

Deep Learning in Liquid argon time projection chambers: Optimizing the Architecture for Particle Classification

MicroBooNE Collaboration

R. Acciarri^g C. Adams^z R. An^h J. Asaadi^w M. Auger^a L. Bagby^g B. Baller^g G. Barr^q M. Bass^q
F. Bay^x M. Bishai^b A. Blake^j T. Boltonⁱ L. Bugel^m L. Camilleri^f D. Caratelli^f B. Carls^g
R. Castillo Fernandez^g F. Cavanna^g H. Chen^b E. Church^r D. Cianci^{l,f} G. H. Collin^m
J. M. Conrad^m M. Convery^u J. I. Crespo-Anadón^f M. Del Tutto^q D. Devitt^j S. Dytman^s
B. Eberly^u A. Ereditato^a L. Escudero Sanchez^c J. Esquivel^v B. T. Fleming^z W. Foreman^d
A. P. Furmanski^l G. T. Garvey^k V. Genty^f D. Goeldi^a S. Gollapinniⁱ N. Graf^s E. Gramellini^z
H. Greenlee^g R. Grossi^e R. Guenette^q A. Hackenburg^z P. Hamilton^v O. Hen^m J. Hewes^l
C. Hill^l J. Ho^d G. Horton-Smithⁱ C. James^g J. Jan de Vries^c C.-M. Jen^y L. Jiang^s
R. A. Johnson^e B. J. P. Jones^m J. Joshi^b H. Jostlein^g D. Kaleko^f G. Karagiorgi^{l,f}
W. Ketchum^g B. Kirby^b M. Kirby^g T. Kobilarcik^g I. Kreslo^a A. Laube^q Y. Li^b A. Lister^j
B. R. Littlejohn^h S. Lockwitz^g D. Lorca^a W. C. Louis^k M. Luethi^a B. Lundberg^g X. Luo^z
A. Marchionni^g C. Mariani^y J. Marshall^c D. A. Martinez Caicedo^h V. Meddageⁱ T. Miceli^o
G. B. Mills^k J. Moon^m M. Mooney^b C. D. Moore^g J. Mousseauⁿ R. Murrells^l D. Naples^s
P. Nienaber^t J. Nowak^j O. Palamara^g V. Paolone^s V. Papavassiliou^o S.F. Pate^o Z. Pavlovic^g
D. Porzio^l G. Pulliam^v X. Qian^b J. L. Raaf^g A. Rafiqueⁱ L. Rochester^u C. Rudolf von Rohr^a
B. Russell^z D. W. Schmitz^d A. Schukraft^g W. Seligman^f M. H. Shaevitz^f J. Sinclair^a
E. L. Snider^g M. Soderberg^v S. Söldner-Rembold^l S. R. Soleti^q P. Spentzouris^g J. Spitzⁿ
J. St. John^e T. Strauss^g A. M. Szelc^l N. Tagg^p K. Terao^f M. Thomson^c M. Toups^g Y.-T. Tsai^u
S. Tufanli^z T. Usher^u R. G. Van de Water^k B. Viren^b M. Weber^a J. Weston^c
D. A. Wickremasinghe^s K. Wierman^r S. Wolbers^g T. Wongjirad^m K. Woodruff^o T. Yang^g
G. P. Zeller^g J. Zennamo^d C. Zhang^b

