UNSUPERVISED DETECTION OF ANOMALOUS SOUNDS VIA PROTOPNET

Technical Report

Yannis Kaltampanidis[1], *Iordanis Thoidis*[1], *Lazaros Vrysis*[1], *Charalampos Dimoulas*[1],

[1]*Aristotle University of Thessaloniki, University Campus, 54124, Thessaloniki, Greece*
{kaltampa, ithoidis, lvrysis, babis}@auth.gr

ABSTRACT

Prototypical part network (ProtoPNet) is a novel method proposed for the task of image classification, offering the ability to interpret the network's reasoning process during classification. The subject of this work is the examination of ProtoPNet as an unsupervised anomaly detection method, through its application at the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 task 2 challenge. It is also showed that ProtoPNet shares common grounds with Deep One-Class Support Vector Data Descriptor (DOCSVDD).

Index Terms— Unsupervised Anomaly Detection, ProtoPNet, DOCSVDD

1. INTRODUCTION

While most condition monitoring approaches focus on vibration signals [1], the use of audio signals for fault detection is appealing as they encapsulate a substantial amount of machinery faults [2]. On the other hand, the machinery's surrounding noise under realistic industrial conditions may lead to low signal to noise ratio [3], thus impairing our ability to detect anomalous operations.

Qualitatively, the term anomaly describes a sample that strays considerably from the majority of the data, arousing the suspicion that it was generated by another mechanism [4]. Given an input space $\mathcal{X}$, the purpose of anomaly detection methods is the formulation of a function $\mathcal{A} : \mathcal{X} \rightarrow \mathbb{R}$ such that:

- $\mathcal{A}(x)$ is high when x represents an anomalous sample.
- $\mathcal{A}(x)$ is low when x represents a normal sample.

The mapping $\mathcal{A}(x)$ is called anomaly score [5]. Finally, a sample x is classified as anomalous if $\mathcal{A}(x) > \phi$, where ϕ is a predetermined threshold. Overlooking the way $\mathcal{A}(x)$ is measured, anomaly detection techniques come under the following categories [6]:

- *Supervised anomaly detection*: The training dataset consists of both normal and anomalous samples. That is equivalent to binary classification.
- *Semi-supervised anomaly detection*: The training dataset consists of only normal samples. The problem is also known as one-class classification.
- *Unsupervised anomaly detection*: The available samples are not labeled.

Since most unsupervised methods operate under the implicit assumption that the majority of the available samples are normal [7], they can find application in a semi-supervised scenario. That often leads to the use of the term unsupervised anomaly detection, even for problems configured as semi-supervised. Conforming with the title of the task, we use the term unsupervised anomaly detection for both of those categories.

DCASE 2020 task 2 poses the challenge of detecting whether the sound emitted for a target machine is normal or anomalous, under the condition that only normal samples constitute the training dataset.

2. PROPOSED METHOD

ProtoPNet is a novel method proposed for the task of image classification, offering the ability to interpret the network's reasoning process during classification [8]. With minimum alterations on the original network's architecture, the model can be of use in anomaly detection problems.

The generic ProtoPNet architecture for anomaly detection is presented in Fig.1. Let $X \in \mathbb{R}^{MxNxC}$ be the input to a convolutional neural network $G_\Theta : \mathbb{R}^{MxNxC} \rightarrow \mathbb{R}^{HxLxD}$ with parameters Θ. Each prototype $\{p_i\}_{i=1}^{k}$ is a tensor in $\mathbb{R}^{H_p x L_p x D}$, $H_p \leq H, L_p \leq L$ with the desired property that it resembles some patch/subtensor of the convolutional output $G_\Theta(X)$. This resemblance corresponds to the similarity between each prototype and some part of the original input X [8].

To compute the distance between the prototype p_i and every patch $z \in \mathbb{R}^{H_p x L_p x D}$ of the convolutional output $G_\Theta(X)$, the generalized L_2 convolution operator is introduced, where the L_2 norm operator replaces the inner product [9]. The resulting distance matrix $M_i \in \mathbb{R}^{(H-H_p+1)x(L-L_p+1)}$:

$$M_i = p_i \odot_{L_2} G_\Theta(X) \tag{1}$$

where $\odot_{L_2}$ is the L_2 convolution operator, can be used as a similarity heatmap if properly transformed and upsampled to the original input's height and width.

