Extraterrestrial Technosignature Detection Using Deep Learning

Petar Kupusarević
Faculty of Technical Sciences
University of Novi Sad
kupusarevic.sw57.2019@uns.ac.rs

Tanita Mošić
Faculty of Technical Sciences
University of Novi Sad
mosic.sw53.2019@uns.ac.rs

***Abstract*— Deep learning techniques are currently the best solution available for recognizing potential extraterrestrial signals, and using them correctly will enable discoveries in the fields such as cosmology, astrophysics, and astrobiology. These discoveries will give us insight into our place in the universe and potentially lead to the first extraterrestrial contact. In this paper, we present four methods to predict whether given digital spectrometer data contains anomalies in the form of abnormal radio waves or noise that indicate extraterrestrial intelligence. To solve this problem, we will use artificial intelligence techniques such as image classification (using convolutional neural networks), anomaly detection, autoencoders, support vector machines, and image processing. The input data was taken from SETI Breakthrough Listen - E.T. Signal Search dataset on Kaggle. No conclusive results have been achieved yet, but we discuss several promising techniques that could lead to a solution.**

Keywords—anomaly detection; autoencoder; computer vision; convolutional neural network; deep learning; neural network; outer space; unsupervised learning;

I. INTRODUCTION

Space exploration has always been propelled by state-of-the-art technology and the latest scientific discoveries. With the advances made in the fields of Artificial Intelligence (AI) and Machine Learning (ML) in recent years, it comes as no surprise that these technologies are being used to further improve our understanding of the extraterrestrial. A major aspect of modern exploration is the search for and detection of technosignatures, as it addresses the existence of complex (i.e., technological) life in outer space [1]. The search for technosignatures had risen from a renewed effort at the search for extraterrestrial intelligence (SETI), which has been a topic of interest since the late 1950s.

As this type of task emerged fairly recently, there have not been any major problems solved in this particular niche. In this case, we rely on simulated data to represent possible future data that could be collected from space. In this paper, we focus on the problem presented in the SETI Breakthrough Listen - E.T. Signal Search competition [4]. The competition problem is presented as such: there is data from the scans of millions of stars, a so-called “haystack”. The objective is to detect “needles”, faint signals of alien transmissions. A two-part filtering method is then used to classify the scans[4]. However, this method is too harsh and eliminates too many actually anomalous scans.

In this paper, we will discuss our ideas for implementing and improving this solution, as well as the differences between supervised and unsupervised learning techniques. We will be using several deep learning techniques for anomaly detection including autoencoders [7], support vector machines, and image classification using convolutional neural networks (CNN) [5][6]. These are some of the state-of-the-art approaches in the field of computer vision that have been used to complete similar tasks [33]. Our hypothesis is that the best results will be achieved by using a combination of these methods. We have no concrete results as of yet, but we have devised multiple approaches to solving the problem that we will discuss at length.

The input to our solution will be the digital spectrometer data and the output will be the probability of that data containing an anomaly. The target metric is the F1 score of the model’s performance on the dataset.

The remainder of our paper is structured as follows. Section II is dedicated to related works, while section III will explain the data we were working with in detail. Section IV presents some of the challenges we came across during data exploration. Section V is dedicated to image classification, while section VI discusses several techniques of unsupervised learning. Section VII talks about image processing and section VIII gives an overview of the final model that combines the aforementioned techniques. In section IX we discuss the results of our experiments and elaborate on our target metric choice. Finally, we give our conclusion in section X.

II. RELATED WORK

As we are dealing with data that is unique to the challenge presented by the SETI Institute, there are no other works dealing with this exact problem. However, both space exploration and deep learning are topics of interest to various researchers, and in recent years progress has been made in space exploration using deep learning methods [9].

M. Huertas-Company et al. [10] used a convolutional neural network to identify galaxies of certain characteristics. They trained the CNN on simulated images of galaxies. As the model successfully identified the galaxies, they surmised that the technique "can be applied to the classification of other astrophysical phenomena". N. Gupta et al. [11] proposed using a combination of deep learning methods to estimate the mass of galaxy clusters. They employed a modified version of the feed-forward algorithm ResUNet [12].

