UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

Automatic detection of visible events in fusion reactors with Deep Learning

Degree Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

David Serrano Lozano

In partial fulfillment
of the requirements for the bachelors's degree in
Telecommunications Technologies and Services **ENGINEERING**

Advisor: Josep Ramon Morros Rubió
Barcelona, Date June 2021

Contents

List of Figures

List of Tables

Acronyms

ADASYN Adaptive Synthetic sampling

AN Anomaly

BBox Bounding Box

CE Cross-Entropy

CNN Convolutional Neural Network

EDICAM Event Detection Intelligent Camera

ENN Edited Nearest Neighbors

FC Fully Connected

FN False Negative

FP False Positive

HS Hot Spot

IPP Max Planck Institute for Plasma Physics

IR Infrared

KNN K-Nearest Neighbors

ML Machine Learning

NHS No Hot Spot

NN Neural Network

OvA One-vs-All

OvO One-vs-One

PFC Plasma-Facing Component

RD Random Displacement

ROI Region Of Interest

RR Random Rotation

SMOTE Synthetic Minority Oversampling Technique

SVM Support Vector Machine

TN True Negative

TP True Positive

UFO Unidentified Flying Object

W7-X Wendelstein 7-X

Abstract

The next decades are crucially important to putting the world on a path of reduced greenhouse gas emissions since energy demand is increasing more and more. That is why fusion power, one of the most environmentally friendly sources of energy, has been getting so much attention lately. Wendelstein 7-X is a stellarator, a experimental fusion reactor build by the IPP which intends to demonstrate the capabilities of fusion power to produce energy.

The main problem with this process is that the working conditions to achieve fusion are extremely dangerous and unstable and sometimes the reactor walls overheat, damaging the structure of the device. For this reason, a continuous real-time data acquisition, analysis and control system is necessary to protect the structure.

This thesis studies the procedure of generating a complete hot spot detector and classifier making use of the visible cameras installed inside the reactor. A Convolutional Neural Network is used to extract features from the bright events to later on use them to classify the incident with a Machine Learning classifier.

The outcome has been very satisfactory as the system has been able to detect all the dangerous events in the data base. In addition, the model has been able to detect the incidents in real time with a delay of less than one and a half seconds from the first occurrence.

Resum

Les pròximes dècades tenen una importància crucial per posar el món en un camí de reducció d'emissions de gasos d'efecte hivernacle, ja que la demanda d'energia augmenta cada cop més. Per això, darrerament la fusió nuclear, una de les fonts d'energia més respectuoses amb el medi ambient, s'està tenint tant en consideració. Wendelstein 7-X és un stellarator, un reactor de fusió experimental construït per l'IPP que té la intenció de demostrar les capacitats d'obtenció d'energia de la fusió.

El principal problema d'aquest procés és que les condicions de treball per aconseguir la fusió són extremadament perilloses i inestables i, de vegades, les parets del reactor se sobreescalfen, danyant l'estructura del dispositiu. Per aquest motiu, és necessari un sistema d'adquisició, anàlisi i control de dades continu en temps real per protegir l'estructura.

Aquesta tesi estudia el procediment per generar un detector i un classificador complet de punts calents fent ús de les càmeres visibles instal·lades a l'interior del reactor. Una xarxa neuronal convolucional s'utilitza per extreure característiques dels esdeveniments brillants i, posteriorment, utilitzar-les per classificar l'incident amb un classificador d'aprenentatge automàtic.

El resultat ha estat molt satisfactori, ja que el sistema ha estat capaç de detectar tots els esdeveniments perillosos de la base de dades. A més, el model ha estat capaç de detectar els incidents en temps real amb un retard inferior a un segon i mig des de la primera ocurrència.

Resumen

Las próximas décadas son cruciales para encaminar al mundo hacia una reducción de las emisiones de gases de efecto invernadero, ya que la demanda de energía aumenta cada vez más. Por eso, la energía de fusión, una de las fuentes de energía más respetuosas con el medio ambiente, está recibiendo tanta atención últimamente. Wendelstein 7-X es un stellarator, un reactor de fusión experimental construido por el IPP que pretende demostrar las capacidades de la fusión nuclear para producir energía.

El principal problema de este proceso es que las condiciones de trabajo para lograr la fusión son extremadamente peligrosas e inestables y, en ocasiones, las paredes del reactor se sobrecalientan, dañando la estructura del aparato. Por este motivo, es necesario un sistema de adquisición, análisis y control de datos en tiempo real y continuo para proteger la estructura.

Esta tesis estudia el procedimiento para generar un detector y clasificador de puntos calientes haciendo uso de las cámaras visibles instaladas en el interior del reactor. Se utiliza una Red Neural Convolucional para extraer características de los eventos brillantes para posteriormente utilizarlas para clasificar el incidente con un clasificador de Machine Learning.

El resultado ha sido muy satisfactorio ya que el sistema ha sido capaz de detectar todos los eventos peligrosos de la base de datos. Además, el modelo ha sido capaz de detectar los incidentes en tiempo real con un retraso de menos de un segundo y medio desde la primera aparición.

Revision history and approval record

Revision	Date	Purpose
0	28/04/2021	Document creation
1	04/06/2021	Document revision
2	15/06/2021	Document revision
3	19/06/2021	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
David Serrano Lozano	david.serrano.lozano@estudiantat.upc.edu
	99d.serrano@gmail.com
Josep Ramon Morros Rubió	ramon.morros@upc.edu

Written by:		Reviewed and approved by:	
Date	28/04/2021	Date	19/06/2021
Name	David Serrano	Name	Josep Ramon Morros Rubió
Position	Project Author	Position	Project Supervisor

1 Introduction

Wendelstein 7-X (W7-X) is a stellarator, a experimental type of fusion reactor built in Germany by the Max Planck Institute for Plasma Physics (IPP) [1]. Though this experimental reactor will not produce electricity, its goal is to prove that this type of device is suitable for a future fusion power plant with steady-state operation. The operational reactors currently generating power use nuclear fission. Both reactor types use nuclear processes to obtain energy, in that they involve nuclear forces to change the nucleus of atoms. Fission splits a heavy element into fragments, while fusion joins two light elements forming a heavier one. In both cases, energy is freed because the mass of the remaining nucleus is smaller than the mass of the reacting nuclei. Nuclear fission power plants have the disadvantage of generating unstable nuclei, some of these are radioactive for millions of years. Fusion on the other hand does not create any long-lived radioactive nuclear waste. A fusion reactor produces helium, which is an inert gas. It also consumes tritium within the plant in a closed circuit. Tritium is radioactive (a beta emitter) but its half life is short. Furthermore, it is only used in low amounts so, unlike long-lived radioactive nuclei, it cannot produce any serious danger. So, the latter is among the most environmentally friendly sources of energy. There are no CO_2 or other harmful atmospheric emissions from the fusion process, which means that fusion does not contribute to greenhouse gas emissions or global warming. That is why so much effort is being put into obtaining energy from nuclear fusion.

However, while it is, relatively speaking, rather straightforward to split an atom to produce energy (which is what happens in fission), it is a grand scientific challenge to fuse two hydrogen nuclei together to create helium isotopes (as occurs in fusion). The sun constantly does fusion reactions all the time, burning ordinary hydrogen at enormous densities and temperatures. But to replicate that process of fusion on Earth, where do not exist the intense pressure created by the gravity of the sun's core, it is needed a temperature of at least 100 million degrees kelvin, or about six times hotter than the sun. For this reason, the W7-X device confines the plasma (a gas of ions) with temperatures of up to 100 million kelvin, discharges lasting up to 30 minutes and a heating power of 10 million watts. To do that the reactor has a toroidal shape with 50 non-planar and 20 planar superconducting magnetic coils, which induces a magnetic field that prevents the plasma from colliding with the reactor walls. Hence, as the working conditions are extremely dangerous, continuous real-time data acquisition, analysis and control system is necessary in order to protect the Plasma-Facing Components (PFCs) from overheating.

Through the whole structure, with its tangential viewports into the plasma vessel, the prototype has distributed 10 visible cameras (EDICAM [2]) allowing to do the video diagnostic of the plasma in its visible spectral range (from 450 to 720 nm) [3]. The position, reference and field of view of each camera inside the reactor is shown on Fig. 1. In Fig. 2, a capture of the interior of the reactor is shown.

Inside of the reactor could occur several visible events within normal operation. Some of them are not directly harmful, considered as Anomalies, such as debris, pellets (impurities injected inside the reactor to control the plasma composition) and other unidentified flying objects (UFOs). Conversely, when some parts of the walls get hotter uncontrollably, called

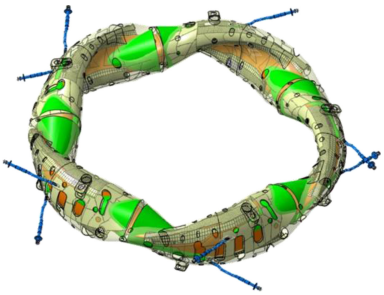


Figure 1: Orientation of the viewing cones (green) of the ten channels of the overview video diagnostics.

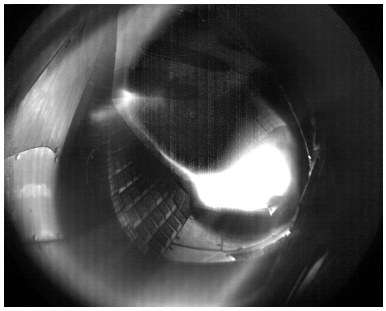


Figure 2: View of one of the EDICAM showing the inside of the toroid.

Hot Spots, is considered a high risk situation . IPP researchers expect to detect and classify both of those abnormal events using the data from the sensors [4].

The overall objective of this thesis is to develop a Hot Spot and Anomaly detector with a semi-automated labeling tool to prepare the database (the extensive definition of each class is done in the Database section). Despite detecting these abnormal events with the visual cameras, even to the human eye, is extremely hard, it is expected that the software detects and classify them with high accuracy to avoid burning out sensors and other elements of the reactor and label automatically more sequences from the cameras alongside a person to check the process to enlarge the database.

1.1 Requirements and specifications

Once a general overview of this thesis has been exposed, the requirements, the specifications and the concrete objectives can be stated.

The software itself has to extract features from the abnormal events using Transfer Learning and Deep Learning techniques and with them, classify the abnormal events using Machine Learning classifiers. One of the major goals of the programme is to prevent overfitting at all cost since the software will be used with new and unseen sequences. Moreover, the project has to be robust against class imbalance due to, as can be seen in the Database section, the data base has an enormous difference between the number of samples of the majority class and the minority classes.

In terms of results, the performance of the software should improve and strengthen the previous detector made by M. Cobos [5]. The classifier should focus on the recall of the hot spots since it is extremely important to detect all the moments when the walls are dangerously overheated. In other words, it should minimize the False Negatives (FN) of the hot spots.

Two points just as important as all the stated above are reasoning and justification. All the code has to be well documented to help for future work and easy to understand to let the reader see how the procedure has been done. The techniques used have to be reasoned to let all the people understand the key points of the project and its argument.

1.2 Work Plan

As can be seen in Figure 3, the project was split into smaller tasks.

The first part of the project consisted in making research in the topic under consideration, just as acquiring knowledge in the areas and techniques that were going to be implemented such as data augmentation, Deep Learning, focusing in transfer learning and fine tuning Convolutional Neural Networks (CNNs) and Machine Learning (ML) classifiers. Then, having a general overview of the project, the next objective was to understand, test and accomplish running the code of the previous work.

After doing the first two tasks, the following consisted in creating the software to accomplish the stated objectives. The process has been in the order of the sections of this document.

The structure and the process of the project has not suffered any big modifications. At the beginning, a few more ways to classify the events were decided to be used, but at the middle of the project, it was decided to focus on improving the results and in creating a real-time detector.

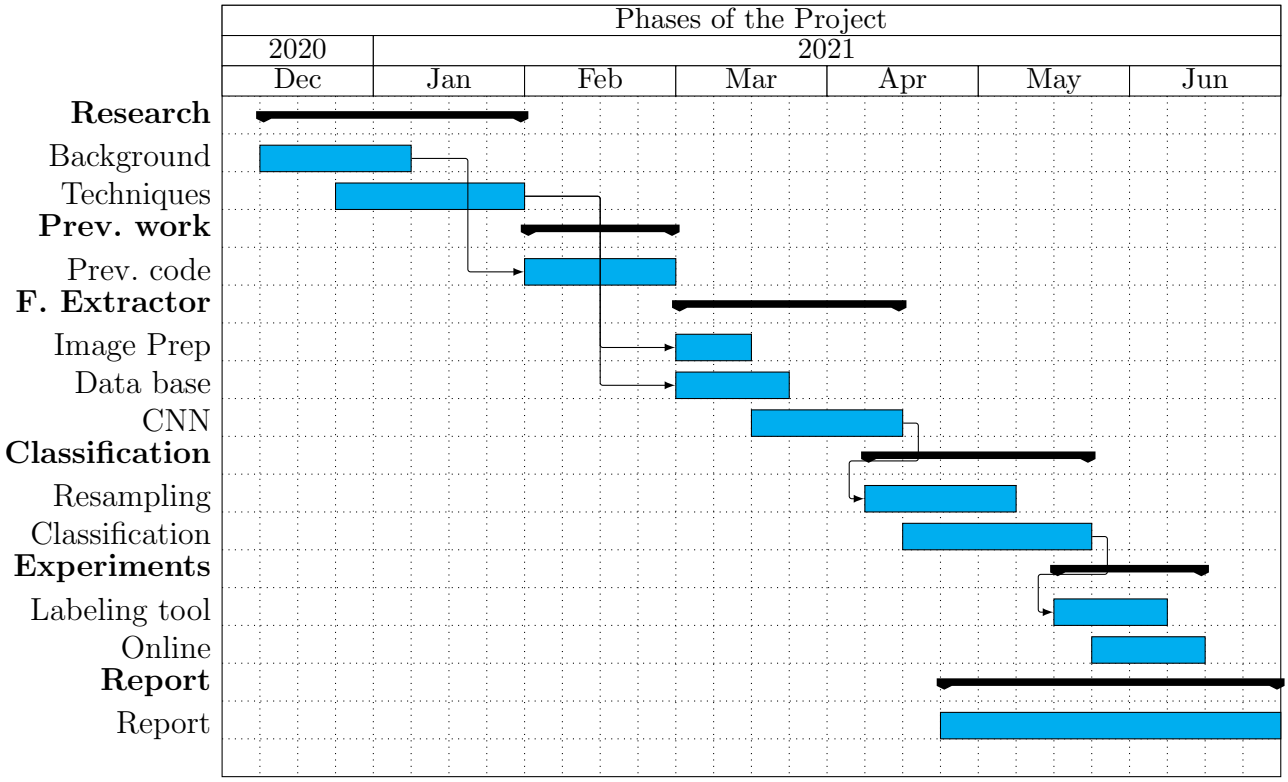


Figure 3: Gantt diagram of the project

2 State of the art

The road map to steady-state operation in W7-X consists of different operating phases (OPs) with increasing machine performance with every phase. The first measurement data was created in the so-called OP1.2, the first operating phase lasting more than 5 seconds, between the second half of 2017 and the first half of 2018 [6]. As a result, the detection of visible plasma events in the W7-X is a relative novel field of study.