^aUniversität Bern, Bern CH-3012, Switzerland

^bBrookhaven National Laboratory (BNL), Upton, NY, 11973, USA

^cUniversity of Cambridge, Cambridge CB3 0HE, United Kingdom

^dUniversity of Chicago, Chicago, IL, 60637, USA

^eUniversity of Cincinnati, Cincinnati, OH, 45221, USA

^fColumbia University, New York, NY, 10027, USA

^gFermi National Accelerator Laboratory (FNAL), Batavia, IL 60510, USA

^hIllinois Institute of Technology (IIT), Chicago, IL 60616, USA

ⁱKansas State University (KSU), Manhattan, KS, 66506, USA

^jLancaster University, Lancaster LA1 4YW, United Kingdom

^kLos Alamos National Laboratory (LANL), Los Alamos, NM, 87545, USA

^lThe University of Manchester, Manchester M13 9PL, United Kingdom

^m*Massachusetts Institute of Technology (MIT), Cambridge, MA, 02139, USA*

ⁿ*University of Michigan, Ann Arbor, MI, 48109, USA*

^o*New Mexico State University (NMSU), Las Cruces, NM, 88003, USA*

^p*Otterbein University, Westerville, OH, 43081, USA*

^q*University of Oxford, Oxford OX1 3RH, United Kingdom*

^r*Pacific Northwest National Laboratory (PNNL), Richland, WA, 99352, USA*

^s*University of Pittsburgh, Pittsburgh, PA, 15260, USA*

^t*Saint Mary's University of Minnesota, Winona, MN, 55987, USA*

^u*SLAC National Accelerator Laboratory, Menlo Park, CA, 94025, USA*

^v*Syracuse University, Syracuse, NY, 13244, USA*

^w*University of Texas, Arlington, TX, 76019, USA*

^x*TUBITAK Space Technologies Research Institute, METU Campus, TR-06800, Ankara, Turkey*

^y*Center for Neutrino Physics, Virginia Tech, Blacksburg, VA, 24061, USA*

^z*Yale University, New Haven, CT, 06520, USA*

ABSTRACT: Liquid Argon Time Projection Chambers (LArTPCs) produce image-like data that may be analyzed with deep neural networks (Deep Learning, or DL). This approach to data analysis is becoming more common in LArTPCs, as well as other high channel-count segmented detectors where events may be interpreted as images. In particular, MicroBooNE is an 8256 wire detector viewing 89 tons of fiducial liquid argon, where studies have demonstrated single particle classification capability using DL as well as progress toward DL neutrino identification [2]. In the case of these previous studies, the performance dependence on a variety of network architecture types was not explored in great depth. Using the same single particle classification metric, this paper seeks to provide a baseline of guidance for choosing the optimal network for the task at hand. We answer key questions about performance as a function of network architectures and show that networks __, __ and __ give functionally equivalent high performance, with gains in network __ in resource consumption that suggest its use.

Contents

1	Introduction	2
2	Particle Classification	2
3	Image Preparation	4
4	Training and Inference	4
5	The Architectures	5
5.1	ResNet	5
5.2	VGG	6
6	Conclusions	7

Figure 1: Example neutrino candidate observed in the MicroBooNE detector. The same interaction is shown in all three wire plane views. The top image is from wire plane U ; the middle from wire plane V ; and the bottom from wire plane Y . The image is at full resolution and is only from a portion of the full event view.

1 Introduction

MicroBooNE is the first large-scale LArTPC detector to collect data in a neutrino beam in the United States. The experiment has a total liquid argon mass of 170 tons and a rectangular active volume of $2.3 \times 2.6 \times 10.4 \text{ m}^3$. The system consists of two subdetectors: a time projection chamber (TPC) for tracking and calorimetry along with a light collection system for the detection of scintillation light. The detector began observing interactions of neutrinos from the FNAL Booster Neutrino Beam (BNB) [22] in October, 2015.

The information we consider in these performance studies is TPC readout only. Here, TPC information comes in the form of 3 wire readout planes, where each plane represents 1 image. Unlike the images normally used in image classification tasks, MicroBooNE’s images are sparse; this means that most of the roughly 9600x3000 image pixels are at their pedestal value for each event. The complications and final handling of such sized images is discussed in detail in a previous paper [2]. A typical event is shown in Figure 1.

In this paper, we survey variants of 2 well-used networks in image recognition literature: ResNet [] and Vgg16 []. We are interested in both their parameters and architectures, which means we will explore the depth and width of these networks, the addition of fully connected layers, stride sizes, input sparsification, and “bottlenecks.” We seek to understand whether or not networks that are less computationally expensive can match their more expensive counterpart networks in performance, in addition to seeking highest classification performance.

The organization of this paper consists of a description of our single particle metrics, the preparation of the images, the training and inference protocol used, and finally a description of the various architectures with results, followed by conclusions.

