# Machine Learning methods for NEWSdm

Artem Golovatiuk

March 2018

## 1 Introduction

One of the large unsolved problems of modern physics is Dark Matter. There are numerous observations both on astrophysical and cosmological length scales pointing to the existing of Dark Matter, but still no evidence in ground experiments. Three ways of looking for DM signatures on the experiment are: direct detection (detection of the DM-nuclei scattering), collider searches (creating of DM particles in the Standard Model collisions) and indirect detection (detection of the DM annihilation signals). The region in parameter space that can be tested by current direct detection experiments corresponds to the class of DM models called WIMPs (Weakly Interacting Massive Particles).

Here we are going to talk about one of the direct detection experiments called NEWSdm (Nuclear Emulsions for WIMP Search directional measure [1]). The main peculiarity of this experiment is an ability to measure the direction of the signal. This gives a possibility to extend Dark Matter searches beyond the neutrino background and to prove the Galactic origin of the DM, since the flow of DM particles (so called 'WIMP wind') is expected to be directed from the Cygnus constellation and the background is expected to be isotropic.

The distinct features of the NEWS experiment among directional experiments are high sensitivity and possibility to scale the target mass due to usage of nuclear emulsions both as solid target and detector instead of gas chambers.

### 1.1 Current classification approach

The experiment is based on the idea to detect tracks of scattered nuclei after WIMP collisions. Nucleus leaves a trace of excited Ag ions, which can be detected after chemical development, thus the data we get from the detector is track images. The expected range of the signal tracks is $O(100nm)$.

No matter how good we shield and clean the detector, there remains some irreducible background contamination. Therefore a classification technology is needed in order to distinguish signal from background.

There were two possible approaches for the emulsion scanning: using an X-ray or optical microscope [1]. The first approach gives good resolution ($\sim$ 60 nm), but is much slower and much more complicated than the optical one. Although, optical microscope is much faster and cheaper, it lacks resolution ($\sim$ 200 nm). An example of tracks scanned with optical and X-ray microscopes can be seen on the Fig. 1.

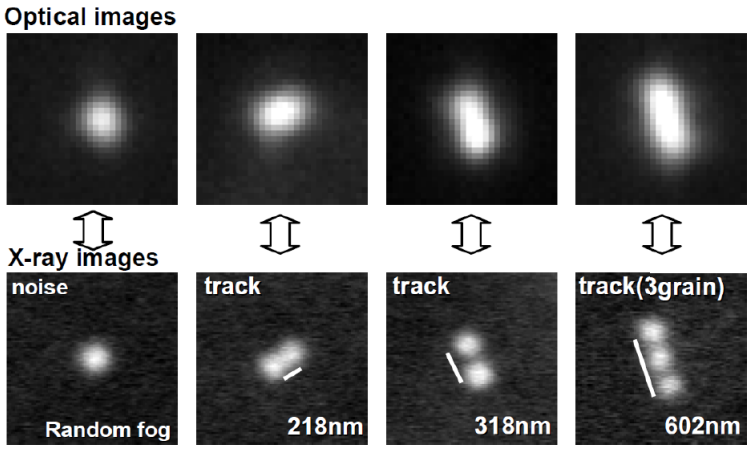



Figure 1: Comparison between reconstructed tracks of a few hundred nanometers length with the optical microscope and with the X-ray microscope.[1]

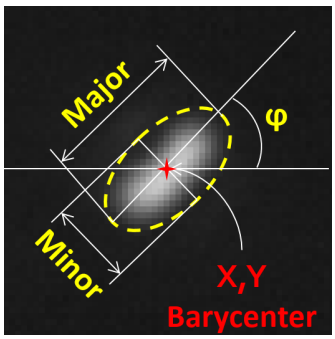



Figure 2: Elliptical fit.

This leads us to the problem of distinguishing the tracks, that are below the optical resolution. Since we have several silver grains merged into one cluster on the picture, one can use a 2D-Gaussian fit to approximate the cluster with an ellipse. Fig. 2 shows the main parameters of the fit. Additionally, we can use parameters like ellipticity (major axis divided by the minor), area of the cluster and brightness to perform physical analysis.

From the WIMP mass range one can obtain the expected range of recoil energies and thus the track lengths. This gives us order of several hundred of nanometers and allows to unambiguously identify long tracks (large major axis) as background (can be caused by dust or some high-energetic radiation). However, we don't have such possibility on the lower bound to separate random fog (temperature fluctuations) from small track, as it can be seen on Fig. 1. Further analysis of the cluster is needed.

A very promising approach is the usage of polarised light. The main idea is following. In the real experiment grains inside the cluster are non-spherical and aligned in the random direction hence they will give resonant peaks of brightness on different polarisation angles for a fixed wavelength. This will result in the shift of the cluster's barycenter and can be detected on the images from optical microscope. Such observation could allow us to distinguish between single grain fog and tracks containing at least two grains. Fig. 3 shows the behaviour of track barycenter's $x$ and $y$ coordinates.

