Electron Classification using deep learning

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Steenu Johnson

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2019

Supervisor: Dr. Sourabh Dube

Certificate

This is to certify that this dissertation entitled Electron Classification using deep learning towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Steenu Johnson at Indian Institute of Science Education and Research under the supervision of Dr. Sourabh Dube, Associate Professor, Department of Physics , during the academic year 2018-2019.

Dr. Sourabh Dube

Committee:

Dr. Sourabh Dube

Dr. Rajeev Bhalerao

This thesis is dedicated to my sister, Stacey Sera Johnson.

Declaration

I hereby declare that the matter embodied in the report entitled Electron Classification using deep learning are the results of the work carried out by me at the Department of Physics, Indian Institute of Science Education and Research, Pune, under the supervision of Dr. Sourabh Dube and the same has not been submitted elsewhere for any other degree.

Steenu

Steenu Johnson

Acknowledgments

I would like to express my special thanks to my thesis advisor, Dr. Sourabh Dube for his sincere and valuable guidance and encouragement. He steered me in the right direction while constantly allowing this to be my own work. I would also like to thank Dr. Rajeev Bhalerao and Dr. Arun Thalapillil for their timely inputs and passionate participation.

I would also like to acknowledge the whole of IISER Pune EHEP group to whom I am gratefully indebted for their comments on this thesis. Special thanks to Anshul Kapoor for his valuable suggestions and stimulating discussions.

Finally, I would like to thank my parents, my sister, and friends for providing me with boundless support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them.

Abstract

In this thesis, we examine to what extend machine learning techniques like convolutional neural networks can differentiate real and fake electrons better than the observables constructed by physicists. Our approach is treating the electron as an image, with pixel intensities given by local calorimeter deposits. Overall, the convolutional neural network outperforms the traditional physics observable used, for most signal efficiencies. We also find that the performance of the trained model was independent of the source of real and fake electrons. The performance of the model matched our expectations while being tested on electrons from different sources even those which the model wasn't trained for. This suggests that the network can extract relevant physical information about the real and fake electrons which the traditional observables cannot. This classifier is also more robust than the data-driven approaches used for fake electron estimation which relies on the source of electrons.

Contents

Abstract xi

1 Introduction 7

1.1 The Standard Model 8

1.2 Seesaw search with multi-leptons 9

1.3 Neural Networks in Quark - Gluon jet discrimination 12

1.4 Pattern recognition for electron classification 14

2 Convolutional Neural Network 17

2.1 Image classification pipeline 17

2.2 Learning - A parametric approach 18

2.3 Neural networks 19

2.4 Convolutional neural network 22

2.5 CNN architecture used 23

3 Electron classification 25

3.1 Dataset 25

3.2 Event selection 25

3.3 Electron Images 28

4 Results **33**

4.1 Low p_{T} electrons passing medium identification criteria 33

4.2 Changing the identification criteria 36

4.3 Changing the p_{T} range 36

4.4 Fake electrons generated using Alpgen generator 37

4.5 Fake electrons generated using Herwig generator 38

4.6 Testing the model 40

5 Conclusion and future direction **43**

List of Figures

1.1 Particles in the standard model. Retrieved from [7]. 9

1.2 Feynman diagrams showing multilepton final state arising from the production and decay of vector-like leptons (left) (Retrieved from [10].) and production and decay of gauginos in supersymmetric models (right). (Retrieved from [11].) 11

1.3 Relative isolation for electrons from Z decay and electrons from QCD jets. . 13

1.4 An η-ϕ space around the jet axis is constructed into a pixelated image where energy deposits become the pixel intensity. Retrieved from [6]. 14

1.5 A two-dimensional image constructed for an electron with the pixel intensity being energy deposited at a particular $\eta - \phi$ region of the calorimeter. 15

2.1 An image classification model using linear score function. Retrieved from [19]. 19

2.2 Optimization using gradient descent. Parameters W are updated after each step till the minimum of the loss function is reached. Retrieved from [20]. . . 20

2.3 Layer wise organization of a simple neural network (left) (Retrieved from [19].) and representation of a neuron in a hidden layer (right) which takes the dot product of previous layer output vector and weight vector and pass it through a non-linear function. 21

2.4 A deep neural network with two hidden layers. (Retrieved from [19].) 21

2.5 An input volume in red and first convolutional layer in blue. Each neuron is connected to only a local region of the input volume. Multiple neurons look at the same region of the input through different filters. (Retrieved from [19].) 23

2.6 Left: Input volume of size [224 x 224 x 64] is downsampled to size [112 x 112 x 64] by taking maximum of every non-overlapping 2 x 2 region of the input volume. Right: Maximum of four numbers being taken in 2 x 2 regions of the input volume .(Retrieved from [19].) 24

3.1 Feynman diagrams of a QCD event (left) and a fully hadronic $t\bar{t}$ decay (right). 26

3.2 The p_{T} of electrons from Z decay (left) and W decay in a $t\bar{t}$ event (right). . . 27

3.3 The p_{T} of electrons found in QCD events (left) and those in $t\bar{t}$ events decaying hadronically (right). 27

3.4 An electron image cropped to a 40 x 40 pixel region around the electron after centering at the position of reconstructed electron. 29

3.5 Example images of real electrons. These real electrons are selected from Z boson decay. 30

3.6 Example images of fake electrons. These fake electrons are selected from QCD events. 31

4.1 Output of Convolutional neural network. Signal - Electrons from Z decay, Background - Electrons from QCD Jets. 34

4.2 ROC curve for relative isolation, CNN , simple ANN and DNN. 35

4.3 Performance of CNN(left) and relative isolation(right) for electrons passing different identification criteria. 37

4.4 Left: Output distribution of CNN for real and fake electrons within a p_{T} range of 30GeV $< p_{\mathrm{T}} <$ 60GeV. Right: ROC curve for CNN classification of real and fake electrons within a p_{T} range of 30GeV $< p_{\mathrm{T}} <$ 60GeV (high p_{T}) compared with those in p_{T} range of 10GeV $< p_{\mathrm{T}} <$ 30GeV (low p_{T}). 38

4.5 Left: Output distribution of CNN where fake electrons are generated by Alpgen. Signal: Real electrons from Z decay. Background: fake electrons from QCD events. Right: ROC curve for CNN and relative isolation when the fake electrons are generated by Alpgen. 39

4.6 Left: Output distribution of CNN where fake electrons are generated by Herwig. Signal: Real electrons from Z decay. Background: fake electrons from QCD events. Right: ROC curve for CNN and relative isolation when the fake electrons are generated by Herwig generator. 39

4.7 Testing on electrons arising from a W boson decay(red). Test on electrons from Z decay is given for reference (blue). 40

4.8 Testing on electrons arising from a ttbar hadronic decay(red). Test on electrons from Z decay is given for reference (blue). 41

4.9 Testing on electrons within bottom quark jet. Test on electrons from Z decay is given for reference (blue). 42

List of Tables

3.1 Simulation samples, available in CERN open data portal, used for getting different sources of real and fake electrons, used in this thesis. 26

Chapter 1

Introduction

Particle physics is the study of the elementary constituents of matter and the interactions between them. It is also referred to as “high energy physics” because these fundamental particles are detected during energetic collisions of particles accelerated to nearly the speed of light in the particle accelerators. The currently known particles and interaction can be explained by a quantum field theory called the standard model (SM).

Most powerful accelerators are used to test the prediction and shortcomings of the standard model. Though the SM is a well-tested theory which has been successful in explaining many experimental results, it explains only 4% of the known universe. Physicists are trying to explain the rest and other unexplained phenomena using models of physics beyond the SM and experimental searches. One such model is described in a later section of this chapter.

In essence, the two main objectives of experimental high energy physics (HEP) is to probe the Standard Model(SM) with further precision and to search for new physics, by exploiting the full potential of the Large Hadron Collider(LHC). Both these objectives require a search for rare signals within an extensive background. Hence it is critical to explore different courses of action which will provide a better signal efficiency while maintaining a good background rejection. One example is better discrimination between particles in the signal and the background events. This leads to the aim of the thesis.

As we are currently dealing with datasets of size petabytes per year, generated by the LHC, the power of the experiments is profoundly affected by the performance of algorithms and other computational resources. This makes machine learning algorithms a useful approach as these are devised to utilize large datasets to find new features in data. The most used machine learning algorithms in this field are boosted decision trees (BDT) and neural networks. Generally, physics variables are used to train a machine learning model for regression or classification. A typical application of these algorithms is the classification of particles or events into signal and background instances. In regression application, the model learns a function, for example, an estimate of particles' energy from other measurements. The past decade saw an increase in the application of neural networks in this field for the improvement of several aspects ranging from electronics of the detector [1] to distinguishing signals from backgrounds [2]-[3]. A type of neural network, namely convolutional neural networks, is being used for this work.

This chapter introduces the problem and the approach being used. Neural networks are described in chapter 2. Chapter 3 explains the steps taken for electron classification followed by results in chapter 4.

1.1 The Standard Model

The SM of particle physics is a quantum field theory based on gauge symmetry which represents the elementary constituents of matter and their interactions. It organizes these particles in three generations and successfully explains matter interactions, i.e., strong, electromagnetic, and weak.

The gauge bosons which mediate these interactions are spin 1 particles obeying Bose-Einstein statistics. In addition, Higgs boson is a spin 0 particle predicted by Higgs mechanism. It was a missing piece which was finally discovered in 2012 by CMS and ATLAS experiments [8]-[9].

The matter constituents are fermions which are spin 1/2 particles obeying Fermi-Dirac statistics. These are of two categories, namely, quarks and leptons. Quarks carry both electric, and color charge and hence undergo both electromagnetic and strong interactions through the exchange of photons and gluons respectively. There are six quarks and six anti-quarks in SM. Unlike quarks, leptons do not carry color charge and hence do not have strong interactions. Charged leptons, namely, electron, muon and tau, and their anti-particles

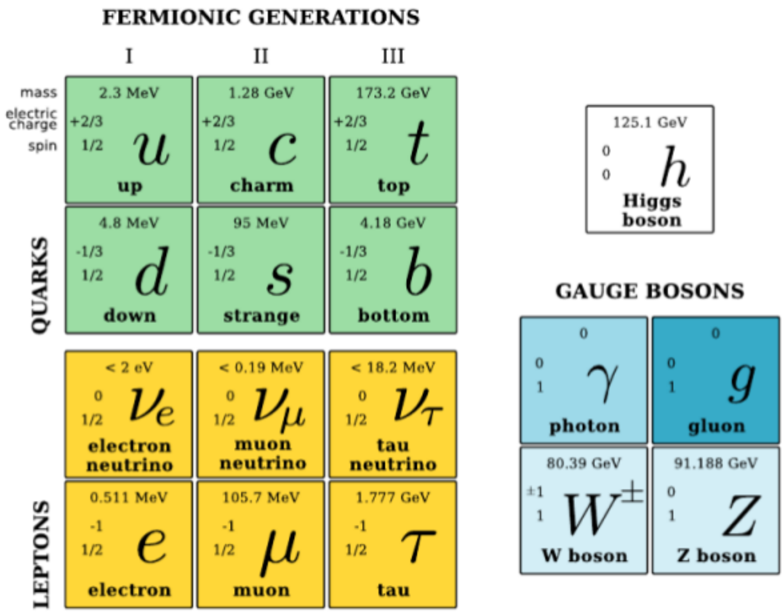

Figure 1.1: Particles in the standard model. Retrieved from [7].

undergo electromagnetic interaction while the neutral leptons, i.e., neutrinos do not interact electromagnetically. All fermions interact weakly through the exchange of W or Z bosons as well. The particles of the SM are summarised in Figure 1.1.

