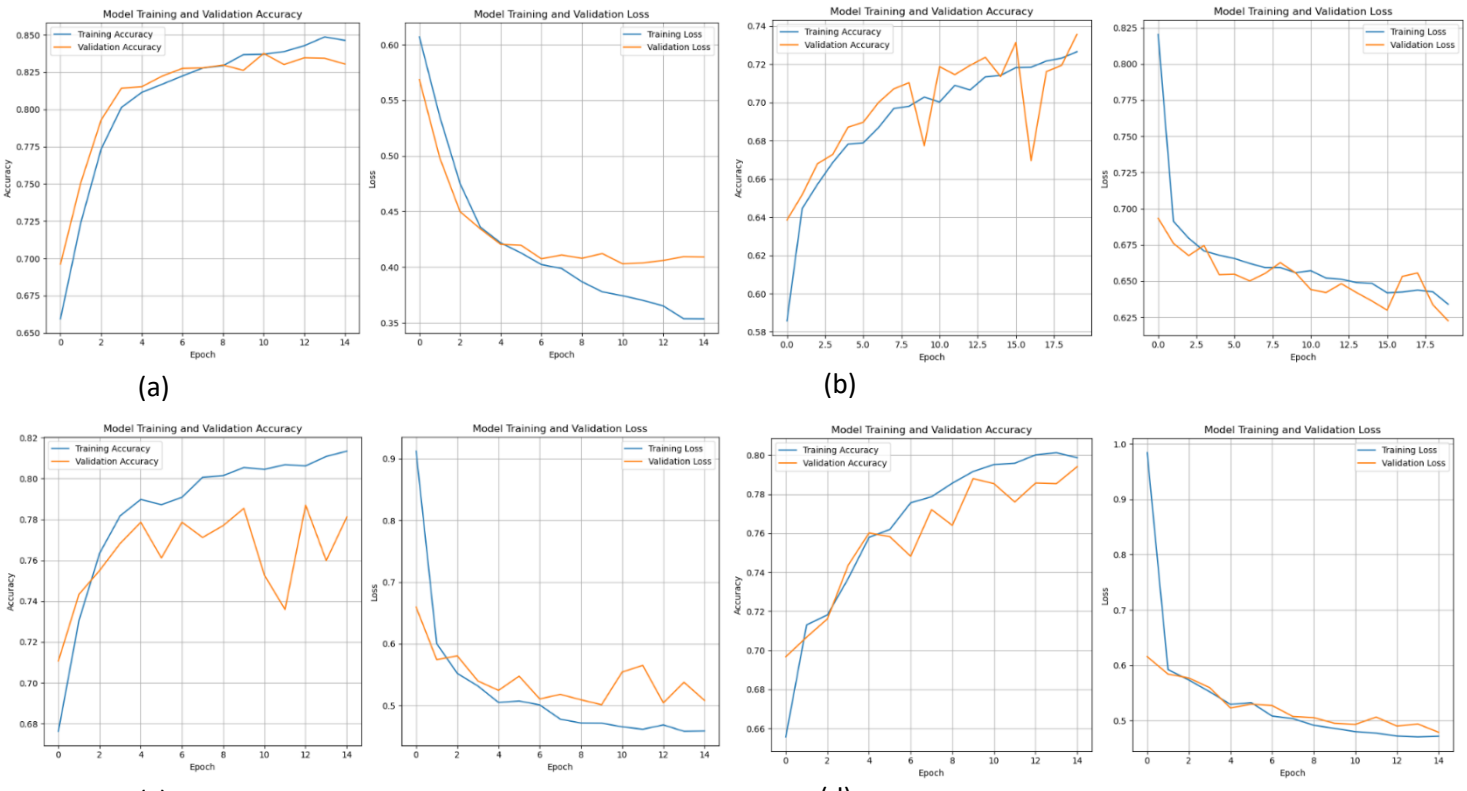

Fig. 8 Accuracies and losses achieved for all models. Regular CNN (a), CNN with L2 Regularisation (b), MVCNN (c) and MVCNN with inception module (d). The regular CNN's validation loss plateaus around epoch 7 while the training loss still decreases which means that it has overfitted, in contrast to the MVCNN with inception module, whose validation loss decreases along with the training loss.