## 1.2 Machine Learning approach

The physical approach in samples classification has two main weaknesses:

- The Gaussian fit is an approximation and does not truly describe the form of the cluster, thus you lose information.
- Overall it is limited by our physical understanding of the system and the models we use to describe it. We first need to find some physical phenomenon (like resonant light scattering)

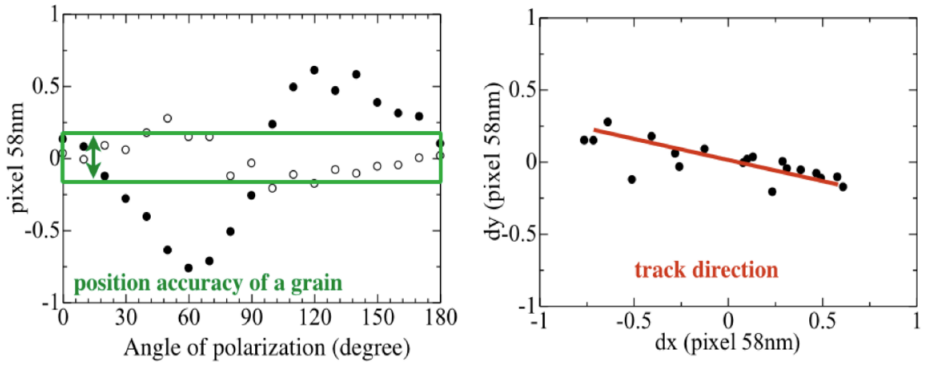



Figure 3: Application of resonant light scattering to an elliptical cluster. Left plot: $dx$ and $dy$ are the displacements of the cluster barycenter for a given polarization in pixel units (1 pixel = 55 nm). Right plot: track slope fit and its length of about 90 nm.[1]

and only then we can create a classification criterion for images (like barycenter shift).

The way to improve this situation is Machine Learning.

- It allows working directly with the cluster images without any additional approximations, preserving all the information in pixels.
- It can discover complex correlations in signal and background, which do not necessarily need a physical model explaining them, even when the difference is minimal.
- ML algorithms can be robust to the small background contaminations in the signal samples.
- A broad variety of algorithms and approaches in Machine Learning gives a possibility to improve the signal purity to a very high level.

# 2 The experimental data

To prepare any classification algorithm for a real experiment we need some data to train it. Usual solution for this case is Monte Carlo simulations. However, more physically motivated solution is using real background and experimentally simulated signal. NEWSdm uses Carbon ions with fixed kinetic energies to irradiate emulsions to simulate an elastic scattering of WIMP on a nucleus [1].

Since we use real emulsions irradiated by Carbon ions as a signal samples, some background contamination is inevitable. Short time of exposure as well as usual shielding techniques are used to reduce the contribution of background. As we mentioned before, Machine Learning algorithms allow neglecting this contamination without much harm to the algorithm's performance.

## 2.1 Current training data

At this point we use 2 emulsion samples to train our ML algorithms:

- **Signal:** sample, irradiated by C 100 keV ions. Contains $\sim$ 15000 images with 8 polarisation angles each after applying physical cuts (discarding too long tracks, large dust samples etc.).
- **Noise:** sample, exposed at the LNGS in 2017. Assumed to have only mixed background tracks. Contains $\sim$ 7000 images with 8 polarisations each after applying cuts.

There are two possible types of algorithms depending on what they take as an input. One takes some handcrafted features, another takes images themselves. We tried them both.

As a **physical features** we used Gaussian fit parameters and some physical parameters of the cluster image for each polarisation:

- $x, y$ – cluster center coordinates
- $l_x, l_y$ – major and minor axes of an elliptical fit
- $\varepsilon$ – ellipticity $(l_x/l_y)$
- $\varphi$ – direction of the cluster
- $n_{px}$ – area of the ellipse in pixels
- $vol$ – brightness volume of the cluster (sum of the pixel brightness inside the cluster)

So it is 64 features in total.

Fig. 4 shows an example of **raw images** used for training the algorithm.

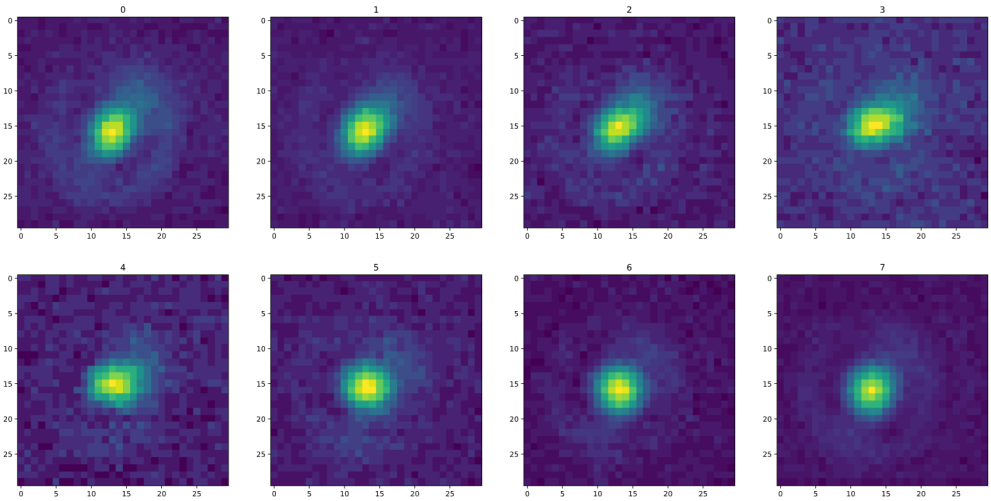



Figure 4: Example of 8 polarisations for a track of 100keV Carbon ion.

