QCD vs W jets recognition using CNNs and SVMs

Besana Emanuele Francesco

1 Introduction

Collimated streams of particles, called jets, are the signature of quarks and gluons being produced in head-on high-energy parton-parton collisions at hadron colliders such as the Large-Hadron-Collider (LHC), and they are a possible interesting source for the search of Beyond Standard Model (BSM) physics.
When gluons or quarks are generated they create a cascade of other particles that arrange into hadrons, which in turn decay into the particles that the detectors finally observe, where they are analyzed and reconstructed by algorithms that combine the information received by the detectors' components in order to cluster it into data and create jet images.

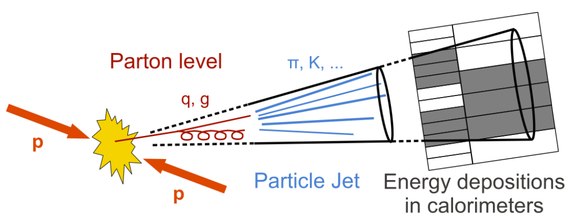

Figure 1: Cartoon of p-p collision resulting in a jet

Jet "tagging" is the job done by algorithms that seek to indentify the particle that initiated the jet from the variety of data obtained.
Traditionally this tagging endeavour was designed to distinguish between three classes of jets: light flavour quarks q=u,d,s,c, gluons g and bottom quarks b, however at the LHC, due to the reach of its high center-of-mass energy, new jet topologies emerged: heavy particles such as the Higgs, weak bosons and top-quarks were produced.
These heavy particles are usually imparted with a large Lorentz boost, and as a results, the particles generated from their decays are highly collimated and may be captured by a single jet; besides the heavy jets having a larger invariant mass, this property distinguishes them from Quantum Chromodynamic (QCD) multijet background (mostly gluon jets), which produces more diffuse radiation patterns. (Techincally speaking, QCD jets display an octet radiation pattern, compared to the singlet radiation patter of W bosons for instance.)
Classifying the origin of these jets and differentiating them from the QCD background is then a fundamental challenge for searches with jets at the LHC.

2 The Dataset

Out of all the data that is received by the detectors and analyzed by algorithms, this work is mainly concerned with the jets' images, which are 2D representations of the radiation pattern within a jet: the distribution of the locations and energies (deposited in the calorimeters of the detectors) of the jet's constituent particles.
Since a jet's formation is naturally over before it can be detected, it is sufficient to consider the said pattern in a 2D surface spanned by the angles $\phi \times \eta$. (ϕ is the real azimuthal angle while η, the pseudorapidity, is only approximately equal to θ).
Basically the image of a jet consists of a grid of pixels in the $\phi \times \eta$ space.
The goal of the present work is to distinguish (through images) between jets originated by a W boson, and the ones that are pure QCD background using Machine Learning (ML) methods for image recognition. To this end we use a dataset that is available online at `https://zenodo.org/record/269622#.YfnMEOrMKUn` which consists of 872666 25×25 grayscale images, of which 6000 were used: 3222 W jets (labeled as 1) and the remaining 2778 QCD multijets (labeled as 0).
The jet images are constructed and pre-processed using the setup described in Ref.[1]; both signal and background were generated with Pythia 8.219 at $\sqrt{s} = 14\ TeV$.
The following are the average over all images for the two classes of jets, the W boson and the QCD background:

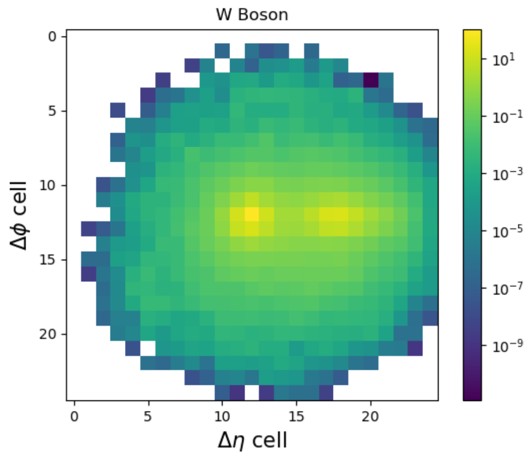

Figure 2: Average of the 3222 25×25 images of W boson jets. The temperature map represents the amount of energy collected in each cell of the calorimeter. Note that the values of $\Delta\eta$ and $\Delta\phi$ are not real and serve just as a representation.