Deep learning has been used to solve anomaly detection problems on Earth as well. Haselmann et al. [13] proposed a method for automating surface inspection in manufacturing using CNNs with unsupervised learning. They used pattern detection and image completion to detect faulty samples. However, this particular approach produced poor results for weakly contrasted anomalies and was thus unsuitable for our purposes.

III. THE DATA

The dataset that was used was the SETI Breakthrough Listen – E.T. Signal Search dataset from Kaggle [4]. The SETI (Search for Extraterrestrial Life) organization uses the Green Bank Telescope to gather data and take pictures of space. The resulting raw data is converted with a Fourier Transform [36] to generate a spectrogram, which is a measurement of signal density as a function of frequency and time. A single observation consists of six of these spectrograms, three of which contain the object in question, and the other three are "control" objects. Firstly, the target object or region is observed for 5 minutes, then another object (such as a star or a region of the sky) is observed for another 5 minutes. The process of observing the target object and then a control object is repeated 2 more times. The reason why for the observation of a random object after every target observation is that our own human technology (such as radio stations, routers, cellphones, etc.) emits radio signals which can cause disturbances and noise in our observations. So, we use the control spectrograms to check whether the noise we are detecting is localized to the object we are monitoring or is interference coming from Earth. These human-generated signals are referred to as "radio frequency interference" or RFI.

The dataset contains 60,000 of these groups of six observations, which are called cadences. Out of the 60,000 examples, 6,000 contain digitally synthesized anomalies in the form of signal noise or concentrated bursts of radio signals, the other 54,000 are normal real observations of space. Figure 1 contains a cadence targeting the Voyager 1 space probe. This is of course a man-made object and emits its own radio signal to communicate with Earth, but it does represent the kind of signal we are searching for to detect extraterrestrial life. There is a clear line visible in the 1st, 3rd, and 5th spectrograms which are pointed to the target object, in this case, Voyager 1. This line is diagonal because during the time it took to make the observations, the Earth and the space probe have moved, imparting a Doppler effect.

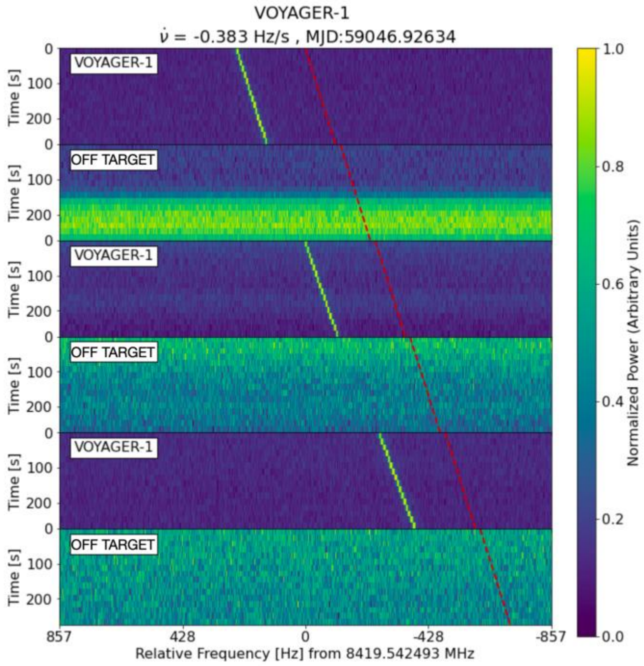

Figure 1 - Cadence showing the Voyager 1 space probe. Contains six spectrograms, 1st 3rd and 5th cadences show spectrograms of Voyager, while the 2nd, 4th and 6th are looking off target.[i]

IV. CHALLENGES

As we worked on the problem, we faced several major challenges; the most prominent of which was the lack of computing power needed to process the data. The datasets were large, as the images representing the scans were in high definition. It seems that the quality and quantity of the data are enough to create a classifier that can distinguish between anomalous and normal data, but there are many challenges with the data that make solving this problem difficult. Each spectrogram image is 273 x 256 pixels, and the entire cadence ends up at 1638 x 256 pixels when they are stacked vertically. This caused long processing times and slow training and testing. Attempts to mitigate this were unsuccessful as lowering the resolution of the scans in any way came with the risk of removing the small anomalies we were trying to detect.