1.2 Seesaw search with multi-leptons

Beyond SM physics explores theories which try to explain the insufficiencies in the SM, such as the nature of dark matter, dark energy, neutrino oscillations e.t.c. One of the leading models addressing the question of neutrino mass is the Seesaw mechanism. The model predicts a heavy particle, with mass near the electroweak scale and coupled to SM leptons, whose mediation give rise to the small neutrino mass. Type I Seesaw mechanism predicts the particle to be a neutrino singlet, whereas type II and type III predicts a scalar and a fermion triplet respectively [13]. Let us consider the search described in Ref. [12]. The predicted massive fermion Σ in type III Seesaw mechanism can be charged ($\Sigma^{\pm}$) or neutral (Σ^{0}) lepton. These form an SU(2) triplet and can be pair produced at the LHC as either charged-charged or charged-neutral pair.

Consider the multilepton final state of type III Seesaw mechanism. Σ can be coupled to leptons via W, Z or H bosons. There are 27 different production and decay channels for the pair production and decay. A possible final state is one with at least three charged leptons, an example of this final state being $\Sigma^{\pm}\Sigma^{0} \rightarrow W^{\pm}\nu W^{\pm}l^{\mp} \rightarrow l^{\pm}\nu\nu l^{\pm}\nu l^{\mp}$ [12].

The background from SM can be classified into reducible and irreducible background. The primary irreducible background arises from the decay of dibosons like ZZ and WZ, which produces prompt leptons just like the final state of the signal. The reducible sources of background include leptonic decay of Z or $t\bar{t}$ accompanied by a lepton within or near to a jet, leptons from a heavy quark decay or a jet misidentified as a lepton. Other processes which contributes to the irreducible background are $t\bar{t}$W, $t\bar{t}$Z, triboson and Higgs boson production.

The search for multilepton events is classified into statistically independent search channels using the number of leptons and opposite sign same flavor (OSSF) pairs. The search channels are further classified as “on Z”, “below Z”, and “above Z” based on the presence of at least one OSSF pair forming an invariant mass relative to a Z boson mass window, 81-101 GeV. The main search channels are (i) three leptons, OSSF1, on Z/above Z/below Z and (ii) four or more leptons, OSSF1/OSSF2, on Z/above Z/below Z. The different search channels have different signal to background ratio. For example, WZ background has exactly three leptons and will fall in three lepton, OSSF1, on Z channel. The ZZ background falls in four leptons, OSSF2, on Z channel because the event has exactly four leptons, and the two OSSF pairs both fall in on Z mass region. Hence background estimation for each channel should be done independently.

The leptons from the decay of a Z or W boson are labelled as “real” leptons. These are prompt and typically isolated. Leptons within or close to a jet, those from a heavy quark decay or in some cases jets misidentified as leptons, are collectively referred here as “fake” leptons. Fake leptons may be non-prompt and are typically non-isolated. Fake leptons contribute highly to the reducible background.

The background from irreducible sources is estimated using simulation samples. But the reducible background is estimated using a data-driven approach called matrix method [26]. In this method, the rate of both real and fake leptons which passes a loose lepton selection, to also pass a tight lepton selection is measured in data. A dilepton sample is used for estimating this rate for real electrons and a trilepton sample (without signal) is used for estimating the

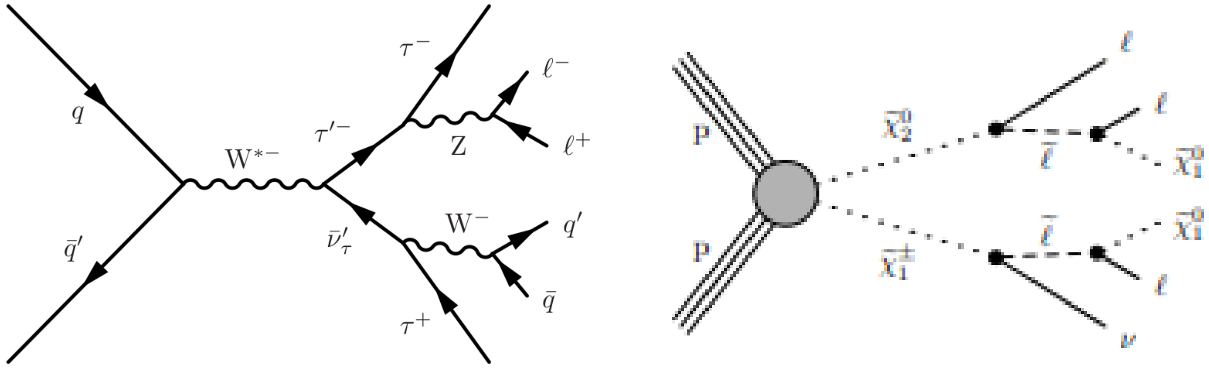

Figure 1.2: Feynman diagrams showing multilepton final state arising from the production and decay of vector-like leptons (left) (Retrieved from [10].) and production and decay of gauginos in supersymmetric models (right). (Retrieved from [11].)

fake rate of fake leptons. The rate for fake leptons is different based on the source of leptons. For example, the rate is different for Z+jets events and $t\bar{t}$ + jets events. This leads to higher uncertainty in the background estimation of these reducible backgrounds. Hence we need a classifier which classifies real leptons from fake leptons irrespective of the source of leptons.

The same observation can be made for the multilepton final state of a vector-like lepton (VLL) model [10] and a supersymmetry model [11] as shown in Figure 1.2. The signal consists of two or more real leptons. The primary reducible background in these cases is leptonic decay of Z or $t\bar{t}$ accompanied by fake leptons. Proper discrimination between real and fake leptons will increase the chances of observing the rare multilepton signal.

To repeat, real leptons are those which arise from a vector boson decay and are prompt and isolated. Fake leptons are those within or close to a jet, or those which originated from a heavy quark decay or jets being misidentified as leptons. Some examples include

- Decay of bottom quark or charm quark. As these leptons arise from a colored particle, they are usually within a spray of particles called jets formed by hadronization of quarks. These are **fake** leptons.

- Decay of a W boson or a virtual W boson. These are **real** leptons as the event is free of any colored particle.

- Decay of a Z boson to **real** lepton and anti-lepton.

- Top quark decay to a W boson and a bottom quark, both of which decays to leptons. W boson gives an **real** lepton while bottom quark can give one or more **fake** lepton.

- New physics models like Seesaw have final states with multiple **real** electrons

One criterion to distinguish real and fake leptons is relative isolation. Relative isolation is a parameter which quantifies activity around a lepton by taking the ratio of the sum of p_T of all non-leptonic particles around the lepton within a cone of radius $R = \sqrt{\eta^2 + \phi^2} = 0.3$ to the p_T of the electron [15].

$$R^{ele}_{iso} = \frac{\Sigma_i p_{\mathrm{T},i}}{p_{\mathrm{T},ele}} \tag{1.1}$$

where the sum is over all non-leptonic particles inside a cone of radius R. Typically, isolation criteria is $R^{ele}_{iso} < 0.1 - 0.2$ [15] - [16]. Electrons from Z decay have comparatively low values of relative isolation compared to electrons within QCD jets. That is, electrons from Z decay are more isolated than those from QCD Jets. Figure 1.3 shows relative isolation for electrons from Z decay and electrons within QCD jets.

One aspect to note is that relative isolation is calculated using reconstructed particles around the particle of interest. It indirectly relies on the performance of underlying reconstruction algorithm. Instead, one could try utilizing information used by the reconstruction algorithm instead of using reconstructed particles. This can include information like hits in silicon tracker, or energy deposits in calorimeter. In this thesis, we aim to classify electrons that are real from those that are fake using energy deposits in the calorimeter.

1.3 Neural Networks in Quark - Gluon jet discrimination

One interesting fact to note is that neural networks have been seen to perform equally good in discriminating two objects when supplied with physics inputs as well as with minimal physical input. An example in this context is the quark-gluon jet discrimination using deep learning [6] and an approach called “jet images” which is introduced in [4]-[5]. The idea is to treat energy deposited in the calorimeter as pixel intensities of a 2D image which then becomes input to an image recognition algorithm. In other words, the image classification

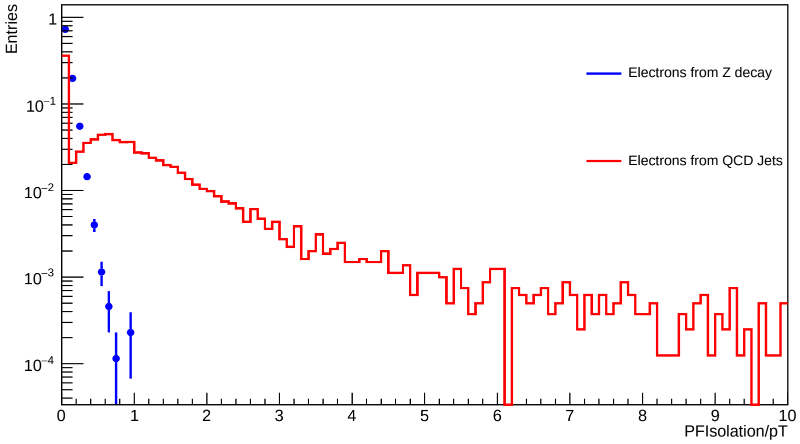

Figure 1.3: Relative isolation for electrons from Z decay and electrons from QCD jets.

model takes a pattern of energy deposited by particles within a jet as shown in Figure 1.4 and assigns probabilities whether it is a quark or gluon jet.

In this work, the images of both quark and gluon jets were 33 pixels wide, 33 pixels tall and has three color channels Red, Green, Blue. Each pixel in these images represents a specific area of the detector $\Delta\eta\Delta\phi$ and the three color channels are transverse momenta of charged particles, neutral particles, and charged particle multiplicity respectively. Therefore an image consisted of 33 x 33 x 3 numbers, or a total of 3267 numbers. Each number ranges from 0 (no energy is deposited in a particular region by any particles) to 1 (all energy is deposited in a specific area) since normalized images were being used. The task was to turn these 3267 numbers into a single label, such as "quark jet".

Non-perturbative effects like hadronization and other factors like pileup make the quark-gluon jet classification challenging. The approach was to provide the algorithm examples of both quark jet images and gluon jet images and let the learning algorithm learn the energy deposit pattern of both. This is called a data-driven approach since this method depends on acquiring a training dataset of both categories first.

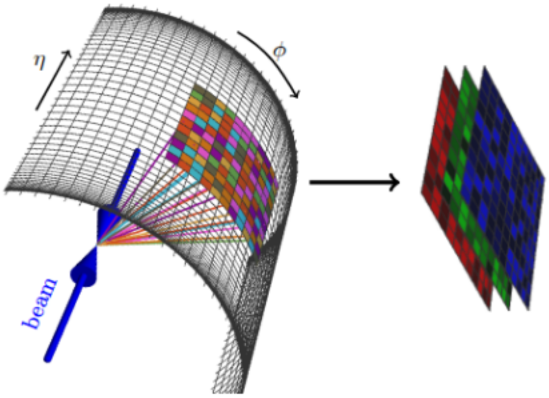

Figure 1.4: An η-ϕ space around the jet axis is constructed into a pixelated image where energy deposits become the pixel intensity. Retrieved from [6].