2.3 Threshold Determination and Classifier Testing

The fourth model was chosen for testing. First, its receiver operating characteristic (ROC) curve was plotted (shown in Fig. 9), using the testing dataset, to determine a threshold for conversion of the network's output into binary values (i.e. 0s and 1s). The ROC curve plots a classifier's sensitivity (the true positive rate) against 1 minus the specificity (the false positive rate) [22]. A random binary classifier with a 50% false positive and true positive rate would result in a straight line through the origin. A successful classifier results in a curve to the left of the random classifier, representing a true positive rate larger than its false positive rate. The chosen threshold was found by finding the threshold which maximised the difference between the true positive and false positive rates, which yielded an optimum threshold of 0.549, where outputs above are labelled as positive CC ν_μ events (1), and outputs below as non CC ν_μ events (0).

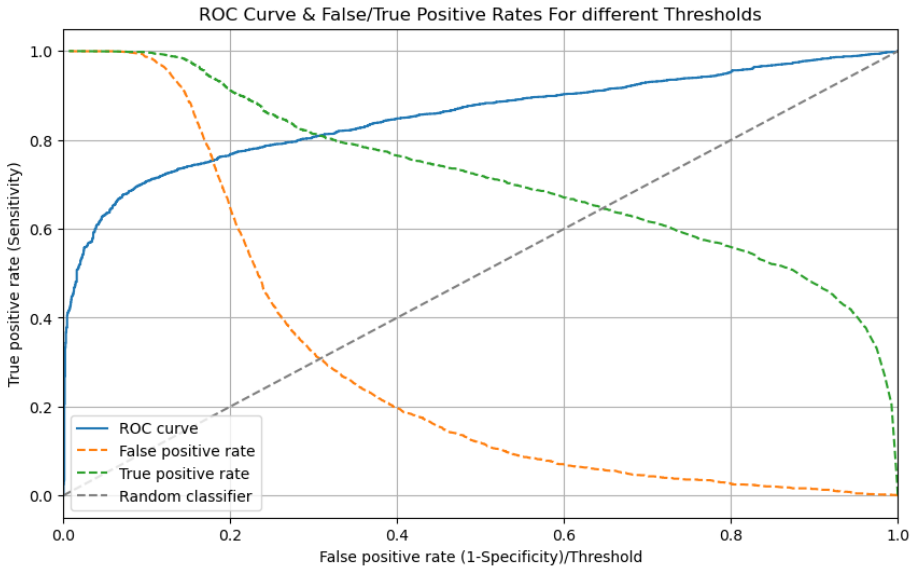

Fig. 9 The fourth model's ROC curve (in blue) and true/false positive rates for different thresholds.

The model was evaluated on the test dataset, achieving 80% accuracy, slightly better than its final training and validation accuracy which further points to it having learned general features and not overfitted to its training dataset. The model's predictions on the test dataset were plotted (shown in Fig. 10) against the actual distribution of events, both as their raw values and as their converted binary values. When converted to binary, the model was found to overpredict on non-CC ν_μ events, and underpredict on positive CC ν_μ events. It also demonstrated what could be interpreted as less certainty for negative events than positive events, as its predictions for non-CC ν_μ events were distributed among 0.2-0.4, but for true positive events the values were closer to 0.9-1.