## 2.2 Yandex dataset

For further research we are planning to use broader dataset, consisting of the following classes:

- **Signal:** C 60keV, C 80keV, C 100keV
- **Noise:** Gamma, Dust, Dust-fog

*Signal* contains samples exposed horizontally (in the plane of emulsion) to the Carbon ions with certain energies. Noise contains 3 types of background. *Gamma* sample was irradiated by gamma rays and therefore contains lots of tracks from electron-nuclear recoils (electrons coming from photon scatterings). *Dust* sample was not exposed to anything, also there was no chemical development, only fixation, it contains mostly dust and impurities, that got into emulsion during its production. *Dust-fog* sample was not exposed as well, contains fog grains from the temperature fluctuations additionally to dust.

Different classes of signal and background would allow us to perform more thorough study of our algorithms. Moreover, this dataset contains much more track samples in each class, than what we have now ($\sim 10^5$ instead of $\sim 10^4$).

These samples has already been scanned before, but they need to be re-scanned because scanning technique has changed significantly by now.

## 2.3 Directionality

During the exposure to Carbon ions the emulsion sample is fixed, therefore all signal tracks are aligned in the same direction (with some variance). This makes cluster direction an easy classification parameter as long as most of background is isotropic.

The goal of Machine Learning in this project is to find some non-trivial dependencies, which could not be found by physics intuition. That is why we try to study the performance of our algorithms for more isotropic signal case.

For the algorithms using the Gaussian fit parameters we simply drop the cluster direction $\varphi$ from the set of training features. For another type of algorithms working with the images directly we try artificially rotating the images. On one hand, this enlarges our dataset, while on the other hand, is allows us to train the algorithm to identify the direction of the track.

Rotating the images after scanning may not reproduce some physical peculiarities of the microscope and the scanning process, so it is one of the further plans to physically rotate the samples while scanning to make data more isotropic.

# 3 Performance metrics

In order to find the best algorithm for our purposes we need some way to compare their performances. There exist various metrics which allow us to quantify the quality of the algorithm.

The simplest metric is *Accuracy*: it divides a number of right answers over the whole number of test samples.

More physically motivated metric for our study is *Precision*: it shows how good is our algorithm in distinguishing some specified class (DM signal in our case) from another.

If we call one class as *positive*, than we can express the above metrics as following:

$$Accuracy = \frac{\sum True\ positive + \sum True\ negative}{\sum all\ samples} \tag{1}$$

$$Precision = \frac{\sum True\ positive}{\sum True\ positive + \sum False\ positive} \tag{2}$$

From (2) follow that the more background we consider as signal, the lower is the *Precision*. In physics it is usually called *Signal purity*.

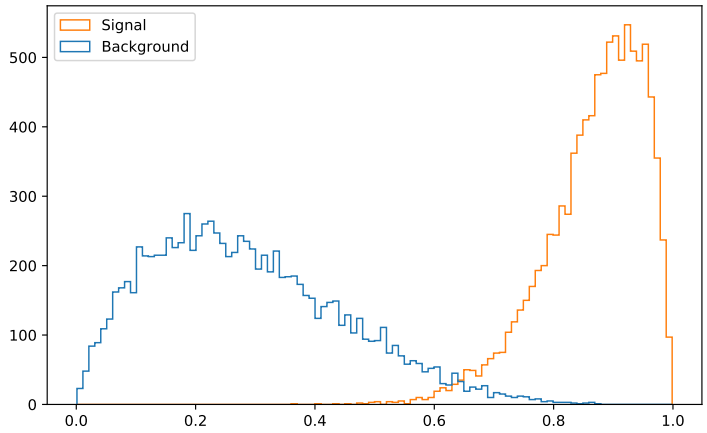



Figure 5: Algorithm's probability output showing optimal threshold different from 0.5

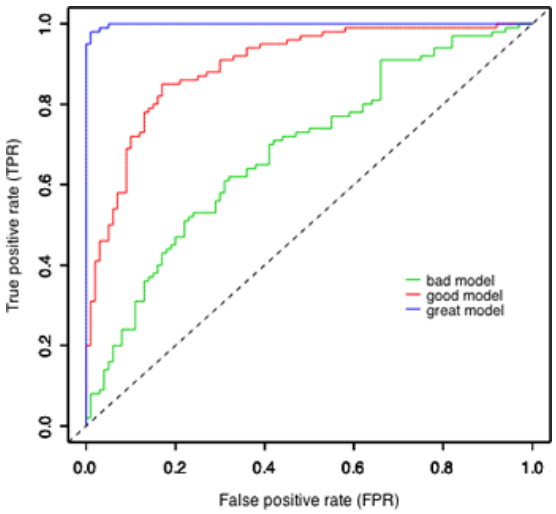



Figure 6: ROC curves.