2 Particle Classification

In this paper we study the performance of five particle classification (e^- , γ , μ^- , π^- , proton) using a variety of Convolutional Neural Networks (CNNs). We simulate these particles in the 50-500 MeV/c momentum range of interest to MicroBooNE using the relevant Geant4-based LArSoft package [8]. The e^- and γ are electromagnetic particles that develop as showery objects in the detector. These are generally distinguishable by eye from their μ^- , π^- and proton track-like counterparts, which form more linear traces in the detector. The networks learn these features and others to make final identification. Such other distinguishing characteristics include dQ/ds variation at track ends (Bragg peak) and total Q deposition differences, per the Bethe-Bloch curves for different particle species. We note, however, that unlike a more traditional case of cat, dog, human-face image data separation, we are faced with some level of irreducible ambiguity in our five particle classification. In particular, μ^- ’s and π^- ’s are minimum-ionizing particles which will only be discernable in the

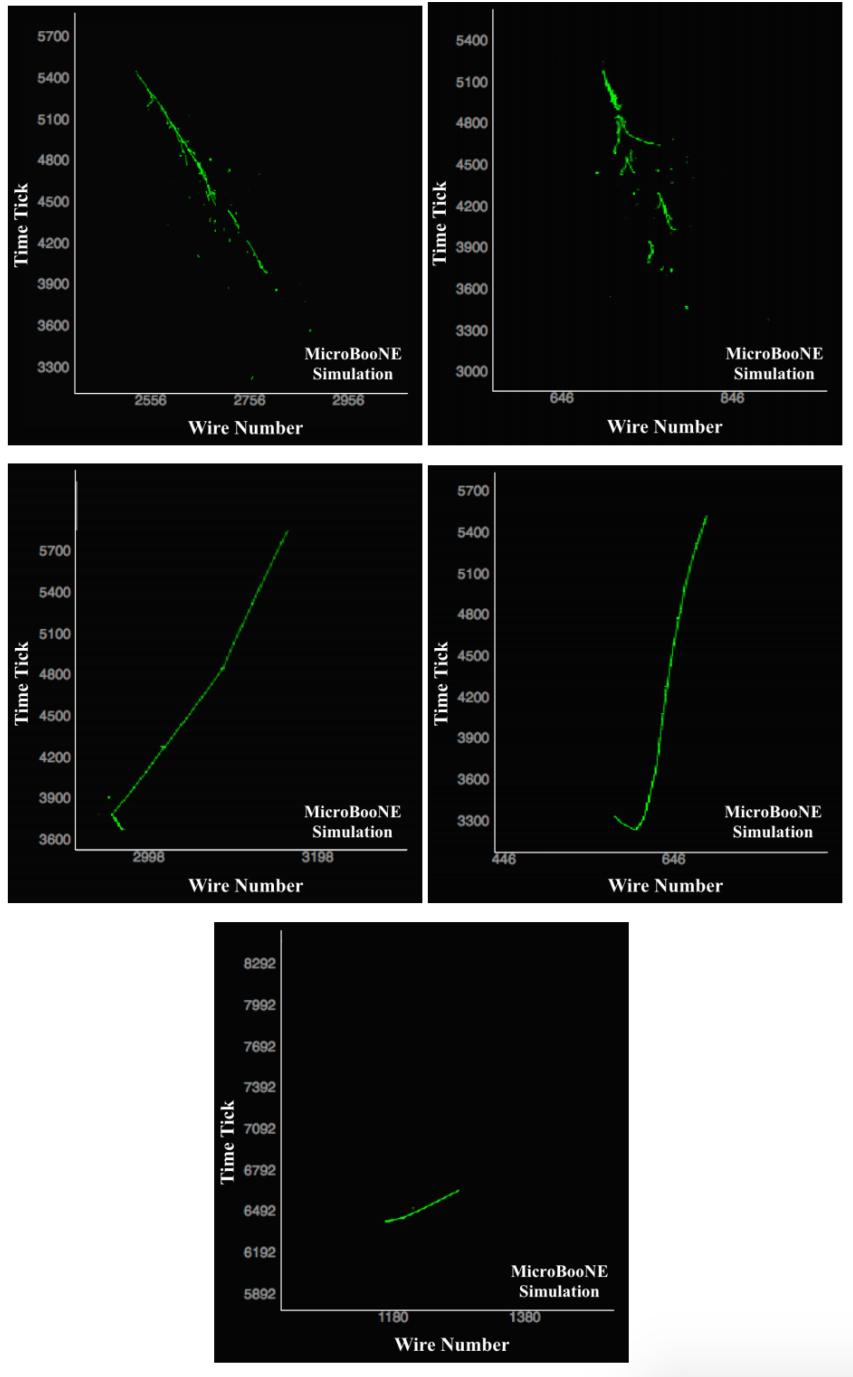


Figure 2: Example display for classification of each particle type in the collection plane. From top left and moving clockwise: e^- , γ , μ^- , proton, and π^- . (Should this cite previous paper again? Image?

cases that, say, the μ^- captures or the π^- scatters and produces a kink or inelastic interaction. Additionally, sometimes the simulated π^- will even decay to a μ^- . Therefore, we expect lower separation power between these particular species.