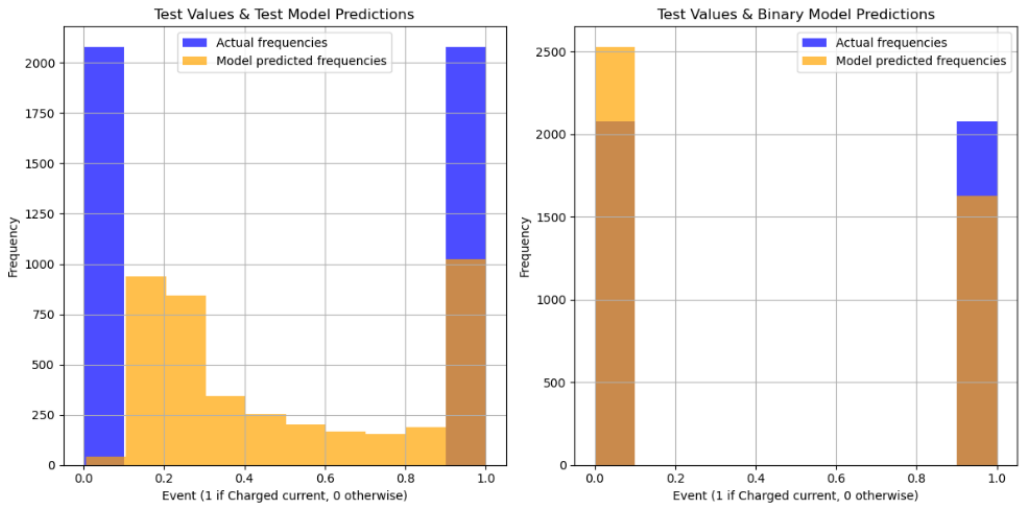

Fig. 10 Fourth model's predictions on test dataset. Raw values (left) and converted to binary values (right).

2.4 Testing Classifier's Energy Dependence

The given interactions from the simulated NOvA images occur at different energies. For each interaction, three energies are considered: the lepton energy, the neutrino energy and their combined energies. Their distribution is shown in Fig. 11.

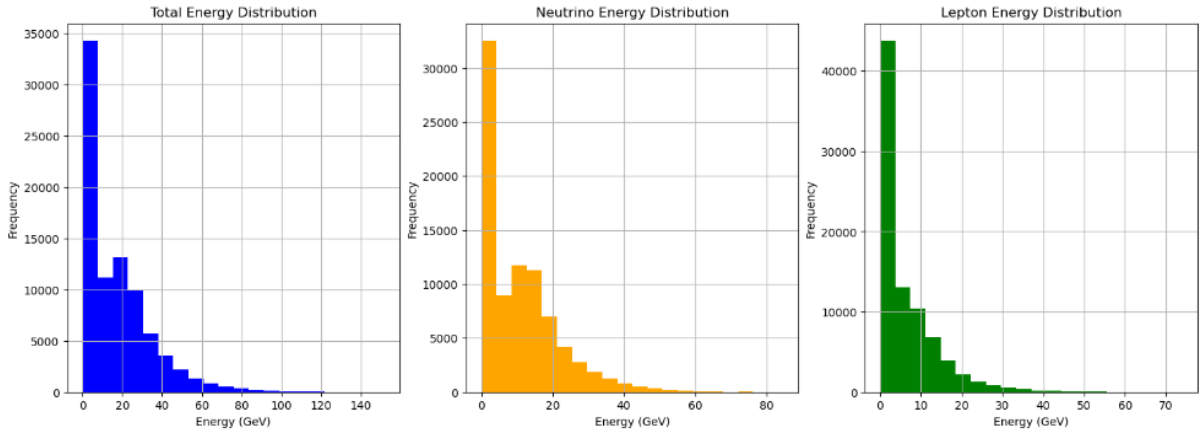

Fig. 11 Energy distributions of simulated NOvA interactions. The distributions seem to loosely follow an exponential decay-like shape, especially the lepton energies.

To test the performance of the model on these energies, the energies were arranged in ascending order and then split into 30 arrays. The model was then evaluated on each array, and the model's accuracies plotted against each array's mean energy (as shown in Fig. 12). A linear fit was performed on the plots to better observe the presence of any trends. In all cases it was found that the model's accuracy heavily depended on the energy, with accuracies growing with energy. Higher energies enable particles to penetrate deeper into the detector, resulting in longer and clearer tracks in the

image, as well as expectedly displaying greater pixel brightnesses, which aids the model in classification.