The problem of the mentioned metrics is that they depend on the probability threshold. Output of the algorithm is the probability for the sample to be signal. The basic threshold to consider something signal is 0.5. Fig. 5 shows possible situation when the optimal dividing threshold is between 0.6 and 0.7. This means that threshold can be different for different algorithms.

A way to solve this problem is ROC curve that illustrates the distinguishing ability of a binary classifier as its discrimination threshold is varied. Fig. 6 shows an example of such curve. Where:

$$TPR = \frac{\sum True\ positive}{\sum all\ positive\ samples}$$

$$FPR = \frac{\sum False\ positive}{\sum all\ negative\ samples}$$

An area under such curve (AUC) can be used as a performance metrics. This can be intuitively understood as a chance that classifier gives a signal sample bigger probability value than a background sample. An ideal classifier would have 100% ROC AUC.

# 4 Tested approaches

As we mentioned before, we tested two types of algorithms: one using some image features as an input and another one using directly images. We can also divide them as decision trees and neural networks approaches.

## 4.1 Decision trees

Before trying heavy weapons like Neural Networks we tried more basic algorithms. The great advantage of them is computational speed. Since number of parameters is usually much smaller than in Neural Networks, it takes much less computational power and time to train the algorithm.

One of the most powerful non-network algorithms are Decision trees ensembles. The concept of the decision tree is pretty simple. On each step it takes some condition on the input features, checks whether the sample fulfils it and hence makes a decision about the sample class (more signal like or more background like). Then it repeats the same procedure with different conditions until it separates the samples good enough. The deeper the tree - the better it separates the training samples, but the more it tends to overfit the training samples and thus worsen the performance on test samples.

The usage of ensembles of trees classifiers is usually a solution which allows increasing the performance on the training set without much loss in the performance on the test set (avoiding overfitting).

**Boosted Decision Trees (BDT):** Is an ensemble of shallow (small number of separation conditions) decision trees, each of them is improving the answer of the previous one. Since it is being built successively, it has very limited possibility to be parallelized.

**Random Forest (RF):** Is an ensemble of very deep decision trees (up to maximal depth, when the final conditions separates samples one by one), each of them is training on the random subsample of training data and random subsample of input features. The answer is an average of the trees answers. Since every tree is being trained independently, it is highly parallelizible on multiple CPUs.

The performance of these algorithms grows with the number of trees in the ensemble, but at some point BDT starts overfitting the training data, while RF does not due to averaging the randomized trees, so it comes to horizontal asymptote. Due to parallelization RF can be much faster during the training, than BDT.

The performance strongly depends on the input features choice, which limits the improvement possibilities.

Fig. 7 shows the ROC AUC dependence on the number of estimators (trees) of 50 runs with 75% random samples (different on each training) as a training set. Lines are average performance and error bars are roots from variance of 50 runs.

The plot confirms the behaviour of BDT and RF mentioned above.

The resulting performance is quite low comparing to our goal, but the largest training time for BDT ($10^4$ trees) is about 15 min and for RF ($10^4$ trees) is about 2 min (due to parallelization), which is both very fast.

Changing the list of input features would possibly improve the performance, if we had other physically motivated features.