2.1 Thermal events in W7-X

Despite the following field may slightly diverge from the events which are in the visible spectral range and from the starting point of this thesis, it is important to know that the IPP studies several ways of detecting the hot spots and other abnormal events. Aside from the EDICAMs, the reactor also contains some IR cameras with which A. Ali et al. [7] developed and successfully tested a method to classify plasma impurities deposited as surface layers. A. Puig Sitjes et al. [8] develop an algorithm which uses a scene model that provides pixel-wise information of the Plasma-Facing Components (PFC) to detect thermal events including overload hot spots and shine-through hot spots due to the heating systems among other issues.

T. Szepesi et al. [9] demonstrated, in the first two campaigns OP1.2 and OP1.2a, that the EDICAM-based visible video system can be simultaneously utilized for safety-related and scientific observation. This is because thanks to the non-destructive readout (NDR) capability of the EDICAM cameras, when reading only a small part of the camera sensor (ROI), the readout speed can be significantly increased and carry valuable extra information, such the time-dependency of an event.

A. Puig Sitjes et al. [10] developed a system for the PFCs protection using both the IR cameras and the EDICAMs. The software is divided into two main applications (see Fig.4) which, with ten workstations per camera type, the sequences are acquired and analyzed online by means of computer vision techniques to detect hot spots and other thermal events. If a critical condition is reached, an alarm is sent to the interlock system, and proper action has to be taken. The hot spots are detected by temperature thresholding with the sequences acquired from the IR cameras and the video streams in the visible spectrum are only analyzed in order to detect bright spots due to localized fast particle losses from the Neutral Beam Injection system.

However, although some of the previous mentioned works used the EDICAM cameras or demonstrated that the EDICAM-based system is extremely useful, the IR cameras were their main focus and in this thesis the study is only done in the sequences obtained in the visible spectral range.

2.2 Visible events in W7-X

The most similar work and the only one which uses only the sequences acquired from the EDICAMs, was done by M. Cobos for his Master Thesis Dissertation in which completed a first version of the project and achieved some tangible results. This thesis uses the

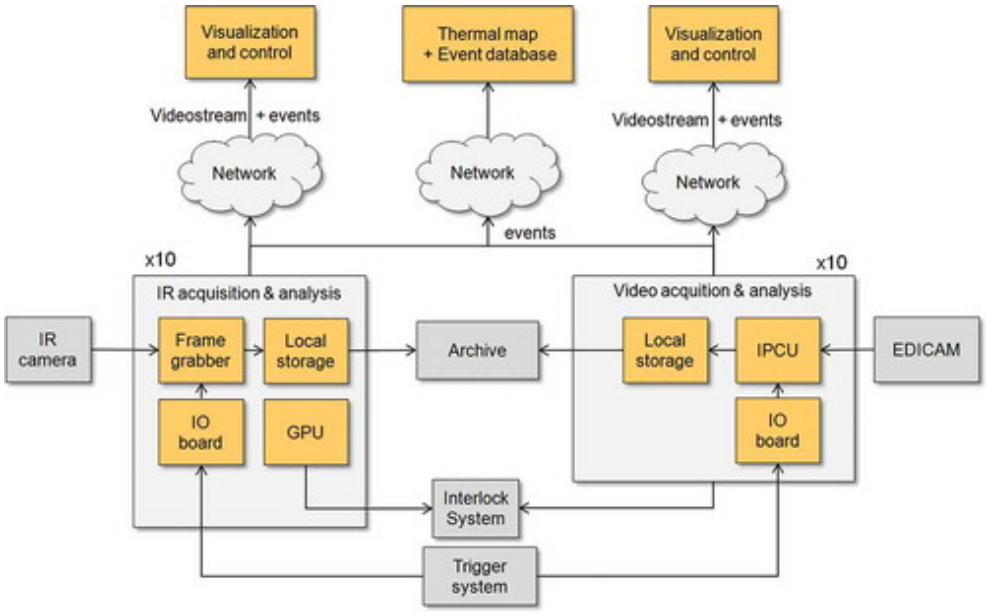


Figure 4: A. Puig Sitjes et al. system overview

database from the mentioned work as starting point since the sequences are exactly the same and only the scientists from IPP are properly qualified to label the detections. Moreover, using the same baseline as Martí allows to do a good comparison between techniques. As a result, to enable the reader have a much wider context, the bright spot detector and the tracker are briefly explained hereunder.

The software takes one video stream and for each frame applies sequentially a Top Hat Transform, an Otsu Binarization or Thresholding Method [11], background subtraction, some morphological operations and a maxima suppression algorithm to merge similar zones. The output of the previous pipeline are a lot of Bounding Boxes (BBoxes) of possible candidates of being hot spot.

Right after, an algorithm takes all the bright events and searches in the 50 following frames for other candidates with an euclidean distance difference between centroids of less than 10 pixels to link and assign them to the same track. Later, all the obtained tracks are manually labeled between No Hot Spot (reflections or background, not harmful for the reactor), Hot Spot and Anomaly.

In Fig.5 a example of the entire process can be seen

2.3 Feature extraction and classification

When the pre-processing and the desired levels of segmentation (ROI) have been achieved, a feature extraction technique is applied to obtain features. This methods have been given as “extracting from the raw data information that is most suitable for classification purposes, while minimizing the within class pattern variability and enhancing the between class pattern variability” [12]

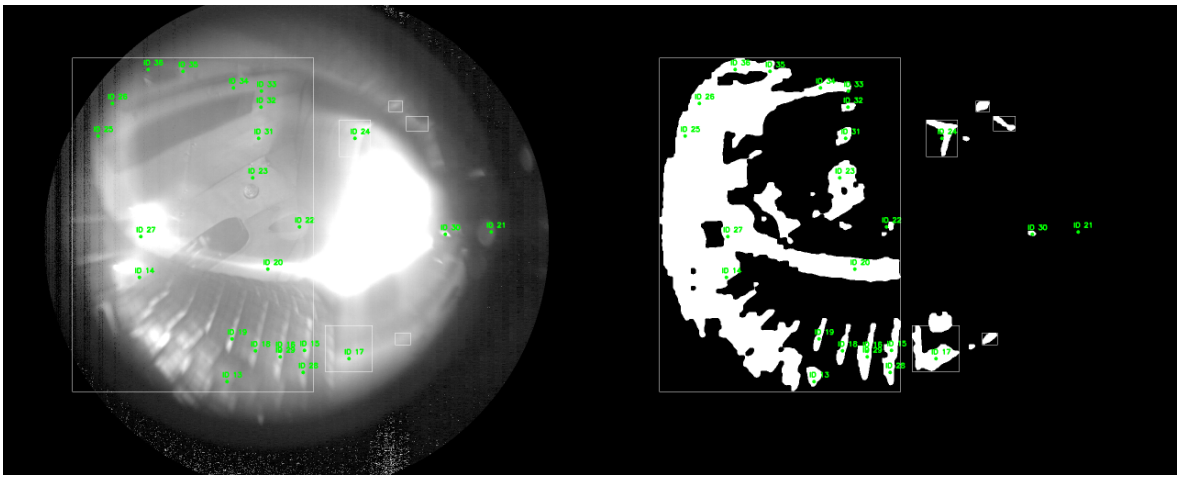


Figure 5: M. Puig Cobos image processing pipeline. On the left a frame of a sequence. On the right the same frame after Mart's pipeline. In green, the centroids of the detections of the previous 50 frames. In white, the Bounding Boxes of the actual frame exclusively.

CNNs learn the basic shapes in the first layers and evolving to learn features of the input image in the deeper layers. Due to the characteristics of the images to analyze in this thesis, a CNN is used to both classify and extract features from the tracks. However, as the data base is small, Transfer Learning and fine-tuning is used. N. Tajbakhsh et al. [13], using a wide variety of CNNs trained in the ImageNet competition and focusing in medical images, demonstrated that deeply fine-tunned CNNs are useful for image analysis, performing as well as fully trained models and even outperforming the latter when limited training data are available. Nevertheless, it is not expected to achieve extremely good results only using a CNN because of the similarity of the images from different classes. Therefore, as it was demonstrated by S. Notley and M. Magdon-Ismail in [14] to improve the performance on classificating images, in addition of the CNN, a variety of ML classifiers are concatenated to the next to last layer of the CNN.

3 Database

This thesis uses the database created by M. Cobos for his Master Thesis Dissertation in which 14 sequences from the first W7-X operation (OP1.2 and OP1.2a) are labeled. The bright events which have linked the frame number and its BBox were manually classified between:

- No Hot Spot (NHS). Small parts of the plasma running through the toroid, reflections from the plasma and other bright points which are not harmful to the reactor (see Fig. 6).
- Hot Spot (HS). Parts of the wall that get hotter uncontrollably. This event has burned several sensors of the reactor and it is the main event to detect for safety reasons (see Fig. 7).
- Anomaly (AN). This class groups all the events that are neither NHS nor HS which are not directly harmful, but are important to detect such as falling debris (small broken parts of the reactor), pellets (impurities injected inside the reactor to control the plasma composition) and other unidentified flying objects (UFOs) (see Fig. 8).

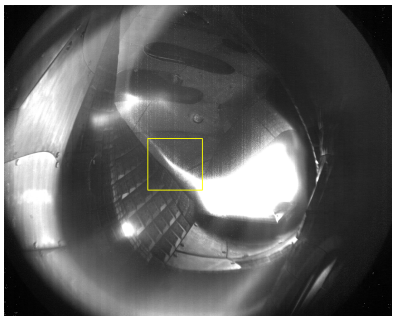


Figure 6: No Hot Spot detection. Part of the plasma.

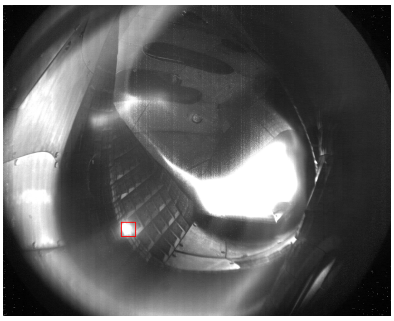


Figure 7: Hot Spot detection. Overheated part of the wall.

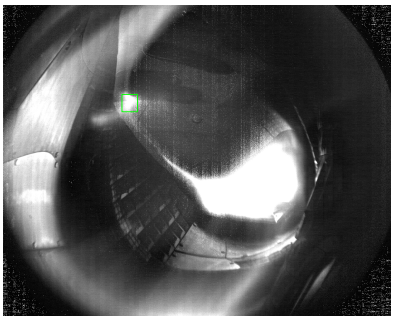


Figure 8: Anomaly detection. Pellet.

The bright events were detected at frame level, but each of them were associated to a track. The frame level detections are all the candidates, while a track is composed by all the frame level detections which are caused by the same event along a number of frames. So, a track is a set of bright events in which the reason for the appearance is the same and the movement between successive frame detections is less than 10 pixels (using the Euclidean distance between the centroids of the BBoxes)(see Fig. 9)

In table 1, the number of detections both at frame and track level can be seen. In this work, the focus is in detecting, classifying and predicting the tracks to exploit the temporal information that can exist. In addition, for future work, in Appendix 1 a more detailed view of the number of detections can be seen separated by files and its names.

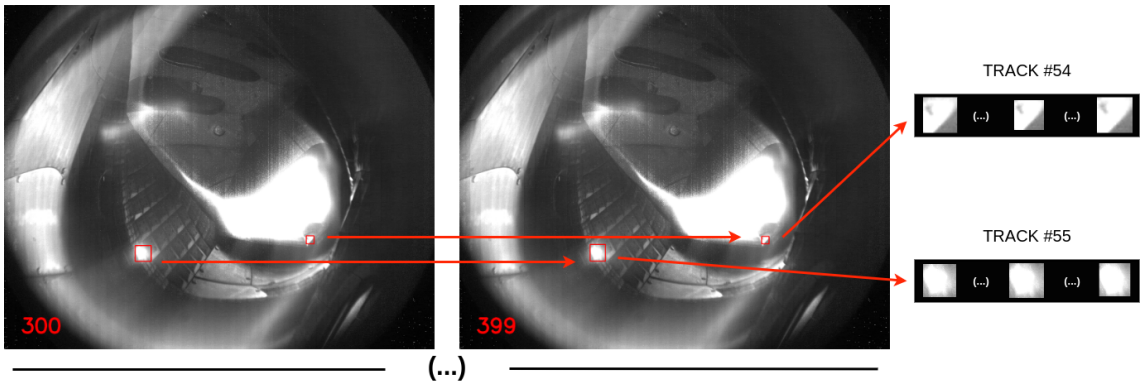


Figure 9: Track of detections. The temporal evolution of two tracks marked with their respective BBox can be seen.