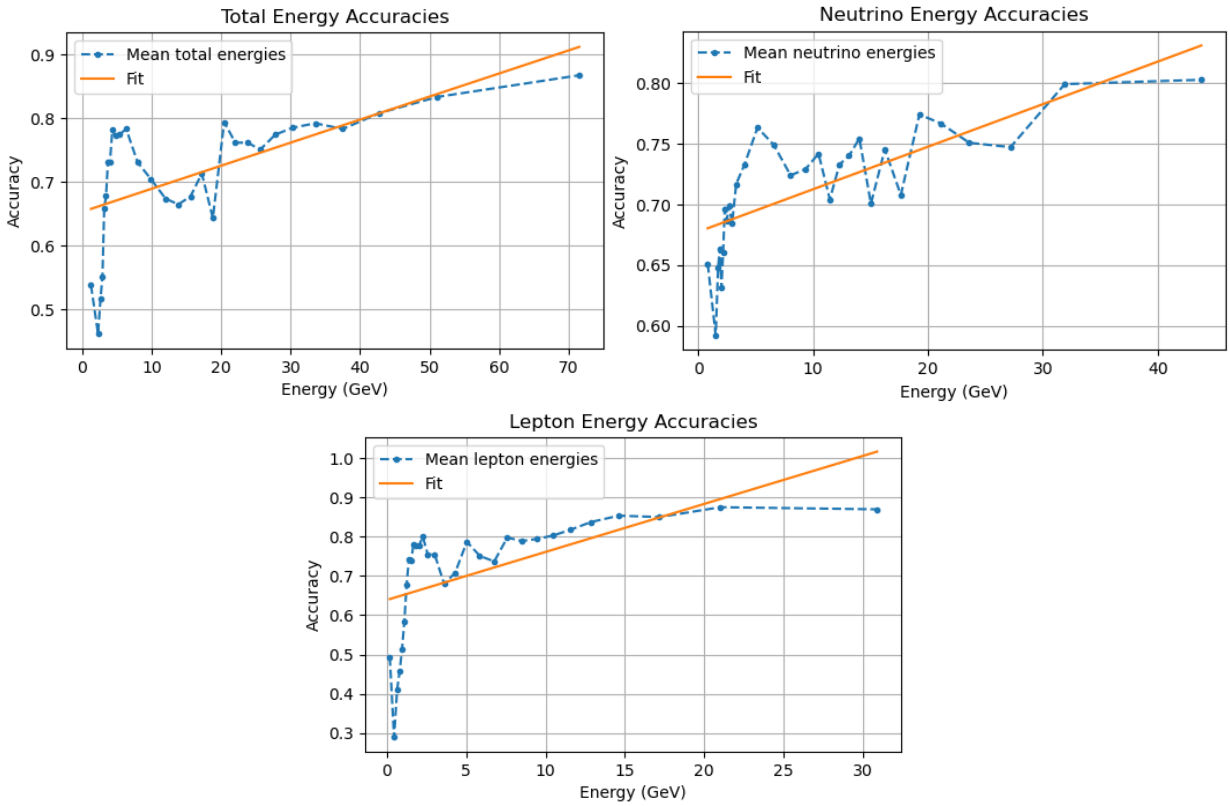

Fig. 12 Binary Classifier's accuracies for different energies.

2.5 Testing Classifier's Interaction Dependence

The model's performance was tested for the three separate types of ν_μ interactions, DIS, QE and Res. It was also tested at distinguishing interactions not belonging to the previous three types and to non-ν_μ interactions altogether (resulting accuracies are shown in Fig. 13).

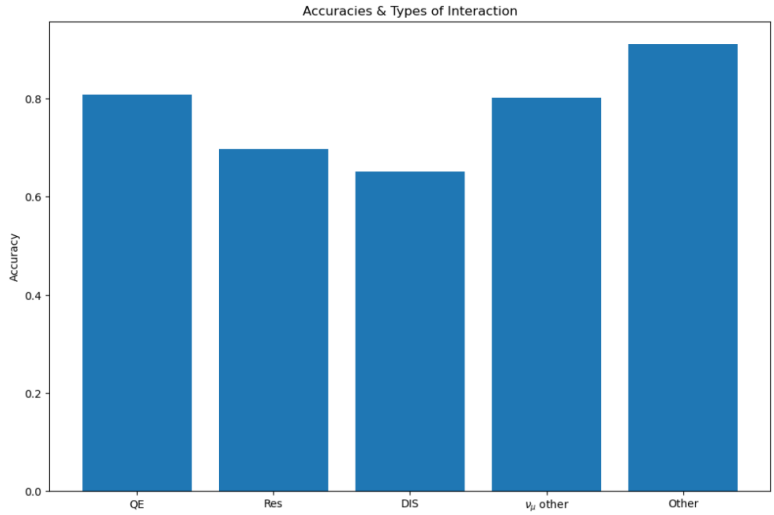

Fig. 13 Binary classifier's accuracies for different interactions. From left to right, the accuracies are: 0.809, 0.698, 0.650, 0.801, 0.911.