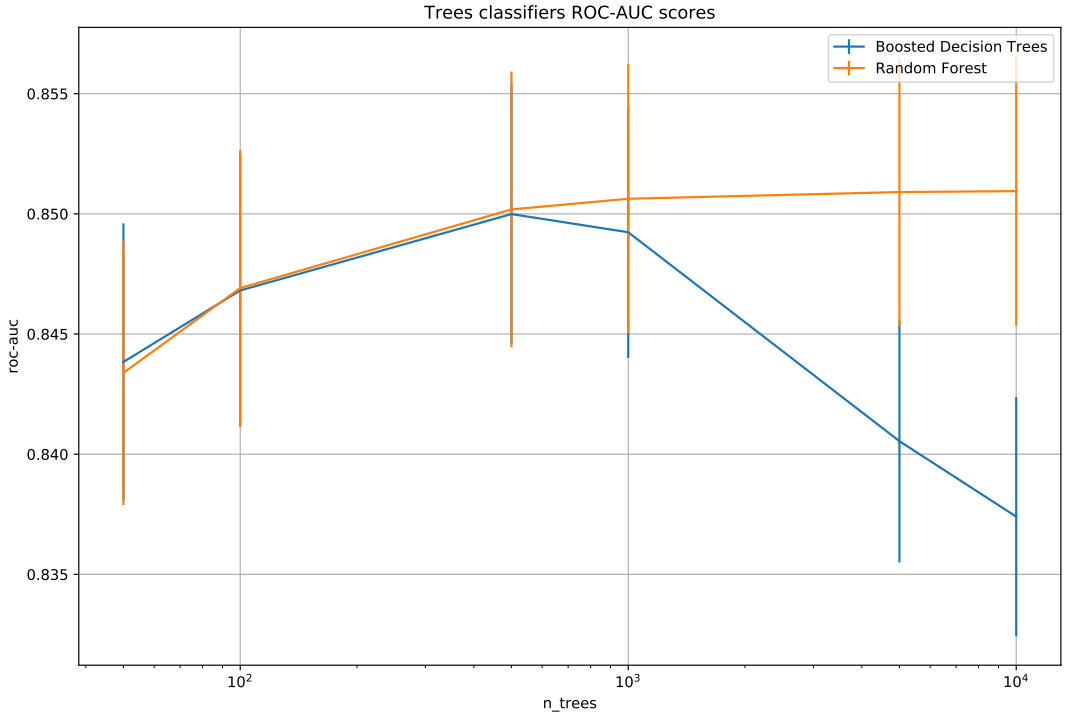



Figure 7: BDT and RF performance for different numbers of estimators, averaged after 50 runs.

## 4.2 Convolutional Neural Networks

Our experimental data is represented by images, and the best class of Neural Networks for working with images is commonly known to be Convolutional Neural Networks. The key part of such networks are Convolution layers, convolving the image (matrix) with the filter matrices to produce an input for the next layer. Another feature of ConvNets is Pooling layer, which combine the outputs of neuron clusters at one layer into a single neuron in the next layer, usually by taking maximum value or average of the cluster.

Since we have multiple images for each cluster from different polarisation angles, we tried two ways of treating the cluster images:

- **2D images:** images for different polarisations are treated as colour channels, like colours for ordinary image, therefore we used 2D ConvNets architecture to process the images.
- **3D images:** images for different polarisations are stacked together creating a 3D picture, where polarisation is changing along the new axis. We use 3D ConvNets architecture for this case with 1 colour channel in the input images.

### 4.2.1 2D ConvNets

We should start this section from the discussion of the previous results of Sergey Shirobokov from Yandex. He was working on the same problem, but with the different dataset, mentioned in sec. 2.2.

He used so called VGG architecture [2] with 4 portions of convolution layers with $3 \times 3$ filters in each and with the number of filters doubled in every next layer, starting from 64 filters. Fig. 8 shows the accuracy of Sergey's network on the validation set during training.

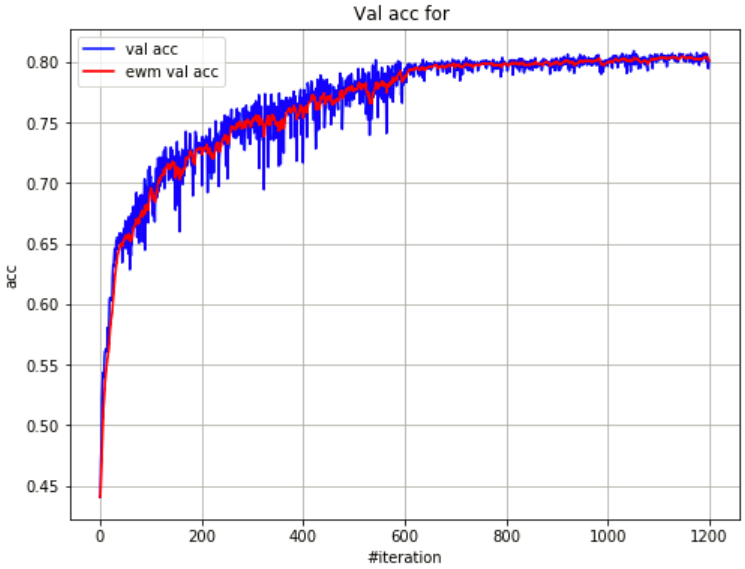
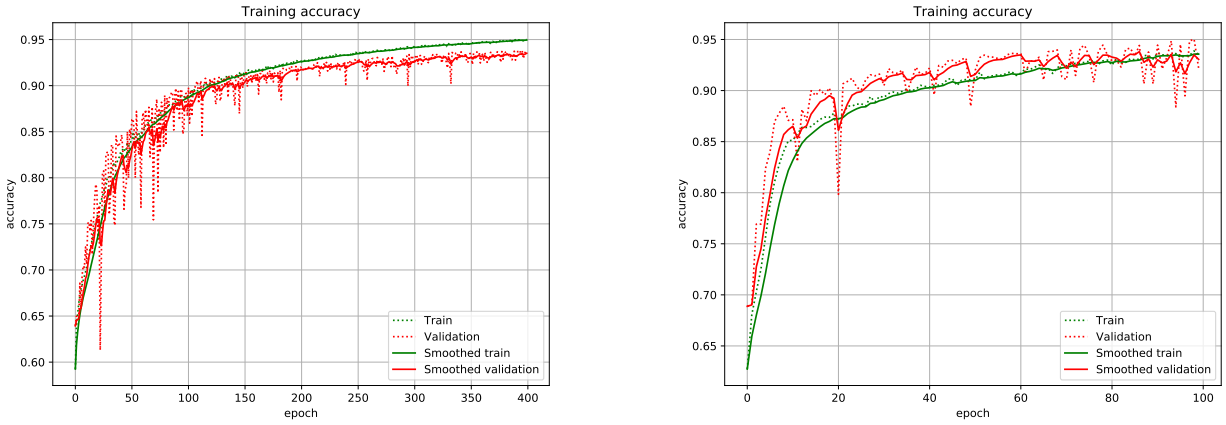



Figure 8: Validation accuracy of Shirobokov's CNN during training.