1.4 Pattern recognition for electron classification

Instead of using the summed energy of reconstructed particles, energy deposited around an electron in the calorimeter can be used for classification. Energy deposits are recorded by the calorimeter crystals. These crystals take up distinct $\Delta\eta\Delta\phi$ region of the calorimeter. For example, each crystal of electromagnetic calorimeter in the barrel subtends an area of 0.0175 x 0.0175 in $\eta - \phi$ space [18]. For our purpose, we are interested in energy deposits, in a specific $\eta - \phi$ region around the electron, along with its spatial location in $\eta - \phi$ space.

This section describes how the information of energy deposit along with its spatial location is condensed in the form of an image. Consider a specific area in $\eta - \phi$ space around the lepton in the calorimeter to be the area of interest. If this area is divided into several cells of smaller area $\Delta\eta\Delta\phi$, then each cell corresponds to a pixel of an image. Spatial structure is preserved, i.e., two neighboring cells in the calorimeter becomes two adjacent pixels in an image. Amount of energy deposited in a particular $\Delta\eta\Delta\phi$ area in the calorimeter becomes the grayscale intensity of the corresponding pixel. Figure 1.5 shows the 2d image constructed for an electron, the two dimensions are pseudorapidity η, and azimuthal angle ϕ and intensity of a pixel is the intensity of energy deposited in a particular calorimeter region.

Once the relevant information is represented in the form of an image, pattern recognition techniques can be applied for solving the electron classification problem. Pattern recognition

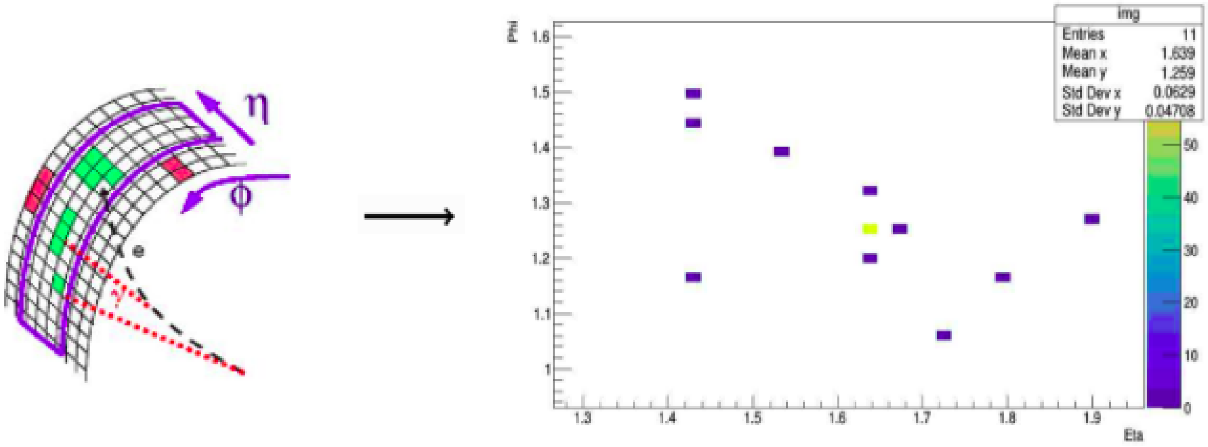

Figure 1.5: A two-dimensional image constructed for an electron with the pixel intensity being energy deposited at a particular $\eta - \phi$ region of the calorimeter.

or image classification is a problem of designating labels to an input image; in this case, labels being real electron or fake electron. The algorithm used for this problem is the convolutional neural network which is explained in Chapter 3.

The following are the factors which make this problem of recognizing visual information about an electron non-trivial even for humans.

- **Pileup**. As the protons are circulated in bunches at the LHC, multiple proton-proton collisions can happen instead of one. Decay products from different collisions in the same bunch other than the collision of interest are called pileup. Energy deposits from pileup can give rise to background noise in the image.
- **Bremsstrahlung radiation**. As shown in Figure 1.5, an accelerating electron can radiate photons which deposits energy close to the electron of interest.

Chapter 2

Convolutional Neural Network

Deep learning has had a significant impact in the field of high energy physics during the past decade. This chapter gives a brief description of machine learning, particularly, for image classification. It also introduces neural networks and convolutional neural networks which has been a promising endeavor when it comes to a large amount of data and features.

2.1 Image classification pipeline

The task of image classification is to designate labels to an input image, the labels, in our case, being real or fake electron. The complete pipeline can be divided into three steps:

- **Input**. Input consists of a set of images each of them labeled as either a real or a fake electron. This is the training set.

- **Learning**. The training set of images is used to teach the algorithm what real or fake electrons look like in the calorimeter. This is henceforth referred to as training the classifier.

- **Evaluation**. After training, the classifier is evaluated. This is done by showing a new set of images called the testing set to the classifier and comparing its prediction with the actual labels.

2.2 Learning - A parametric approach

The learning algorithm has two major components:

- **Score function**: maps the input image to the required output.
- **Loss function**: quantifies agreement between prediction and the correct label.

The desired output can be the probability for the input image to be a real electron so that an ideal classifier outputs one if the input is a real electron image and zero if the input is a fake electron image. This now becomes an optimization problem in which we optimize the parameters of the score function so that we get the desired output by minimizing loss function with respect to the parameters.

The simplest example of a score function is a linear mapping:

$$f(x_i, W, b) = W x_i + b \tag{2.1}$$

For an image classification problem, x_i in equation 2.1 can be pixel intensities where all pixels are flattened out into a column vector. The matrix W and vector b are the parameters of the function, often called weights and biases respectively. Figure 2.1 represents image classification using a linear model. The output gives a score for each class. A good classifier is expected to give the highest score for the correct category. The parameters W and b are learned during training which uses the training data. After the training is complete, the entire training set can be discarded because only the learned parameters are required for further classification. The model can then be tested on new images.

Loss function measures the agreement between the prediction and the true label. An example of a loss function is

$$L_i = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + \Delta) \tag{2.2}$$

where s_{y_i} is the score for the correct class, and s_j is the score for other classes, for one specific example in the training data. The function requires the correct class to have a higher score than incorrect categories by a margin of Δ. A higher loss function implies poor performance. For a good classifier, we expect the loss to be low. The objective is to find parameters that

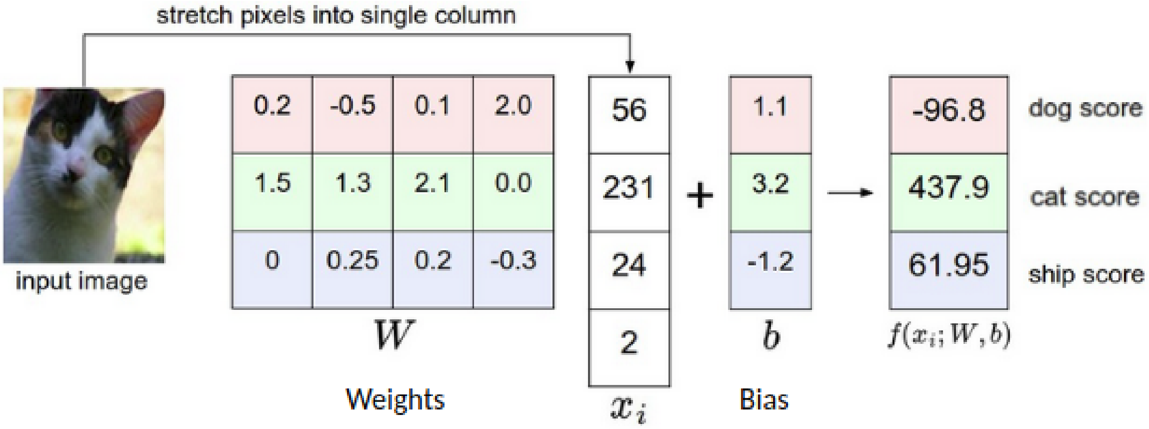

Figure 2.1: An image classification model using linear score function. Retrieved from [19].

will give minimum total loss for all examples in the training data. This leads to the final component of the parametric approach for learning, i.e., optimization.

Optimization is the process of finding the parameters, e.g., W and b, to minimize loss function. This will be an iterative process. We start with random values of the parameters and refine them in each iteration until the loss is minimized. The most commonly used approach is gradient descent. In this approach, we compute the gradient of loss function with respect to parameters $\Delta_w L$. The gradient of a function gives the steepest descent direction. Hence gradient will provide the best direction for updating parameters to minimize the loss function. (See Figure 2.2).

2.3 Neural networks

Neural networks are designed as a collection of nodes connected in an acyclic graph, organized in distinct layers. The nodes are also called neurons. The output of neurons from one layer becomes input to those in the next layer. The simplest neural network representation is shown in Figure 2.3. The most common type of layer is a fully connected layer in which all neurons from two adjacent layers are pairwise connected, but neurons within the same layer have no connections. The output of a neuron in a fully connected hidden layer is determined by first taking the dot product of the previous layer output vector and the weight vector, and

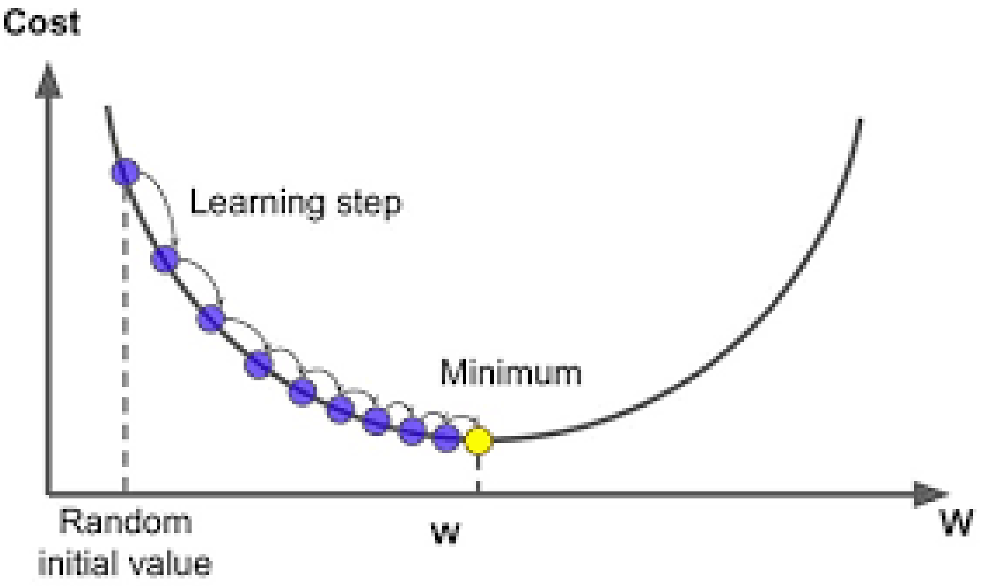

Figure 2.2: Optimization using gradient descent. Parameters W are updated after each step till the minimum of the loss function is reached. Retrieved from [20].

passing it to a non-linear function called the activation function. Representation of a neuron in a hidden layer is shown in Figure 2.3. Neural networks with more than one hidden layer is called a deep neural network and is shown in Figure 2.4. Introductions to deep learning and neural networks can be found in [40] and [41].

