Detection of sunquakes in Egression Power Maps using Deep Autoencoders

Vanessa M. Mercea [1] Anca M. Marginean [1] Diana Besliu Ionescu [2] Alin R. Paraschiv [3]

[1]Technical University of Cluj-Napoca [2]Romanian Academy Astronomical Institute [3]High Altitude Observatory, NCAR

Introduction

Solar flares involve the release of the Sun's magnetic energy as radiation, particle beams and high-speed plasma flows. These flares also affect the Sun's interior, generating seismic waves similar to earthquakes. Sunquakes have been studied in detail for cause and signatures. Yet, an automatic ML based tool for detecting these signatures has not been established. We attempt to detect sunquakes based on their observational signatures in Egression power maps.

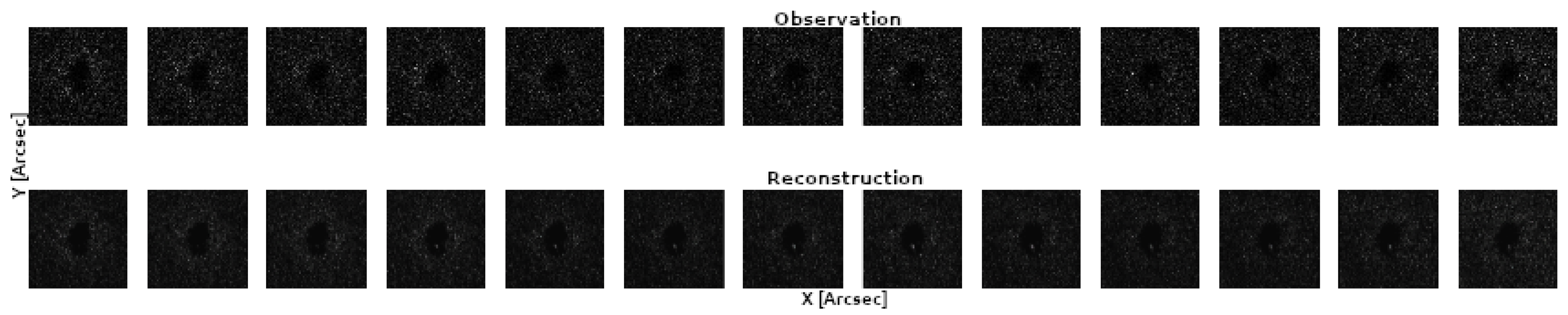

Figure 1. Source and Reconstruction of sunquake observations

We use a dataset consisting of timeseries of 35 sunquakes over solar cycle 23, represented as FITS files, along with numerical observations and metadata used to label the frames.

Challenges and approaches

The main challenges we faced include:

- **class imbalance**: Each sunquake observation file contains a minimum of 8 frames where a sunquake is visible. The remaining, up to 256, are considered negative samples. We introduced class weights, imbalance specific loss functions, and down/upsampling.
- **A low data regime**: Working in a low data regime means that we cannot use very deep architectures. We start from small models, and gradually increase their size and complexity.
- **The sunspot morphology:** The sunspot areas are not consistent between sunquake events. To mitigate this, we further increase the weight of positive events by oversampling our positive frames using 10 different transforms.

Machine Learning models

In our solution, we used both supervised and unsupervised Machine Learning methods to construct our model. The main components include:

- A **Variational Auto Encoder** used to learn representation from the Egression Power maps
- A **Residual Network** as a backbone for the AutoEncoder
- A **classifier** used to perform automated detection over encoded samples. Our experiments include: **KNN** (with and without bagging), **Feed Forward NN**, **SVC**, **Logistic Regression**, **fine-tuned unfrozen layers of the VAE**, **SGD**.