In our study we compared the performance of the most simple CNN (only 1 convolution layer) with the CNN of the same architecture as Shirobokov used, but working with the dataset we have right now. Figure 9 present results of the Simple CNN and VGG-like CNN. Epoch is one iteration of gradient descent over the whole training set.




(a) Simple CNN

(b) VGG-like CNN

Figure 9: Accuracy on the training and validation set for 2D ConvNets.

The resulting performance of 2 approaches is quite close and in both cases much better than the Shirobokov's result, which should be due to the different datasets. However, possible reason of the

improvement is the purification of current data using physical cuts, that was mentioned in sec. 2.1. VGG-like CNN's performance on the validation set on fig. 9b is a bit unstable probably due to high learning rate of the gradient descent and because of usage of the learning rate decay it should stabilise in the later epochs around the same value. The result on fig. 9a can be used as a benchmark for this dataset, since it is almost the simplest architecture possible.

The resulting test accuracy is $\sim 93\%$ for both Simple and VGG-like model.

If we compare the time needed for 1 training epoch, Shirobokov's CNN took $\sim 100$ sec, VGG-like CNN took $\sim 60$ sec (which could be caused by smaller dataset) and Simple CNN took $\sim 5$ sec.

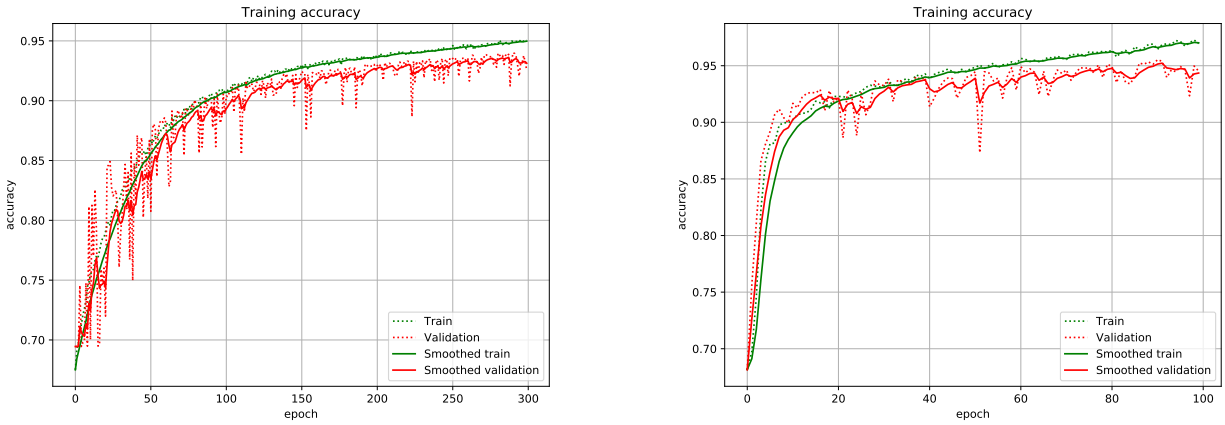
### 4.2.2 3D ConvNets

For 3D Convolutional Neural Networks we tested the same architectures as for 2D, but using 3D layers.

Using 3D convolutions could discover more interactions between different polarisation, hence be more physically motivated. Moreover, such networks have many more trainable parameters and therefore more possibilities to discover some complex correlations.

Figure 10 present the results of training of the Simple 3D and 3D VGG-like models. The resulting accuracy of the Simple 3D model is almost the same as for 2D, while for VGG-like 3D model accuracy increased to $\sim 95\%$.

The training time for 1 epoch of the Simple 3D model is $\sim 10$ sec, while for VGG-like 3D is $\sim 150$ sec, so it doubled comparing to 2D models.