Score function for a linear classification was $s = Wx$ where x was the input vector and W was the learned parameter. A neural network is a similar learning algorithm with a slightly more complicated score function. The simplest neural network has score function $s = W_2 max(0, W_1 x)$, where W_1 and W_2 are both parameters to be learned using gradient descent. A non linear function called relu/ max(0,x) applies elementwise non-linearity. This can be replaced by other non-linear functions like sigmoid. This non-linear functions make the neural network different from a linear classification model which can approximate only linear functions. Another layer can be added to the neural network, and the score function then becomes $s = W_3 max(0, W_2 max(0, W_1 x))$, where W_3, W_2 and W_1 are all parameters to be learned.

Neural networks can approximate any continuous function [21]. Given any continous function $f(x)$ and $\epsilon > 0$, there exists a function represented by neural network of one hidden layer $g(x)$ such that $\forall x, |f(x) - g(x)| < \epsilon$.

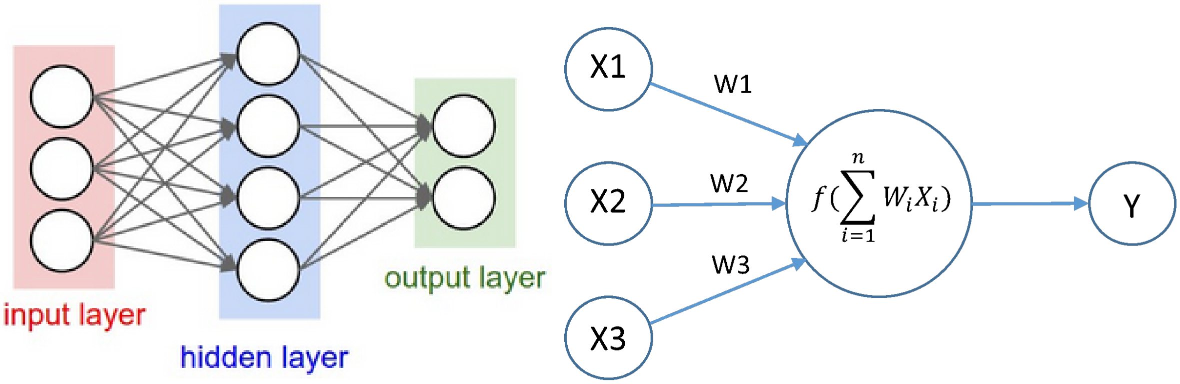

Figure 2.3: Layer wise organization of a simple neural network (left) (Retrieved from [19].) and representation of a neuron in a hidden layer (right) which takes the dot product of previous layer output vector and weight vector and pass it through a non-linear function.

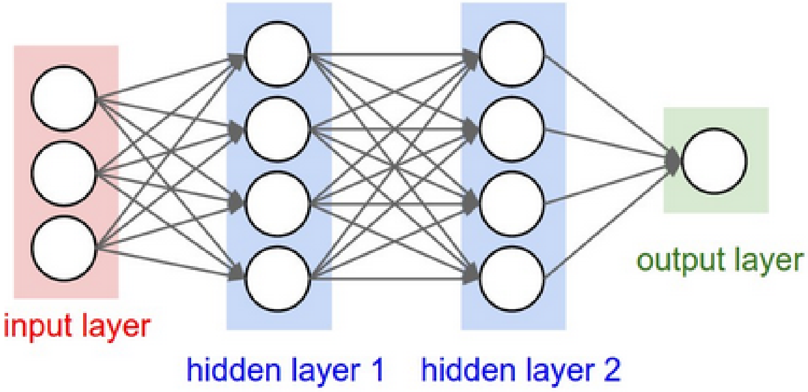

Figure 2.4: A deep neural network with two hidden layers. (Retrieved from [19].)

2.4 Convolutional neural network

A Convolutional Neural Network (CNN) is the category of deep neural networks designed specifically for image classification [25]. Most features of neural networks are valid for CNN as well: CNNs are made up of neuron which receives some input, perform a dot product followed by non-linearity. CNN also represents a differentiable score function. The significant difference in CNNs is that there are layers in which neurons are arranged in a 3D volume: width, height, and depth, unlike neural network where the hidden layer is a 1D collection of neurons. These layers are called convolutional layers. Neurons in a convolutional layer will only be connected to a small region of the previous layer rather than the entire layer, which is the case for a fully connected layer.

The three main types of layers in a CNN are Convolutional layer, Pooling layer, and Fully-Connected layer. The simplest CNN architecture would be:

- INPUT layer: contains pixel values of an image.
- CONV layer: composed of neurons which are connected to only specific regions of the input which outputs dot product of weights and the connected area of the input.
- RELU layer: applies elementwise non-linear function, called the activation function. The common function used is relu which is max(0,x) where x is the input to the function.
- POOL layer: applies downsampling along width and height.
- FC(fully-connected) layer: computes class scores just like in the case of regular neural network.

The convolutional layer is the most crucial part of a CNN which distinguishes it from regular neural networks. The parameters to be learned in a convolutional layer is a set of filters. A filter is a smaller matrix than the input layer (smaller dimensions in terms of width and height but extends through a full depth of input volume). Each filter slides across the input volume and computes dot product between filter entries and a specific region in the input. This gives a two dimensional set of neurons also called activation maps. Each neuron is connected only to a local region of the input. The connected region is called the receptive field of the neuron. This will be equal to filter size of a neuron.

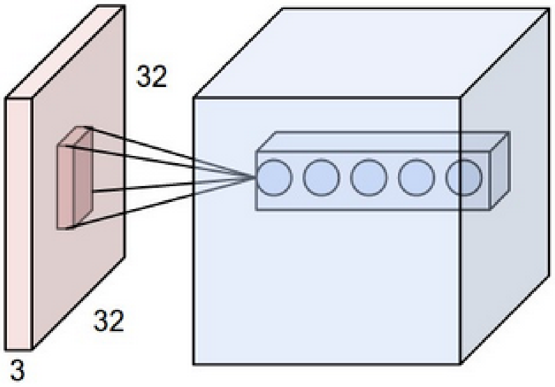

Figure 2.5: An input volume in red and first convolutional layer in blue. Each neuron is connected to only a local region of the input volume. Multiple neurons look at the same region of the input through different filters. (Retrieved from [19].)

Each filter produces a separate 2D activation map. These activation maps are stacked to create the 3D volume of neurons in the convolutional layer, as shown in Figure 2.5. In other words, the depth of a convolutional layer is equal to the number of the filters used. Each filter is expected to learn a specific visual feature like edge, color etc.

A pooling layer is inserted, preferentially after each convolutional layer, to reduce the spatial size (width and height) of the previous layer thereby reducing the number of parameters using MAX operation. This reduces the computational complexity. Also once a feature is learned its exact spatial location is unimportant. Its relative position with respect to other features are preserved mostly during max poolimg while reducing the number of parameters. The most common way of pooling is by applying the max operation to every non-overlapping 2 x 2 region of the input volume. This reduces the spatial size of the previous layer by half. The depth dimension remains the same. An example of max pooling is shown in Figure 2.6.

2.5 CNN architecture used

CNN architecture used in this work was input layer followed by a convolutional layer which was follwed by an element wise non-linear function RELU and a pool layer. The layers following the input layer (till pool layer) was repeated M times with $M \geq 2$. This was

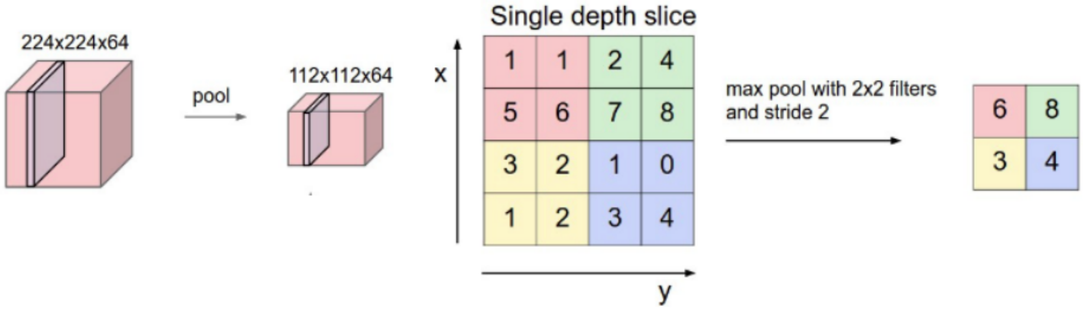

Figure 2.6: Left: Input volume of size [224 x 224 x 64] is downsampled to size [112 x 112 x 64] by taking maximum of every non-overlapping 2 x 2 region of the input volume. Right: Maximum of four numbers being taken in 2 x 2 regions of the input volume .(Retrieved from [19].)

followed by a fully connected layer after which non-linear function RELU was applied. This was followed by an output layer.:
$INPUT \rightarrow [CONV \rightarrow RELU \rightarrow POOL] * M \rightarrow [FC \rightarrow RELU] \rightarrow FC$
where * indicates repetition and $M \geq 2$.

The learning process, as explained above, is through optimizing the parameters (filters and weights connected to the fully connected layer) through gradient descent of a loss function. The training of all neural networks used in this study was performed with deep learning library Tensorflow [42] using NVIDIA Denver2 , 256-core Pascal GPU.

Chapter 3

Electron classification

This chapter mentions the simulation samples used for collecting real and fake electrons in the initial section. The later sections of this chapter describes the event selection for electrons and construction of electron images.

3.1 Dataset

The CERN Open Data portal presents data and simulation datasets used by CMS and ATLAS at 7 TeV and 8 TeV. The source of real electrons are Z boson decay, leptonic decay of a W boson in a $t\bar{t}$ decay or a WZ decay. The source of fake electrons are QCD jets if any hadron within the jet decays to an electron. Another source of fake leptons is a fully hadronic $t\bar{t}$ decay in which any jet can give rise to a fake electron. The Feynman diagrams of these sources are shown in Figure 3.1. The details of the datasets used in this study, are given in Table 3.1.

3.2 Event selection

Electrons coming from a Z boson decay are selected from the Drell Yan events (row 1 in Table 3.1) by requiring the dielectron invariant mass to be $80 < m_{ee} < 120$ GeV, since

Number	Process	Dataset
1	$Z/\gamma*$	DYToEE-M-20-8TeV-powheg-pythia6 [27]
2		DYJetsToLL-M-50-TuneZ2Star-8TeV-madgraph-tarball-tauola-tauPolarOff [28]
3	QCD	QCD6Jets-Pt-100to180-8TeV-alpgen [29]
4		QCD6Jets-Pt-250to400-8TeV-alpgen [30]
5		QCD4Jets-Pt-400to5600-8TeV-alpgen [31]
6		QCD6Jets-Pt-400to5600-8TeV-alpgen [32]
7		QCD-Pt-30to50-8TeV-herwig6 [33]
8		QCD-Pt-50to80-8TeV-herwig6 [34]
9		QCD-Pt-170to300-8TeV-herwig6 [35]
10		QCD4Jets-Pt-180to250-8TeV-alpgen [36]
11	$t\bar{t}$	TTJets-HadronicDecays-8TeV-madgraph-tauola [37]
12		TTJets-FullLeptMGDecays-TuneP11mpiHi-8TeV-madgraph-tauola [38]
13	WZ	WZJetsTo3LNu-8TeV-TuneZ2Star-madgraph-tauola [39]

Table 3.1: Simulation samples, available in CERN open data portal, used for getting different sources of real and fake electrons, used in this thesis.