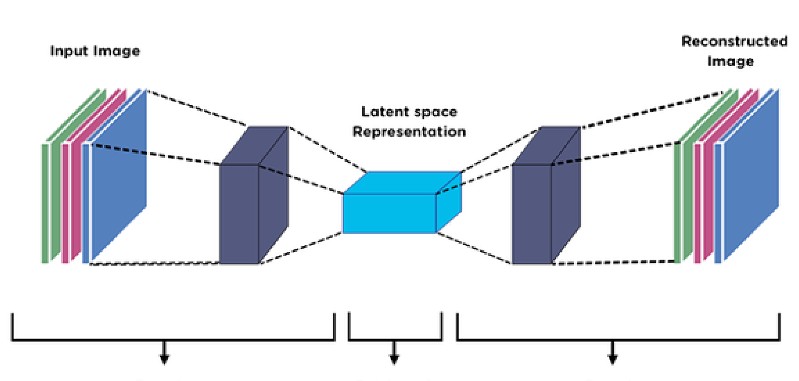

Figure 2. AutoEncoder architecture

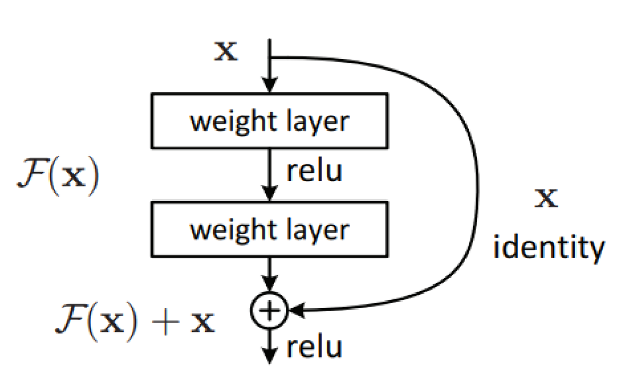

Figure 3. Residual Network Block

By combining the above, we aim to show that an AutoEncoder pretrained on the entire dataset leads to good reconstructions and may be connected to smaller models for solving downstream tasks, such as automated detection.

Analysis Methodology

We processed the initial data files into a comprehensive dataset. We computed the sunquake margin frames based on metadata extracted timestamps. We started with an AutoEncoder with a few convolutional layers to extract information from the quake signatures, which proved insufficient for classification. We then implemented a Residual Network architecture.

Deep Auto Encoders were chosen because of the complex morphological structure of the data samples, so that the model would learn the representation and distribution of data samples. AutoEncoders are trained in an unsupervised manner, so they benefit from being able to learn from the entire dataset, and no information is lost. In this way, downstream models such as classifiers can use the encoder as a backbone and be trained on downsampled data.

After training and evaluation, we manually validated reconstructions for data samples to confirm that the quake information is preserved. Due to varying morphological structure between events, we updated our AutoEncoder so that the Encoder outputs a distribution in the latent space with explicitly modeled variance. We used dimensionality reduction techniques, specifically PCA and UMAP to interpret clustering of the encoded data points. UMAP showed good clustering of the latent data. Finally, we fed the encoded latent space preserving the quake signatures and it's principal components to various classifiers.

Interpreting UMAP Representations

The figures below represent UMAPs built from the the latent encoding of sunquake samples. The color coding is viridis, going from purple (frames distant from the sunquake), to yellow (sunquake frames), symmetrically.

Quake clustering as noise

From initial UMAPs generated using 15 neighbors, we hypothesized that sunquake observations are clustered with noise when no downsampling is used. We can see that after downsampling the frames that are far from the sunquake's peak intensity, the cluster count diminishes and the quake observations are grouped with the remaining frames. This shows that downsampling the negative samples in the training data could be used to mitigate the model training effects due to observational and method background noise.

Impact of Morphological Structure

From Fig. 7 we inferred that the observation's background noise and the morphological structure of the Active Region play a significant role in the Autoencoder output. Counts in the active region are systematically lower, and morphologies are not consistent between the sunquake events. These hinder a proper cluster-like classification of the sunquakes by using just the Variational AutoEncoder.