DIS events produce 'messier' images, which hinder the classifier's performance, while QE events produces significantly clearer tracks which allow for easier classification. Res interactions have clearer images than DIS but messier than QE, but they are more messier than they are clear, such that the accuracy lies between DIS and QE but closer to DIS than QE. The model's high accuracy for interactions not involving the ν_μ suggests that it has learnt some core features which distinguish ν_μ events from others.

3. Further Experiments

3.1 Predicting Neutrino Energies

A model with a similar architecture as the MVCNN was trained to predict neutrino energies given the same NOvA-like detector images as input. The loss was changed to mean squared error (MSE) and the metric used was the mean absolute error (MAE) to obtain estimates of how much the model's predictions diverged from the actual energy values. 32 filters, each 3x3, were used in both branches. The number of units in the final 3 dense layers were 32, 16, and 1, with a linear activation on the output layer, with 372,033 parameters. The model was trained on 66,989 pairs of images for 15 epochs using a batch size of 100 and a 0.2 validation split.

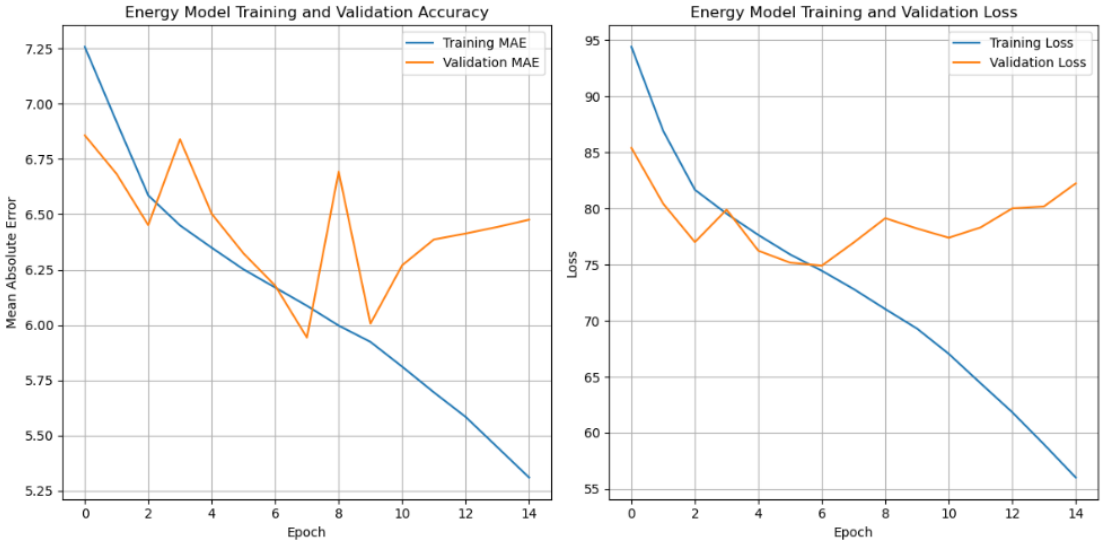

Fig. 14 Modified MVCNN model's metrics for neutrino energy predictions: MAE (left) and loss (right).

The final validation loss and MAE obtained were 82.2409 and 6.4757 respectively (see Fig. 14), with a test loss and MAE of 85.9182 and 6.5971 (evaluated on 16748 test image pairs). This means that the model deviates roughly by 6.5 GeV, which for higher energies (e.g. 25-40GeV) yields a 26%-16.25% relative error, which may be acceptable for some purposes. However, for low energies these errors

become much more significant such that the model's predictions become unreliable, especially when considering that lower energy interactions significantly outnumber higher energy interactions.

To better visualise the model's predictive capacity, its predicted distribution was plotted (see Fig. 15) next to the actual distribution, as well as its probability density function (PDF) and cumulative probability distribution (CPD).

