ANOMALY DETECTION USING THE MIDDLE LAYER OF THE CNN-CLASSIFICATION MODEL

Technical Report

Motonobu Uchikoshi

The Japan Research Institute, Limited
2-18-1 Higashi-Gotanda, Shinagawa-ku,
Tokyo 141-0022
uchikoshi.motonobu@jri.co.jp

ABSTRACT

I have confirmed the effectiveness of anomaly detection using the middle layer of the CNN-classification model. The model is 3-level classification model using 3 different normal datasets (normal data, normal data with noise, and a different type of normal data). The abnormality is defined for the distance from the normal data outputs area in latent space of the middle layer of the CNN-classification model. For characterizing the region occupied by normal data, clustering with a mixed Gaussian model. Though the average scores of the CNN model were below the AE-baselines, some tasks better scores than the baselines. So I tried ensembling the CNN model and the AE model.

***Index Terms*— CNN, KNN, GMM, AE**

1. INTRODUCTION

In this technical report, the model was trained using ToyADMOS and MIMII Dataset [1], [2], [3].

Unsupervised anomaly detection is expected to have the ability to detect unknown anomalies, however, the accuracy of detection tends not to be high when there are not enough “normal data”. For example, AE learns the latent space of normal data and outputs the degree of abnormality as a reproduction error, but since the model can only learn normal data during training, the model tends to regard as the abnormal one the data that is not related to any abnormalities.To avoid this, I created a CNN-based model that classifies 3 normal data (normal data, normal data with random noise, and different type of normal data).

By using latent space of the CNN model, We can use the information that does not exist at the time of learning and is not related to any abnormalities. In creating the CNN-based model, I referred to the model that used the intermediate output of CNN [4].

2. THREE NORMAL DATASETS

This section describes the three data sets to be input.

The first is the target normal data (fan_id_00). The second is the data with random noise in the Hz axis direction (Figure 1). The third is the another normal data (pump_id_00).

As an example, the model inputs for fan_id_00 are:

・Fan_id_00 normal data
・Fan_id_00 normal data + random noise
・Normal data of another data type (pump_id_00, etc.)

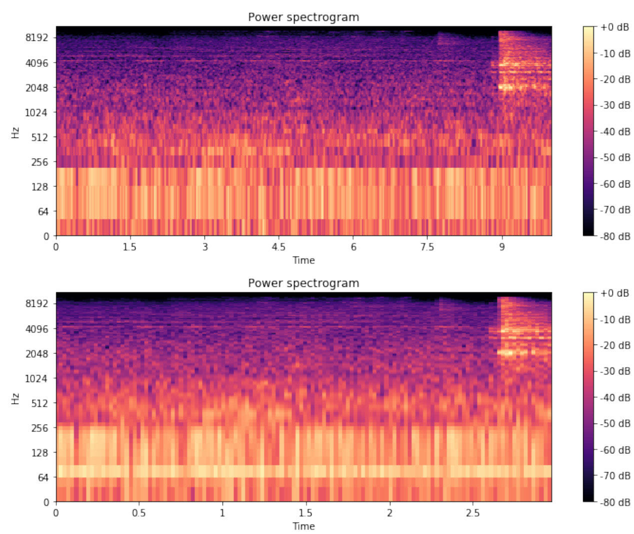

Figure 1: Example of the data with random noise in the Hz axis direction. (The figure below is the one with noise added to the figure above.)

3. CNN MODEL CONFIGURATION

This section describes the configuration of the CNN model. (Figure 2) The input data is the wav data converted into log mel spectrogram and reshaped into two-dimensional data of (224, 224). There are three types of input data: normal data set, data set with noise added to normal data, and data set of different type from normal data. Input it to the 2d-CNN model and perform ternary classification.

Input test data to the trained model and take out the output of the hidden layer. It is expected that the output can be classified into three clusters, which is the number of types of input data.

Then, the distance between the point group of the training normal data and the point of the test data is calculated by KNN. If the test data is abnormal data, the distance is expected to be large.