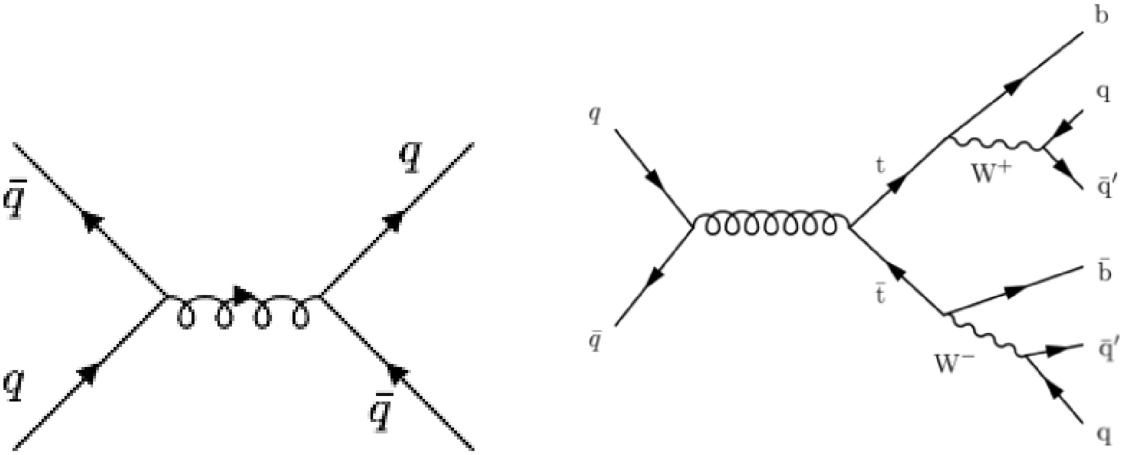

Figure 3.1: Feynman diagrams of a QCD event (left) and a fully hadronic $t\bar{t}$ decay (right).

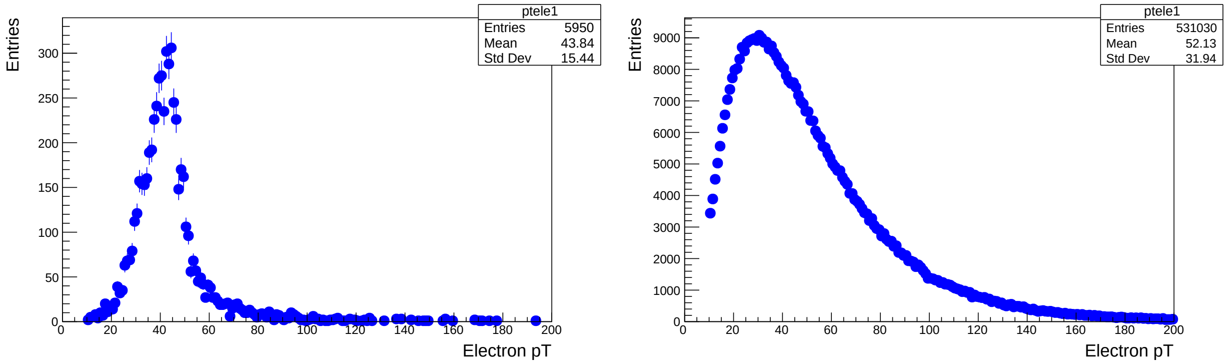

Figure 3.2: The p_{T} of electrons from Z decay (left) and W decay in a $t\bar{t}$ event (right).

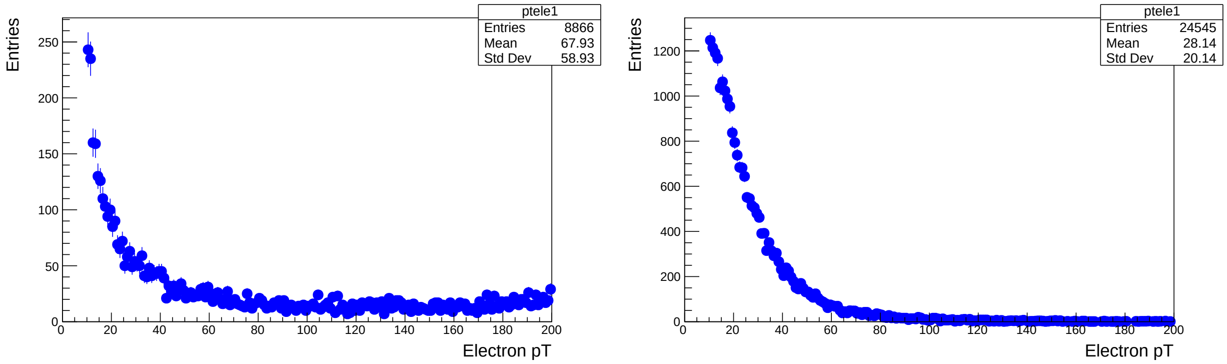

Figure 3.3: The p_{T} of electrons found in QCD events (left) and those in $t\bar{t}$ events decaying hadronically (right).

$m_Z = 91.2$ GeV.

Leading electrons from a Z decay and electrons from leptonic decay of a W boson from a $t\bar{t}$ decay (row 12 in Table 3.1) are the main sources of real electrons used in this work. Figure 3.2 shows the p_{T} of these electrons. Similarly, electrons from a QCD event (rows 3-10 in Table 3.1) and a $t\bar{t}$ hadronic decay (row 11 in Table 3.1) are the main sources of fake electrons used in this work. Figure 3.3 shows the p_{T} of these electrons. This work chooses electrons from two different ranges of p_{T}, $10 < p_{\mathrm{T}} < 30$ GeV and $30 < p_{\mathrm{T}} < 60$ GeV. The electrons falling in the former range of p_{T} will be referred to as low p_{T} electrons and those falling in the latter p_{T} range will be referred to as high p_{T} electrons.

A set of identification criteria is tuned to get three separate working points. These are called, in the increasing order of background rejection and decreasing order of signal efficiency, as Loose, Medium, and Tight. These working points are inclusive, Loose $\subset$ Medium $\subset$ Tight.

In essence, the identification requirement for electrons is as follows:

- 10 GeV $< p_{\mathrm{T}} <$ 30 GeV (OR 30 GeV $< p_{\mathrm{T}} <$ 60 GeV)
- $\mid \eta \mid < 2.4$
- Passes the Medium (OR Loose/Tight) identification criteria excluding relative isolation [24]

3.3 Electron Images

For each electron in a given p_{T} range, electron images are constructed following. The images are square arrays in $\eta - \phi$ space in which pixel intensities are given by energy deposited in the corresponding region of the calorimeter. These images become the input to the neural network. The image has size $2R \times 2R$ where R=0.348. The image size is 40 x 40 in terms of the number of pixels. Hence each pixel corresponds to a $\Delta\eta = \Delta\phi = 0.0174$. The discretization of the image into $\Delta\eta \times \Delta\phi = 0.0174 \times 0.0174$ is to account for the discretization of calorimeter whose crystals subtends an area of $\Delta\eta \times \Delta\phi = 0.0175 \times 0.0175$ in the barrel [18]. Figure 1.5 shows the construction of an electron image.

3.3.1 Pre-processing

The following pre-processing steps were applied to electron images:

- **Center**: Center the electron image at the position of the reconstructed electron so that $(\eta, \phi) = (0, 0)$ in the image corresponds to the location of the reconstructed electron.
- **Crop**: Crop to a 40 x 40 pixel region centered at $(\eta, \phi) = (0, 0)$, which pick up the region $\eta, \phi \in (-R, R)$ for R=0.348.
- **Normalize**: Scale the pixel intensities such that $\Sigma_{ij} I_{ij} = 1$ in the image, where the sum is over the pixels.

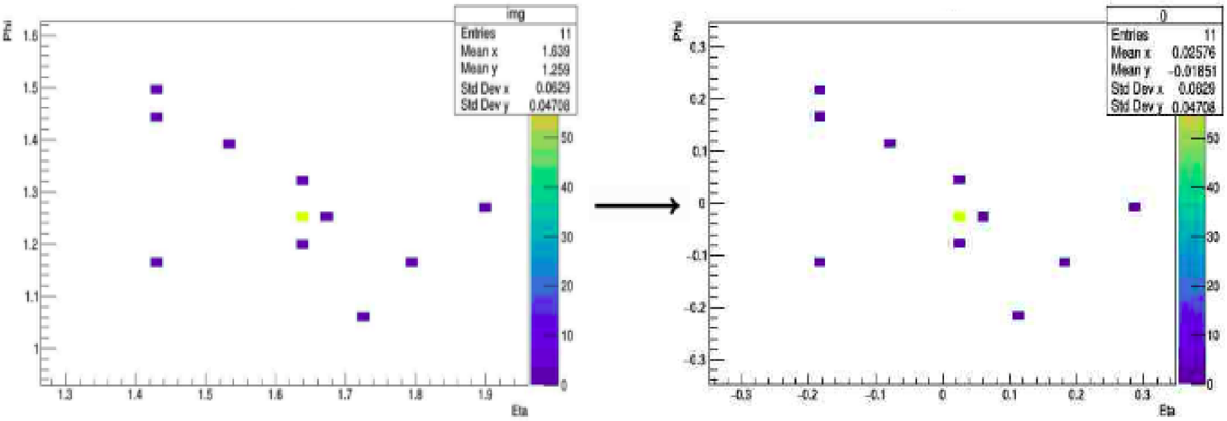

Figure 3.4: An electron image cropped to a 40 x 40 pixel region around the electron after centering at the position of reconstructed electron.

Figure 3.4 shows an example of a preprocessed image.

3.3.2 Real and fake electron images

Figure 3.5 and 3.6 shows images constructed for real and fake electrons respectively. For training of medium low p_{T} electrons, 11600 electron images of both categories were used. For tight and loose electrons with low p_{T}, 3500 electron images of both categories were used for training. A 1000 electron images were used for training the CNN for high p_{T} tight electrons.