Figure 3: One example of GPU tower employed in the training (is this the representative image referred to?)

3 Image Preparation

Throughout all studies in this work, we use one of the most popular open-source CNN software frameworks, CAFFE [11], for CNN training and analysis. Input data is in a ROOT file format [16] created by the LArCV software framework [10], which we developed to act as the interface between LArSoft and CAFFE. LArCV is also an image data processing framework and is used to further process and analyze images as described in the following sections. One can find our custom version of CAFFE that utilizes the LArCV framework in [10]. The computing hardware used in this study consists of multiple servers at various institutions. We have two dual-GPU 12 GByte memory, NVIDIA GPU servers of similar specifications at the Massachusetts Institute of Technology, Columbia University, University of Michigan, Yale University. We also employ a DGX-1 10-GPU 18 GB/GPU server at Pacific Northwest National Laboratories.

... representative image picture here, as from the original CNN paper ... Figure 3

4 Training and Inference

A disciplined training and inference procedure is required in order to compare the performance of various networks. This procedure is as follows: First, we organize a training sample of 2000 events of each particle type. We additionally create 2 testing samples A and B: test A contain 10000 events and will be used to choose a weight file from the training, while test B contain 30000 and will be used for final performance reporting. Our training uses a batch size of 20 images, and runs iteratively where the complete forward/backward pass through each batch marks an iteration. During this time, the testA set's loss curve is monitored. When the loss has achieved its global minimum and begins to rise again, such that this minimum is reached at 85% or less of the total number of iterations, we stop training and note the iteration value. The goal is to ensure that learning has plateaued before we stop training. To ensure that the learning is stable at this observed minimum, we run the inference stage on the test A sample at the minimum iteration \pm 5 epochs.

The best inference run on a given epoch in test sample A is then identified as the weight file to be used in the final inference stage. This final inference is run on test sample B, and is used to extract final performance numbers.

5 The Architectures

Broad descriptions here of ResNet and Vgg16 and whatever else we have results for. GoogleNet, AlexNet? Why we want to stick with flavors of ResNet and Vgg here. Perhaps we don't want each flavor of ResNet and Vgg16 to be its own subsection.

5.1 ResNet

The first networks considered here are ResNet networks [31]. ResNets are made up primarily of residual convolutional modules, where each module represents a layer of the network. The general purpose of these modules is to preserve information learned in previous layers using an identity mapping, and to optimize from there. In other words, the modules refocus the network training from a function over feature space to a residual of that function over feature space. Networks constructed from these modules have been shown to improve performance as network depth increases while also to decrease computational training time [30].

ResNet is built on the idea that an increase in depth will lead to an increase in performance. For this reason, one of our primary goals here is to measure the computational costs of added depths to assess whether this depth is worth the extra resources it consumes. We additionally explore the width of the network and the addition of bottleneck feature (citation). An example of inference results on the testA sample is shown in Figure 4; the weight file used to calculate final performance on testB is extracted from this plot. This weight file is then used to run inference on testB. The results from this second inference run are shown in Table 1. Additional efficiency vs purity curves are shown in Figure 5.

Table 1: The right five columns denote the classification performance per particle type. Errors shown are statistical and assume a binomial distribution.

Network	Classified Particle Type				
	e^- [%]	γ [%]	μ^- [%]	π^- [%]	proton [%]
ResNet14b	84.8 ± 0.2	85.0 ± 0.2	90.5 ± 0.2	76.0 ± 0.2	94.0 ± 0.1
ResNet29b	87.2 ± 0.2	86.7 ± 0.2	90.7 ± 0.2	78.8 ± 0.2	96.8 ± 0.1
ResNet29b_w4	87.9 ± 0.2	86.0 ± 0.2	90.6 ± 0.2	78.9 ± 0.2	96.2 ± 0.1
ResNet50b	87.7 ± 0.2	85.9 ± 0.2	93.1 ± 0.1	78.5 ± 0.2	96.2 ± 0.1