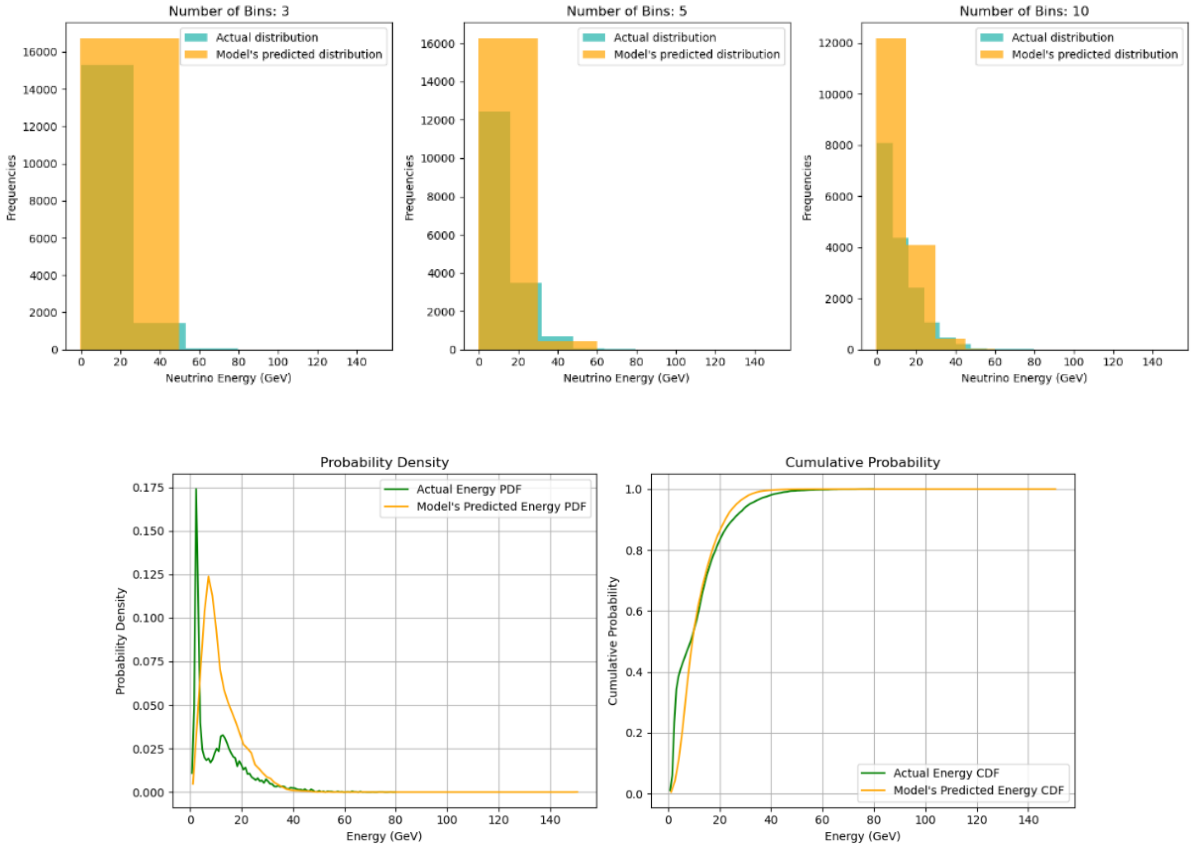

Fig. 15 The neutrino energy predictor's predicted distributions and statistics. Energy distributions (as histograms) for different numbers of bins (above) and cumulative/probability statistics (below).

Visually it becomes apparent that the model overpredicts frequencies at lowers energies. Its cumulative probability closely matches the actual shape. It also struggled to emulate the double peaked shape as lower energies, though it learnt the overall trend of higher frequencies at lower energies.

3.2 Predicting Interaction Mode

A model with the same architecture as the binary classifier was trained as a multi-class classifier to identify the interaction mode (DIS, QE or Res, not strictly limited to events involving the ν_μ). The output layer's activation was altered to have 3 units (for the three interaction types) and a softmax

activation which serves best for multi-class classification problems. Labels were constructed as one-hot encoded vectors (for further discussion on this topic see Ref [23]). DIS events greatly dwarfed the number of events of type QE and Res, such that, as with the binary classifier, the dataset was equalised for an even distribution of events. The model was trained (results are in Fig. 16) on 40,449 image pairs for 15 epochs with a batch size of 100 and a validation split of 0.2, with accuracy as the metric and categorical cross-entropy as the loss function.