	Frame level			Track level		
	NHS	HS	AN	NHS	HS	AN
Detections	170430	14872	6416	873	41	37

Table 1: Total number of detections divided in its labels: NHS, HS and AN.

At first sight, it can be seen some of the biggest challenges of this project. The database is quite small (951 tracks), it is a multiclass classification and two of the three classes are extremely imbalanced in relation to the majority class

	NHS	HS	AN
Percentages (%)	91.80	4.31	3.89

Table 2: Percentages of class imbalance

4 Feature Extraction

Feature extraction is a type of dimensionality reduction by which an intial set of raw data is reduced to more a manageable group. It is the name for methods that select and combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set. As images have a large number of variables that require a lot of computing resources to process, in this section how and with which techniques the feature extraction has been done is explained.

4.1 Track Subsampling

The main task of this project is to classify the tracks of the bright visual events found on the reactor and, as seen in the Database section, the tracks are sets of detections that have appeared for the same reason. So, the first step is to subsample the tracks since they can last up to 1200 frames, as can be seen in the histograms of the frame-lengths of the tracks separated by classes in Fig. 10. Moreover, the subsampling takes on more importance because the tracks have little movement. So, instead of taking all the detections of the tracks, only n of them equispaced are taken (in all the experiments done, $n = 5$).

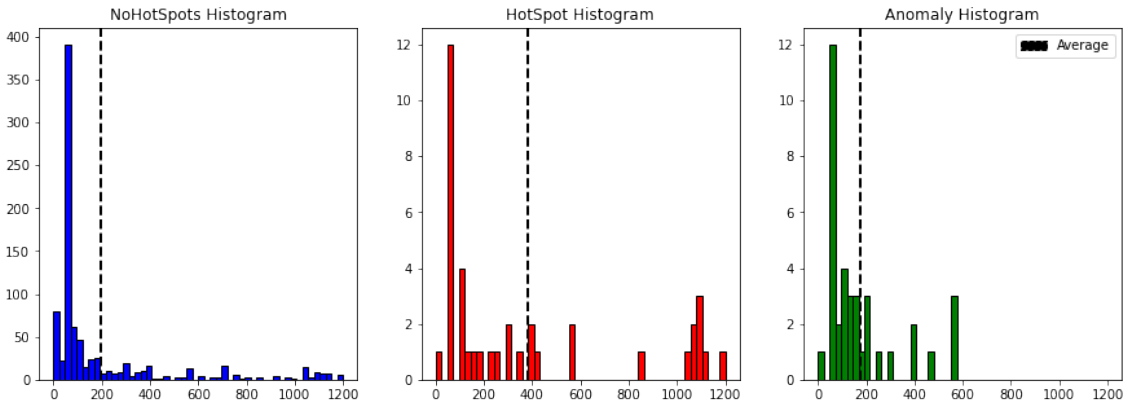


Figure 10: Histograms of the track's frame-lengths separated by classes.

4.2 Image Preprocessing

Before working with the images taken from the sequences some preprocessing has to be done. The features are not extracted from the entire frame, but from the BBox of each detection. This means that all the detections are cropped to a square shape with size of the maximum BBox length and resized to the desired size (224 x 224 pixels since ResNet50 is used)(see Fig. 11).

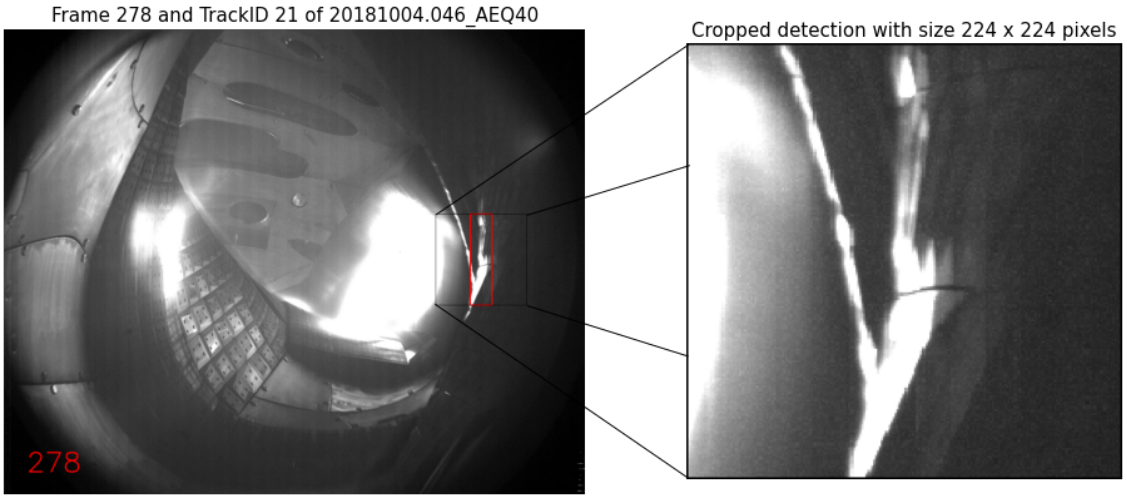


Figure 11: On the left, the original image of a sequence with a detection's BBox in red. On the right, the cropped image of the shown detection.

4.3 Convolutional Neural Network

As it was said in State of the art section, this thesis uses a CNN to obtain features from its logits (the raw outputs from a CNN layer). A Neural Network (NN) is a series of algorithms that endeavors to recognize underlying relationships and patterns in a set of data through a process that mimics the way the human brain operates. NNs are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer in which its output is a series of logits that can be transformed into class probabilities. Each node, or neuron, connects to another and has associated a weight and a threshold value (see Fig. 12). This weights and values have to be adjusted by training on the basis of a set of training data in a way that solves a specific problem. This is done with backpropagation, the essence of NN training, that aims to minimize the cost function by adjusting the previous mentioned weights and values [15]. The backpropagation algorithm computes the gradient of the loss function for a single weight by the chain rule starting from the end and adjusts the weights such that the error is decreased (see Fig. 12).

Over the last few decades, NNs have been considered to be one of the most powerful tools, and have become very popular in the literature as it is able to handle a huge amount of data. One of the most popular NNs is the CNN [17] taking its name from one type of layer which applies a mathematical linear operation between matrices called convolution. The particular reason this type of NN is used is as a consequence of performing extremely well at images and computer vision applications.

However, as today's CNNs have millions of parameters to train requiring a large amount of data and the database is quite small with a high class imbalance ratio, a CNN cannot be trained from scratch to achieve its full operational capacity. Therefore, Transfer Learning is used [18]. Transfer learning is a supervised learning technique that reuses parts of a previously trained model on a new network tasked for a different but similar problem. Using a pre-trained model significantly reduces the time required for feature engineering

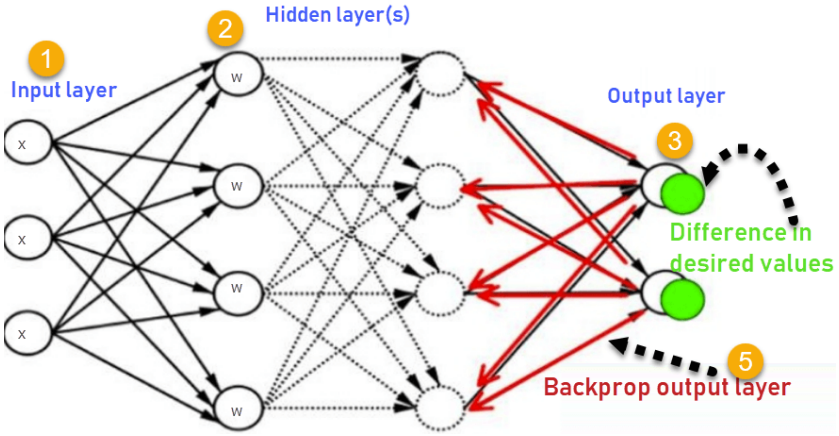


Figure 12: Structure of a NN and how backpropagation algorithm works. (1) Input data x is (2) modeled using the current weights w to (3) get an output. (4) The error in the outputs is calculated with $Error_B = ActualOutput - DesiredOutput$ and (5) travels back to the first layer adjusting the weights such the error is decreased. Image extracted from [16]

and training. The first step is to select a source model, ideally one that has been trained with a large dataset. The goal is to create a framework that is at least better than a naive model selecting only some layers to reuse in the custom model.

Plain CNNs have convolutional layers followed by some fully connected (FC) layers. The convolutional layers are the mayor building blocks which their main focus is to extract and segment the characteristics of the input data, while the FC layers are those layers where all the inputs from one layer are connected to every activation unit of the next layer which are used for the classification task. It is expected that the deeper the CNNs are, the more accurate they will be. However, when the layers of these networks are increased, the problem of vanishing gradients occurs [19].

During backpropagation, each of the neural network's weights receives an update proportional to the partial derivative of the error function with respect to the current weight in each iteration of training. The problem is that in some cases, the gradient is vanishingly small, effectively preventing the weight from changing its value, that is to say, when the network is deep, and multiplying a few of these small numbers the gradient becomes zero. In Fig. 13, extracted from the paper which introduced Residual Networks (ResNets) [20], the test errors of a 20-layer and a 56-layer plain networks can be seen. The degradation problem due to vanishing gradients is easy to see. To solve this problem, a variant of the ResNet family is used.

4.3.1 ResNet50

In this project the ResNet50 [20] is used to extract features from the BBoxes of the bright events found on the reactor sequences. ResNet became the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2015, as well as winner of MS COCO 2015 [21].

ResNet50, trained on a subset of ImageNet, is a pre-trained Residual Network, a variant of the CNN model which introduces shortcut or identity connections between layers to be able to create deep models without the vanishing gradients problem [22]. In these connections, the output of the previous layer is added to the output of a more advanced layer. Consequently, even if there is vanishing gradients for the weights layers, the identity x is always transferred back to the previous ones (see Fig. 14).

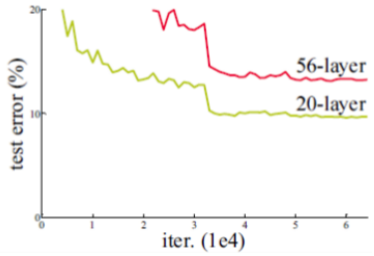


Figure 13: Test errors of two plain CNNs trained in the CIFAR-10 dataset. The deepest network performs worse than the 20-layer network due to the vanishing gradients. Image extracted from [20]

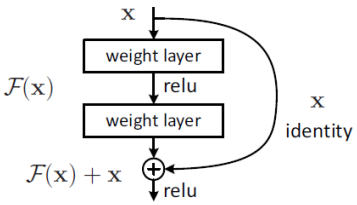


Figure 14: Example of an identity connection skipping two layers. Image extracted from [20]

ResNet50 is based on 5 main stages using a bottleneck design to reduce its time complexity since the network is very deep. This design adds a 1 x 1 convolutional layer at the start and at the end of each block. It turns out that doing this the number of connections and parameters can be reduced while not degrading the performance of the network so much. In Appendix 2 the whole structure of the ResNet50 can be seen.

4.3.2 Fine-tuning

Fine-tuning is a widely used technique for model reuse which consists in unfreezing a few of the last layers of an already trained model, and jointly training both the newly added part of the model and the still frozen layers. Some weights remain the same and cannot be modified when training (freeze), while the others are edited according to the training data set. ResNet50 is trained in a subset of ImageNet and its final layer has an output layer of 1000 logits (number of classes of the data base). As the used data base has only 3 different classes and the main objective of the network is to extract features, the entire FC layer is created from scratch. Moreover, the images from the sequences differ a lot from the images used to train the ResNet50 since the streams from W7-X are in gray-scale and the ImageNet data base is composed of color pictures of everyday items. For that reason, not only the FC layers are fine-tuned, but also the last block of the model (see Fig. 15)

4.3.3 Evaluation

The modified ResNet50 has to be trained with some epochs of the data base to be refined and fine-tuned. An epoch is a term used in ML that quantifies the number of passes of the entire training data set the algorithm has completed. The outputs of the model are unnormalised predictions which can give results, but interpreting their raw values is

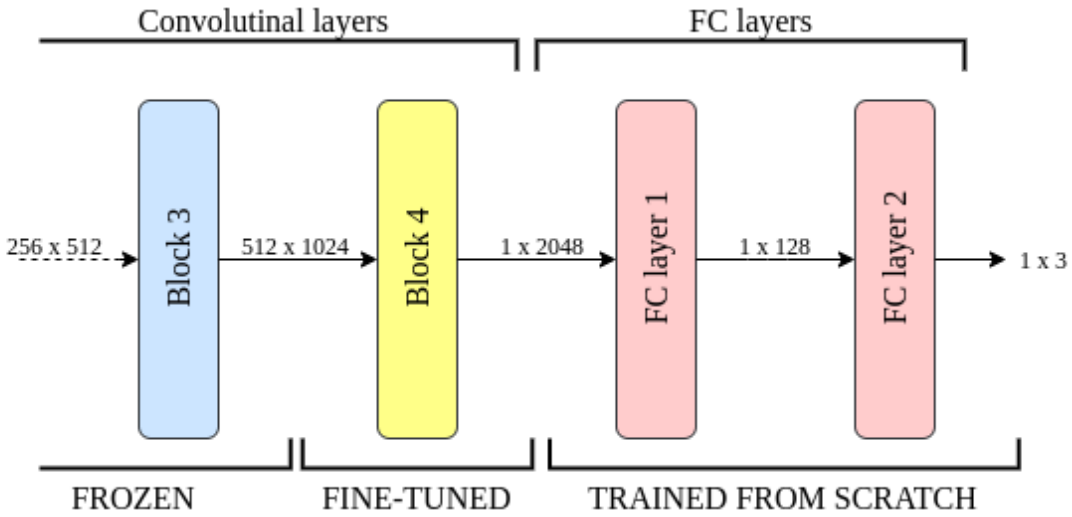


Figure 15: Last layers of the modified ResNet50

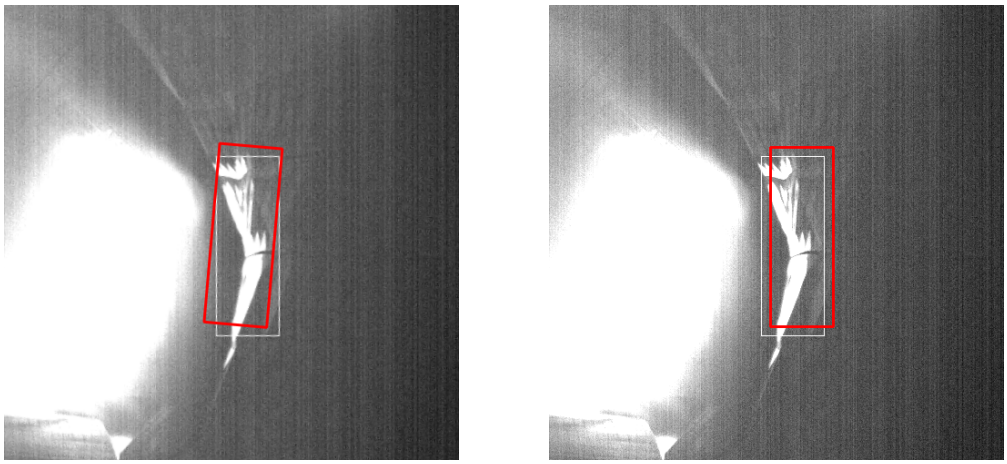
not easy. So, to know when the feature extractor is fully trained, a softmax function is concatenated to the last FC layer to transfer the last logits to probabilities. The softmax function is a generalization of the logistic function to multiple dimensions which takes as input a vector z of K (K is the number of classes, 3) real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input logits. The standard softmax function $\sigma : \mathbb{R}^K \rightarrow [0,1]^K$ is defined by the formula