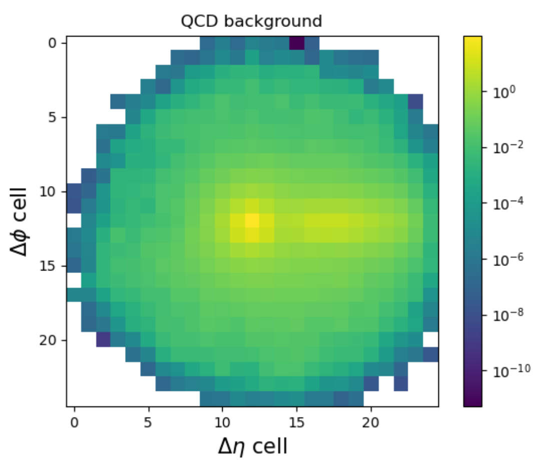

Figure 3: Average of the 2778 25×25 images of QCD jets. The temperature map represents the amount of energy collected in each cell of the calorimeter. Note that the values of $\Delta\eta$ and $\Delta\phi$ are not real and serve just as a representation.

In this work we compare the performances of two different Convolutional Neural Networks (CNNs) architectures and Support Vector Machines (SVMs) using different kernels.

3 CNNs

In the case of CNNs, the input is $(6000, 25, 25, 1)$ and the targets are either 1 or 0. In this work we considered two architectures, that are summarized in Table 1. The details concerning the training of the networks (batches, number of epochs, losses ...) can be found in the python programs. The networks are trained on 80% of the dataset and tested on the remaining 20%. Both CNNs were trained using a basic SGD optimizer with fixed learning rate.
In the first case (CNN1), we grid searched for the optimal (l2 norm) regularizer λ and the learning rate η and subsequently, using the best parameters, a search in the discrete range of $[5, 10]$ for the best number of feature maps (filters) to use was made:

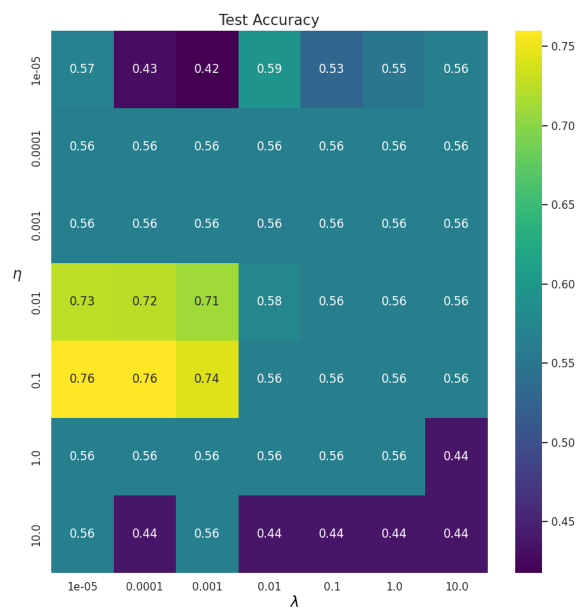

Figure 3: Grid search for λ and η. Using these a search for the best number of filters was made.

For the second architecture (CNN2), a grid search was made for both the number of filters to use in the first and second Conv2D layers in the discrete range $[30, 40]$ and the parameters η and λ as before. This time these latter parameters were optimized before the filters:

CNN	**Specifications**
CNN 1	Convolutional2D layer, 2×2 receptive field, activation = ReLU, regularizer = regularizers.l2
	MaxPooling2D layer, pool size = 2×2
	Dense layer of 70 neurons, activation = ReLU
	Dense layer of 2 neurons, activation = softmax
CNN 2	Convolutional2D layer, 3×3 receptive field, activation = ReLU, regularizer = regularizers.l2
	MaxPooling2D layer, pool size = 2×2
	Convolutional2D layer, 3×3 receptive field, activation = ReLU, regularizer = regularizers.l2
	MaxPooling2D layer, pool size = 2×2
	Convolutional2D layer, 3×3 receptive field, activation = ReLU, regularizer = regularizers.l2, 35 filters
	Dense layer of 70 neurons, activation = ReLU
	Dense layer of 2 neurons, 3×3 receptive field, activation = softmax

Table 1: The two CNN architectures used in this work. Strides are equal to 1.