By selecting the minimum value of each distance matrix, the output of the ProtoPNet is the vector:

$$o = \begin{bmatrix} \min(M_1) \\ \min(M_2) \\ \vdots \\ \min(M_k) \end{bmatrix} \tag{2}$$

The output vector o quantifies the dissimilarity of every prototype with regard to some part of the convolutional output $G_\Theta(X)$. As there might exist different prototypes that characterize different sets of normal samples, the loss function and anomaly score for a sample X are formulated as follows:

$$\mathcal{A}(X) = \mathcal{J}_{\Theta,p}(X) = \min(o) \tag{3}$$

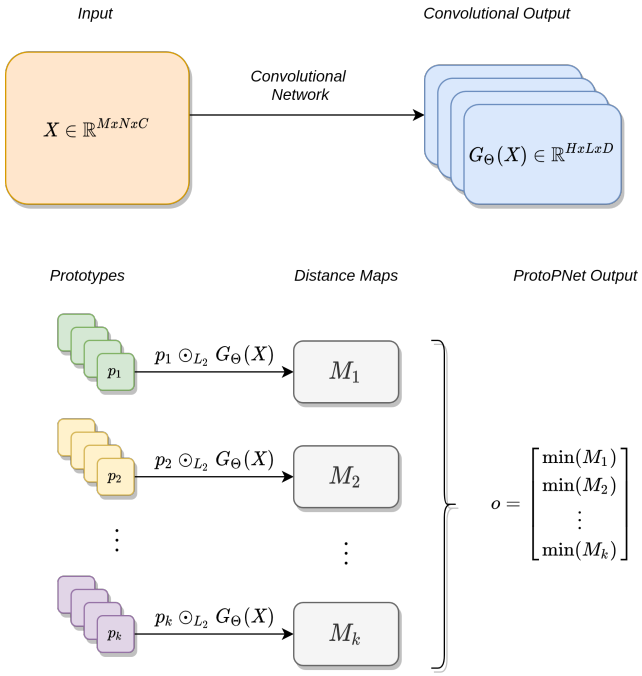

Figure 1: Generic ProtoPNet Architecture

The prototypes are trainable parameters that are initialized with random values. To impose the desired property of similarity between each prototype and some patch of the convolutional output, prototype projection is incorporated. Every r training epochs, each prototype is replaced by its most similar convolutional output patch for all training samples. More Formally, let $D_{\text{train}} = \{X_i\}_{i=1}^{N}$ be the available training dataset and $Z(X)$ be the set of all possible patches $z \in \mathbb{R}^{H_p x L_p x D}$ of $G_\Theta(X) \in \mathbb{R}^{HxLxD}$. Then:

$$p_i \leftarrow \text{argmin}_z(||p_i - z||^2) \; \forall z \in Z(X), \forall X \in D_{\text{train}} \qquad (4)$$

where $\leftarrow$ denotes the assignment operator and $\text{argmin}_z(w)$ returns the value of z that minimizes w.

2.1. ProtoPNet and DOCSVDD

DOCSVDD can be briefly described as a neural network $\phi_W : \mathbb{R}^{MxNxC} \rightarrow \mathbb{R}^K$, that maps the input space to a vector in $\mathbb{R}^K$. The training objective is the estimation of the network's parameters W, that map normal samples within the volume of a hypersphere of center $c \in \mathbb{R}^K$ and radius R [10]. A sample X is classified as an anomaly if it is mapped outside the hypersphere's volume. Both the cost function and the anomaly score for a sample X equate to the squared distance of the network's output from the center of the hypersphere:

$$\mathcal{A}(X) = \mathcal{J}_W(X) = ||\phi_W(X) - c||^2 \qquad (5)$$

Let $G_\Theta : \mathbb{R}^{MxNxC} \rightarrow \mathbb{R}^{HxLxD}$ be the convolutional network of a ProtoPNet with prototypes $\{p_i\}_{i=1}^{k}, p_i \in \mathbb{R}^{H_p x L_p x D}$. In the special case where $k = 1, H_p = H, L_p = L$, the L_2 convolution turns into the L_2 distance between the convolutional output and the prototype p_1. In this manner, the ProtoPNet turns into a convolutional DOCSVDD with center p_1. Furthermore, by utilizing the prototype projection technique, the center of the hypersphere shifts adaptively to its nearest point.