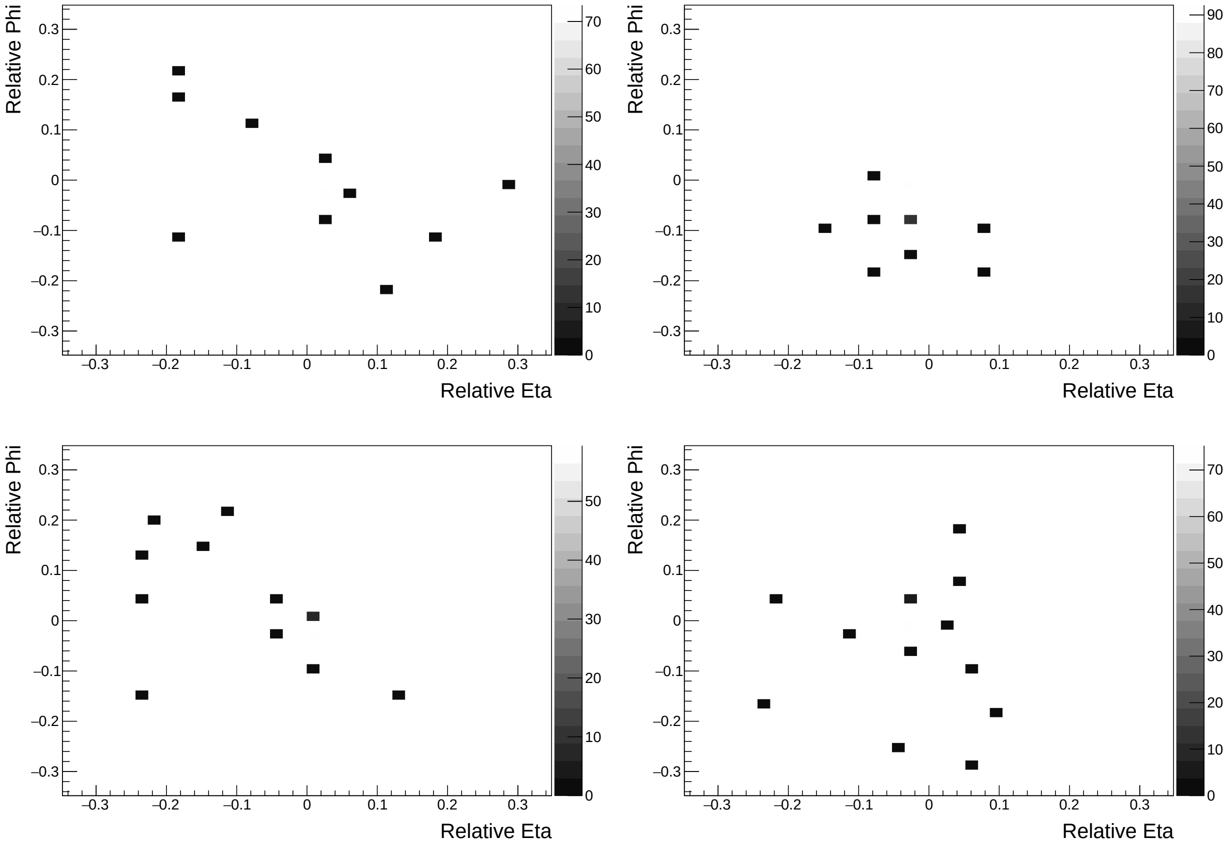

Figure 3.5: Example images of real electrons. These real electrons are selected from Z boson decay.

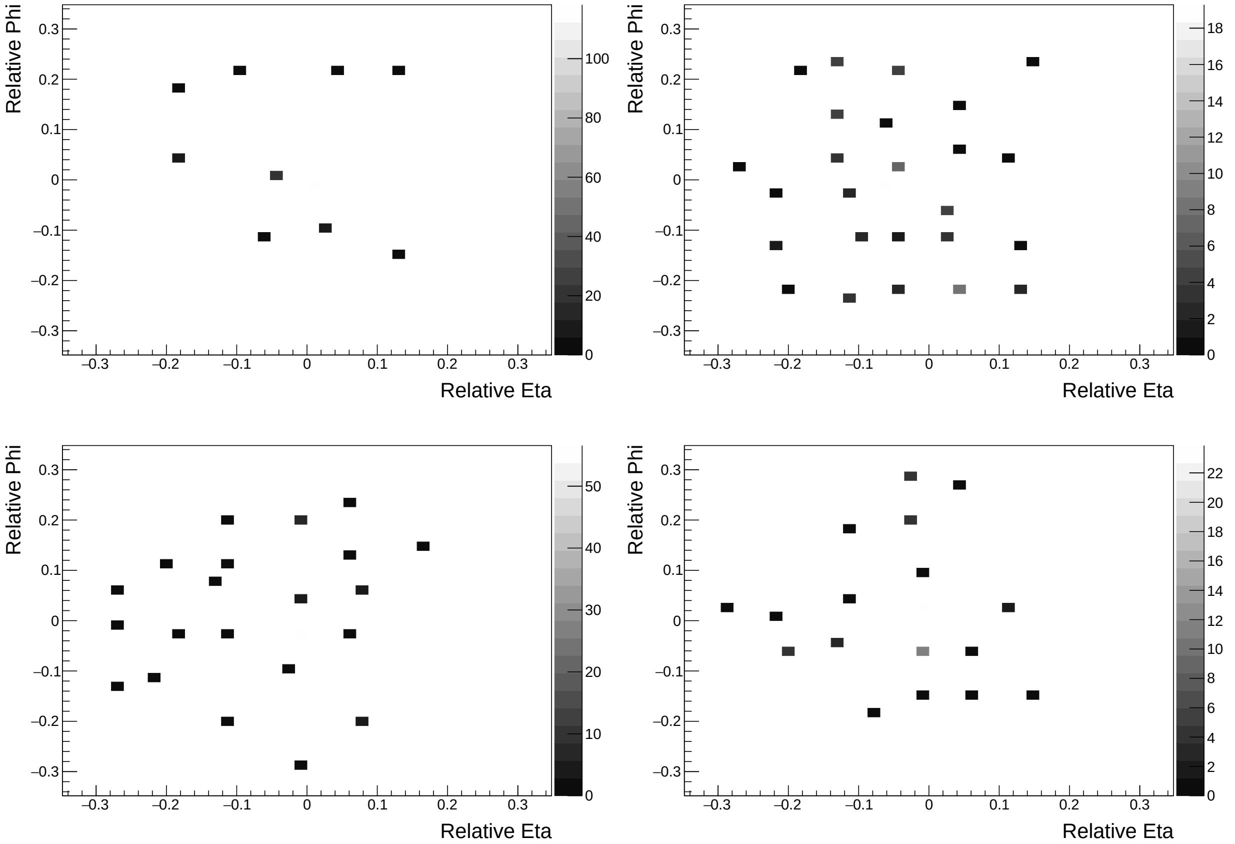

Figure 3.6: Example images of fake electrons. These fake electrons are selected from QCD events.

Chapter 4

Results

4.1 Low p_{T} electrons passing medium identification criteria

The CNN was trained on 11600 electron images of real and fake electrons. Real electrons were the leading electrons from a Z boson decay (row 1 in Table 3.1). Fake electrons were those from a QCD event (rows 3-10 in Table 3.1). These electrons were required to pass the following criteria:

- 10 GeV $< p_{\mathrm{T}} <$ 30 GeV
- $|\ \eta\ | < 2.4$
- Passes the Medium identification criteria excluding relative isolation [24]

After training,the model was tested on the training set of images as well as a new set of 11600 electron images of both real and fake electrons called the testing set. The output distribution is shown in Figure 4.1. The CNN was designed to give the probability of the input image being a signal/real electron image, as output. In an ideal case, we will have the peak of signal distribution at 1 and the background distribution at 0.

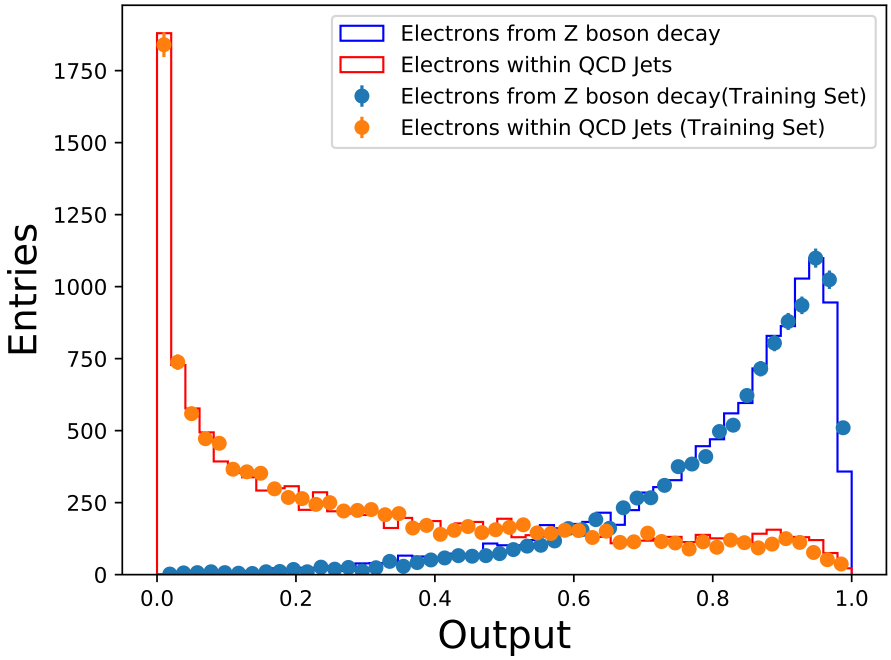

Figure 4.1: Output of Convolutional neural network. Signal - Electrons from Z decay, Background - Electrons from QCD Jets.

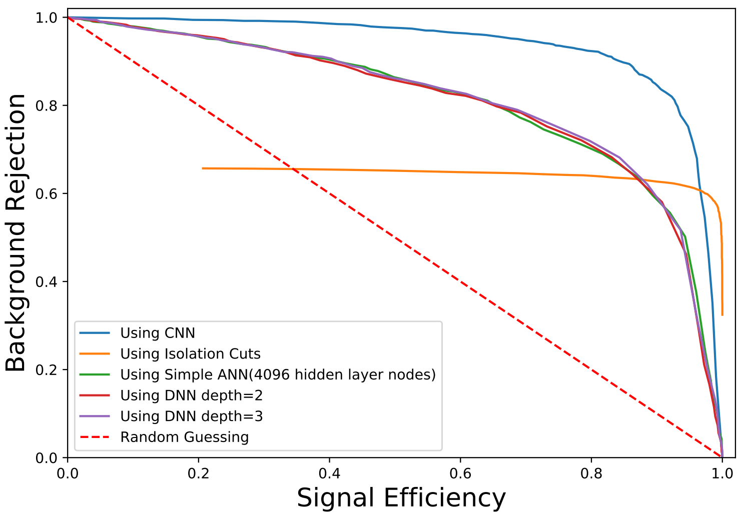

Figure 4.2: ROC curve for relative isolation, CNN , simple ANN and DNN.

4.1.1 Comparing results

The performance of both classifiers is evaluated by plotting background rejection versus signal efficiency, commonly known as the ROC curve. If we decide to divide the distribution at a particular value by making a selection, say output > 0.5 or $R^{ele}_{iso} < 0.3$, then the signal efficiency is the ratio of the signal which passes the selection, and background rejection is the ratio of background which fails the selection. An ideal classifier have at least one such selection criteria which will give both signal efficiency and background rejection as one. In other words, distributions of signal and background don't overlap in the case of an ideal classifier.

ROC curve for relative isolation, CNN, and other simpler algorithms like a simple neural network, and deep neural networks with two and three hidden layers, are shown in Figure 4.2. CNN is found to outperform relative isolation, a simple neural network and a deep neural network in classifying real electrons and fake electrons.

4.2 Changing the identification criteria

The medium identification cuts used to select electrons in the above case was changed to tight and loose identification cuts. All the electrons were now required to pass the following identification requirements.

- 10 GeV $< p_{\mathrm{T}} <$ 30 GeV
- $|\ \eta\ | < 2.4$
- Passes the Tight Identification cuts excluding isolation [24]

OR

- 10 GeV $< p_{\mathrm{T}} <$ 30 GeV
- $|\ \eta\ | < 2.4$
- Passes the Loose Identification cuts excluding isolation [24]

In both cases, CNN was trained and tested on 3500 electrons of both real and fake set. Performance of CNN is found to be independent of the identification criteria used while performance of relative isolation is found to vary with respect to the different identification criteria used as shown in Figure 4.3.