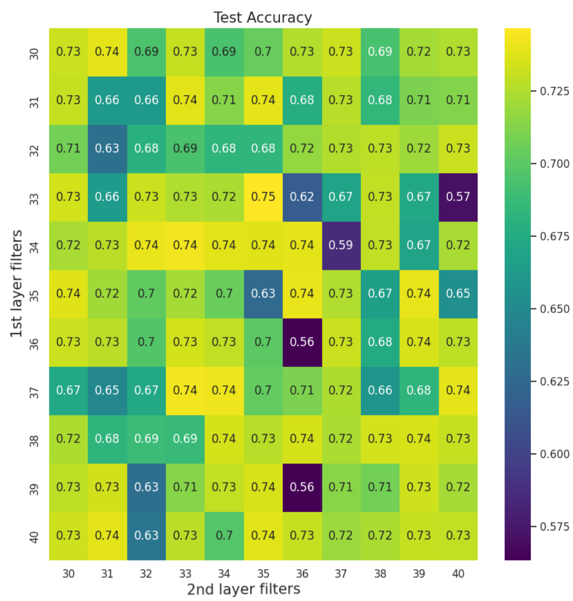

Figure 4: Grid search for the number of filters to use in the first and second convolutional layer. This was done after having searched for the best λ and η.

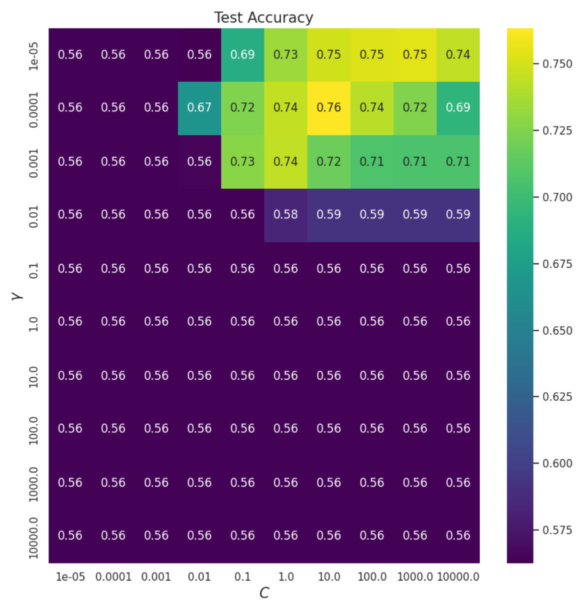

Figure 5: Grid search for the best γ and C for the rbf kernel.

4 SVMs

For the SVMs the input is $(6000, 625)$ and three kernels were used: the radial basis function (rbf), the polynomial and the linear kernels. For the rbf, we grid searched for the γ parameter and the regularization one C. The grid search produced:

For the polynomial kernel we grid searched for the best degree in the discrete range $[3, 6]$ and for C, whereas for the linear kernel the search was made just for the best C.

5 Outlook

All in all, the following table summarizes the results for the various methods employed:

Classifier	Parameters	Test accuracy score
CNN 1	Filters = 10 $\eta = 0.1$ $\lambda = 0.0001$	0.77
CNN 2	Filters in first conv. layer = 35 Filters in second conv. layer = 33 $\eta = 0.1$ $\lambda = 0.0001$	0.75
SVM (rbf)	$\gamma = 0.0001$ $C = 10$	0.76
SVM (poly)	polynomial degree = 3 $C = 100$	0.75
SVM (linear)	$C = 10$	0.73

Table 2: Results for the parameters utilized in every method and the corresponding accuracy reached.

Better performances than the best found in this work are achieved through more sophisticated means, such as in Ref. [2] where the identification algorithm was based on interaction networks (JEDI-nets).

References

[1] `https://arxiv.org/abs/1501.05968`
[2] `https://arxiv.org/abs/1908.05318`