Recurrent convolutional neural network for coronal jets identification

Julie Charlet, Carlos Collado Capell, and Adrien Joliat
École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland - CS-433
European Space Agency (ESA), Collaboration with Sophie Musset

***Abstract*—This report presents a study on the development and application of a Recurrent Convolutional Neural Network (RCNN) for identifying solar coronal jets using data from the Solar Dynamic Observatory (SDO). The study focuses on the challenges of processing extensive solar image sequences and proposes an innovative machine learning solution to automate the classification of solar jets. While prior algorithms struggled to detect solar those structures, our RCNN effectively demonstrates a significant advancement in solar jet detection.**

***Keywords*: Solar Jets, Recurrent Convolutional Network, Computer Vision, Solar Physics.**

I. Introduction

In recent years, there has been an exponential increase in the volume of data available for research in solar physics. This surge is largely due to the launch of the Solar Dynamic Observatory (SDO) in 2010, equipped with the Advanced Imaging Assembly (AIA) instrument. The SDO/AIA enables the acquisition of full-disk images of the Sun with a temporal resolution of 12 seconds [1], resulting in an immense data stream.

In the field of coronal jets, creating a consistent and reliable database for these events is crucial for advancing research in various solar phenomena. These events were being manually reported in the Heliophysics Events Knowledgebase (HEK) [2]. However, given the overwhelming volume of data, manual inspection by experts is neither feasible nor efficient.

A citizen initiative, Solar Jet Hunter was launched in 2021 in the platform Zooniverse [3]. In this project, volunteers were shown a series of movie strips of different regions of the Sun, with the goal of identifying solar jets. This approach, requiring minimal training for participants, significantly scaled up the analysis of solar data. By September 2023, approximately 21% of the data from 2011 to 2016 had been processed, leading to the identification of 883 solar jets by volunteers. However, a substantial portion of the HEK catalogue still remains unclassified.

While some algorithms have been developed to detect and track some of the features in this data, these have not been very successful at detecting solar jets. This is believed to be due to the fact that jets have various shapes, sizes, duration, and brightness. In particular, when seen on the solar disk, they sometimes are not brighter than the background.

This report investigates the potential of a novel machine learning algorithm designed to predict the presence of coronal jets in sequences of solar images. This algorithm, trained by data obtained from the citizen-driven Solar Jet Hunter initiative, aims to automate and enhance the classification process for the vast array of data within the HEK catalogue.

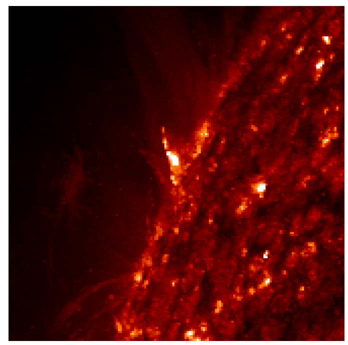

Fig. 1: Example of a jet event in the HEK data.
Click the image for an animated version.

II. Data Collection and Preprocessing

The dataset from the Zooniverse project includes dates and locations of 21% of solar data from 2011 to 2016, with the duration and locations of solar jets. We used the subset of 883 solar jets identified by the volunteers as positive training data (label 1). For the negative data, we used intervals that volunteers had marked as not containing jets.

Coronal jets are easily detectable by humans due to their dynamic motion, but present a challenge for algorithmic detection. To capture the temporal aspect, we analyzed sequences of 30 images, taken every 24 seconds, thus covering 12 minutes. Given that jet durations range from 5 to 182 minutes, not all images in a sequence display a jet, but as long as some does, we want the model to detect it as such.

We employed Sunpy [4] [5], a Python library for solar physics data, to retrieve images from the JSOC server based on specific locations and dates [6]. The images were converted into numpy arrays to use them in the model. Due to the large size of the downloaded data (883 positive and 883 negative sequences, each with 30 images of 500x500 pixels), we implemented compression techniques. These included converting the arrays from float64 to float32 and applying a pooling layer to reduce the image size to 166x166 pixels, preserving a good image quality while reducing the image weight by a factor of 9. The arrays were then saved as NPZ files, using gzip compression.