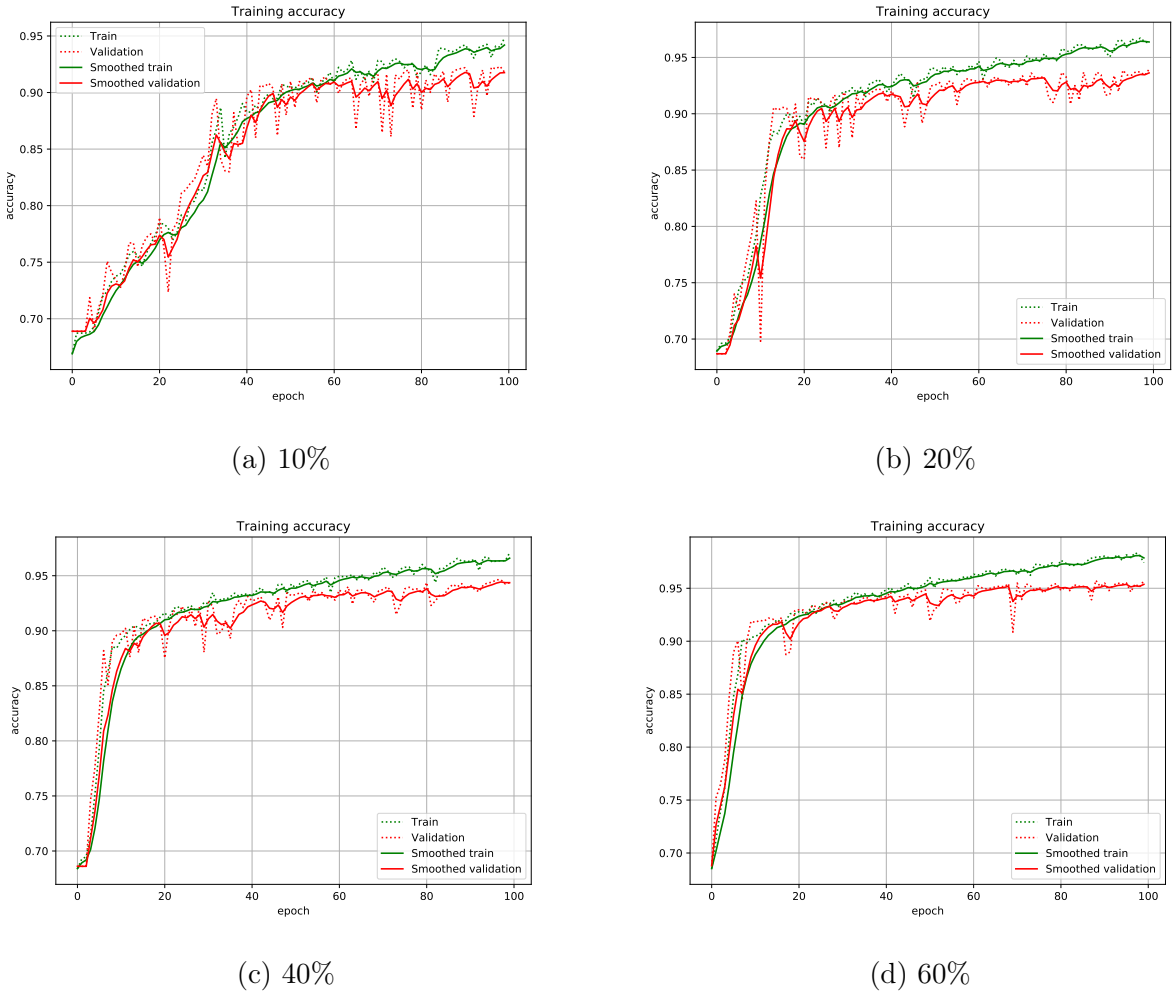

(a) Simple CNN

(b) VGG-like CNN

Figure 10: Accuracy on the training and validation set for 3D ConvNets.

- **Varying the number of training samples**

In order to better understand the performance of our algorithm, we checked its behaviour with decreasing the number of training samples. We used the VGG-like 3D CNN architecture during this part of study.




(a) 10%

(b) 20%

(c) 40%

(d) 60%

Figure 11: Accuracy of the VGG-like 3D CNN for different portion of training samples.

Figure 11 shows that even for only 10% of samples in the training set ($\sim$ 2000 samples) CNN reaches $\sim$ 92% validation accuracy, increasing to $\sim$ 95% at training set of 60% of samples.

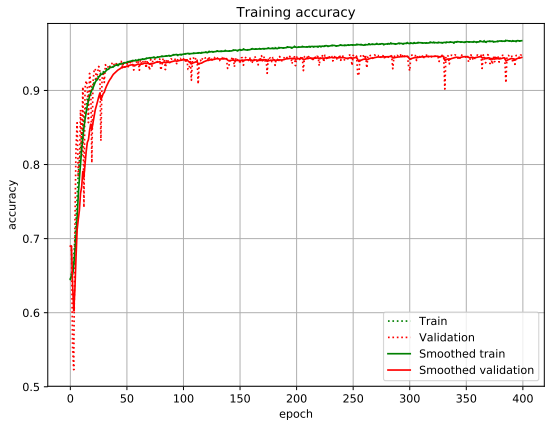
This means that further increasing the number of samples that we mentioned in the sec. 2.2 can potentially improve the performance.

### 4.2.3 Image rotation for isotropy assumption

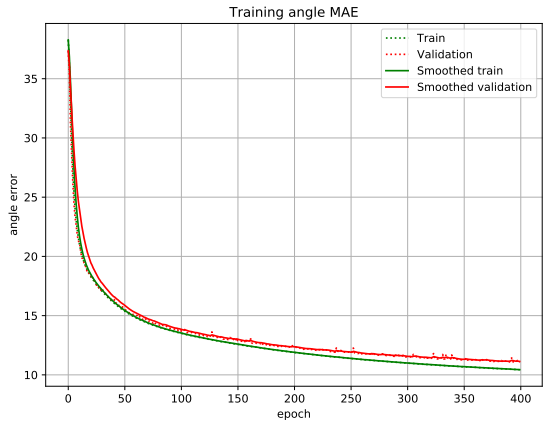
To perform study on pseudo-isotropic images two main transformations were made:

- **From the data point of view:** we rotated every image for 6 fixed random angles in range from $-90°$ to $90°$, since we do not distinguish whether the track is forward or backward, saving the rotation angle as a separate feature.
- **From the CNN point of view:** we added a new output layer calculating the rotation angle and modified the loss function in such a way that angle error would have a bit less weight than a classification error.