That being said, we ran into multiple problems that did stem from the dataset. Some of the cadences containing anomalies are quite difficult to differentiate even for a trained human eye, which means that it can be difficult for a deep learning model to differentiate between irregular and regular data. Furthermore, not all spectrograms of the target have to contain the anomaly; it can be contained within a small region of just one.

The biggest problem, however, is the fraction of anomalous data, which is only 10%. This makes it quite difficult to train a deep learning model. We had to either rely on unsupervised learning techniques to create a classifier or heavily augment the anomalous data to make up for the low percentage of positive examples. Traditionally, with image data, we would add noise to the data to make the model more robust, but in this case, we cannot do this because the goal of the experiment is to detect what is essentially noise. The only augmentations we could do would be to flip the images horizontally or vertically since space doesn't have a preference for a direction. Even after augmentation, the ratio of positive to negative examples is still lower than desirable.

V. IMAGE CLASSIFICATION

To solve the problem, a powerful model which can detect the anomalies and differentiate between positive and negative examples will be needed. In recent years, deep learning has been used to solve these kinds of problems, specifically deep neural networks, where the word *deep* implies that the neural network contains a large number of layers, thus giving it the ability to learn more complex functions and find more obscure patterns in the data. A vectored form of the data is fed into the neural network, where through a series of matrix multiplications and nonlinearities (non-linear functions) a resulting output value is computed. The parameters of the network, also called the weights, are matrices that are multiplied by the resulting output from each layer. The aim of the training process is to find good values for these weights according to a predefined criterion. This is achieved using backpropagation [18] and gradient descent [19].

A convolutional neural network (CNN) is a type of neural network that is used for image data and other spatially dependent data, the first convolutional neural network is considered to be the Neocognitron network [20] which first introduced the two main layers in CNNs, convolutional and downsampling layers. But the implementation of LeCun et al. [21] in 1989 laid the foundation for the modern CNN. The CNN gets its name from the convolution operation which uses sliding filters to detect features in the data. At the shallow part of the network, these filters detect small features such as lines and dots, but in the further layers they can detect higher-level representations such as whole objects. The training process tries to learn the exact values of these filters. The most important parameters a convolution operation has are the filter size and the stride (the step the filter takes from one operation to the next).

After the next representation is calculated, a nonlinearity is applied to the output, for instance, ReLu or Sigmoid [22]. Accompanying this is usually a pooling operation, MaxPooling [15] is used often [16]. MaxPooling takes the highest value of a region of the resulting output, this region makes its way around the image much like a filter in a convolution operation does, but with pooling, the stride is usually the same as the filter region size. Multiple blocks of these operations are stacked until the resulting data is small enough to be flattened (representing a tuple with height, width, and depth as a vector). This vector can then be used later by a neural network or some other model.

Neural networks can approximate very difficult functions, and when the complexity of the network is far greater than the target function requires, overfitting can occur. Overfitting is when the network finds it easier to remember the training data instead of trying to generalize, this results in better training performance but harms performance during inference. To mitigate and lessen the effects of overfitting, some regularization techniques are used such as Dropout [8], Batch Normalization [23], and L1/L2 regularization [24]. In practice, instead of trying to find the exact size for the neural network according to the complexity of the function it is trying to approximate, usually a bigger architecture is selected and is then regularized to prevent overfitting [14].

Dosovitskiy et al. [25] introduced visual transformers, which is a new architecture that can be used for image data. Visual transformers are built around transformers, introduced in the famous paper "Attention Is All You Need" [26]. ViTs (Visual Transformers) work much like regular transformers except that they represent the input image as a series of vectorized patches.

A. Implementation

There are two approaches we devised that show promise for using CNNs for the task of anomaly detection on the given dataset. Unfortunately, we were not able to fully implement and test them because of time constraints. The first is that we can represent the input as a tuple of size 6 x h x w, where h is the height, w is the width, and 6 represents each of the spectrograms that make a cadence. This should work better than simply stacking all the spectrograms vertically because we are separating each spectrogram. This implementation is illustrated in Figure 2.