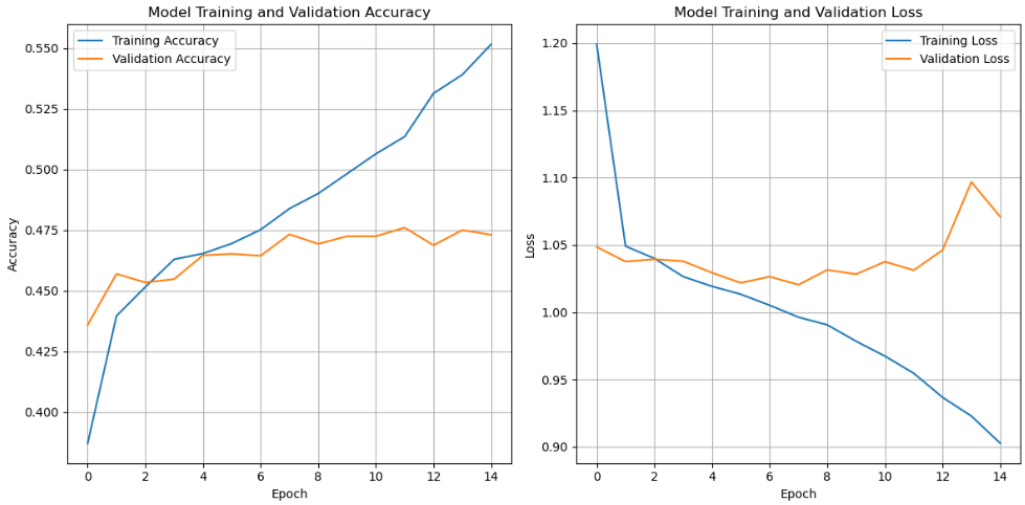

Fig. 16 Multi-classifier's training metrics.

The model was tested on 9,987 image pairs, achieving an accuracy and loss of 0.4646 and 1.1006, respectively. The classifier's confusion matrix (CM) is shown in Fig. 17.

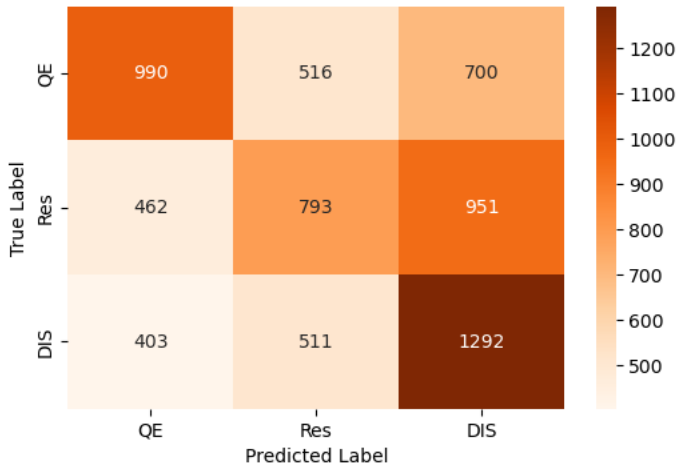

Fig. 17 Multi-classifier's confusion matrix.

Accuracies for each interaction mode were calculated from the CM and were found to be 0.44 (DIS), 0.44 (Res) and 0.53 (QE). Both classifiers discussed in this report achieved their highest accuracies on QE interactions, further supporting the notion that QE interactions offer clearer/cleaner images and

tracks than DIS and Res interactions, with features that are easier to recognise and to extract by convolutional filters.

4. Conclusion

Four models were trialled as binary-classifiers for simulated NOvA-like CC ν_μ events. It was found that a multi-view CNN using an inception module yielded the best results, with an accuracy of 0.80. Its performance was found to be highly dependent on event energies and interaction modes, with higher energies yielding greater accuracies and QE interactions (accuracy of 0.809) being significantly more accurately classified than DIS (0.650) or Res (0.698) interactions. A similar model was used as a multi-class classifier for predicting interaction modes, resulting in an accuracy of 0.4646, with different accuracies for QE (0.53), DIS (0.44) and Res (0.44) interactions. A multi-view CNN was also trained to predict neutrino energies (MAE of 6.5971).

The techniques used in this study have proved successful at classification CC ν_μ events, and may be transferred over for use in other neutrino-involved phenomena such as tidal disruption events used to probe for supermassive black holes [24]. With slight modifications they may also be applied to other types of particles such as protons, mesons etc.

Improvements on the binary classifier used could be achieved through a more detailed implementation of late-fusion methods by performing feature vector concatenations before one fully connected classification layer, which has been found to be an effective method at increasing classification accuracies [25].