We compared the performance of the Simple 2D model and VGG-like 3D model. The results for classification accuracy and mean absolute angle error are presented on figures 12 and 13.

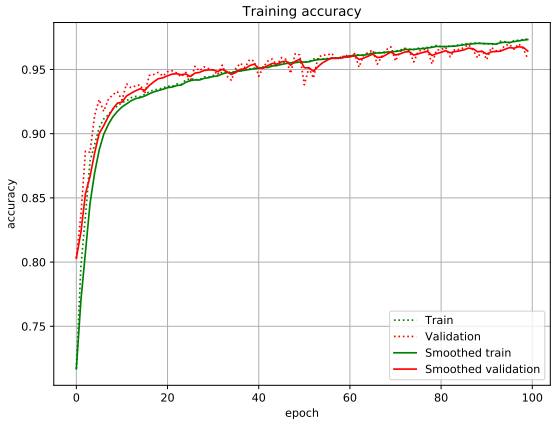



(a) Classification accuracy

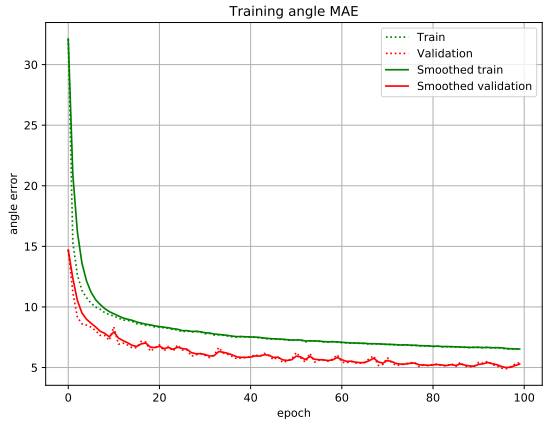



(b) Mean absolute angle error

Figure 12: Performance of the Simple 2D CNN for 6 random rotation angles.




(a) Classification accuracy




(b) Mean absolute angle error

Figure 13: Performance of the VGG-like 3D CNN for 6 random rotation angles.

The fig. 12a shows final validation accuracy abound $\sim 95\%$ and fig. 12b shows the angle error around $\sim 11°$. Both errors are approximately twice the corresponding errors on fig. 13. On the other hand, this algorithm is order of magnitude faster and can be used as a benchmark for rotational analysis.

The fig. 13a shows that accuracy of our model stabilised around $\sim 97\%$ on the validation set. This improvement comparing to fig. 10b is due to the increased dataset. Since the enlargement was artificial, the impact of physically enlarged dataset should be even bigger. From fig. 13b that angle detection error is around $\sim 5°$, which seems to be a good starting point for the directional study.

# 5 Summary and conclusions

Machine Learning is a very promising approach for image classification in Dark Matter search. Fast and computationally cheap algorithms like Boosted Decision Trees and Random Forests showed some improvement (fig. 7) comparing to the physical cuts used to sort out the background. At the same time, even the simplest Convolutional Neural Network outscored them (fig. 9a) in the classification problem.

Changing the Network architecture from 2D layers to 3D allowed better accounting of the correlations between different polarisations, which increased the ConvNet's performance (fig. 10b comparing to fig. 9b).

The amount of training data is another important factor for the algorithms performance, that was shown on fig. 11 and confirmed with the artificially increased dataset by rotations on fig. 13a.

The best performance achieved at this point is 97%. Further increasing of the performance is expected by studying the architecture and parameters of the networks, as well as changing the dataset to a broader one mentioned in the sec. 2.2.

Image rotations allowed us to teach our algorithm to identify the rotation angle of the image with quite good accuracy (fig. 13b). However, this is not the physical direction of the track, but the difference between the initial direction and rotated one. Two main problems are:

- The background is already isotropic, but we rotate it as well and algorithm succeed in identifying this angle.
- The angular deviation of nuclear recoils is $\sim 0.5$ rad [3], which is much bigger than the $5^\circ$ error of our algorithm.

Further research is needed in order to better understand the physical sense of the output angle of our algorithm and to get the output angle we are really interested in.

Analysis of the best metrics for our task is also a part of the further research.

# References


[1] Aleksandrov, A., et al. "NEWS: Nuclear Emulsions for WIMP Search." arXiv:1604.04199

[2] K. Simonyan, A. Zisserman "Very Deep Convolutional Networks for Large-Scale Image Recognition" arXiv:1409.1556

[3] N. Agafonova, et al. "Discovery potential for directional Dark Matter detection with nuclear emulsions" arXiv:1705.00613