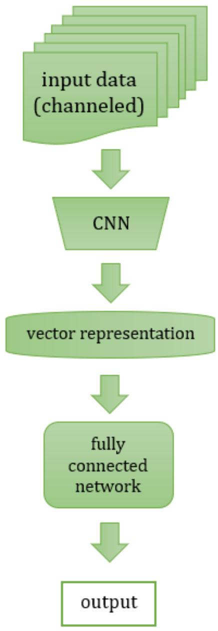

Figure 2 - First proposed model architecture. The input is represented as a 6 x h x w tensor and is given as input to a CNN. The resulting tensor is then flattened and given to a neural network to compute the probability of the input containting an anomaly.

The second approach is to use two separate encoding CNNs for the target and off-target data. Each CNN would create a representation of its corresponding part of the data, and after flattening these vectors would be concatenated and passed to a neural network for classification. This approach completely separates the target and off-target data, but we then have to rely on both of the encoders to create meaningful representations of their respective areas. This implementation is illustrated in Figure 3.

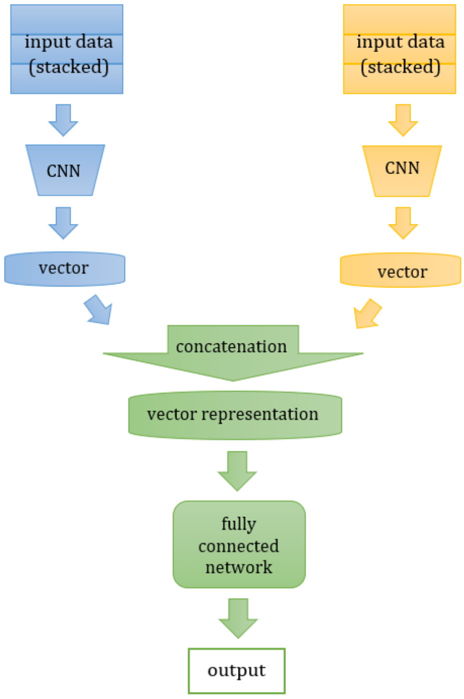

Figure 3 - Second proposed model architecture. Stacked anomalous and non-anomalous data are fed into respective Convolutional Neural Networks and are flattened at the end. The two resulting vectors are concatenated and then the output probability is computed via the final Neural Network.

VI. UNSUPERVISED LEARNING

In section V we used the labels provided by the dataset to help the model differentiate between the positive and negative examples. This is called supervised learning. However, our dataset is heavily imbalanced, so it is much harder to take this approach. We propose that a possibly better way of solving the problem is by using unsupervised learning, where we can use techniques such as clustering [27], dimensionality reduction [28], probability estimation [29], and autoencoders [30].

Primarily, we will be using autoencoders to get a smaller vectorized representation of the input data (and use that for further classification) or to use its generative capability to find outliers. In the following subsections, we will discuss different approaches to using various Unsupervised Learning techniques to solve the problem.

A. Autoencoder

An Autoencoder [30] is a neural network consisting of an encoder and a decoder (Figure 4). The encoder encodes the input data to a smaller representation. The decoder is trained to reproduce the input using the intermediate representation. The encoder will learn to represent the incoming data as a vector of the same size as the "bottleneck" layer. The smaller the vector, the more information is lost, but the bigger the vector, the less meaningful representation is acquired, and the data remains of higher dimensionality.

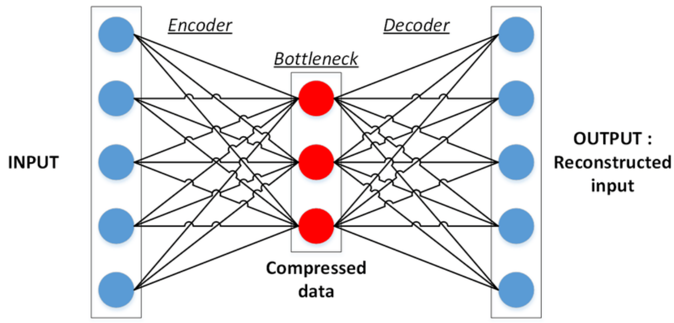

Figure 4 - Architecture of an autoencoder. On the left is the input layer which accepts the incoming data. In the middle is the so-called bottleneck layer. And on the right is the reconstructed input.[ii]