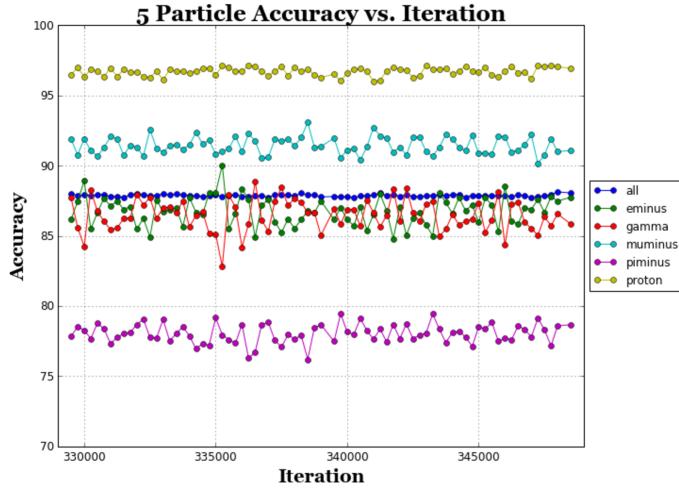


Figure 4: Example of inference results on the testA sample for ± 5 epochs around global loss minimum. The weight file for final performance assessment on testB is chosen based on the point of overall maximum accuracy in this plot.

Table 2: Computational performance for various ResNet architectures

Network	NUMBER CURRENTLY WRONG				
	Mem (MB)	Wall Time (s)	IO Wait Time (s)	IO Thread Time (s)	IO Thread Copy Time
ResNet14b	84.8 ± 0.2	85.0 ± 0.2	90.5 ± 0.2	76.0 ± 0.2	94.0 ± 0.1
ResNet29b	87.2 ± 0.2	86.7 ± 0.2	90.7 ± 0.2	78.8 ± 0.2	96.8 ± 0.1
ResNet29b_w4	87.9 ± 0.2	86.0 ± 0.2	90.6 ± 0.2	78.9 ± 0.2	96.2 ± 0.1
ResNet50b	87.7 ± 0.2	85.9 ± 0.2	93.1 ± 0.1	78.5 ± 0.2	96.2 ± 0.1

From Table 1 and Table 2 we can conclude several things. First, we see that in general, performance increases as depth increase. (We see from Kevin's slides that resources usage also increases with depth). From the Resnet29b_w4 results we see that increasing the width of the Resnet results in minimal performance increase at the high cost of computational resources. We also see that the bottleneck architecture increases performance minimally at the expense of resources (from Kevin's slides). We conclude here that resnet29 is the best choice of the ResNet in terms of computational and overall performance.

5.2 VGG

Not to be included, but for convenience, all results are should-be referenced at

<https://docs.google.com/presentation/d/1Rvo2Yt9KqRQk2eDy01WeA5GwaWamLvZR2CR3txfYbBE/edit#>

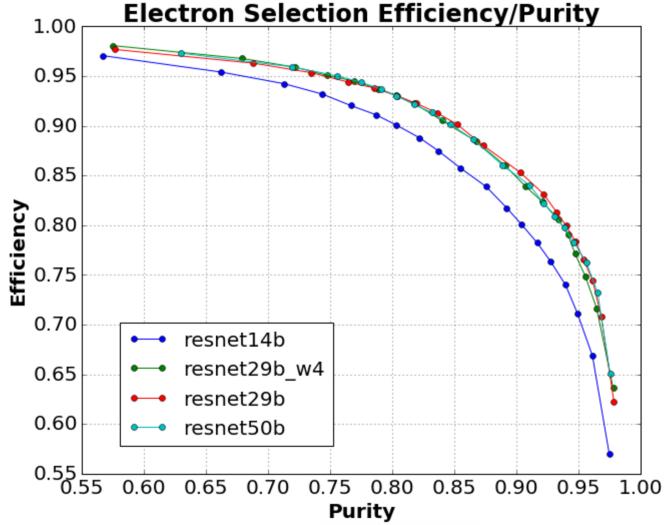


Figure 5: Efficiency vs purity curves for all resnet networks. As expected, ResNet14b is on the lower end, while the others are fairly comparable.

6 Conclusions

We have surveyed a variety of CNNs for the sake of classifying five charged particles as they appear in LArTPCs. Some interesting conclusions are that while Ref. [2] achieved formidable performance, a few other networks do as well or better, and it is seen now that RestNetXYZ and VGG16XYZ are better choices.

We plan to start with these networks for future physics-identification tasks in MicroBooNE; we believe similar guidance for general TPCs applies there, as well. We advocate, per this study, that the full and infinite (hyper) parameter space for CNNs need not be explored in those future analyses in favor of the identified best-performing networks here.