Finally, by utilizing more prototypes, ProtoPNet turns into a DOCSVDD with multiple hyperspheres/clusters that might encapsulate more complex structures in the dataset, or be used as a deep clustering algorithm.

3. EXPERIMENT SETTINGS

The dataset used is comprised partly of the datasets ToyADMOS [11] and MIMII [12]. It consists of six machine types (Fan, Pump, Valve, Slider, ToyCar, ToyConveyor), whereas each machine type corresponds to several physical machines of the same type, identified by a unique id. For each valid combination of machine type and machine id, the training dataset consists of only normal audio samples, while the testing dataset consists of both normal and anomalous samples.

It should be noted that a distinct model is trained and evaluated per machine id. The models are evaluated on the area under the curve (AUC) and the partial area under the curve (pAUC) where $0 \leq \text{FPR} \leq 0.1$. For the brevity of presentation, the scores of all machine ids of the same type are averaged.

3.1. Audio representation

Each audio sample is represented by its mel-spectrogram $S \in \mathbb{R}^{128x157}$ of 128 frequency bands, a window length of 128 ms and 50% overlap. The resulting spectrogram is log-scaled after adding an offset a:

$$S_p = 10 \log_{10}(S + a) \qquad (6)$$

where $a \approx 2.2\ 10^{-16}$ for the ToyADMOS machine types, and $a = 1$ for the MIMII machine types.

3.2. Training procedure

In general, a different architecture was selected per machine type as presented in T.1-T.4. The gradient descent optimization is conducted via the Adam optimizer for a batch size of 32 samples. The network parameters are optimized for 100 training epochs, and the parameters that result in the lowest validation loss are restored. Finally, prototype projection is performed every $r = 5$ epochs.

Table 1: Fan, Pump, Slider Architecture

Layer	Output Shape
Conv2D $8@(5,5,1)$	$(128, 157, 8)$
ReLU	$(128, 157, 8)$
MaxPool2D $(3,3)$	$(42, 52, 8)$
Conv2D $8@(5,5,8)$	$(42, 52, 8)$
ReLU	$(42, 52, 8)$
MaxPool2D $(3,3)$	$(14, 17, 8)$
Conv2D $16@(5,5,8)$	$(14, 17, 16)$
ReLU	$(14, 17, 16)$
MaxPool2D $(3,3)$	$(4, 5, 16)$
Conv2D $16@(5,5,16)$	$(4, 5, 16)$
ReLU	$(4, 5, 16)$
MaxPool2D (3,3)	$(1, 1, 16)$
Proto $1@(1,1,16)$	$\mathbb{R}$

Table 2: Valve Architecture

Layer	Output Shape
Conv2D $8@(5,5,1)$	$(128, 157, 8)$
ReLU	$(128, 157, 8)$
MaxPool2D $(2, 2)$	$(64, 78, 8)$
Conv2D $8@(5,5,8)$	$(64, 78, 8)$
ReLU	$(64, 78, 8)$
MaxPool2D $(2, 2)$	$(32, 39, 8)$
Conv2D $16@(5,5,8)$	$(32, 39, 16)$
ReLU	$(32, 39, 16)$
MaxPool2D $(2, 2)$	$(16, 19, 16)$
Conv2D $16@(5,5,16)$	$(16, 19, 16)$
ReLU	$(16, 19, 16)$
MaxPool2D (2,2)	$(8, 9, 16)$
Proto $1@(4,4,16)$	$\mathbb{R}$