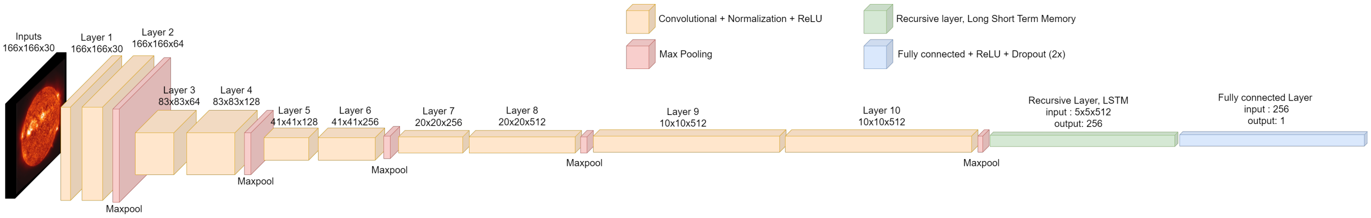

Fig. 2: RCNN Architecture: Convolutional and LSTM layers for binary classification of (166, 166, 30) input tensors.

Some data was unavailable for download, leading to incomplete sequences. Since our RCNN required consistent input sizes, we excluded these incomplete sequences. Based on literature, we opted for recurrent convolutional networks over architectures like transformers, which accommodate varying sequence sizes. The final dataset comprises 866 jet events and 846 non-jet events. We allocated 70% of the data for training, 15% as validation set for hyperparameter tuning, and 15% as a test set for final model validation.

To enhance model robustness and realism, we implemented data augmentation techniques, particularly random rotations and flips using torchvision's transforms module. This also helped to handle the limited amount of data available, and the initial overfitting shown by the in the model. It was crucial to apply the same transformation to every frame within a sequence. This consistency is vital because the primary identifiable characteristic of a jet is its motion, and altering the orientation of images between frames could impede the model's ability to detect it.

Furthermore, we standardized the data. Initial standardization led to some values exceeding 15, inconsistent with a normal distribution. To rectify this, we clipped values to a maximum of 1000 before normalization, effectively reducing the top 1% of the data but ensuring sufficient brightness for jet detection. We also explored logarithmic scaling prior to standardization, treating the data according to a power law. However, we found that clipping yielded better results. This adjustment allowed the data to more closely approximate a normal distribution, enhancing the benefits of standardization in the model.

III. Deep-Learning Framework

A. Model Architecture

In the field of deep learning for sequence image classification of astronomical events, literature shows Recursive Convolutional Neural Networks (RCNN) to be highly effective [7], [8]. The architecture of the model, influenced by the model developed by John A. Armstrong et al., is shown in Figure 2.

The model begins with two convolutional layers, each followed by max pooling using a 2x2 kernel and a stride of 2. This sequence is repeated five times, successively halving the image's height and width with each iteration, while increasing the number of channels. After these transformations, an LSTM (Long Short-Term Memory) recursive layer is employed to extract temporal information from a set of 30 images.

Finally, the architecture includes a fully connected layer with dropout, producing a singular output for binary classification. The output is then processed through a sigmoid layer, converting the values to a range between 0 and 1. This enables the interpretation of the model's output as the probability of the image sequence containing a jet.

B. Hyperparameter Tuning

The selection of hyperparameters such as the batch size, learning rate, epoch count, and optimizers is vital in neural network optimization. Considering the vast array of potential combinations, we identified the most pertinent hyperparameters for our problem and network architecture. A grid search was then conducted on 18 selected combinations to optimize the model. The following section provides a detailed description of these hyperparameters, as well as a summary of them, as presented in Table I.

We also fine-tuned the threshold parameter. Initially, we set the benchmark such that outputs over 0.5, following the application of the sigmoid function, are classified as positive. However, this threshold is adaptable based on specific scientific objectives, which may prioritize different performance metrics. In this work with the European Space Agency, the priority was to minimize false negatives, accepting a higher rate of false positives as a trade-off. Consequently, we modified the threshold to enhance jet detection, aligning with these specific project goals.