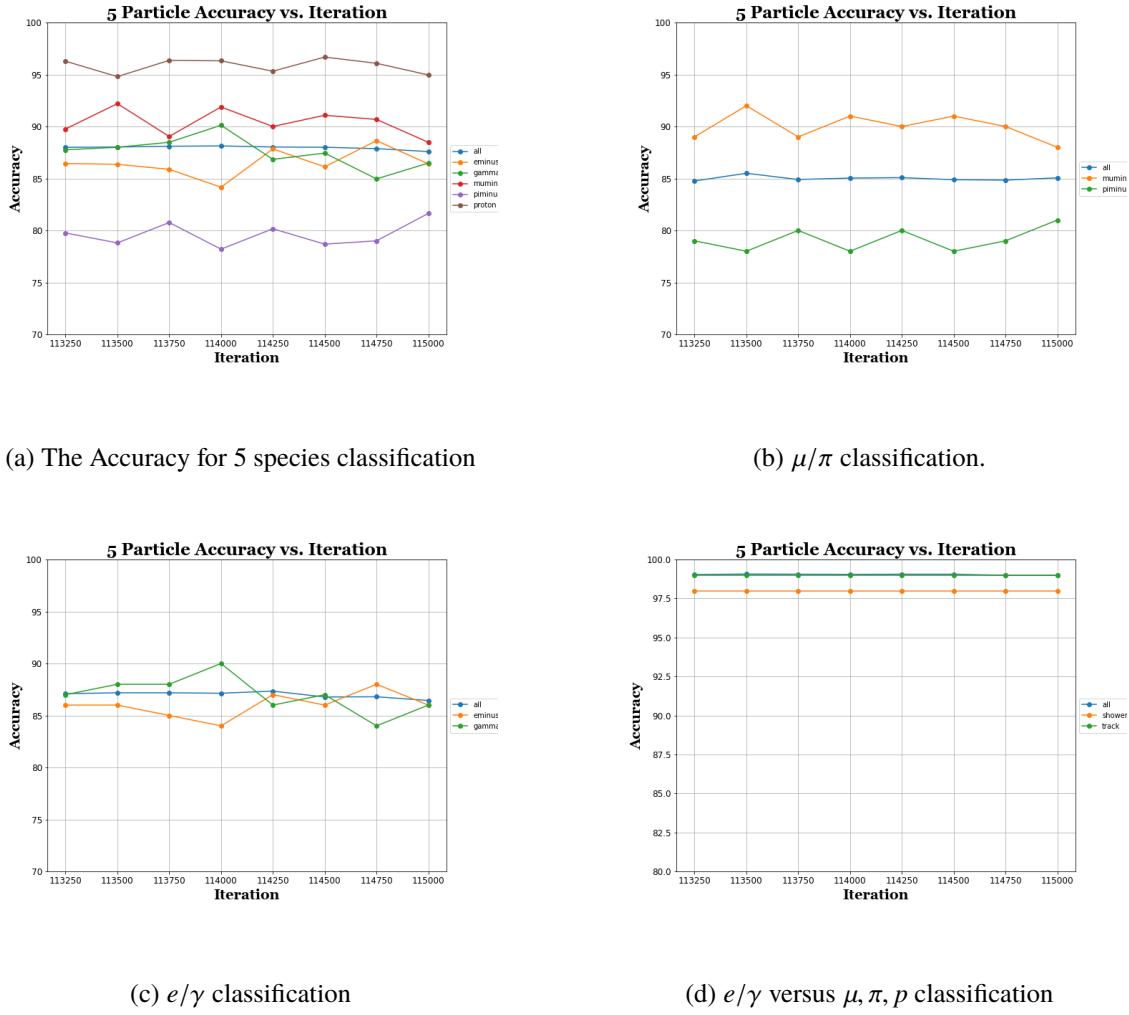


Figure 6: This is the (a) five particle classification, and (b) μ^-/π^- , (c) e/γ , and (d) shower/track separations for the vgg16b network.