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, ..., K \text{ and } \mathbf{z} = (z_1, ..., z_K) \in \mathbb{R}^K \tag{1}$$

4.4 Image Augmentation

As it is said in the Database section, the data base is only made up of 951 tracks with a lot of class imbalance. As a consequence of this and the fact that deep networks need large amount of training data, image augmentation is used to boost its performance. Image augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of the images [23]. Training deep networks on more data can result in more skillful model, and the augmentation techniques can create variations of the original frames that can improve the ability of generalizing what they have learned to new images and reduce the overfitting.

Two image augmentation techniques and the combinations between them are used. A random displacement and a random rotation of the Bounding Box, both with an uniform distribution centered in 0 (see Fig. 16).



(a) Random rotation (b) Random displacement

Figure 16: In (a) an example of a rotation of a BBox of 5 degrees. In (b) an example of a displacement of 10 pixels in the x axis and -10 pixels in the y axis. The white boxes are the ground truth detections and the red boxes are the same ones after applying image augmentation.

5 Classification

Classification is the process of predicting the classes of a set of given data points. To obtain this data points or features, the last layer of the fully trained ResNet50 has to be removed. That is why the structure of the FC layers was modified. So, the output of the feature extractor is 128 features for each input image. As per each track n detections are subsampled, the output at track level is nx128. In this section, how these features and with which techniques are analysed to classify the tracks is explained.

5.1 Resampling methods

As it is explained in the Database section, the data base has an extremely class imbalance between NHS and both classes of abnormal events. Using the features directly from the NN there exists an algorithmic bias. Algorithmic bias describes systematic and repeatable errors that create unfair outcomes, such as privileging one arbitrary class over others which can emerge due to may factors, being one of them the way how the data is structured. Consequently, in this thesis, some resampling methods are used to level the number of features per class and later on classify the track in question. Resampling methods can generate different versions of the training set that can be used to simulate how well models would perform on new data. These techniques differ in terms of how the resampled versions of the data are created and how may iterations of the simulation process are conducted. These methods are divided between over-sampling and under-sampling. Over-sampling focuses in generating new synthetic samples of the minority classes, while under-sampling focuses in deleting samples from the majority class. However, as the dataset is small, under-sampling techniques are not used alone, but combined with other over-sampling techniques to improve the overall performance as demonstrated by A. Agrawal et al. creating and testing SCUT: SMOTE and cluster-based under-sampling [24].

In (Fig. 17), all the resampling methods used and explained in the following subsections can be seen using the first and second of all the features of the training dataset.

5.1.1 SMOTE

SMOTE, described by N. Chawla et al. in their 2002 paper [25], is the most widely used approach to synthesizing new examples. This algorithm applies KNN approach where it selects K nearest neighbors, joins them and creates the synthetic samples in the space. Specifically, this method first selects a minority class instance a at random and finds its K nearest minority class neighbors. The synthetic instance is then created by choosing one the K nearest neighbors b at random and connecting a and b to form a line segment in the feature space. The synthetic instances are generated instances as a convex combination of the two chosen instances a and b [26].

SMOTE has a big issue when there are observations of the minority class which are outlying and appear in the zone where the majority class is prevailing since it creates samples formed in line bridges inside the majority class zone.

5.1.2 Borderline-SMOTE

Bordeline-SMOTE is a variation of the SMOTE, presented by H. Han et al. in [27], which solves the issue stated above. This algorithm classifies any minority observation as a noise point if all the neighbors are majority class points and such an observation is ignored when creating data. Further, it classifies a few points as border points which have both majority and minority class instances as neighborhood and resample completely from these points.

5.1.3 ADASYN

ADASYN, stated by H. He et al. in their 2008 paper [28], is based on the idea on adaptively generating minority data samples according to their distributions using KNN. The algorithm adaptively updates the distribution and there are no assumptions made for the underlying distribution of the data. The key difference between ADASYN and SMOTE is that the former uses a density distribution, as a criterion to automatically decide the number of synthetic samples to compensate for the skewed distributions. The latter generates the same number of synthetic samples for each original minority sample.

5.1.4 SMOTEENN

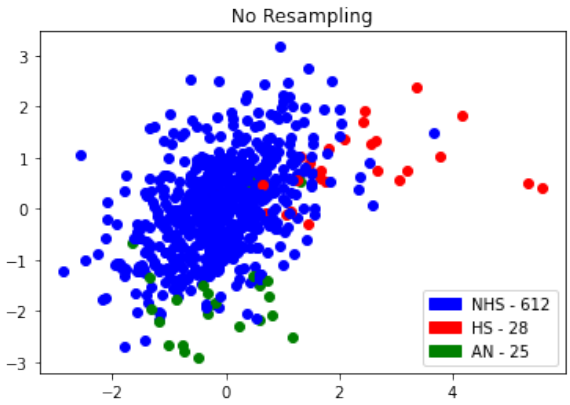
ENN, presented by D. L. Wilson in [29], can be used as an under-sampling method or as a data cleaning and as said in the introduction of this section, this method is used combined with an over-sampling method, the SMOTE. ENN is a rule for finding ambiguous and noisy examples which by finding the K-nearest neighbor of each observation first, then checks wether the majority class from the observation's K-nearest neighboor is the same as the instance's class or not. If the majority class of the observation's K-nearest neighbor and the observation's class is different, then the observation and its K-nearest neighboor are deleted from the dataset. So, before applying the ENN, the SMOTE algorithm is implemented to the output features [30].

5.1.5 SMOTETomek

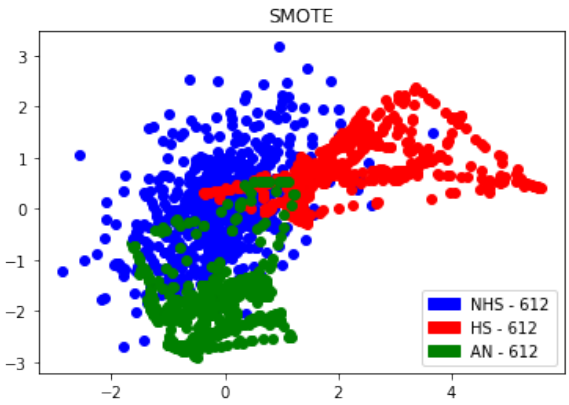
Tomek links can be used in the same applications as ENN, and it is used with SMOTE as well. Tomek links, introduced by I. Tomek in [31], are pairs of instances of opposite classes who are their own nearest neighbors. In other words, they are pairs of opposing instances that are very close together. Tomek's algorithm looks for such pairs and removes by choosing between only the majority instance of the pair or both of them. The idea is to clarify the border between the minority and majority classes, making the minority regions more distinct. So, before applying the Tomek links, the SMOTE algorithm implemented to the output features [30].

5.2 Temporal classification

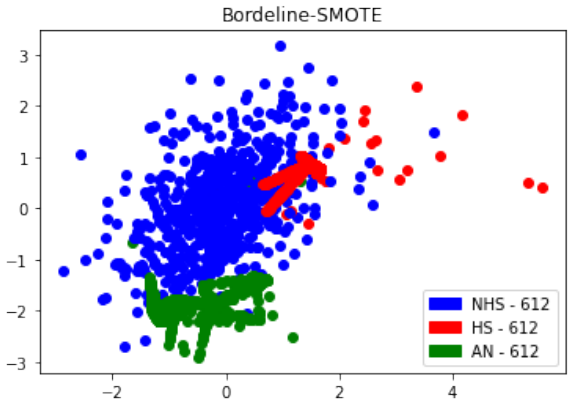
The final output of the system has to be a class prediction of the entire track. To do this, three different ways of merging the probabilities or features has been done. The first one uses the NN and the probabilities from the softmax function, while the other two



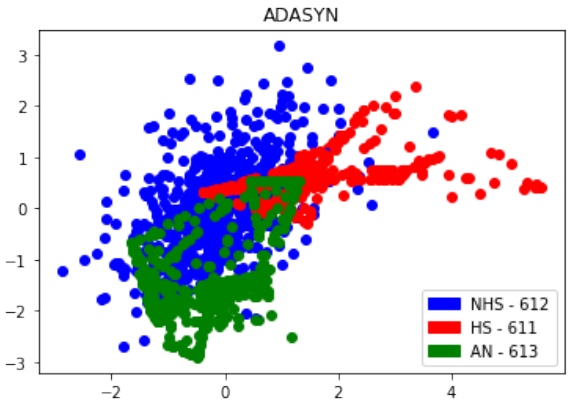
(a) No Resampling



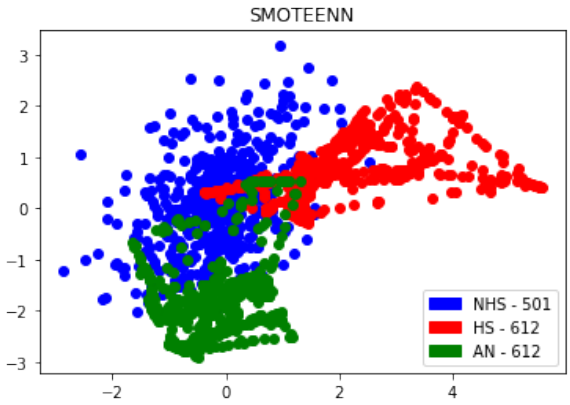
(b) SMOTE



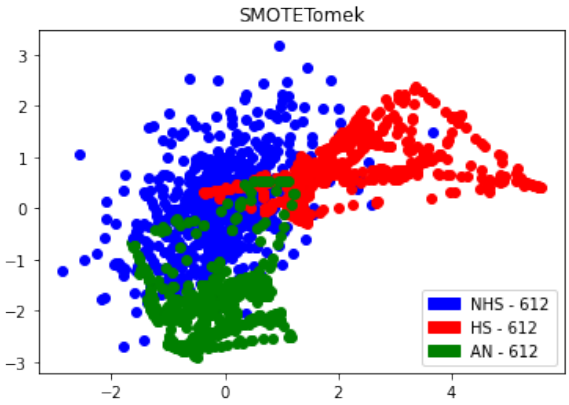
(c) Bordeline-SMOTE



(d) ADASYN



(e) SMOTEENN



(f) SMOTETomek

Figure 17: Resampling methods in the training dataset with a Standard feature scaling using the first (x axis) and second (y axis) of the 128 features. In the legends of the subgraphs it is shown the number of instances per class. Despite in the figures it may seem that all the features are very close together, it is important to remember that there are a total of 128 or 640 features.

techniques remove the the softmax layer and the last FC layer to use the 128 output features per image.

5.2.1 Average of the probabilities

The main focus of this thesis is to use a ML classifier concatenated right after a NN expecting to give better results than using only the NN. But since the final outcome is not known and it is very useful to see the difference in performances, the NN is used as a classifier too.

After the softmax function, the outcome is a class probability for each class and for each image in the track. So, to take advantage of the temporal information between frames, the probabilities are average along the image, obtaining only a class probability for each class of the entire track. Then the class with the highest probability is taken as the track class (see Fig. 18).

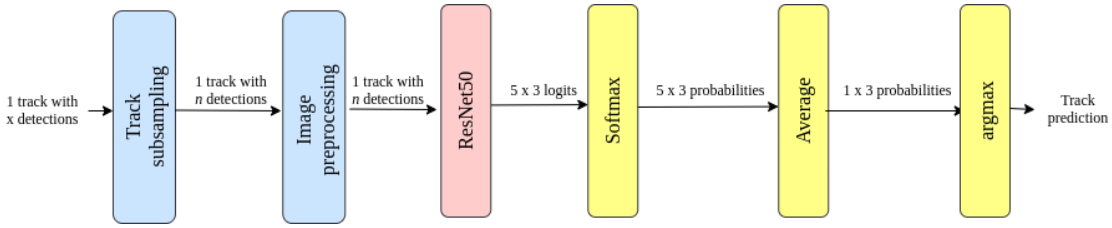


Figure 18: Feature extractor system and analysis. The yellow blocks are used as a classifier to know when the NN is completely trained

5.2.2 Average of the features

This section and the following one use the features extracted from the second last layer of the NN. Doing this, 128 features per image or nx128 per track are obtained. To try to reduce the noise from the features, this technique makes the average of all the features of all detections of the track obtaining always 128 features. This method do not exploit as well as the other method used the temporal information as all image features are compressed in the same one.