1) Batch size: count of data samples used to compute the gradient and update the model's weights at each iteration. An ideal batch size is problem-specific, and typically depends on experimentation. Smaller batch sizes can introduce a form of generalization due to the fact that it update weights more frequently and may converge faster, but are prone to noisier gradient estimates. After evaluating various options, we opted for a notably small batch size of 2.

2) Learning rate (scheduler): magnitude of weight updates during model training, essentially controlling the step size in the gradient descent process. The learning rate is typically set experimentally and often decreases over time (based on a scheduler), starting with a broader search to approach the area of minimum loss, and then slow down to stabilize and refine the training. We tested three schedulers:

(a) *CosineAnnealingLR*: decrease smoothly (cosine curve).
(b) *ExponentialLR*: decrease by a factor γ=0.9 each epoch.
(c) *ReduceLROnPlateau*: adjust when validation loss stops decreasing.

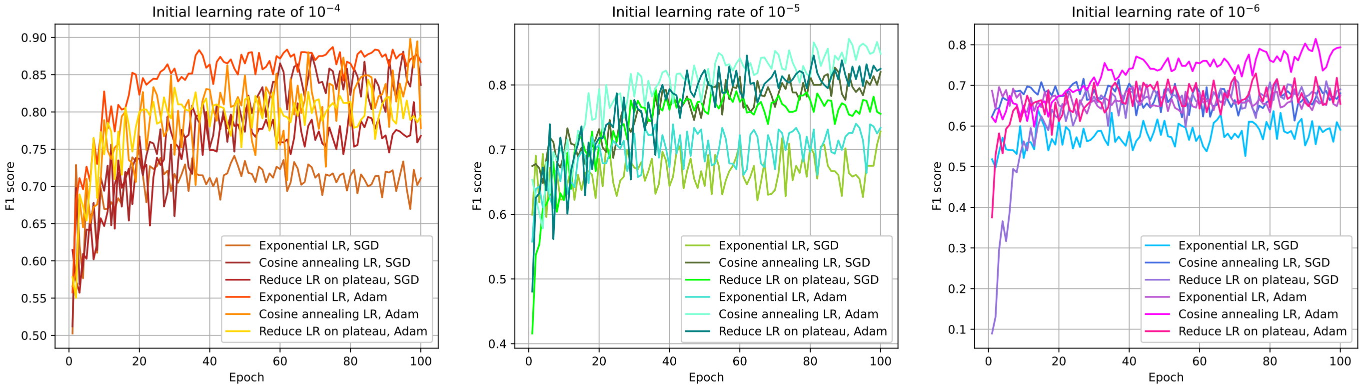

Fig. 3: Cross Validation over the hyperparameters. F1 score evolution on validation data over epochs, each subplot representing a different initial learning rate.

3) Epochs: number of times model's weights are updated by passing through the entire training dataset. The optimal number of epochs depends on various factors, such as the model's complexity, the task's nature, the size and quality of the training data, and the available computational resources. Based on performance under initial runs, we settled on 100 epochs. Although we employed early stopping, it was not triggered during the training process, and the epoch with best validation loss was selected.

4) Optimizer: determines the update of model parameters during training. The simplest is gradient descent, which adjusts weights against the loss gradient. We experimented with two optimizers:

(a) *SGD with momentum*: updates weights using one training point, and momentum (moving average of past gradients) to accelerate convergence and dampen oscillations.

(b) *AdamW*: optimizer most widely used in the literature. Combines momentum, adapting LR, and regularization through weight decay to avoid overfitting.

5) Loss function (criterion): difference between the predicted output and actual target values. Binary Cross-Entropy (BCE) is the most common loss for binary classification tasks, measuring the cross-entropy loss. The criterion used (BCEWithLogitsLoss) combines BCE with a sigmoid activation function (can either be implemented as part of the loss calculation or integrated within the network's architecture). The BCE formula is given by:

$$\mathrm{BCE}(y, p) = -\left(y \cdot \log(p) + (1 - y) \cdot \log(1 - p)\right)$$

y is the true label, and p the probability, output of the sigmoid function (increases numerical stability, especially handling extreme output values).