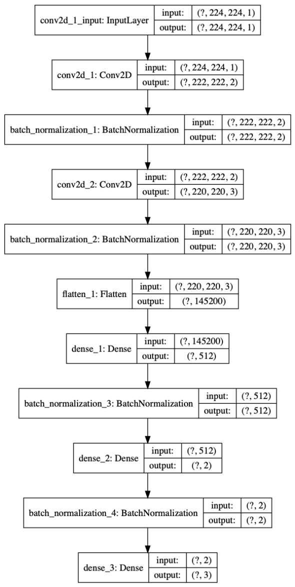

Figure 2: The CNN model structure (hidden layer, dense_2, is the layer for KNN)

4. USE OF GAUSSIAN MIXTURE DISTRIBUTION

The output of the CNN middle layer of normal data tends to have high variance. This is probably because the training normal data contains noise and data close to other types of data. In the case that the variance is high, the accuracy is not stable, so the output of the CNN hidden layer was clustered with a Gaussian mixture model. (The number of kernels is fixed at 2) By using a cluster with a large number of data as normal data for KNN, the variance was low, the accuracy was stable, and the accuracy was high on average. The number of kernels is fixed at 2 so as not to improve the accuracy by judging the distribution of test data.

5. ENSEMBLE WITH AUTO-ENCODER

As a result of checking AUC and pAUC in the development dataset for the CNN model, it was found that there are valid data and invalid data. Therefore, for the test data, we output the degree of the abnormality for each model, and summed up a fixed ratio for all the data. As a result, validation on the development data resulted in an average improvement in pAUC compared to the baseline.

6. RESULTS

The test result of the development data set is described. (Figure 3) the average scores of the CNN model were below the AE-baselines, some tasks better scores than the baselines.

dataset_name	CNN_AUC	CNN_pAUC	base_AUC	base_pAUC
ToyCar_id_01	0.51921	0.51970	0.81439	0.670663
ToyCar_id_02	0.58671	0.51803	0.86595	0.798553
ToyCar_id_03	0.62135	0.53147	0.63368	0.552844
ToyCar_id_04	0.74002	0.60517	0.85266	0.653217
ToyConveyor_id_01	0.50313	0.50281	0.78816	0.658355
ToyConveyor_id_02	0.51302	0.50718	0.64395	0.568110
ToyConveyor_id_03	0.68221	0.58328	0.72827	0.602969
fan_id_00	0.55619	0.49448	0.53971	0.491659
fan_id_02	0.57395	0.53232	0.72267	0.549333
fan_id_04	0.57077	0.49955	0.60483	0.527223
fan_id_06	0.71050	0.54258	0.73278	0.525004
pump_id_00	0.73070	0.61556	0.67028	0.545275
pump_id_02	0.69117	0.58697	0.61928	0.576808
pump_id_04	0.52090	0.49128	0.89460	0.646150
pump_id_06	0.55853	0.50226	0.75824	0.546505
slider_id_00	0.74433	0.54307	0.96238	0.815642
slider_id_02	0.74292	0.54442	0.79139	0.585130
slider_id_04	0.68112	0.54595	0.93865	0.610486
slider_id_06	0.52652	0.51023	0.71382	0.491789
valve_id_00	0.96580	0.85520	0.66235	0.519737
valve_id_02	0.52775	0.53014	0.67183	0.512820
valve_id_04	0.85058	0.51368	0.72942	0.512820
valve_id_06	0.56950	0.48889	0.54075	0.489740

Figure 3: The test result of the development data set (the CNN model is "CNN", and AE-baseline model is "base".)

7. REFERENCES

[1] Yuma Koizumi, Shoichiro Saito, Hisashi Uematsu, Noboru Harada, and Keisuke Imoto. *ToyADMOS: a dataset of miniature-machine operating sounds for anomalous sound detection.* In Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 308–312. November 2019. URL: https://ieeexplore.ieee.org/document/8937164.

[2] Harsh Purohit, Ryo Tanabe, Takeshi Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi. *MIMII Dataset: sound dataset for malfunctioning industrial machine investigation and inspection.* In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019), 209–213. November 2019. URL: http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Purohit_21.pdf.

[3] Yuma Koizumi, Yohei Kawaguchi, Keisuke Imoto, Toshiki Nakamura, Yuki Nikaido, Ryo Tanabe, Harsh Purohit, Kaori Suefusa, Takashi Endo, Masahiro Yasuda, and Noboru Harada. *Description and discussion on DCASE2020 challenge task2: unsupervised anomalous sound detection for machine condition monitoring.* In arXiv e-prints: 2006.05822, 1–4. June 2020. URL: https://arxiv.org/abs/2006.05822.

[4] Golan, Izhak and El-Yaniv, Ran "Deep Anomaly Detection Using Geometric Transformations" arXiv preprint arXiv:1805.10917, 2018