5.2.3 Concatenation of the features

To try to exploit the temporal information of the tracks, this technique concatenates all the features obtaining a vector of nx128 (as n=5, 640) features. This method is more susceptible than the other one to fail due to noise or little displacements of the features when trying to classify them. Moreover, the number of total features is n times when using the other technique and the curse of dimensionality could occur. The curse of dimensionality refers to a set of problems that arise when working with high-dimensional data. Some of them are data sparsity (the model learns from the most frequently occurring combinations of attributes) and distance concentration (all the pairwise distances between samples in the space converging to the same value as the dimensionality of the data increases).

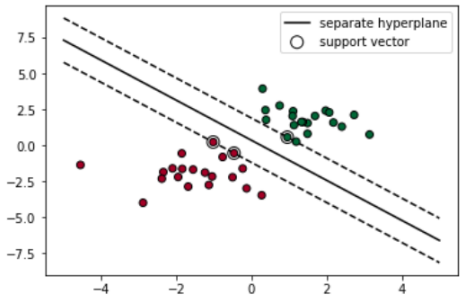


Figure 19: SVM example with a binary classification case. The continuous line is the hyperplane with highest margin between classes. The underlines points are the support vectors. Image extracted from [33]

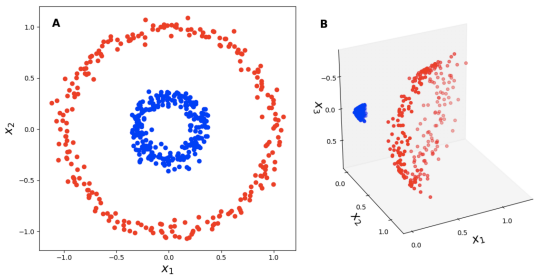


Figure 20: The "lifting trick". In the left a binary classification problem that is not linearly separable in $\mathbb{R}^2$. In the right a lifting of the data into $\mathbb{R}^3$ using a polynomial kernel, $\varphi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$. Image extracted from [34]

5.3 Machine Learning classifiers

As explained in the State of the art, the key point to obtain good results is the classifier. By concatenating a classifier after the NN, performance is expected to improve compared to when the ResNet50 is used with the Softmax function. In this section, the two ML classifiers used are explained.

5.3.1 SVM

SVM, introduced by V. Vapnik and C. Cortes in [32], is a supervised binary machine learning algorithm and one of the most robust prediction methods, based on statistical learning. The SVM algorithm finds and constructs a hyperplane or set of hyperplanes between instances of two classes. There exists infinite hyperplanes to separate the data, but SVM intend to find the most optimal hyperplane possible, the one with biggest margin between the support vectors. The support vectors are the instances that are closer to the hyperplane and influence the position and orientation of the subspace divisor (see Fig. 19).

Sometimes the data is not linearly separable and cannot be well divided with a hyperplane. That is why SVM uses a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form (see Fig. 20).

The SVM is a binary classifier, however the database has more than 2 classes. Two techniques are used to solve this problem: One-vs-All (OvA) and One-vs-One (OvO). Both techniques, analysed in the obtained features, involve splitting the multi-class dataset into multiple binary classification problems. The former, OvA, trains one classifier per class, it assumes that the class in question are positive labels and the rest as negative. On the other hand, the latter, OvO, trains a separate classifier for each different pair of labels. This is much less sensitive to the problems of imbalanced datasets but is much more computationally expensive.

5.3.2 XGBoost

Extreme Gradient Boosting, better known as XGBoost [35] is a decision-tree-based ensemble ML algorithm that uses a gradient boosting framework. Decision trees build classification or regression models in the form of a tree structure to break down the data set into smaller and smaller subsets depending on a set of criteria of the features to facilitate the prediction.

When separating the data in a tree model, instead of taking all the features at once, several subsets of data chosen randomly from the training dataset can be taken. This is called Bagging (Bootstrap Aggregation) and it is used to reduce the variance of a decision tree [36]. Each collection of data is used to train their trees ending up with an ensemble of different models which is more robust than a single decision tree. To decide the final prediction, the average of all the predictions from the different trees are used. Random Forest [37] is a bagging-based algorithm with a key difference wherein only a subset of features at random are taken.

Boosting is another ensemble technique to create a collection of predictors. In this algorithm, learners are learned sequentially with early learners fitting simple models to the data and then analyzing data for errors. In other words, consecutive trees are fitted, and at every step, the goal is to solve the error from the prior tree. When an input is misclassified by a hypothesis, its weight is increased so that next hypothesis is more likely to classify it correctly. When instead of increasing the weight the algorithm tries to optimize a loss function is called Gradient Boosting Decision Tree (Gradient Descent and Boosting) [38].

XGBoost creates several decision training datasets to build sequentially more than one tree such that each subsequent tree aims to reduce the errors of the previous ones. Each tree learns from its predecessors and updates the residual errors (see Fig. 21) [39] .

The reason of using XGBoost is that using the previous techniques and taking an extremely good software optimization yields superior results using less computing resources in the shortest amount of time (see Fig. 22). Since its introduction, this algorithm has not only been credited with numerous Kaggle competitions but also for being the driving force under the hood for several cutting-edge industry applications.

5.4 Stratified K-fold cross-validation

When evaluating the classifiers, stratified K-fold cross-validation [41] is used to obtain less biased and less optimistic estimate of the model skill. The procedure has a single parameter called K that refers to the number of groups that the entire data set is split. In all the experiments carried out the number of divisions has been of K=5. Moreover, all the divisions keep the class balance of the original data set to ensure that each fold is representative of all strata of the data. Once the divisions are done, of the K subsamples, a single one is retained as the validation data for testing the model, and the remaining K - *1* subsamples are used as training data. The cross-validation process is then repeated K times, with each of the K subsamples used exactly once as the validation data. Then, the K results are averaged to produce a single estimation.

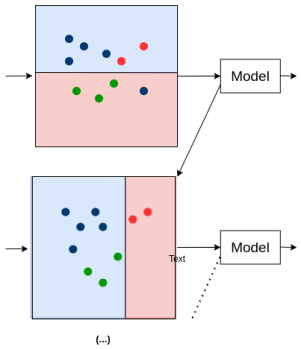


Figure 21: Two of all the trees XGBoost creates to sequentially train them using an optimization function.

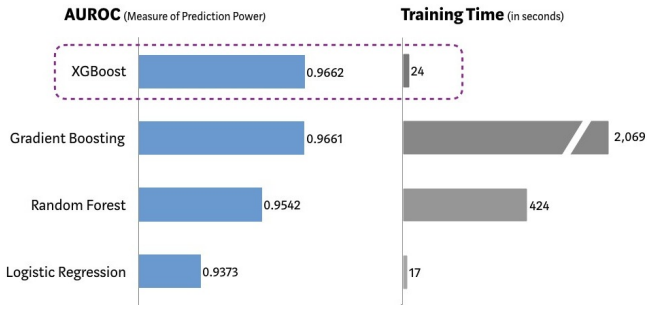


Figure 22: Test of several algorithms using Scikit-learn's 'Make Classification' data package to create a random sample of 1 million data points with 20 features. Image extracted from [40]

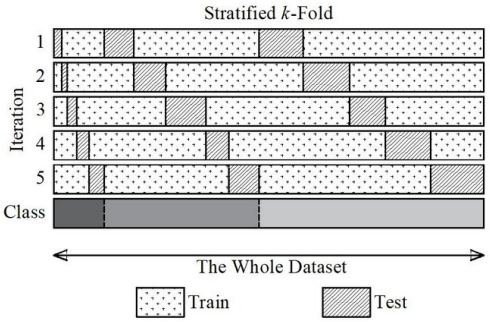


Figure 23: Stratified K-fold cross-validation.

6 Metrics

A classifier is only as good as the metric used to evaluate it. Choosing an appropriate metric or set of metrics is extremely important to evaluate the models. So, once all the techniques and procedures of how to predict the tracks detected on the sequences, the metrics used to evaluate this methods can be stated.

In binary classifications, only 4 situations in terms of results can happen. Taking one of the two classes as positive and the other one as negative there exist:

- True positive (TP). A TP is an outcome where the model *correctly* predicts the *positive* class.
- True negative (TN). A TN is an outcome where the model *correctly* predicts the *negative* class.
- False positive (FP). A FP is an outcome where the model *incorrectly* predicts the *positive* class.
- False negative (FN). A FN is an outcome where the model *incorrectly* predicts the *negative* class.

When the classification is a multiclass problem, like in this thesis, instead of a binary challenge, the previous conditions or terminologies can be used with some modifications. The ground truth class can be taken as the positive class and the other two as negative. Knowing that, the 4 well-known metrics used to evaluate the models can be stated:

- Precision. It quantifies the number of correct positive predictions made and evaluates the fraction of correct classified instances among the ones classified as positive.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

- Recall. It quantifies the number of correct positives predictions out of all the positive predictions that could have been made. It provides an indication of missed positive predictions.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

- F1 Score or F Score. It is a way of combining the precision and recall of a model, and it is defined as the harmonic mean of the model's precision and recall. Therefore, this score takes both FPs and FNs into account. F1 Score is a good metric to evaluate the overall model since sometimes it is difficult to determine if one algorithm is superior to another only looking to precision and recall.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{4}$$

- Confusion matrix. It is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in an

actual class while each column represents the instances in a predicted class. It gives insights not only into the errors being made by the classifier bur more importantly the types of errors that are being made.

So, after having stated and explained the metrics used, the expected results and the conditions that were optimized are declared. As the database is made of 3 different classes, the first three metrics explained, can be calculated for each label. The most important visual event to detect is the hot spot and it is extremely vital to detect all of them. Then, the recall of the HS class has to be maximized in order to classify all the occurring hot spots as HS. As stopping the reactor is a very cost-consuming task, it is important to avoid detecting HS that are not from that class. Hence, the overall performance of the model should be consistent, obtaining the F1 Score of the HS class as high as possible. In previous work made by M. Cobos only the F1 Score-weighted is mentioned, which is not significant due to the existing class imbalance on the validation set. The F1 Score-weighted (F1 W, in the tables) is calculated averaging all the F1 Score of all the classes according to the number of true instances for each label. In other words, the more instances the class has, the more importance the class takes. That means that the NHS label which is not the main event to detect takes approximately the 90% of the relevance. However, to compare between techniques, the F1 Score-weighted is shown too. Furthermore, to have a better overview of the performance of the models, the F1 Score-macro is used. F1 Score-macro (F1 M, in the tables) is the mean average between ever class regardless of the number of instances each class has. In other words, despite the fact that the NHS class has more instances than the other two, all three take on equal importance.

7 Experiments and results

During this document two divided blocks have been clearly differentiated: the development of a NN to extract features of the abnormal bright events and a classifier which is able to predict the label of the detection using the temporal information. In this section, the technical aspects and parameters of the experiments done are shown as well as the results obtained.

To create the software, python, pytorch and scikit-learn among other libraries have been used.

7.1 Feature extraction

7.1.1 Train/validation splits

To fully train the feature extractor, the entire data set has to be split in a training set and in a validation set. Usually, to train NNs one more set called test is used, but as the minority classes have so few instances, this set is removed to have more labels in the validation set. The models take the training set to learn and refine their parameters from its data, the test set is used once per epoch to know how well the model performs and the validation set helps to know how well the model works once the whole training is done. To train the feature extractor the training set has been created with the 80% of the instances and the validation set with the rest, 20%, maintaining the class proportion. Furthermore, images from the same track must not be placed in both database divisions due to the model would be training with the same examples as the validation set and could not be evaluated with unseen bright events.

7.1.2 Image augmentation

All the models have been tested with and without image augmentation. When using image augmentation the following combinations of the two methods explained in the Image Augmentation subsection have been used:

- RD5. Random Displacement (RD) between -5 and 5 pixels for each side.
- RR5. Random Rotation (RR) between -5 and 5 degrees from the center of the BBox
- RD5 RR5. RD between -5 and 5 pixels for each side and RR between -5 and 5 degrees from the center of the BBox.
- RD10 RR10. RD between -10 and 10 pixels for each side and RR between -10 and 10 degrees from the center of the BBox.
- RD% RR5. RD of the 10% of the side pixel length BBox and RR between -5 and 5 degrees from the center of the BBox.

7.1.3 Training

To train the feature extractor, a custom DataLoader function has been created. The DataLoader is a function which loads only a few samples of the data set instead of loading

all the data to have enough memory space to process it. Even the state-of-the-art configurations cannot carry all the data at once since the images of the sequences are in 16 bits per pixel (in reality the sequences have a dynamic range of 12 bits but as there is not a such variable type, the data was converted to 16 bit in order not to lose quality). For that reason, the DataLoader only loads a batch of the data set. The batch size determines how many training or validation examples are processed in parallel for training or inference. The batch sizes taken for all the models after some experiments were, for training, 7 tracks, and for validation, as many as the memory could handle, 12 tracks. The batch size in training time can affect how fast and how well the training converges. Thus, for the training batch size, it is worth picking a batch size that is neither too small nor too large. It has been observed in practice that when using a larger batch there is a significant degradation in the quality of the model, as measured by its ability to generalize. The lack of generalization ability is due to the fact that large-batch methods tend to converge to sharp minimizers of the training function. These minimizers which are the functions used to refine the weights of the NN, are characterized by large positive eigenvalues in $\nabla^2 f(x)$ and tend to generalize less well when the batch is too large. On the other hand, when the batch size is to small, the training time is bigger [42].

To refine the model a loss function has to be optimized. In all the models the Cross-Entropy (CE) Loss has been used. CE builds upon the idea of information theory entropy and measures the differences between two probability distributions for a given random set of events [43]. It is used after applying the softmax function of the last layer of the NN and it is defined as:

$$CE = -\sum_{i=1}^{C} t_i log(s_i) \tag{5}$$

where t_i ans s_i are the ground truth and the output of each class i respectively.