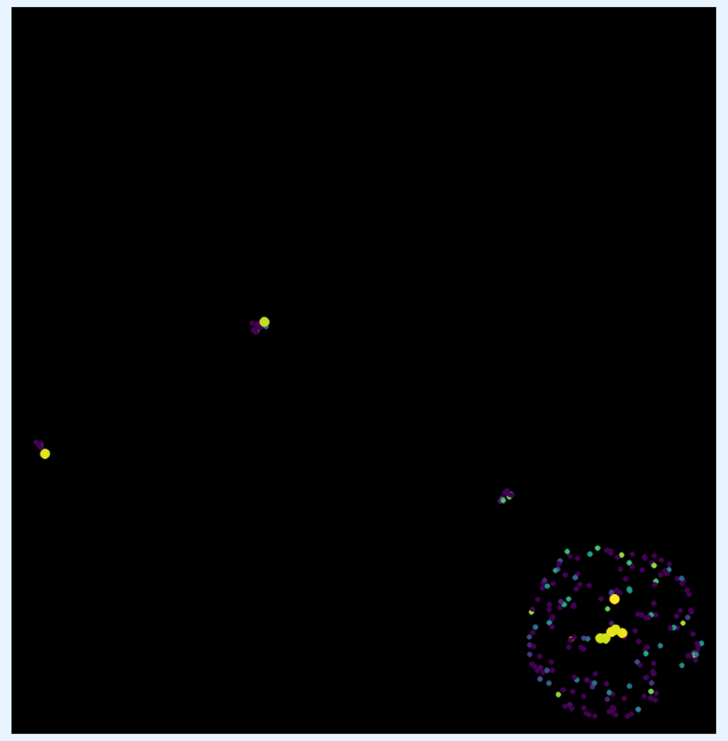

Figure 4. UMAP plot of all observations of an event using 250 neighbors. Sample Size: 256

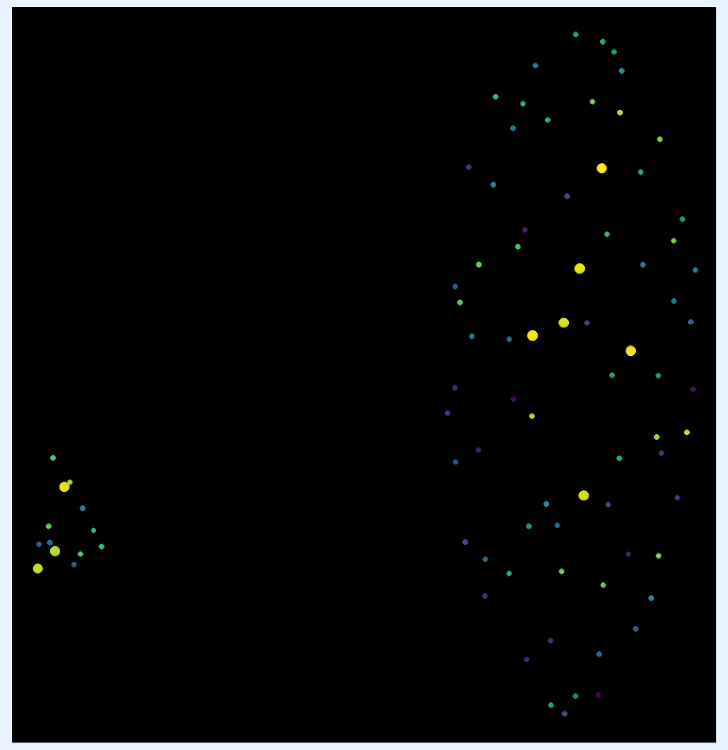

Figure 5. UMAP plot of downsampled observations of an event using 250 neighbors. Sample Size: 94

Higher Complexity yields to concentric quake clustering

To mitigate the issues presented at the previous point, and improve the hindered clustering, a more complex classification model is required. We have increased the number of neighbors used to generate the UMAP representation of the data and shown that higher complexity leads to clustering of the reconstructed sunquake observations. In Fig. 4 we observe that even without downsampling, the data is now clustered in concentric circles, having the sunquake observations at the center. A grid search over UMAP parameters shows that without downsampling, a more complex model is needed, which is confirmed by our classification experiments.

Experiments and Results

For the AutoEncoder training, the hyperparameters below yielded the best learning curve and classification results so far:

- Learning Rate = 1e-5
- Batch Size = 16
- Number of Residual Blocks = 6
- Number of Levels = 4
- Number of Latent Dimension filters = 32
- Bottleneck Dimension = 256
- Multi Residual Block Skips = True

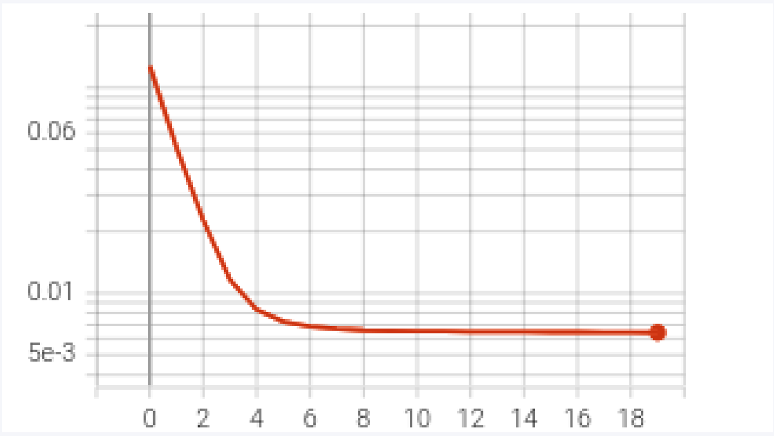

Figure 6. Training loss curve for the AutoEncoder

A smaller learning rate worked better when combined with a smaller batch size. In terms of the AutoEncoder architecture, we allowed for residual block skips, so that the feature maps can be added to the latent output at each resolution.