Table 3: ToyCar Architecture

Layer	Output Shape
Conv2D $8@(5,5,1)$	$(128, 157, 8)$
ReLU	$(128, 157, 8)$
MaxPool2D $(2, 2)$	$(64, 78, 8)$
Conv2D $8@(5,5,8)$	$(64, 78, 8)$
ReLU	$(64, 78, 8)$
MaxPool2D $(2, 2)$	$(32, 39, 8)$
Conv2D $16@(5,5,8)$	$(32, 39, 16)$
ReLU	$(32, 39, 16)$
MaxPool2D $(2, 2)$	$(16, 19, 16)$
Conv2D $16@(5,5,16)$	$(16, 19, 16)$
ReLU	$(16, 19, 16)$
MaxPool2D (2,2)	$(8, 9, 16)$
Proto $3@(6,6,16)$	$\mathbb{R}^3$

Table 4: ToyConveyor Architecture

Layer	Output Shape
Conv2D $8@(5,5,1)$	$(128, 157, 8)$
ReLU	$(128, 157, 8)$
MaxPool2D $(2, 2)$	$(64, 78, 8)$
Conv2D $8@(5,5,8)$	$(64, 78, 8)$
ReLU	$(64, 78, 8)$
MaxPool2D $(2, 2)$	$(32, 39, 8)$
Conv2D $16@(5,5,8)$	$(32, 39, 16)$
ReLU	$(32, 39, 16)$
MaxPool2D $(2, 2)$	$(16, 19, 16)$
Conv2D $16@(5,5,16)$	$(16, 19, 16)$
ReLU	$(16, 19, 16)$
MaxPool2D (2,2)	$(8, 9, 16)$
Proto $1@(6,1,16)$	$\mathbb{R}$

4. RESULTS AND DISCUSSION

The average model performance for five independent runs is shown at T.5, where the $\pm$ sign denotes the deviation between different machine-ids of the same type.

Table 5: Model Performance (Avg $\pm$std)

Machine Type	AUC	pAUC
Fan	0.8557 ± 0.08	0.7868 ± 0.11
Pump	0.7733 ± 0.07	0.7369 ± 0.07
Slider	0.8303 ± 0.14	0.6959 ± 0.19
Valve	0.8724 ± 0.09	0.7648 ± 0.15
ToyCar	0.8211 ± 0.12	0.7235 ± 0.14
ToyConveyor	0.7035 ± 0.08	0.5992 ± 0.07

The model's reasonably high pAUC for the majority of the machine types is encouraging as to the competence of ProtoPNet in an unsupervised anomaly detection scenario. We suspect that the relatively lower results for ToyConveyor are due to the class imbalance imposed on the dataset (greater than 2:1 normal to anomalous test samples). Finally, spectrogram preprocessing has proven to be of immense importance regarding the model's performance while there is no one size fits all method since machine types from different datasets required unique preprocessing approaches.

5. REFERENCES

[1] Henriquez, Patricia, et al. "Review of automatic fault diagnosis systems using audio and vibration signals." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.5 (2013): 642-652.

[2] Márquez, Fausto Pedro García, et al. "Condition monitoring of wind turbines: Techniques and methods." *Renewable Energy* 46 (2012): 169-178.

[3] Yadav, Sandeep Kumar, et al. "Audio signature-based condition monitoring of internal combustion engine using FFT and correlation approach." *IEEE Transactions on instrumentation and measurement* 60.4 (2010): 1217-1226.

[4] Aggarwal, Charu C. "Outlier analysis." *Data mining.* Springer, Cham, 2015.

[5] Koizumi, Yuma, et al. "Unsupervised detection of anomalous sound based on deep learning and the Neyman–Pearson lemma." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.1 (2018): 212-224.

[6] Goldstein, Markus, and Seiichi Uchida. "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data." *PloS one* 11.4 (2016): e0152173.

[7] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41.3 (2009): 1-58.

[8] Chen, Chaofan, et al. "This looks like that: deep learning for interpretable image recognition." *Advances in Neural Information Processing Systems*. 2019.

[9] Ghiasi-Shirazi, Kamaledin. "Generalizing the convolution operator in convolutional neural networks." *Neural Processing Letters* 50.3 (2019): 2627-2646.

[10] Ruff, Lukas, et al. "Deep one-class classification." *International conference on machine learning.* 2018.

[11] Koizumi, Yuma, et al. "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection." *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).* IEEE, 2019.

[12] Purohit, Harsh, et al. "MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection." *arXiv preprint arXiv:1909.09347* (2019).