7.2 Classifiers

After passing the images through the NN, using the features extracted, the track has to be classified between the three possible classes. As explained in the classification section, three classifiers have been used, one of them transforming the output logits to class probabilities and the other two merging the features in different ways. In this section, the obtained results using the NN as a classifier using the softamax function and both ML classifiers are shown.

7.2.1 NN as a classifier

This technique concatenates a softmax function to the NN and transforms the logits to probabilites. Then the n class probabilities are averaged and the class with highest probability is selected as the track prediction.

So, using the ResNet50 as a feature extractor and a classifier at the same time the best HS recall is 0.67, obtained with almost every image augmentation technique. The best HS F1 Score is 0.75 using RD5 since this technique improves the HS precision by far.

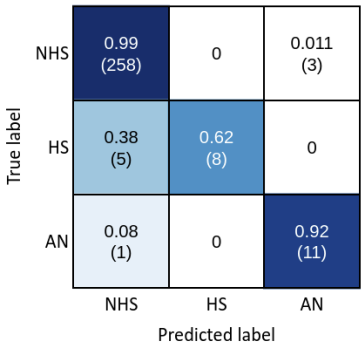
In Appendix 3 all the results obtained using all the image augmentation techniques can be seen.

7.2.2 Machine Learning classifiers

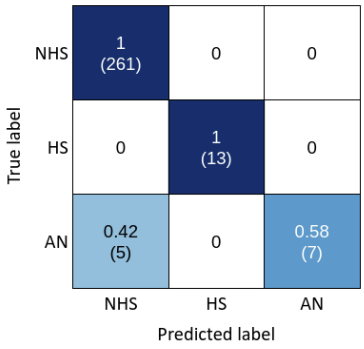
The other two classifiers used are the SVM and XGBoost. Both of them learn from instances, so, as explained in the classification section, the features are extracted from the next-to-last layer of the NN. Although, for each image 128 features are obtained, the class imbalance still remains. That is why one of the resampling techniques explained above is used on the training set before training the ML classifiers. To train them the stratified 5-fold cross-validation has been used.

The best performance using the SVM is obtained averaging the features since this classifier is very liable to missclassify if there exist noise in the instances because it is harder to find the hyperplane with the biggest margin. Moreover, the best results are also obtained using the same image augmentation technique as the best result using the NN as a classifier (RD5). Borderline-SMOTE is the resampling method that obtains the best results since is the one which focus on the points of the border of the instances of different classes and doing this helps SVM to find the best hyperplane. The best HS recall and F1 Score are 0.85 and 0.92 respectively, which indicates that the performance has been improved by a lot in respect of using the NN as a classifier. In Fig. 24 (a), the confusion matrix of the experiment with the best results can be seen.

However, the best performance using XGBoost are obtained concatenating the features, just the opposite of SVM. That is because the more features XGBoost has, the better classifies since more and deeper trees can be done. In addition, the best HS F1 Score has been obtained using the same image augmentation technique as the best result obtained on the other techniques used (RD5). That means that this exactly image augmentation combination is the best for this specific data set since it is the one which obtains the best results on all three classifiers. Moreover, XGBoost performs better when no resampling methods are used. This is because this classifier has some parameters to deal with class imbalance. The best HS recall and F1 Score are of 1 for both of the metrics, meaning that the 13 HS of the validation set were predicted correctly and no other events were missclassified as HS either. In Fig. 24 (b), the confusion matrix of the experiment with the best results can be seen. The results obtained using XGBoost were the best that could be obtained with the given database and initial conditions.



(a) SVM confusion matrix



(b) XGBoost confusion matrix

Figure 24: SVM and XGBoost confusion matrices of the experiments with best results. In each square the percentage of ground truth instances predicted correctly of each class is shown as well as the number of instances in brackets.

The best results obtained are in the following table. Nevertheless, in Appendix 4, 5, 6 and 7 all the results in full detail can be seen.

	Precision			Recall			F1 Score				
	NHS	HS	AN	NHS	HS	AN	NHS	HS	AN	M	W
SVM	0.98	1.00	0.79	0.99	**0.62**	0.92	0.98	**0.76**	0.85	0.86	0.97
XGBoost	0.98	1.00	0.88	1.00	**1.00**	0.58	0.99	**1.00**	0.70	0.90	0.98

Table 3: Best classification results.

7.3 System-wide tests

At the time when both the feature extractor and the classifier were fully trained and tested, some tests of the two blocks concatenated were made. 4 unseen sequences and without labels were taken to test how the system works all together. The new sequences were passed through the detector of bright events, the tracker, the feature extractor and the classifier. Then, with the help of A. Puig Sitjes, a computer vision and machine learning engineer of the IPP, the predictions were reviewed to know the system performance. In one of these 4 sequences one of the hot spots was from the Neutral Beam Injection System as can be seen in Fig. 25

The HS recall is 1, all the hot spots that occur were correctly classified, but the HS F1 Score is 0.82 due to some reflections were missclassified as HS. In Fig. 26 the confusion matrix of the results of all the sequences can be seen.

So, taking the results into account the system is a good tool to help label the sequences, one of the objectives of this thesis. Instead of having to look carefully at the whole sequence

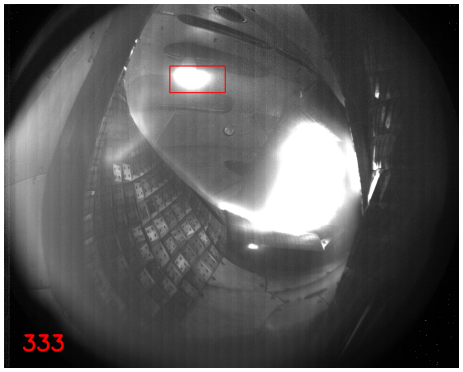


Figure 25: Hot spot from one of the system-wide test sequences because of the Neutral Beam Injection.

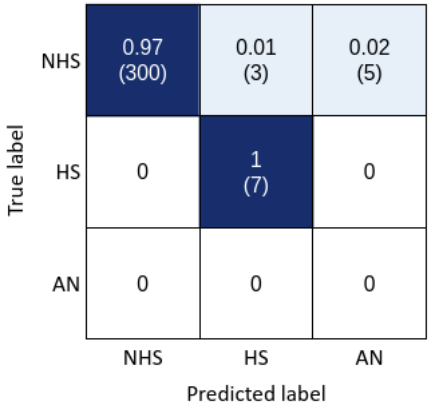


Figure 26: Confusion matrix of the system-wide test sequences. In each square the percentage of ground truth instances predicted correctly of each class is shown as well as the number of instances in brackets.

to detect bright events and decide whether they are HS or not, the system can be used and then review the results to go faster and have the same outcome.

7.4 Online classification

So far all the tests and experiments have been done in a forensic or offline way, which means that the entire sequence or track has to happen to be able to predict its class. This is good for analysing the sequences and knowing where the most dangerous zones are and helping to label new sequences for future use. However, being able to know when the reactor walls are overheating in real time could save a lot of resources by shutting down the reactor before reaching excessively high temperatures. That is why the system has been modified to be able to predict in real time the bright events occurring in the reactor. In this section how it is done and the experiments and results obtained are explained.

In the system explained so far the tracks were subsampled to n images (n=5), and as the models have been trained this way, the online classifier uses the same number of images per track. But rather than waiting to the end of the track before predicting its class, the system now starts analysing the tracks from the nth image onwards. The online system makes a new prediction for each new detection corresponding to the track in question. In other words, this system makes as many predictions as there are frames in the track minus n. The online model takes 5 images of the track equispaced from all the images detected at that time. If a bright event has been detected on 5 consecutive frames, the system takes all the detections, but if the track is extended up to 40 frames the system would take the frames 1, 10, 20, 30, 40. That is to say that, in each frame, all the existing tracks are analysed as if it was the last shot of the sequence.

The online system has been tested with the 261 NHS tracks and 13 HS tracks of the validation set and all the bright events found on a sequence newly labeled obtaining a

data set composed of 320 NHS and 14 HS with different track lengths. It is important to correctly predict, but also to do it as early as possible.

As the NHS class is the majority one and when the bright events start appearing are not very intense, the classifier always starts predicting the classes as NHS. 5 out of the 320 NHS were in some point of the track missclassified as HS and 2 out of the 320 NHS were poorly predicted as AN, as can be seen in Fig. 27. All the HS were correctly classified in some point of the tracks. Moreover, the 14 HS were predicted with an average time of 1.38 seconds and with an average of 62.03% of the track Fig. 28.

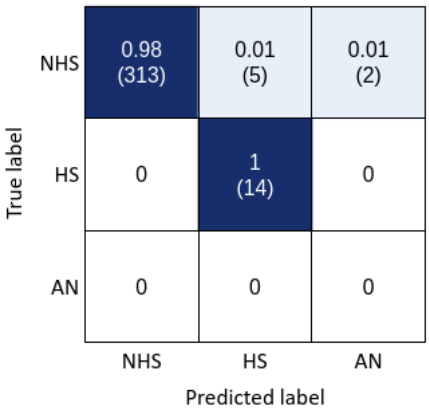


Figure 27: Confusion matrix of the online system. In each square the percentage of ground truth instances predicted correctly of each class is shown as well as the number of instances in brackets.

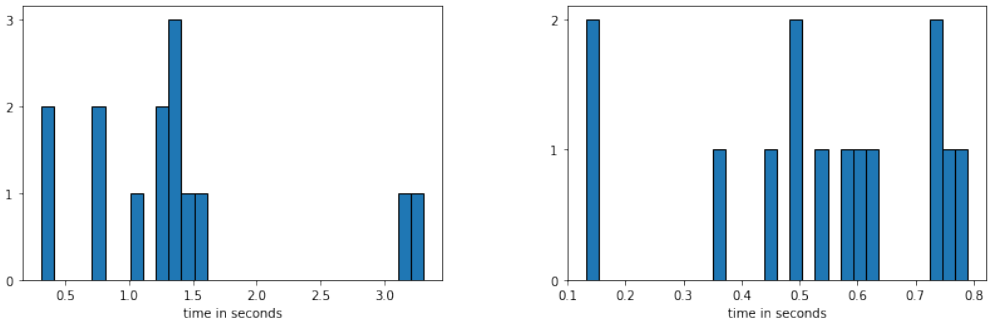


Figure 28: Histograms of the online predictions of the HS. In (a) the time is evaluated in seconds, while in (b) the time is evaluated by the percentage of track has already passed.

The IPP researchers expect to detect the HS in half a second (50 frames as the EDICAM works at 100 frames per second). So, the online system do not work fast enough as only the 7% of the tested HS tracks were predicted under that time.

8 Conclusions

This thesis has reported several studies about different strategies to classify bright events occurring inside the reactor. The lack of researching about thermal events detection with visible range cameras or EDICAMs has lead to experiment with some of the best and most recognised computer vision and Deep Learning techniques. Since the beginning of the project a lot of knowledge in state-of-the-art computer vision algorithms has been gathered to try to achieve an steady-state operation in W7-X using the EDICAM-based system.

Firstly, although it is normal for there to be many more reflections and not harmful bright events than hot spots, the data base is rather small. So this has led to the need to study some techniques to improve this problem.

Therefore, taking into account the initial conditions on which this thesis is based on, the obtained results and the project itself have been extremely successful. The ResNet50 has proven how good it is at dealing with images even though the images from W7-X are very different from the ones the pre-trained model has been trained on. Both ML classifiers have obtained extrmely good results and it has been demonstrated how well the classifiers perform concatenating them after a NN.

In terms of results, the outcome has been excellent. The XGBoost has performed as well as possible obtaining a HS F1 Score of 1, proving why it is one of the most widely used. Despite the fact that the results are extremely good, it is necessary to take into account the small dimensions of the data base and therefore of the validation set. If the data base was better and bigger the results would be even stronger since the models could be trained with more sequences and examples. The results obtained in this thesis are much better than the ones obtained in the first detector by M. Cobos as can be seen comparing the F1-Score-weighted.

The forensic system works extremely well, however, the online system is quite slow and could not be used to detect the hot spots under the required time. That is why, the detection speed would need to be increased a bit.

All things considered, the experiments have shown that the feature extractor as well as the classifiers were on the right way and hopefully it helps to define the way to follow in the forthcoming investigations.

9 Future Work

In this thesis a feature extractor using a NN and a classifier has been done to detect and classify bright events in W7-X. Following this point, there is still work to do, such as implement other NNs and other classifiers. Other Deep Learning techniques could be used such as object detection, object recognition techniques and recurrent networks (Faster R-CNN, YOLO, DetectoRS...). This models could even be used alongside the system done in this thesis.

A good starting point for future research would be enlarging and improving the actual data base to obtain better results focusing only on the hot spots instead of using the AN class since as it contains many different types of events it is very difficult to predict.

In addition, it would be interesting to change the approach of this thesis and focus more on the online classification instead the forensic one. This is not an easy problem, because with real-time information it is more complicated to predict the events.

References

[1] Guru. Wendelstein 7-x. `https://www.ipp.mpg.de/w7x`.

[2] S. Zoletnik et al. EDICAM (Event Detection Intelligent Camera). *Fusion Engineering and Design*, 88(6):1405–1408, 2013.

[3] G. Kocsis et al. Overview video diagnostics for the w7-x stellarator. *Fusion Engineering and Design*, 96-97:808–811, 2015.

[4] S. Zoletnik et al. First results of the multi-purpose real-time processing video camera system on the Wendelstein 7-X stellarator and implications for future devices. *Review of Scientific Instruments*, 89, 2018.

[5] M. Cobos. Anomalies detection in the visible spectrum of plasma physics at wendelstein 7-x. *Master in Computer Vision*, 2020.

[6] Ascasíbar E., editor. *Wendelstein 7-X in the European Roadmap to FusionElectricity*, Nara, Japan, 5 2015. EUROfusion.

[7] A. Ali et al. Initial results from the hotspot detection scheme for protection of plasma facing components in wendelstein 7-x. *Nuclear Materials and Energy*, 19:335–339, 2019.