References

- [1] Alex Krizhevsky *et al.* , “Imagenet Classification with Deep Convolutional Neural Networks,” *NIPS 25*, pages 1106–1114, 2012.
- [2] R. Acciari *et. al.* [MicroBooNE collaboration], “Convolutional Neural Networks Applied to Neutrino Events in a Liquid Argon Time Projection Chamber,” *1611.05531* [phsics.ins-det].
- [3] R. Acciari *et al.* [SBN Collaboration], “A Proposal for a Three Detector Short-Baseline Neutrino Oscillation Program in the Fermilab Booster Neutrino Beam,” *arXiv:1503.01520* [physics.ins-det].
- [4] R. Acciari *et al.* [DUNE Collaboration], “Long-Baseline Neutrino Facility (LBNF) and Deep Underground Neutrino Experiment (DUNE) : Volume 1: The LBNF and DUNE Projects,” *arXiv:1601.05471* [physics.ins-det].
- [5] E. Racah *et al.* “Revealing Fundamental Physics from the Daya Bay Neutrino Experiment using Deep Neural Networks,” *arXiv:1601.07621*.

- [6] A. Aurisano *et al.*, “A Convolutional Neural Network Neutrino Event Classifier,” 2016 *JINST* **11** P09001.
- [7] J. Renner *et al.* [NEXT Collaboration], “Background rejection in NEXT using deep neural networks,” *arxiv:1609.06202*.
- [8] LArSoft release *redmine 05.08.00*
<https://cdcvn.fnal.gov/redmine/projects/larsoft/wiki/ReleaseNotes050800>.
- [9] uboonecode release *redmine 05.08.00*
<https://cdcvn.fnal.gov/redmine/projects/uboonecode/wiki/ReleaseNotes050800>.
- [10] Genty, Terao and Wongjirad, LArCV: LArTPC Image Processing/Analysis Framework for Deep Learning <https://www.github.com/LArbys/LArCV>
- [11] Y. Jia *et al.*, “Caffe: Convolutional Architecture for Fast Feature Embedding,” *arXiv:1408.5093*, 2014.
- [12] Christian Szegedy *et al.*, “Going Deeper with Convolutions,” *arXiv:1409.4842*, 2014.
- [13] Ren, Shaoqing and He, Kaiming and Girshick, Ross and Sun, Jian, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *NIPS* **28** pages 91-99, 2015.
- [14] Szegedy *et al.*, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” *arXiv:1602.07261*.
- [15] Kaiming He *et al.*, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385*.
- [16] Rene Brun and Fons Rademakers, “ROOT – An object oriented data analysis framework,” *NIM A* **389** 81-86, 2015.
- [17] NVIDIA GeForce Titan X GPU Card Specification, <http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/specifications>
- [18] S. Agostinelli *et al.*, “GEANT4: A Simulation toolkit,” *NIM A* **506** 250-303, 2003.
- [19] MicroBooNE Collaboration, “TPC Noise Filtering,” MicroBooNE Public Note 1016-PUB.
<http://www-microboone.fnal.gov/publications/publicnotes/index.html>
- [20] MicroBooNE Collaboration. “TPC Signal Processing,” MicroBooNE Public Note 1017-PUB.
<http://www-microboone.fnal.gov/publications/publicnotes/index.html>
- [21] Montavon *et al.*, “Neural Networks: Tricks of the Trade,” Springer 2012.
- [22] A. A. Aguilar-Arevalo *et al.*, “The Neutrino Flux Prediction at MiniBooNE,” *PRD* **78**, 072002, 2009
- [23] C. Andreopoulos *et al.*, “The GENIE Neutrino Monte Carlo Generator,” *it NIM A* **614** 87-104, 2010.
- [24] Ioffe *et al.*, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” Proc. of the 32nd Intl. Conf. on Machine Learning 37, 2015.
- [25] Srivastava *et al.*, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” Journal of Machine Learning Research 15 1929-1958, 2014.
- [26] D. Silver *et al.*, “Mastering the game of Go with deep neural networks and tree search,” *Nature* **529**, 484–489, 2016.
- [27] Tieleman, Tijmen and Hinton, Geoffrey. “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” COURSERA: Neural Networks for Machine Learning, 4, 2012.
- [28] MicroBooNE Collaboration, “Selection and kinematic properties of ν_μ charged-current inclusive events in 5×10^{19} POT of MicroBooNE data,” <http://www-microboone.fnal.gov/publications/publicnotes/MICROBOONE-NOTE-1010-PUB.pdf>

- [29] Zheng *et al.*, “Improving the Robustness of Deep Neural Networks via Stability Training,” *arXiv:1604.04326*.
- [30] Szegedy *et al.*, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” *arXiv:1602.07261*.
- [31] Kaiming He *et al.*, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385*.