B. Mean Squared Error

Mean squared error (MSE) [31] is defined as the average of all the squared differences between the generated and the target output. The formula for MSE is as follows:

$$L(f(x;\theta), y) = \frac{1}{n}\sum_{i=1}^{n}(f(x;\theta)_i - y_i)^2$$

1. $f(x;\theta)$ - Output of model with input x parameterized by weights θ
2. y - Target output
3. n - Dimensionality of data
4. i - Index of the current pixel

It is primarily used as a loss function for regression tasks, but we can also use it as the final output of the model upon inference, since it will represent the network's ability to reconstruct the input image. The idea is to use a sliding window (much like with convolutions, except we will use a much bigger window) to gather input images for the Autoencoder network. For this approach we would only use the non-anomalous target data. The Autoencoder will learn to reconstruct these images, but when shown an image containing an anomaly, it will have a harder time since it hasn't seen positive examples during training. This will result in a higher MSE score, indicating an outlier. After computing all the values of the training set, a threshold MSE value will be chosen such that it best divides the two labels of the data

C. Support Vector Machine

A Support Vector Machine's [31] goal is to separate data points of different classes. It does this by finding a hyperplane that best maximizes the margin between these classes. This is illustrated in Figure 5.

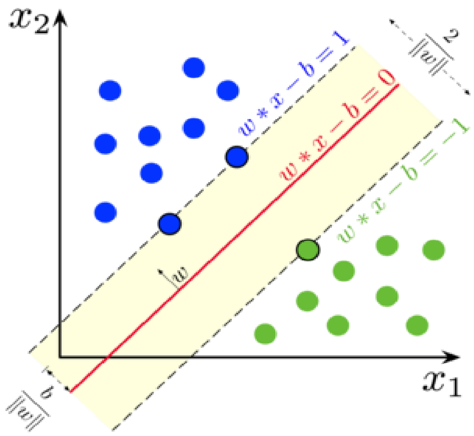

Figure 5 - End result of the SVM algorithm on a simple problem. The 2 classes of blue and green are represented as dots with x1 and x2 values. The hyperplane (represented by the red line) separates these two classes.[iii]

If it is possible to represent a single cadence using an *n*-dimensional vector (where *n* is relatively small compared to the dimensionality of the input cadence) using an autoencoder, we can then represent each data sample as a dot in an *n*-dimensional space and use an SVM to separate the two classes.

VII. IMAGE PROCESSING

As was discussed in section III, the anomaly that we are trying to find doesn't have to be in all of the target spectrograms, this substantially makes the problem more difficult. To solve this problem, we can train another image classification model on processed images. These processed images would contain only the differences between the three target spectrograms (Figure 6).

Using this technique, we can isolate only the differences and try to predict a label using only the differentiated data. This solution should outperform the others when the anomaly is not fully present in all the target observations. Another advantage of this approach is that the new input image will have lower dimensionality.

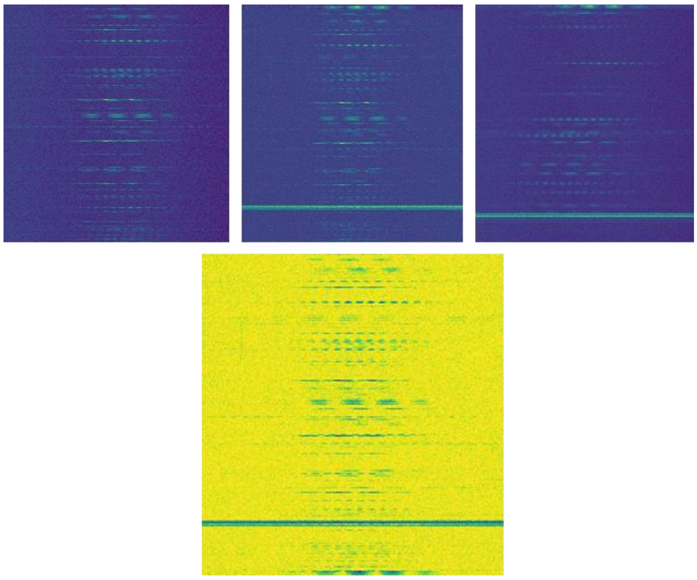

Figure 6 – The top row depicts the three target observations while the bottom picture shows the absolute difference of their corresponding pixels. The thick line at the bottom represents the anomaly which is not present in the first observation.