4.3 Changing the p_{T} range

The electrons were required to have a higher p_{T} of 30-60 GeV instead of 10-30 GeV. All electrons were now required to pass the following identification criteria:

- 30 GeV $< p_{\mathrm{T}} <$ 60 GeV

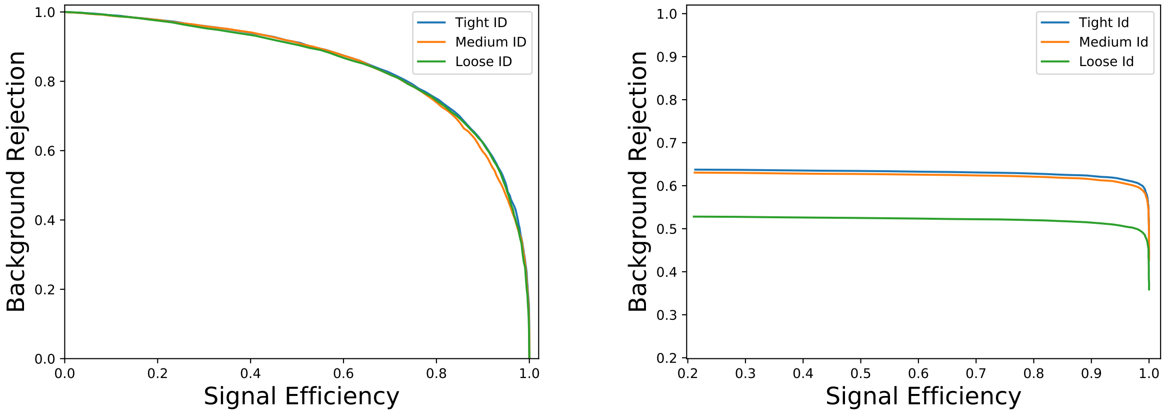

Figure 4.3: Performance of CNN(left) and relative isolation(right) for electrons passing different identification criteria.

- $\mid \eta \mid < 2.4$
- Passes the Tight Identification cuts excluding isolation [24]

CNN was trained and tested on 1000 electrons of both real and fake set. The result is shown in Figure 4.4. The performance of CNN was found to be independent of p_{T} range.

4.4 Fake electrons generated using Alpgen generator

Note that electrons from QCD events, which was selected as the fake electrons, were generated using two different generators, Alpgen and Herwig. This section uses fake electrons generated using only Alpgen (rows 3 - 6 in Table 3.1) and shows the performance of the CNN and relative isolation. The real electrons are again the leading electron from a Z boson decay. All the electrons were required to pass the following selections:

- 10 GeV $< p_{\mathrm{T}} <$ 30 GeV
- $\mid \eta \mid < 2.4$
- Passes the Tight identification criteria excluding relative isolation [24]

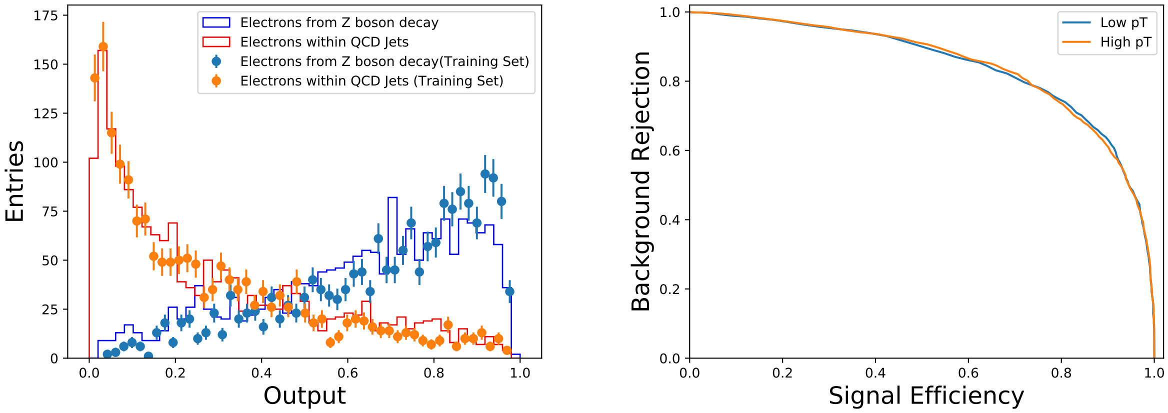

Figure 4.4: Left: Output distribution of CNN for real and fake electrons within a p_{T} range of 30 GeV $< p_{\mathrm{T}} <$ 60 GeV. Right: ROC curve for CNN classification of real and fake electrons within a p_{T} range of 30 GeV $< p_{\mathrm{T}} <$ 60 GeV (high p_{T}) compared with those in p_{T} range of 10 GeV $< p_{\mathrm{T}} <$ 30 GeV (low p_{T}).

The CNN was trained and tested on 3500 images each of both real and fake electrons. Figure 4.5 shows the output distribution of CNN and the ROC curve for CNN and relative isolation.

4.5 Fake electrons generated using Herwig generator

This section uses fake electrons generated using Herwig generator (rows 7-10 in Table 3.1). The real electrons are the leading electrons from a Z boson decay. All the electrons were required to pass the selection:

- 10 GeV $< p_{\mathrm{T}} <$ 30 GeV
- $\mid \eta \mid < 2.4$
- Passes the Tight identification criteria excluding relative isolation [24]

The CNN was trained and tested on 3500 images each of both real and fake electrons. Figure 4.6 shows the output distribution of CNN and the ROC curve for CNN and relative isolation. It was found that the network performed better when trained on electrons generated using Alpgen than using Herwig generator.

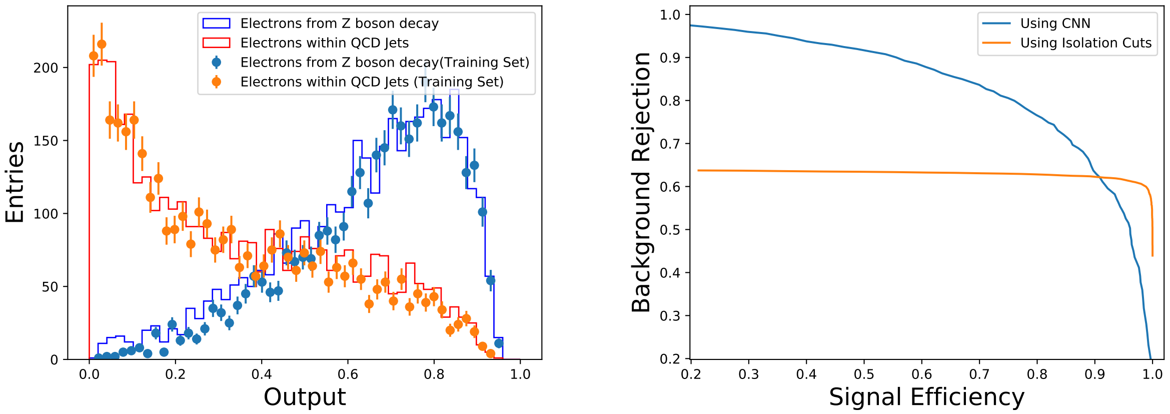

Figure 4.5: Left: Output distribution of CNN where fake electrons are generated by Alpgen. Signal: Real electrons from Z decay. Background: fake electrons from QCD events. Right: ROC curve for CNN and relative isolation when the fake electrons are generated by Alpgen.

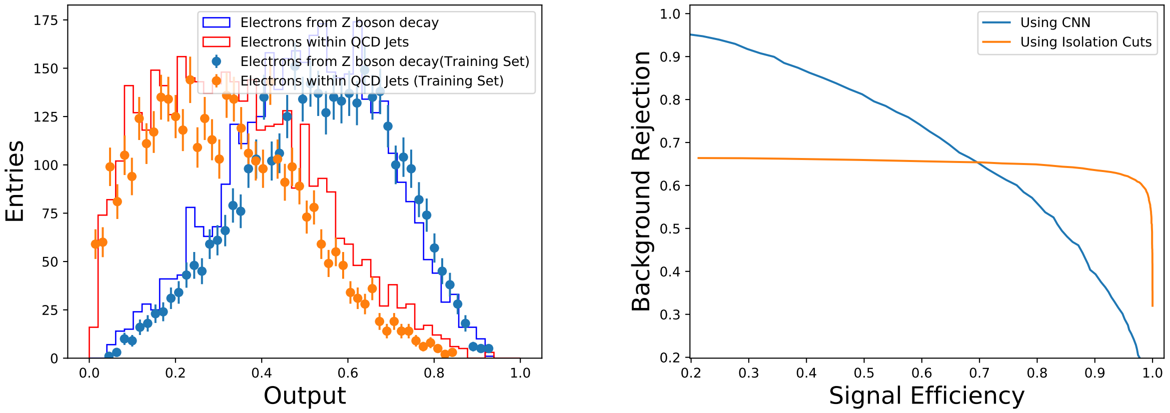

Figure 4.6: Left: Output distribution of CNN where fake electrons are generated by Herwig. Signal: Real electrons from Z decay. Background: fake electrons from QCD events. Right: ROC curve for CNN and relative isolation when the fake electrons are generated by Herwig generator.

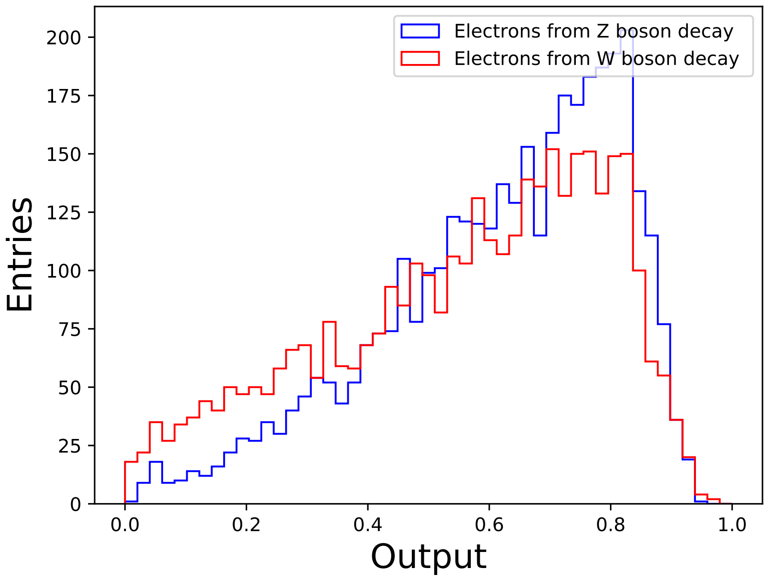

Figure 4.7: Testing on electrons arising from a W boson decay(red). Test on electrons from Z decay is given for reference (blue).

4.6 Testing the model

The model trained on electrons from Z decay and on those from QCD events, can be tested on electrons from different processes. For example, an electron arising from the decay of a W boson is expected to be real. It is interesting to check whether the CNN identifies the electron from W boson as a real electron even without training on it. Another example is that an electron within a light or heavy flavor jet from a $t\bar{t}$ decay is expected to be fake. Whether the CNN recognizes them to be fake is a question to be asked.

4.6.1 Electrons from a W boson decay

The trained model was tested on electrons from W boson decay. Row 12 in Table 3.1 was used to collect these electrons. Figure 4.7 shows the output of CNN when tested on electrons from a W boson decay. Test on electrons from Z decay is also shown for reference. The output distribution peaks at one, showing that the model recognized these to be real electrons.