[8] A. Puig Sitjes et al. Observation of thermal events on the plasma facing components of wendelstein 7-x. *Journal of Instrumentation*, 14:C11002–C11002, 2019.

[9] T. Szepesi et al. Combining research with safety: Performance of the Wendelstein 7-X video diagnostic system. *Fusion Engineering and Design*, 146:874–877, 2019.

[10] A. Puig Sitjes et al. Wendelstein 7-x near real-time image diagnostic system for plasma-facing components protection. *Fusion Science and Technology*, 74(1-2):116–124, 2018.

[11] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66, 1979.

[12] G. Kumar and B. Pradeep Kumar. A Detailed Review of Feature Extraction in Image Processing Systems). In *2014 Fourth International Conference on Advanced Computing Communication Technologies*, pages 5–12, 2014.

[13] N. Tajbakhsh et al. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016.

[14] S.Notley and M. Magdon-Ismail. Examining the Use of Neural Networks for Feature Extraction: A Comparative Analysis using Deep Learning, Support Vector Machines, and K-Nearest Neighbor Classifiers, 2018.

[15] L. Shen et al. Relay Backpropagation for Effective Learning of Deep Convolutional Neural Networks. *Computer Vision – ECCV 2016*, 9911, 2016.

[16] G. Krishna. Back propagation neural network: What is backpropagation algorithm in machine learning? `https://www.guru99.com/backpropogation-neural-network.html`.

[17] T. A. Mohammed S. Albawi and S. Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[18] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[19] Boris Hanin. Which Neural Net Architectures Give Rise To Exploding and Vanishing Gradients?, 2018.

[20] H. Kaiming et al. Deep Residual Learning for Image Recognition, 2015.

[21] ImageNet Large Scale Visual Recognition Challenge 2015 (ILSVRC2015), 2015.

[22] L. Tianyi et. al. Towards Understanding the Importance of Shortcut Connections in Residual Networks, 2019.

[23] C. Shorten and T. M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6, 2019.

[24] A. Agrawal et al. SCUT: Multi-class imbalanced data classification using SMOTE and cluster-based undersampling. In *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, volume 1, pages 226–234, 2015.

[25] N. Chawla et al. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, Jun 2002.

[26] H. He et al. Resampling methods. In *Imbalanced Learning: Foundations, Algorithms, and Applications*, volume 1, pages 40–59, 2013.

[27] H. Han et al. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887, 2005.

[28] H. Haibo et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[29] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421, 1972.

[30] Zeng et al. M. Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data. In *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, pages 225–228, 2016.

[31] I. Tomek. Two Modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, 1976.

[32] V. Vapnik and C. Cortes. Support-vector networks. *Mach Learn 20*, pages 273–297, 1995.

[33] L. Chen. Support vector machine — simply explained. `https://towardsdatascience.com/`.

[34] G. Gundersen. Implicit lifting and the kernel trick. `https://gregorygundersen.com/`.

[35] C. Tianqi and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794, 2016.

[36] L. Breiman. Bagging predictors. *Mach. Learn.*, 24:123–140, 1996.

[37] L. Breiman. Random Forests. *Mach. Learn.*, 45(1):5–32, 2001.

[38] K. Guolin et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[39] T. Chen. XGBoost: Overview and Latest News. LA Meetup Talk, 2016.

[40] V. Morde. XGBoost Algorithm: Long May She Reign! *Towards data science*, 2019.

[41] T. Hastie et al. Model assessment and selection. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, chapter 7, pages 219–260. Springer, 2004.

[42] N. S. Keskar et al. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima, 2017.

[43] P. T. de Boer et al. A Tutorial on the Cross-Entropy Method. *Ann Oper Res*, 134:19–67, 2005.

10 Appendices

10.1 Appendix 1

	Frame level			Track level		
	NHS	HS	AN	NHS	HS	AN
(1) 20171207.039_AEQ11	5473	0	191	49	0	3
(2) 20171207.043_AEQ11	7887	0	279	74	0	3
(3) 20180918.036_AEQ50	8592	673	877	56	3	8
(4) 20180918.038_AEQ50	7325	433	829	62	2	0
(5) 20180918.040_AEQ50	6702	255	398	83	1	4
(6) 20180919.007_AEQ40	7302	0	0	63	0	0
(7) 20180920.034_AEQ11	8663	2020	1110	61	5	4
(8) 20181002.028_AEQ20	15152	52	53	43	1	1
(9) 20181004.038_AEQ10	14574	2296	0	73	5	0
(10) 20181004.038_AEQ20	25491	2481	0	64	5	0
(11) 20181004.038_AEQ40	14899	1161	785	72	3	4
(12) 20181004.046_AEQ20	21141	2495	0	60	6	0
(13) 20181004.046_AEQ40	11361	1273	1331	55	4	3
(14) 20181004.046_AEQ50	15868	1733	563	58	6	1
Total	170430	14872	6416	873	41	37

Table 4: Total number of detections divided in its labels and its sequences: NHS, HS and AN.

10.2 Appendix 2

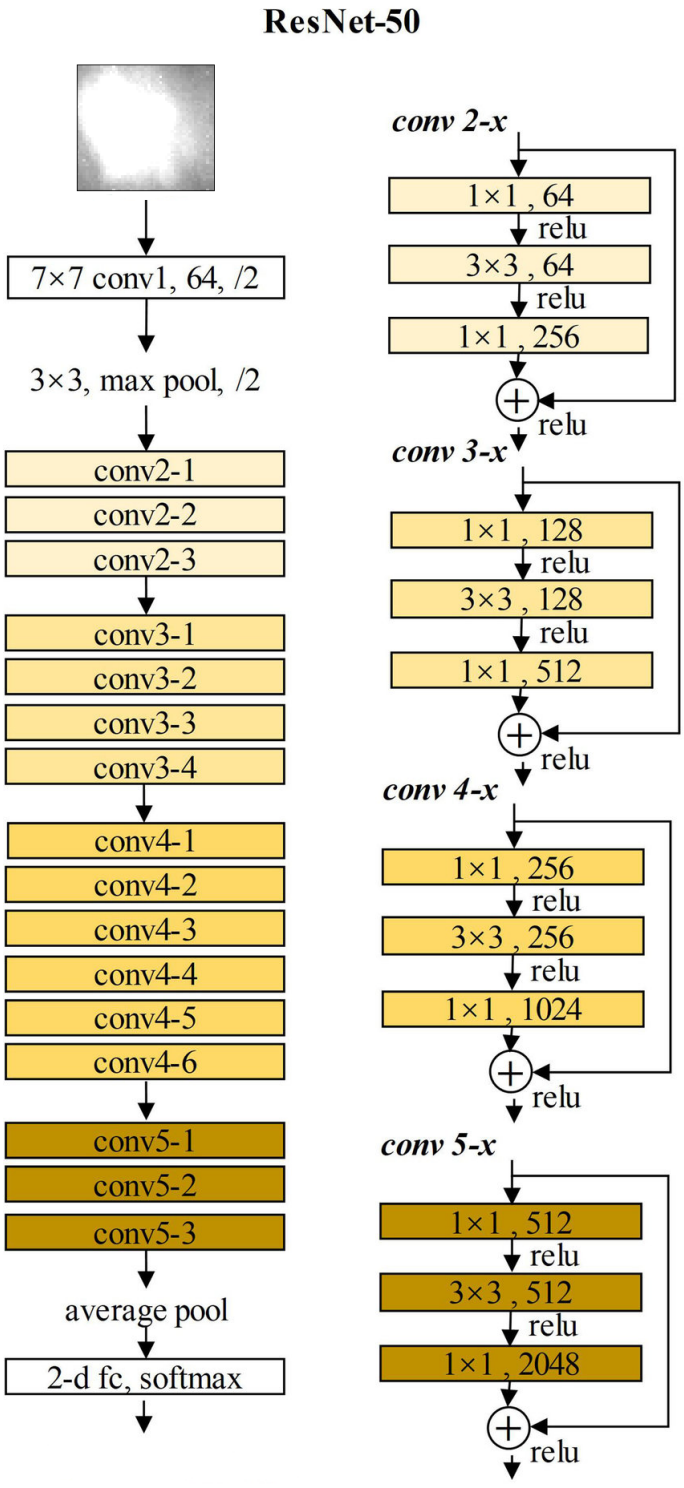


Figure 29: Structure of the ResNet50.

10.3 Appendix 3

Softmax	Precision			Recall			F1				
Img. Aug.	NHS	HS	AN	NHS	HS	AN	NHS	HS	AN	M	W
-	0,96	0,55	0,33	0,94	**0,67**	0,38	0,95	0,60	0,35	0,63	0,91
RD5	0.95	0.86	0.17	0.94	**0.67**	0.25	0.94	**0.75**	0.20	0.63	0.90
RR5	0.95	0.60	0.40	0.97	**0.67**	0.25	0.96	0.63	0.31	0.63	0.92
RD5 RR5	0,96	0,35	0,43	0,92	**0,67**	0,38	0,94	0,46	0,40	0,60	0,89
RD10 RR10	0,95	0,83	0,22	0,96	0,56	0,25	0,95	0,67	0,24	0,62	0,91
RD% RR5	0,96	0,55	0,33	0,94	**0,67**	0,38	0,95	0,60	0,35	0,63	0,91

Table 5: Results of the NN used as classifier. The numbers in bold are from the models which performed better in the important metrics to optimise.

10.4 Appendix 4

SVM averaging the features:

		Precision			Recall			F1				
Img. Aug.	Resampling T.	NHS	HS	AN	NHS	HS	AN	NHS	HS	AN	M	W
-	SMOTE	0,97	1,00	0,82	0,99	0,62	0,75	0,98	0,76	0,78	0,84	0,96
	B. SMOTE	0,98	1,00	0,83	0,99	0,77	0,83	0,99	0,87	0,83	0,90	0,97
	ADASYN	0,98	1,00	1,00	1,00	0,77	0,83	0,99	0,87	0,91	0,92	0,98
	SMOTEENN	0,98	1,00	0,55	0,96	0,62	1,00	0,97	0,76	0,71	0,81	0,95
	SMOTETomek	0,97	1,00	0,82	0,99	0,62	0,75	0,98	0,76	0,78	0,84	0,96
RD5	SMOTE	0,98	1,00	0,79	0,99	0,62	0,92	0,98	0,76	0,85	0,86	0,97
	B. SMOTE	0,98	1,00	0,83	0,99	**0,85**	0,83	0,99	**0,92**	0,83	0,91	0,98
	ADASYN	0,97	1,00	0,75	0,99	0,62	0,75	0,98	0,76	0,75	0,83	0,96
	SMOTEENN	0,98	0,77	0,65	0,97	0,77	0,92	0,97	0,77	0,76	0,83	0,96
	SMOTETomek	0,98	1,00	0,79	0,99	0,62	0,92	0,98	0,76	0,85	0,86	0,97
RR5	SMOTE	0,96	1,00	0,88	1,00	0,46	0,58	0,98	0,63	0,70	0,77	0,95
	B. SMOTE	0,97	1,00	0,80	0,99	0,69	0,67	0,98	0,82	0,73	0,84	0,96
	ADASYN	0,97	0,83	0,89	0,99	0,77	0,67	0,98	0,80	0,76	0,85	0,96
	SMOTEENN	0,97	1,00	0,77	0,99	0,54	0,83	0,98	0,70	0,80	0,83	0,96
	SMOTETomek	0,96	1,00	0,88	1,00	0,46	0,58	0,98	0,63	0,70	0,77	0,95
RD5 RR5	SMOTE	0,95	1,00	0,80	1,00	0,46	0,33	0,97	0,63	0,47	0,69	0,93
	B. SMOTE	0,97	1,00	0,86	1,00	0,77	0,50	0,98	0,87	0,63	0,83	0,96
	ADASYN	0,95	1,00	1,00	1,00	0,46	0,33	0,97	0,63	0,50	0,70	0,94
	SMOTEENN	0,96	0,88	0,54	0,97	0,54	0,58	0,96	0,67	0,54	0,72	0,93
	SMOTETomek	0,95	1,00	0,80	1,00	0,46	0,33	0,97	0,63	0,47	0,69	0,93
RD10 RR10	SMOTE	0,97	1,00	0,85	1,00	0,46	0,92	0,98	0,63	0,88	0,83	0,96
	B. SMOTE	0,98	0,89	0,85	0,99	0,62	0,92	0,98	0,73	0,88	0,86	0,97
	ADASYN	0,97	0,75	0,85	0,99	0,46	0,92	0,98	0,57	0,88	0,81	0,96
	SMOTEENN	0,98	0,78	0,60	0,97	0,54	1,00	0,97	0,64	0,75	0,79	0,95
	SMOTETomek	0,97	1,00	0,85	1,00	0,46	0,92	0,98	0,63	0,88	0,83	0,96
RR% RD5	SMOTE	0,97	0,86	0,91	0,99	0,46	0,83	0,98	0,60	0,83	0,82	0,96
	B. SMOTE	0,97	0,90	0,90	0,99	0,69	0,75	0,98	0,78	0,82	0,86	0,97
	ADASYN	0,98	0,91	0,90	0,99	0,77	0,75	0,98	0,83	0,82	0,88	0,97
	SMOTEENN	0,98	0,82	0,58	0,96	0,69	0,92	0,97	0,75	0,71	0,81	0,95
	SMOTETomek	0,97	0,86	0,91	0,99	0,46	0,83	0,98	0,60	0,87	0,82	0,96

Table 6: Results of the SVM classifier averaging the features. The numbers in bold are from the models which performed better in the important metrics to optimise.