In terms of classification our experiments show that dimensionality reduction techniques that we used capture relevant data for classification, both by looking at the UMAP results, and by experimenting with a variety of classifiers. Results need yet to be improved by further updates to the autoencoder. The input data we used was varied to include: i. random transforms on negatives and an addition of 5-10 transforms for each positive sample, ii. downsampling to 40 frames around the sunquake for training.

We achieved best results with combined sampling and sliding windows on current and previous frame encodings and UMAP components. For a total of two events in the test dataset, the models correctly marked 10 sunquake frames (out of 13) for the first event event as positive but only 2 frames (out of 13) from the second event.

We experimented with the following models: KNN, MLP, SGD, SVC, Logistic Regression, unfreezing and finetuning the top layers of the autoencoder, and also soft voting between these models. The losses we used include BCE and focal loss.

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma log(p_t)$$

Some of our results can be visualized in the tables below. SVC with a poly kernel showed best results for the negative class. For most models, there was a trade-off between precision and recall for positive sunquake samples. The presented results underline a work in progress. By analyzing them, we have uncovered specific issues to address, such as the metrics for the positive class and the reduced clustering of encodings when fitting UMAP on the entire dataset.

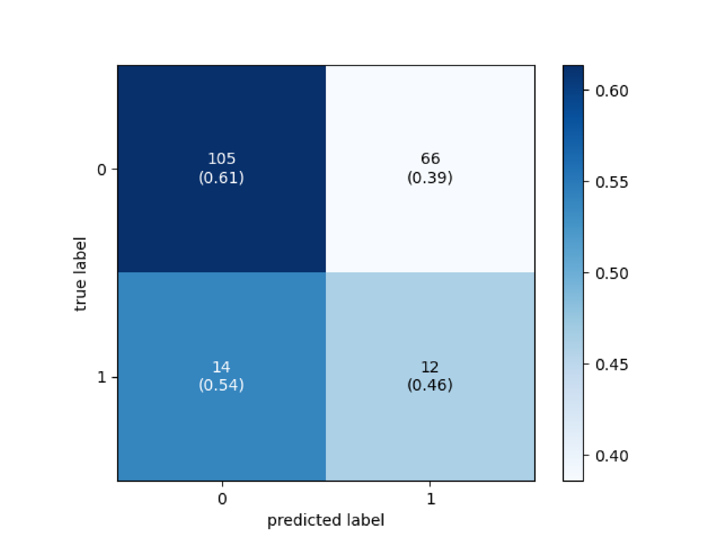

Figure 7. Confusion Matrix for the Test data (sampled)

Table 1. Classification Report SVC Poly

Class	Precision	Recall	F-score	Support
0	0.89	0.98	0.93	171
1	0.62	0.19	0.29	26
accuracy			0.88	197
Train 0	0.75	1.00	0.86	1950
Train 1	0.98	0.32	0.48	930
Train accuracy			0.95	2680

Table 2. Classification Report KNN

Class	Precision	Recall	F-score	Support
0	0.88	0.61	0.72	171
1	0.15	0.46	0.23	26
accuracy			0.59	197
Train 0	0.68	1.00	0.81	975
Train 1	1.00	0.73	0.84	1705
Train accuracy			0.83	2680

Future Work and Conclusions

- The second test event is not correctly recovered. We aim to analyze if observational data is acting in an outlying way when compared to our sample, and if not, plan to perform cross validation to review the learned features and predictions over other observations.
- Dimensionality reduction techniques showed promising results, indicated by the fact that the encoded data is clustered. Moreover, positive results when classifying the encoding show that the relevant aspects in the initial trained data are indeed captured by the encoder.
- Classification results indicate typical class imbalance and low data regime problems. We believe the current presented metrics have great potential for improvement.