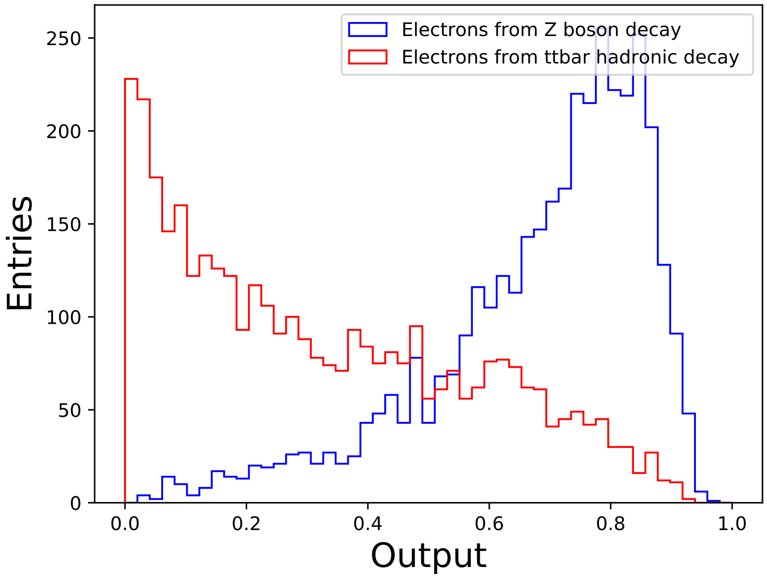

Figure 4.8: Testing on electrons arising from a ttbar hadronic decay(red). Test on electrons from Z decay is given for reference (blue).

4.6.2 Electrons within jets in a $t\bar{t}$ hadronic decay

The model was tested on electrons from a $t\bar{t}$ hadronic decay (row 11 in Table 3.1) in which the electrons are expected to be fake as they are arising from hadrons within jets. Figure 4.8 shows the output of CNN when tested on these electrons. Test on electrons from Z decay is also shown for reference. The output distribution peaks towards zero, showing that the model recognized these to be fake electrons.

4.6.3 Electron within heavy flavour jets

The model was tested on electrons within a jet from a bottom quark. Row 12 in Table 3.1 was used to collect these electrons. The b-jet was defined as the jet with a bottom meson which carries at least 20% of the jet p_{T}. Figure 4.9 shows the output of CNN when tested on these electrons. Test on electrons from Z decay is also shown for reference. As expected, the output distribution peaks towards zero showing that the model recognized these to be fake electrons.

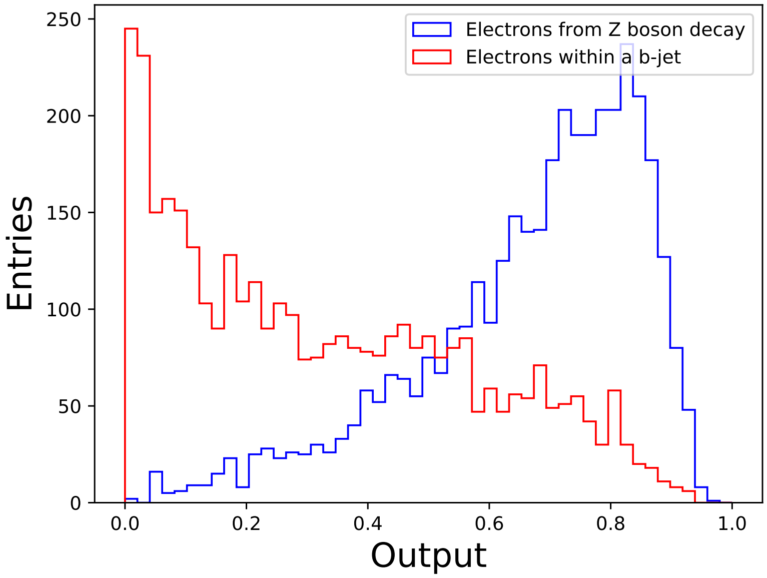

Figure 4.9: Testing on electrons within bottom quark jet. Test on electrons from Z decay is given for reference (blue).

Chapter 5

Conclusion and future direction

The ability to discriminate between real and fake electrons would be of immense application for discriminating signal and background in the field of high energy physics. For example, many signals of beyond SM physics, like the search for Seesaw mechanism, include real electrons, while their backgrounds are dominated by events with fake electrons. The task is challenging mainly because of the excess number of fake electrons which looks isolated. In this thesis, we applied a machine learning technique namely deep convolutional neural network, to the problem of real/fake electron discrimination. The input to the CNN was a 2D image, the two dimensions being pseudorapidity and azimuthal angle, with pixel intensities being energy deposited in a particular region of the detector. We found that the CNN outperformed physically motivated variable, relative isolation in classifying real and fake electrons in most region of the ROC curve.

The future directions this project can take are:

- **More input channels**: Adding more channels to the input image like energy deposits of charged hadrons, neutral hadrons e.t.c. These channels can become intensity of a color channel of the pixels, i.e., RGB values of the pixel. The input to the network will then be a colored image rather than a grayscale image.
- p_{T}: Explore different p_{T} ranges of electrons and analyze the performance of the classifier.
- η: Explore different η range of electrons. For example, it will be interesting to separate

electrons detected in the barrel and the endcap region of the detector because of the difference in material and geometry of crystals in these regions.

Bibliography

[1] G. Aad *et al.* [ATLAS Collaboration], JINST **9**, P09009 (2014) doi:10.1088/1748-0221/9/09/P09009 [arXiv:1406.7690 [hep-ex]].

[2] J. Gallicchio, J. Huth, M. Kagan, M. D. Schwartz, K. Black and B. Tweedie, JHEP **1104**, 069 (2011) doi:10.1007/JHEP04(2011)069 [arXiv:1010.3698 [hep-ph]].

[3] S. Chatrchyan *et al.* [CMS Collaboration], Phys. Rev. D **87**, no. 7, 072001 (2013) doi:10.1103/PhysRevD.87.072001 [arXiv:1301.0916 [hep-ex]].

[4] J. Cogan, M. Kagan, E. Strauss and A. Schwarztman, JHEP **1502**, 118 (2015) doi:10.1007/JHEP02(2015)118 [arXiv:1407.5675 [hep-ph]].

[5] L. G. Almeida, M. Backovi, M. Cliche, S. J. Lee and M. Perelstein, JHEP **1507**, 086 (2015) doi:10.1007/JHEP07(2015)086 [arXiv:1501.05968 [hep-ph]].

[6] P. T. Komiske, E. M. Metodiev and M. D. Schwartz, JHEP **1701**, 110 (2017) doi:10.1007/JHEP01(2017)110 [arXiv:1612.01551 [hep-ph]].

[7] M. Missiroli, CERN-THESIS-2017-014, CMS-TS-2017-005.

[8] S. Chatrchyan *et al.* [CMS Collaboration], Phys. Lett. B **716**, 30 (2012) doi:10.1016/j.physletb.2012.08.021 [arXiv:1207.7235 [hep-ex]].

[9] G. Aad *et al.* [ATLAS Collaboration], Phys. Lett. B **716**, 1 (2012) doi:10.1016/j.physletb.2012.08.020 [arXiv:1207.7214 [hep-ex]].

[10] [CMS Collaboration], CMS-PAS-EXO-18-005.

[11] A. M. Sirunyan *et al.* [CMS Collaboration], JHEP **1803**, 166 (2018) doi:10.1007/JHEP03(2018)166 [arXiv:1709.05406 [hep-ex]].

[12] A. M. Sirunyan *et al.* [CMS Collaboration], Phys. Rev. Lett. **119**, no. 22, 221802 (2017) doi:10.1103/PhysRevLett.119.221802 [arXiv:1708.07962 [hep-ex]].

[13] F. del Aguila, J. A. Aguilar-Saavedra and J. de Blas, Acta Phys. Polon. B **40**, 2901 (2009) [arXiv:0910.2720 [hep-ph]].

[14] Matt Strassler *Multi-Lepton Events: A Good Place to Look for New Physics* Retrieved from https://profmattstrassler.com/articles-and-posts/lhcposts/multi-lepton-events-a-good-place-to-look-for-new-physics

[15] S. Chatrchyan *et al.* [CMS Collaboration], JHEP **1106**, 077 (2011) doi:10.1007/JHEP06(2011)077 [arXiv:1104.3168 [hep-ex]].

[16] [ATLAS Collaboration], ATLAS-CONF-2013-070.

[17] A. M. Sirunyan *et al.* [CMS Collaboration], JINST **12**, no. 10, P10003 (2017) doi:10.1088/1748-0221/12/10/P10003 [arXiv:1706.04965 [physics.ins-det]].

[18] D. J. A. Cockerill [CMS ECAL Collaboration], arXiv:0810.0381 [physics.ins-det].

[19] Fei-Fei Li, Justin Johnson, Serena Yeung (2018), *Convolutional Neural Networks for Visual Recognition*[Lecture notes]. Retrieved from http://cs231n.github.io/

[20] Saugat Bhattarai, *What is Gradient Descent in machine learning?* Retrieved from https://saugatbhattarai.com.np/what-is-gradient-descent-in-machine-learning

[21] Cybenko, G. Math. Control Signal Systems (1989) 2: 303. https://doi.org/10.1007/BF02551274

[22] *Useful Diagrams of Top Signals and Backgrounds* Retrieved from https://www-d0.fnal.gov/Run2Physics/top/top_Public_web_pages/top_feynman_diagrams.html

[23] *WW/WZ&rarr e&nu jj at CDFII* Retrieved from https://www-cdf.fnal.gov/s̃fyrla/dibosons/

[24] CMS-EgammaCutBasedIdentification

[25] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. ”Deep learning.” nature 521.7553 (2015): 436.

[26] S. Chatrchyan *et al.* [CMS Collaboration], JHEP **1211**, 067 (2012) doi:10.1007/JHEP11(2012)067 [arXiv:1208.2671 [hep-ex]].

[27] DYToEE sample. Retrieved from http://opendata.cern.ch/record/7735.

[28] DYJetsToLL sample. Retrieved from http://opendata.cern.ch/record/7731.

[29] QCD sample. Retrieved from http://opendata.cern.ch/record/8874.

[30] QCD sample. Retrieved from http://opendata.cern.ch/record/8876.

[31] QCD sample. Retrieved from http://opendata.cern.ch/record/8873.

[32] QCD sample. Retrieved from http://opendata.cern.ch/record/8877.

[33] QCD sample. Retrieved from http://opendata.cern.ch/record/8891.

[34] QCD sample. Retrieved from http://opendata.cern.ch/record/8895.

[35] QCD sample. Retrieved from http://opendata.cern.ch/record/8885.

[36] QCD sample. Retrieved from http://opendata.cern.ch/record/8871.

[37] TTJets-HadronicDecays sample. Retrieved from http://opendata.cern.ch/record/9584.

[38] TTJets-FullLeptMGDecays sample. Retrieved from http://opendata.cern.ch/record/9579.

[39] WZJetsTo3LNu sample. Retrieved from http://opendata.cern.ch/record/9978.

[40] Nielsen, Michael A. Neural networks and deep learning. Vol. 25. USA: Determination press, 2015.

[41] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, U.S.A. (2016).

[42] Abadi, Martn, et al. "Tensorflow: A system for large-scale machine learning." 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). 2016.