10.5 Appendix 5

SVM concatenating the features:

		Precision			Recall			F1				
Img. Aug.	Resampling T.	NHS	HS	AN	NHS	HS	AN	NHS	HS	AN	M	W
-	SMOTE	0,93	1,00	1,00	1,00	0,23	0,25	0,96	0,38	0,40	0,58	0,91
	B. SMOTE	0,98	1,00	1,00	1,00	0,77	0,75	0,99	0,87	0,86	0,91	0,98
	ADASYN	0,97	1,00	1,00	1,00	0,69	0,67	0,98	0,82	0,80	0,87	0,97
	SMOTEENN	0,94	1,00	0,75	1,00	0,31	0,25	0,96	0,47	0,38	0,60	0,92
	SMOTETomek	0,93	1,00	1,00	1,00	0,23	0,25	0,96	0,38	0,40	0,58	0,91
RD5	SMOTE	0,92	0,00	1,00	1,00	0,00	0,25	0,96	0,00	0,40	0,45	0,89
	B. SMOTE	0,97	1,00	1,00	1,00	0,69	0,75	0,99	0,82	0,86	0,89	0,97
	ADASYN	0,98	1,00	1,00	1,00	0,69	0,83	0,99	0,82	0,91	0,91	0,98
	SMOTEENN	0,93	0,00	0,83	1,00	0,00	0,42	0,96	0,00	0,56	0,51	0,90
	SMOTETomek	0,92	0,00	1,00	1,00	0,00	0,25	0,96	0,00	0,40	0,45	0,89
RR5	SMOTE	0,92	1,00	0,50	1,00	0,15	0,08	0,96	0,27	0,14	0,46	0,89
	B. SMOTE	0,92	1,00	1,00	1,00	0,15	0,08	0,96	0,27	0,15	0,46	0,89
	ADASYN	0,96	1,00	1,00	1,00	0,62	0,42	0,98	0,76	0,59	0,78	0,95
	SMOTEENN	0,94	1,00	0,50	0,99	0,38	0,17	0,96	0,56	0,25	0,59	0,91
	SMOTETomek	0,92	1,00	0,50	1,00	0,15	0,08	0,96	0,27	0,14	0,46	0,89
RD5 RR5	SMOTE	0,93	1,00	1,00	1,00	0,23	0,08	0,96	0,38	0,15	0,50	0,90
	B. SMOTE	0,92	1,00	1,00	1,00	0,08	0,08	0,96	0,14	0,15	0,42	0,89
	ADASYN	0,95	1,00	1,00	1,00	0,69	0,25	0,98	0,82	0,40	0,73	0,94
	SMOTEENN	0,93	1,00	0,50	0,99	0,23	0,17	0,96	0,38	0,25	0,53	0,90
	SMOTETomek	0,93	1,00	1,00	1,00	0,23	0,08	0,96	0,38	0,15	0,50	0,90
RD10 RR10	SMOTE	0,94	1,00	1,00	1,00	0,15	0,42	0,97	0,27	0,59	0,61	0,92
	B. SMOTE	0,99	1,00	0,86	1,00	0,69	1,00	0,99	0,82	0,92	0,91	0,98
	ADASYN	0,97	1,00	1,00	1,00	0,62	0,83	0,99	0,76	0,91	0,89	0,97
	SMOTEENN	0,94	1,00	1,00	1,00	0,15	0,50	0,97	0,27	0,67	0,63	0,92
	SMOTETomek	0,94	1,00	1,00	1,00	0,15	0,42	0,97	0,27	0,59	0,61	0,92
RR% RD5	SMOTE	0,92	1,00	1,00	1,00	0,15	0,08	0,96	0,27	0,15	0,46	0,89
	B. SMOTE	0,97	1,00	1,00	1,00	0,69	0,75	0,99	0,82	0,86	0,89	0,97
	ADASYN	0,97	0,90	1,00	1,00	0,69	0,67	0,98	0,78	0,80	0,86	0,97
	SMOTEENN	0,93	1,00	1,00	1,00	0,23	0,17	0,96	0,38	0,29	0,54	0,91
	SMOTETomek	0,92	1,00	1,00	1,00	0,15	0,08	0,96	0,27	0,15	0,46	0,89

Table 7: Results of the SVM classifier concatenating the features. The numbers in bold are from the models which performed better in the important metrics to optimise. As all the results were better averaging the features there are not results in bold.

10.6 Appendix 6

XGBoost averaging the features:

		Precision			Recall			F1				
Img. Aug.	Resampling T.	NHS	HS	AN	NHS	HS	AN	NHS	HS	AN	M	W
-	-	0,98	0,86	0,78	0,98	0,92	0,58	0,98	0,89	0,67	0,85	0,96
	SMOTE	0,99	0,81	0,53	0,96	**1,00**	0,75	0,97	0,90	0,62	0,83	0,95
	B. SMOTE	0,98	0,87	0,73	0,98	**1,00**	0,67	0,98	0,93	0,70	0,87	0,97
	ADASYN	0,99	0,81	0,42	0,93	**1,00**	0,83	0,96	0,90	0,56	0,80	0,94
	SMOTEENN	0,99	0,76	0,38	0,92	**1,00**	0,83	0,96	0,87	0,53	0,78	0,93
	SMOTETomek	0,99	0,81	0,53	0,96	**1,00**	0,75	0,97	0,90	0,62	0,83	0,95
RD5	-	0,98	0,93	0,67	0,98	**1,00**	0,50	0,98	0,96	0,57	0,84	0,96
	SMOTE	0,98	0,65	0,47	0,95	**1,00**	0,58	0,96	0,79	0,52	0,76	0,94
	B. SMOTE	0,98	0,81	0,50	0,97	**1,00**	0,58	0,97	0,90	0,54	0,80	0,95
	ADASYN	0,99	0,68	0,45	0,93	**1,00**	0,75	0,96	0,81	0,56	0,78	0,94
	SMOTEENN	0,99	0,57	0,42	0,92	**1,00**	0,67	0,95	0,72	0,52	0,73	0,93
	SMOTETomek	0,98	0,65	0,47	0,95	**1,00**	0,58	0,96	0,79	0,52	0,76	0,94
RR5	-	0,97	0,86	0,57	0,98	0,92	0,33	0,97	0,89	0,42	0,76	0,95
	SMOTE	0,98	0,76	0,47	0,95	**1,00**	0,58	0,97	0,87	0,52	0,78	0,94
	B. SMOTE	0,98	0,87	0,46	0,97	**1,00**	0,50	0,97	0,93	0,48	0,79	0,95
	ADASYN	0,98	0,72	0,44	0,94	**1,00**	0,67	0,96	0,84	0,53	0,78	0,94
	SMOTEENN	0,99	0,72	0,31	0,90	**1,00**	0,75	0,94	0,84	0,44	0,74	0,92
	SMOTETomek	0,98	0,76	0,47	0,95	**1,00**	0,58	0,97	0,87	0,52	0,78	0,94
RD5 RD5	-	0,96	0,86	0,60	0,98	0,92	0,25	0,97	0,89	0,35	0,74	0,94
	SMOTE	0,98	0,68	0,38	0,93	**1,00**	0,67	0,95	0,81	0,48	0,75	0,93
	B. SMOTE	0,98	0,67	0,50	0,95	0,92	0,58	0,96	0,77	0,54	0,76	0,94
	ADASYN	0,98	0,62	0,33	0,91	**1,00**	0,67	0,94	0,76	0,44	0,72	0,92
	SMOTEENN	0,99	0,62	0,28	0,89	**1,00**	0,67	0,94	0,76	0,39	0,70	0,91
	SMOTETomek	0,98	0,68	0,38	0,93	**1,00**	0,67	0,95	0,81	0,48	0,75	0,93
RD10 RR10	-	0,97	0,93	0,71	0,99	**1,00**	0,42	0,98	0,96	0,53	0,82	0,96
	SMOTE	0,98	0,87	0,44	0,95	**1,00**	0,67	0,97	0,93	0,53	0,81	0,95
	B. SMOTE	0,98	0,86	0,47	0,96	0,92	0,67	0,97	0,89	0,55	0,80	0,95
	ADASYN	0,98	0,72	0,39	0,94	**1,00**	0,58	0,96	0,84	0,47	0,75	0,93
	SMOTEENN	0,93	0,76	0,35	0,95	**1,00**	0,50	0,96	0,87	0,41	0,75	0,94
	SMOTETomek	0,98	0,87	0,44	0,95	**1,00**	0,67	0,97	0,93	0,53	0,81	0,95
RD% RR5	-	0,98	0,86	0,78	0,99	0,92	0,58	0,98	0,89	0,67	0,85	0,97
	SMOTE	0,99	0,76	0,47	0,95	**1,00**	0,67	0,97	0,87	0,55	0,80	0,95
	B. SMOTE	0,98	0,86	0,78	0,99	0,92	0,58	0,98	0,89	0,67	0,85	0,97
	ADASYN	0,99	0,68	0,45	0,94	**1,00**	0,75	0,96	0,81	0,56	0,78	0,94
	SMOTEENN	1,00	0,76	0,40	0,93	**1,00**	0,83	0,96	0,87	0,54	0,79	0,94
	SMOTETomek	0,99	0,76	0,47	0,95	**1,00**	0,67	0,97	0,87	0,55	0,80	0,95

Table 8: Results of the XGBoost classifier averaging the features. The numbers in bold are from the models which performed better in the important metrics to optimise.

10.7 Appendix 7

XGBoost concatenating the features:

		Precision			Recall			F1				
Img. Aug.	Resampling T.	NHS	HS	AN	NHS	HS	AN	NHS	HS	AN	M	W
-	-	0,99	0,86	0,88	0,99	0,92	0,58	0,98	0,89	0,70	0,86	0,97
	SMOTE	0,99	0,87	0,71	0,98	**1,00**	0,83	0,98	0,93	0,77	0,89	0,97
	B. SMOTE	0,98	0,86	0,89	0,99	0,92	0,67	0,98	0,89	0,76	0,88	0,97
	ADASYN	0,98	0,76	0,57	0,96	**1,00**	0,67	0,97	0,87	0,62	0,82	0,95
	SMOTEENN	0,99	0,72	0,53	0,95	**1,00**	0,83	0,97	0,84	0,65	0,82	0,95
	SMOTETomek	0,99	0,87	0,71	0,98	**1,00**	0,83	0,98	0,93	0,77	0,89	0,97
RD5	-	0,98	1,00	0,88	1,00	**1,00**	0,58	0,99	**1,00**	0,70	0,90	0,98
	SMOTE	1,00	0,81	0,55	0,95	**1,00**	0,92	0,97	0,90	0,69	0,85	0,96
	B. SMOTE	0,99	0,87	0,80	0,99	**1,00**	0,67	0,99	0,93	0,73	0,88	0,97
	ADASYN	0,98	0,67	0,64	0,96	0,92	0,75	0,97	0,77	0,69	0,81	0,95
	SMOTEENN	0,99	0,68	0,50	0,94	**1,00**	0,83	0,96	0,81	0,62	0,80	0,94
	SMOTETomek	1,00	0,81	0,55	0,95	**1,00**	0,92	0,97	0,90	0,69	0,85	0,96
RR5	-	0,97	0,79	0,71	0,98	0,85	0,42	0,97	0,81	0,53	0,77	0,95
	SMOTE	0,99	0,76	0,60	0,96	**1,00**	0,75	0,97	0,87	0,67	0,84	0,96
	B. SMOTE	0,98	0,81	0,64	0,97	**1,00**	0,58	0,98	0,90	0,61	0,83	0,96
	ADASYN	0,98	0,76	0,62	0,97	**1,00**	0,67	0,97	0,87	0,64	0,83	0,96
	SMOTEENN	0,99	0,68	0,40	0,93	**1,00**	0,67	0,96	0,81	0,50	0,76	0,93
	SMOTETomek	0,99	0,76	0,60	0,96	**1,00**	0,75	0,97	0,87	0,67	0,84	0,96
RD5 RD5	-	0,97	0,86	0,67	0,98	0,92	0,33	0,98	0,89	0,44	0,77	0,95
	SMOTE	0,98	0,65	0,50	0,95	**1,00**	0,58	0,97	0,79	0,54	0,76	0,94
	B. SMOTE	0,98	0,87	0,64	0,98	**1,00**	0,58	0,98	0,93	0,61	0,84	0,96
	ADASYN	0,99	0,62	0,50	0,94	**1,00**	0,67	0,96	0,76	0,57	0,77	0,94
	SMOTEENN	0,99	0,62	0,38	0,92	**1,00**	0,67	0,95	0,76	0,48	0,73	0,93
	SMOTETomek	0,98	0,65	0,5	0,95	**1,00**	0,58	0,97	0,79	0,54	0,76	0,94
RD10 RR10	-	0,98	0,92	1,00	1,00	0,92	0,58	0,99	0,92	0,74	0,88	0,97
	SMOTE	0,99	0,72	0,69	0,97	**1,00**	0,75	0,98	0,84	0,72	0,85	0,96
	B. SMOTE	0,98	0,86	0,75	0,98	0,92	0,75	0,98	0,89	0,75	0,87	0,97
	ADASYN	0,99	0,72	0,62	0,97	**1,00**	0,67	0,98	0,84	0,64	0,82	0,96
	SMOTEENN	0,99	0,68	0,53	0,94	**1,00**	0,83	0,97	0,81	0,65	0,81	0,95
	SMOTETomek	0,99	0,72	0,69	0,97	**1,00**	0,75	0,98	0,84	0,72	0,85	0,96
RD% RR5	-	0,98	1,00	0,90	1,00	0,92	0,75	0,99	0,96	0,82	0,92	0,98
	SMOTE	1,00	0,81	0,62	0,97	**1,00**	0,83	0,98	0,90	0,71	0,86	0,97
	B. SMOTE	0,98	1,00	0,82	0,99	0,92	0,75	0,99	0,96	0,78	0,91	0,98
	ADASYN	0,99	0,76	0,69	0,97	**1,00**	0,75	0,98	0,87	0,72	0,86	0,97
	SMOTEENN	0,99	0,76	0,56	0,96	**1,00**	0,75	0,98	0,87	0,64	0,83	0,96
	SMOTETomek	1,00	0,81	0,62	0,97	**1,00**	0,83	0,98	0,90	0,71	0,86	0,97

Table 9: Results of the XGBoost classifier concatenating features. The numbers in bold are from the models which performed better in the important metrics to optimise.