VIII. FINAL MODEL

We presented four different models that each could be used to solve the problem. It is our hypothesis that combining all of these models will yield the best performance. Using multiple models to solve a problem is called Ensemble Learning [32].

The final model we propose is as follows:

$$a(x) = w_1 * IC(x) + w_2 * MSE(x) +$$

$$w_3 * SVM(x) + w_4 * IP(x) + b$$

$$f(x) = \frac{1}{1 + e^{-a(x)}}$$

1. x - cadence
2. f(x) - final output of the model
(<0.5 : non anomalous, > 0.5 : anomalous data)
3. $w_{1,2,3,4}$ - weights associated with each model
4. b - bias term
5. IC(x) - output from Image Classification model
6. MSE(x) - output from Mean Squared Error model
7. SVM(x) - output from Support Vector Machine model
8. IP(x) - output from Image Processing model

IX. RESULTS

Due to time constraints and aforementioned challenges, we have acquired no concrete results at the time of completion of the first draft of this paper. Further testing and adjustments are needed.

The target metric we are after while training this model is not necessarily accuracy, since there are a lot more positive than negative examples so the model can get a high accuracy score by simply classifying every data point as non-anomalous. What is most needed from this classifier is to not misclassify positive examples, and it is acceptable if as a result of this the model classifies more images as anomalous that were actually non-anomalous. In other words, a strong recall is needed. A good all-around measure is the F1 score. It is calculated as such:

$$F_1 = 2\frac{precision*recall}{precision+recall}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

1. TP - number of positive examples labeled correctly
2. FP - number of negative examples labeled incorrectly
3. FN - number of positive examples labeled incorrectly

X. CONCLUSION

The search for extraterrestrial intelligence is a very important problem, as it will show us whether we are alone in the universe. Answering this question will give us insight into our place in the cosmos and potentially lead to the first extraterrestrial contact.

In this paper we have showcased four methods to solve the problem of detecting extraterrestrial technosignatures:

1. Image classification using CNN
2. Mean squared error
3. Support vector machine
4. Image processing

Our hypothesis was that combining these methods will produce the best results. The next step will be to implement this model and make improvements through testing. Future work could apply this method to similar datasets, as well as trying more ensemble methods utilizing a different configuration of machine learning techniques.

REFERENCES

[1] Jason Wright, Veronica Allen, Julián D. Alvarado-Gómez, Daniel Angerhausen, Daniel Apai, et al. "Searches for Technosignatures: The State of the Profession", 2019

[2] Marc Kaufman “Technosignatures and the search for extraterrestrial intelligence”, 2018, https://astrobiology.nasa.gov/news/technosignatures-and-the-search-for-extraterrestrial-intelligence/

[3] Hector Socas-Navarro, Jacob Haqq-Misra, Jason Wright, Ravi Kopparapu, James Benford, Ross Davis, the TechnoClimes 2020 workshop participants “Concepts for future missions to search for technosignatures”, 2021

[4] SETI Breakthrough Listen - E.T. Signal Search, Available: https://www.kaggle.com/c/seti-breakthrough-listen

[5] Chein-I Chang, Shao-Shan Chiang “Anomaly detection and classification for hyperspectral imagery”, 2002

[6] Stefania Matteoli, Marco Diani, Giovanni Corsini “A tutorial overview of anomaly detection in hyperspectral images”, 2020

[7] Chong Zhou, Randy C. Paffenroth “Anomaly Detection with Robust Deep Autoencoders”, 2017

[8] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov “Dropout: a simple way to prevent neural networks from overfitting”. Journal of Machine Learning Research, 2014.