The overview of all hyperparameters can be seen in the Table I.

Initial LR	10^{-4}	10^{-5}	10^{-6}
Scheduler	Exponential decay	Cosine annealing	Reduce on plateau
Optimizer	SGD with momentum	AdamW	
Criterion	BCE with logits		
Batch size	2		
# epochs	100		

TABLE I: Overview of the hyperparameters

IV. Results and discussion

A. Model Selection

The results of the hyperparameter tuning are presented in Figure 3. The optimal F1 score was achieved using a configuration consisting of an initial learning rate of 10^{-4}, a cosine annealing learning rate scheduler, and the AdamW optimizer. This configuration reached a peak F1 score of 0.9 prior to any threshold adjustment for classification. However, it is noteworthy from Figure 3 that the model configured with an exponential learning rate scheduler demonstrated faster convergence to a comparable F1 score. Moreover, this model exhibited a smoother increase in F1 score over the epochs relative to the model using the cosine annealing learning rate scheduler.

Given these observations, both models were subjected to further analysis to determine the optimal classification threshold. As depicted in Figure 4, the model employing the cosine annealing learning rate scheduler consistently outperformed the exponential scheduler model in terms of F1 score when varying the threshold. In aligning with our aim to ensure a more proactive stance in identifying jet events, we selected a classification threshold that yielded a high F1 score while favoring the identification of positive events. The chosen threshold of 0.1 demonstrated a preference for classifying events as positive, while maintaining a final F1 score of 0.8949 on the validation set.

B. Application of the Model to Unseen Data

To evaluate the generalizability and unbiased performance of our final model, we conducted tests on a previously unseen dataset comprising 260 image sequences, balanced between positive and negative instances.

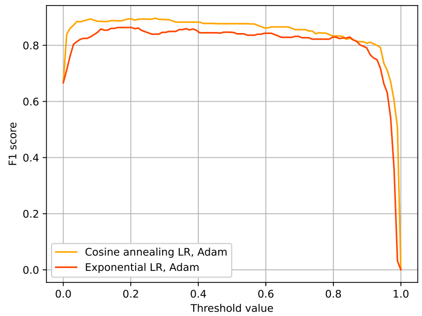

Fig. 4: F1 score versus threshold value

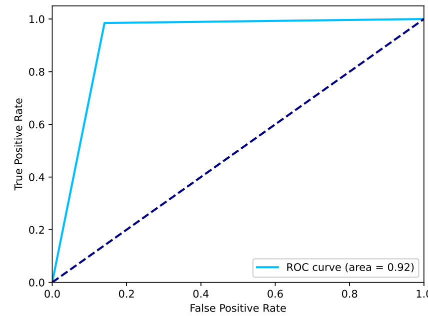

Fig. 5: Receiver Operating Characteristic (ROC) curve.

The model exhibited an accuracy on the test set of 92.31% and an F1 score of 0.9286. Notably, the confusion matrix derived from this dataset revealed 18 false positives and 2 false negatives, as detailed in Table II. This outcome aligns with our decision to intentionally skew the output distribution towards a higher sensitivity in detecting positive instances.

The model's consistent accuracy across both test and validation sets demonstrates robust generalization and confirms no overfitting on the training data.

		Predicted	
		Class 0	**Class 1**
True	**Class 0**	110	18
	Class 1	2	130

TABLE II: Confusion Matrix

The Receiver Operating Characteristic (ROC) curve of our model, depicted in Figure 5, provides a visual and quantitative analysis of its classification ability in a binary context. The dashed blue line represents a random classifier as a reference. Our model's Area Under the Curve (AUC) is 0.92, significantly higher than 0.5, the AUC of a random classifier. This high AUC score signifies the model's strong ability to differentiate between classes. Notably, the ROC curve exhibits a clear distinct shoulder value, indicating the optimal threshold to enhance the model's performance.

The enhanced performance of this model for this specific data compared to others in the literature can be largely attributed to the utilization of high-resolution images from the SDO/AIA. This approach contrasts with the norm in astronomical machine learning literature, where models often train on more extensive datasets comprising lower-resolution images. For instance, the model developed by Carrasco-Davis et al. [7] was trained using over half a million image sequences, each at a resolution of 21x21 pixels. In comparison, our model employs a dataset of fewer than 2000 image sequences, with each image of a resolution of 166x166 pixels.