References

[1] P. Abreu et al. (1992) 'Classification of the hadronic decays of the Z0 into b and c quark pairs using a neural network'. *Phys. Lett*., B295:383—395.

[2] Hermann Kolanoski. (1995) 'Application of artificial neural networks in particle physics'. *Nucl. Instrum. Meth*., A367:14—20.

[3] Leif Lonnblad, Carsten Peterson, and Thorsteinn Rognvaldsson. (1990) 'Finding Gluon Jets With a Neural Trigger'. *Phys. Rev. Lett*., 65:1321—1324.

[4] Carsten Peterson, Thorsteinn Rognvaldsson, and Leif Lonnblad. (1994) 'JETNET 3.0: A Versatile artificial neural network package'. *Comput. Phys. Commun*., 81:185—220.

[5] Carsten Peterson. (1989) 'Track Finding With Neural Networks'. *Nucl. Instrum. Meth*., A279:537.

[6] Kuo, T. K. and Pantaleone, J. (1989) 'Neutrino oscillations in matter'. *Rev. Mod. Phys.* American Physical Society, 61, pp. 937–979. doi: 10.1103/RevModPhys.61.937.

[7] Kustom, R. L. et al. 'Quasielastic Neutrino Scattering'. *Phys. Rev. Lett.* American Physical Society, 22, pp. 1014–1017. doi: 10.1103/PhysRevLett.22.1014, 1969.

[8] Bellini, G.B. & Ludhova, L. & Ranucci, Giorgia & Villante, F. (2014) 'Neutrino Oscillations'. *Advances in High Energy Physics*. 2014. 10.1155/2014/191960.

[9] B.R. Martin. (2009) *Nuclear and Particle Physics*, 2nd Edition. Wiley, West Sussex.

[10] Gardner, M. W. and Dorling, S. R. (1998) 'Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences', *Atmospheric Environment*, 32(14), pp. 2627–2636. doi: 10.1016/S1352-2310(97)00447-0.

[11] O'Shea, Keiron & Nash, Ryan. (2015). 'An Introduction to Convolutional Neural Networks'. ArXiv e-prints.

[12] A. Krizhevsky, I. Sutskever and G.E. Hinton. (2017) 'Imagenet classification with deep convolutional neural networks'. *Communications of the ACM 60*.

[13] Ioffe, S. and Szegedy, C. (2015) 'Batch normalization: accelerating deep network training by reducing internal covariate shift', in Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. Lille, France: JMLR.org (ICML'15), pp. 448–456.

[14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting'. Journal of Machine Learning Research, 15(56), 1929–1958. [Online] Available: http://jmlr.org/papers/v15/srivastava14a.html

[15] I. Goodfellow, Y. Bengio and A. Courville. (2016) *Deep learning*. MIT press.

[16] Szegedy, C. et al. (2015) 'Going deeper with convolutions', in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.

[17] Psihas, F., Groh, M., Tunnell, C. and Warburton, K. (2020). 'A review on machine learning for neutrino experiments'. *International Journal of Modern Physics A*, [online] 35(33), p.2043005. Available at: https://arxiv.org/pdf/2008.01242.pdf.

[18] Catano-Mur, Erika. (2022). 'Recent results from NOvA'. 10.48550/arXiv.2206.03542.

[19] Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). 'Multi-view Convolutional Neural Networks for 3D Shape Recognition'. arXiv:1505.00880.

[20] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). 'Rethinking the Inception Architecture for Computer Vision'. Retrieved from arXiv preprint arXiv:1512.00567.

[21] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). 'Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning'. Retrieved from arXiv preprint arXiv:1602.07261.

[22] Hoo, Z.H., Candlish, J., Teare, D. (2017). 'What is an ROC curve?' *Emergency Medicine Journal* 34, 357–359. doi:10.1136/emermed-2017-206735

[23] Samuels, Jamell. (2024). 'One-Hot Encoding and Two-Hot Encoding: An Introduction'. 10.13140/RG.2.2.21459.76327.

[24] Hayasaki, K. (2021) 'Neutrinos from tidal disruption events'. *Nat Astron* 5, 436–437. https://doi.org/10.1038/s41550-021-01309-z

[25] Seeland M, Mäder P (2021) Multi-view classification with convolutional neural networks. *PLoS ONE* 16(1): e0245230. https://doi.org/10.1371/journal.pone.0245230 pmid:33434208