[9] Ting-Yun Cheng, Christopher J Conselice, Alfonso Aragón-Salamanca, Nan Li, Asa F L Bluck, Will G Hartley, James Annis, David Brooks, Peter Doel, Juan García-Bellido, David J James, Kyler Kuehn, Nikolay Kuropatkin, Mathew Smith, Flavia Sobreira, Gregory Tarle “Optimizing automatic morphological classification of galaxies with machine learning and deep learning using Dark Energy Survey imaging”, Monthly Notices of the Royal Astronomical Society, Volume 493, Issue 3, April 2020, Pages 4209–4228

[10] M. Huertas-Company et al. “Deep Learning Identifies High-z Galaxies in a Central Blue Nugget Phase in a Characteristic Mass Range”, Astrophysics Journal 858, 2018

[11] N. Gupta, C. L. Reichardt “Mass Estimation of Galaxy Clusters with Deep Learning I: Sunyaev-Zel’dovich Effect”, 2020

[12] Foivos I. Diakogiannis, François Waldner, Peter Caccetta, Chen Wu “ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data”, 2020

[13] M. Haselmann, D.P. Gruber, P. Tabatabai “Anomaly Detection using Deep Learning based Image Completion”, 2018

[14] Andrew Ng “Machine Learning Yearning”, 2018

[15] Jawad Nagi, Frederick Ducatelle, Gianni A. Di Caro, Dan Ciresan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jurgen Schmidhuber, Luca Maria Gambardella “Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition”, 2011

[16] Hossein Gholamalinezhad, Hossein Khosravi “Pooling Methods in Deep Neural Networks, a Review”, 2020

[17] Jie Yang, Ruijie Xu, Zhiquan Qi, Yong Shi “Visual Anomaly Detection for Images: A Survey”, 2021

[18] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams “Learning representations by back-propagating errors”, Nature Volume 323, 1986

[19] Sebastian Ruder “An overview of gradient descent optimization algorithms”, 2017

[20] Kunihiko Fukushima “Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”, 1980

[21] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackal “Backpropagation Applied to Handwritten Zip Code Recognition”, 1989

[22] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, Stephen Marshall “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning”, 2018

[23] Sergey Ioffe, Christian Szegedy “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, 2015

[24] Andrew Y. Ng “Feature selection, L1 vs. L2 regularization, and rotational invariance”, 2004

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, 2020

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin "Attention Is All You Need", 2017

[27] Dongkuan Xu, Yingjie Tian "A Comprehensive Survey of Clustering Algorithms", 2015

[28] Andrzej Mackiewicz, Waldemar Ratajczak "Principal components analysis (PCA)", 1992

[29] Sheng Liu, Aakash Kaku, Weicheng Zhu, Matan Leibovich, Sreyas Mohan, Boyang Yu, Haoxiang Huang, Laure Zanna, Narges Razavian, Jonathan Niles-Weed, Carlos Fernandez-Granda "Deep Probability Estimation", 2022

[30] Ian Goodfellow, Yoshua Bengio, Aaron Courville "Deep Learning", 2016, Chapter 14

[31] Gareth James, Daniale Witten, Trevor Hastie, Robert Tibshirani "An Introduction to Statistical Learning", Springer New York Heidelberg Dordrecht London, 2013.

[32] David Opitz, Richard Maclin "Popular Ensemble: An Empirical Study", Journal of Artificial Intelligence Research 11, 1999

[33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie "A ConvNet for the 2020s", 2022

[34] YeongHyeon Park "Concise Logarithmic Loss Function for Robust Training of Anomaly Detection Model"

[35] Zhang, Xueqin & Gu, Chunhua & Lin, Jiajun. "Support Vector Machines for Anomaly Detection." , 2006

[36] Bracewell, R. N., & Bracewell, R. N. (1986). "The Fourier transform and its applications" (Vol. 31999). McGraw-Hill New York.

[i] image source: https://www.kaggle.com/competitions/seti-breakthrough-listen/overview/data-information

[ii] image source: https://medium.com/analytics-vidhya/autoencoders-with-tensorflow-2f0a7315d161

[iii] image source: https://en.wikipedia.org/wiki/Support-vector_machine