The animations of the 20 misclassified events can be found in the GitHub repository at this link. The study of the misclassified events reveals the potential for amelioration of the network. Animations 1 and 2 represent false negative events, while the remaining animations correspond to false positive occurrences. For example, one jet was classified as negative when it was located in one of the edges of the frame. With the use of the right transforms, and a more diversly framed training set, our model may have been able to recognise this event.

Among the 18 instances of false positive classifications, the identification of bright spots on the sun as solar jets is a direct outcome of our decision to adopt a lower threshold. In the context of the research conducted on solar jets, those false classifications will have to be manually discarded in a further processing task.

Enhancements to the model could involve a more comprehensive optimization of hyperparameters, as well as implementing additional data augmentation techniques to enhance the model's robustness and compensate for limited training data. Moreover, revising the network's architecture could further enhance performance and accuracy.

V. Conclusion

Our study introduces a Recurrent Convolutional Neural Network (RCNN) for identifying solar jets, utilizing high-resolution SDO/AIA data. Our network directly uses sequences of images as a way of capturing the temporal dynamic of solar jets. This approach is distinct from existing models that predominantly use single frames and lower-resolution images. The model demonstrates robust generalization, achieving an F1 score over 0.92, and effectively minimizes false negatives. Its main application will be the automatic processing of the large Heliophysics events knowledgbase.

Future work will focus on enhancing model accuracy through techniques like the ones describes in the previous section and by implementing unsupervised learning, allowing for the use of more extensive data, as well as exploring the model's capability in identifying jet locations and shapes.

VI. Ethical Risks

Our research project primarily involves scientists or students who may come across and use our algorithm. In this context, ethical concerns are minimal. The algorithm's purpose is to distinguish between images containing coronal jets and those that don't. These events hold scientific significance but don't directly impact people's lives.

The data used for training, as well as the entirety of the litterature consulted, is available in open-source, and no sensitive or personnal information were part of the training process. Consequently, we can affirm that our algorithm poses no threats to data privacy or introduces discriminatory biases.

Nevertheless, typical ethical considerations associated with scientific projects, such as environmental impact from data collection and computing resources during the neural network training phase, still apply. This means that, like many scientific contributions, our project must be mindful of potential environmental consequences linked to the methods employed in gathering and processing data.

VII. Acknowledgment

Many thanks to Sophie Musset, who actively monitored and supervised this project !

References

[1] J. R. Lemen, A. M. Title, D. J. Akin, P. F. Boerner, C. Chou, J. F. Drake, D. W. Duncan, C. G. Edwards, F. M. Friedlaender, G. F. Heyman *et al.*, "The atmospheric imaging assembly (aia) on the solar dynamics observatory (sdo)," *Solar Physics*, vol. 275, pp. 17–40, 2012.

[2] C. S. N. Hurlburt, M. Cheung. (2010) Heliophysics event knowledgebase and datasearch. [Online]. Available: https://www.lmsal.com/heksearch/

[3] S. Musset, P. Jol, R. Sankar, S. Alnahari, C. Kapsiak, E. Ostlund, K. Lasko, L. Glesener, L. Fortson, G. D. Fleishman, N. K. Panesar, Y. Zhang, M. Jeunon, and N. Hurlburt, "Solar jet hunter: a citizen science initiative to identify coronal jets in euv data sets," 2023.

[4] Sunpy. [Online]. Available: https://sunpy.org/about/index.html

[5] Astropy. [Online]. Available: https://www.astropy.org/

[6] R. L. K. Musset, Sophie; Sankar. (2023) Solar jet hunter: Jet catalog from hek events 2011-2016. [Online]. Available: https://hdl.handle.net/11299/257209

[7] R. Carrasco-Davis, G. Cabrera-Vives, F. Förster, P. A. Estévez, P. Huijse, P. Protopapas, I. Reyes, J. Martínez-Palomera, and C. Donoso, "Deep learning for image sequence classification of astronomical events," *Publications of the Astronomical Society of the Pacific*, vol. 131, no. 1004, p. 108006, Sep. 2019. [Online]. Available: http://dx.doi.org/10.1088/1538-3873/aaef12

[8] J. A. Armstrong and L. Fletcher, "Fast solar image classification using deep learning and its importance for automation in solar physics," *Solar Physics*, vol. 294, no. 6, Jun. 2019. [Online]. Available: http://dx.doi.org/10.1007/s11207-019-1473-z

VIII. Appendix

A. Code Repository

For detailed information on our project's code structure, the libraries utilized, and the implementation of the model in predicting new data, please visit our GitHub repository at https://github.com/CS-433/ml-project-2-jac.git.

B. Digital Ethics Canvas

The following pages contain the digital ethical canvas given during the course.

Context	Beneficence	Non-maleficence		Solution
		Risks	Mitigation	
□ **In which context is the solution evaluated?** The context in which the solution is evaluated is research in astrophysics. In particular, research about the formation and the properties of solar coronal jets.	□ **What are the expected benefits of the solution in this context?** Our model is able to define whether or not a solar jet is present in a sequence of satellite images. It eliminates the need for manual classification for the biggest part of the heliophysic events knowledgebase. The expected benefit of this algorithm is the ability to speed up scientific research about solar jets by utilizing the enormous amount of data that requires processing.	□ **Can the solution be used in harmful ways, in particular with regards to vulnerable populations?** No, the solution has no particular ties with a population in general. □ **What kind of impacts can errors from the solution have?** The output of the algorithm will decide whether a particular event will be discarded or not for solar jet research. As each event classified as positive will be further manually processed, false positive do not have a large impact. They will be identified later in the research process, and will in the worst case waste the time of researchers. In the case of false negative, these events will be discarded for solar research. If the model is biased, a given kind of solar jets could be discarded, hence biasing scientific research about those events. □ **What type of protections does the solution have against attacks or misuse?** It doesn't have, as the user of the solution will be researchers only.		□ **What are the characteristics of the solution under evaluation?** The solution is an image sequence binary classifier, aimed at predicting whether or not a coronal jet is present in a given sequence of 30 images of the Sun.

	Privacy		Fairness		
	Risks	Mitigation	Risks	Mitigation	
	□ **What data does the solution collect?** Satelite data from the SDO/AIA mission. □ **Is it collecting personal or sensitive data?** No, this data is scientific data and is available in open-source. □ **Who has access to the data?** Anyone. □ **How is the data protected?** The data is not protected. □ **Could the solution disclose / be used to disclose private information?** No, the solution only contains information about solar events. No private information can be disclosed or derived from the solution.		□ **How accessible is the solution?** The solution is in a private GitHub repository. Access must be granted by one member of the group. □ **What kinds of biases may affect the results?** It may be some biases as jets are not always on the same face as the satellite so jets events will be missed. Some biases may come from the training dataset as well, if one given type of coronal jet is over- or underrepresented. □ **Can the outcomes of the solution be different for different users or groups?** No. □ **Could the solution contribute to discrimination against people or groups?** No.		

	Sustainability		Empowerment		
	Risks	Mitigation	Risks	Mitigation	
	□ **What is the carbon footprint of the solution?** If the acquisition of data is taken into account, the carbon footprint is relatively large due to the launch of satellites. Additionally, large computing resources were required to train the model. □ **What types of resources does it consume (e.g. water) and produce (e.g. waste)?** The launch of satellite produces significant amount of CO_2 by fuel combustion. Computing consumes energy in the form of electricity. □ **What type of human labor is involved?** Scientific work is behind the launch of the satellites. Human labor is involved in the manufacture of the technology for computing ressources (primarly the computers) and of the satellites. Human labor is involved as well in the fabrication/extraction of the components of those technologies.		□ **Can users understand how the solution works and what its limits are?** Yes, as users are scientists, they are able to grasp the limits of the solution. Moreover, this project is well-documented, and its capacities and limits are discussed □ **Are users able to make choices (e.g. consent, settings) in their use of the solution and how?** Users can modify the threshold for classification. Therefore, they can change the sensibility of the model. □ **How does the solution affect user autonomy and agency?** Both autonomy and agency may be seen as limited as the task is straightforward. Nevertheless, some control is allowed with the change of threshold. However, users will have access to the source code, which they will be able to locally modify. In that case, both autonomy and agency are almost complete.		

Dataset	Beneficence	Non-maleficence	
		Risks	Mitigation
□ **Who created the dataset?** The labelled dataset comes from volunteers annotations on a website called Zooniverse. The images in themselves are images form the SDO/AIA mission, which is part of the NASA program. □ **For what purpose was the dataset created?** To advance research on various astrophysical events and understanding the Sun's influence on Earth. □ **What mechanisms or procedures were used to collect the data?** The data was collected by satellites at a regular frequency. For more information, visit https://sdo.gsfc.nasa.gov/mission/. □ **Who was involved in the data collection process?** NASA's scientists were involved in the satellite imagery. ESA's scientist and volunteers were involved in the annotation of the dataset □ **Over what timeframe was the data collected?** The SDO/AIA mission has been launched in 2010 and is still in progress to this day. Data used in our algorithm specifically comes from observations taken between 2011 and 2016. □ **Was any preprocessing of the data**	□ **What are the expected benefits of analyzing this data? For whom?** Analyzing this data benefits astrophysics researcher that will have access to a larger library of classified jet events. The gain of time in comparison to manual classification is expected to be extremely significant.	□ **Does the dataset contain unsafe data (violence, nudity…)?** No. □ **What kind of impacts can errors in the data or in the analysis have?** Errors in the data or in the analysis could penalize researchers if the set of events they study is incomplete or biased. □ **Could the data or the conclusions from the analysis be used in harmful ways?** No.	

<table>
<tr><td rowspan="6">done? Yes, the images were compressed and their quality downgraded for storage purposes. The images were normalized and the data pixels clipped to reduce the skewness of their distribution.

□ Are there any missing data or data errors? There were missing images for some jet events. Those were discarded from the training set.

□ Where is the data stored?

The data is stored on the JSOC server (http://jsoc.stanford.edu/), which can be accessed either on their webpage or through python’s library Sunpy.</td><th colspan="2">Privacy</th><th colspan="2">Fairness</th></tr>
<tr><th>Risks</th><th>Mitigation</th><th>Risks</th><th>Mitigation</th></tr>
<tr><td colspan="2">□ Does the data contain personal or sensitive information? No, all data used is available in open-source, and does not present privacy or security issues.
□ Can personal or sensitive information be derived or inferred from the data or from the analysis? No, the data has no link to any personal or sensitive information.</td><td colspan="2">□ Is the data representative from a larger set (population)? How are subgroups represented?
□ What kinds of biases may affect the data?
□ Can the outcomes of the analysis be different for different groups?
□ Could the data or analysis results contribute to discrimination against people or groups?</td></tr>
<tr><th colspan="2">Sustainability</th><th colspan="2">Empowerment</th></tr>
<tr><th>Risks</th><th>Mitigation</th><th>Risks</th><th>Mitigation</th></tr>
<tr><td colspan="2">□ What is the carbon and water footprint generated by the storage of the data and by the computation in the analysis process? This footprint is probably relatively large, as the size of the data is considerable.
□ What type of human manual labor is involved in the data (e.g. labeling)? Volunteers annotated the training dataset. The annotation process was supervised by ESA’s researchers and took place on the website https://www.zooniverse.org/.
□ Does the data or the analysis require updates? No, the data being satellite images of the Sun, it does not require updates.</td><td colspan="2">□ How are the people concerned involved with the data or the analysis: have they been notified, have they consented? The data is available in open-source for research purposes. No special authorization was needed as our project falls onto the former category. The server managers have been notified of the downloading of the data as an email address had to be provided to do so.

□ Are the people concerned able to make choices (e.g. revoke consent, modify or delete data) regarding the data or the analysis? Yes, users will have access to the source code. From there, they will be able to make local changes, and retrain the model on a different dataset if needed.</td></tr>
</table>