Senvis-Net: Learning from Imbalanced Machinery Data by Transferring Visual Element Detectors

Qingwei Guo, Yoshinori Miyamae, Zhongjun Wang, Koji Taniuchi, Huazhong Yang, and Yongpan Liu

***Abstract*—With the development of sensor network technologies and the popularization of Industry 4.0, data-driven machine health monitoring has become increasingly important, not only to save maintenance costs of factory machinery, but also to guarantee the safety of factories. However, traditional data-driven algorithms are limited by two aspects. Firstly, the information fusion of multiple sensors heavily relies on domain knowledge. Secondly, imbalanced distribution of machinery data brings a challenge for the machine learning algorithm performance. In order to tackle these issues, we propose a general methodology to organize collected sensor data into image form and utilize visual element detectors learned by a pre-trained convnet to explore meaningful information hidden in data. We also design a Convolutional Neural Network (CNN) model, named Senvis-Net. Applied to an imbalance learning task of remaining useful life (RUL) prediction, our model outperforms the state-of-the-art CNN that learns directly from sensor data. Moreover, transferring visual element detectors can bring another 20.1% ~ 97% performance benefits depending on severity of imbalance.**

***Index Terms*—Machine health monitoring, convolutional neural network, imbalanced data, transfer learning.**

I. Introduction

Significant development of sensor networks has generated large amount of data in factories. On the other hand, deep learning has made great achievements in fields of computer vision, speech recognition, and natural language processing, due to the emergence of massive data and powerful computing resource. While revolutionary changes brought by deep learning to the area of machine health management should be expected, the corresponding progress is slow. There are two major challenges to overcome. Firstly, machine domain knowledge is required to identify representative features in data on a case by case basis by individual applications. Secondly, the class distribution of machinery data in real life normally follows a highly-skewed pattern, which means most data samples belong to a few categories (e.g., faulty data is much harder to collect than healthy data). Similar problems occur in some computer vision area, such as medical image analysis [1] and it has been proven that they can be partly solved by transfer learning from irrelevant ImageNet dataset [2].

Convolutional neural network (CNN) can automatically learn a hierarchical feature representation, which makes it an appealing tool to explore high dimensional data. Since deep CNN has been applied successfully to the task of ImageNet classification challenge, low-level filters in trained CNN have been found out to be detectors for elementary components in natural images [3]. Considering these low-level filters learned by CNNs in computer vision area may be applied to artificial images constructed by sensor data, we propose a methodology that organizes collected sensor data into image form and then transfers those visual information to help learn meaningful information hidden in data. The methodology can be broadly applied to different machine health management systems.

We apply the proposed approach on CMAPSS (Commercial Modular Aero-Propulsion System Simulation) datasets [4], which are run-to-failure datasets from a turbofan engine simulation model generated and disseminated by the prognostic center of excellence at NASA Ames Research Center (https://ti.arc.nasa.gov/tech/dash/). High variability of the datasets due to sensor noise, effects of operating conditions and simultaneous presence of multiple fault modes makes it suitable to study two challenges we mentioned above. In this paper, we organize CMAPSS datasets into images following the principle of maximizing local correlation, and adopt supervise learning with RUL labels offered in the datasets. In the experiments section, it would be showed that when training with data of highly imbalance, generalization ability of models utilizing transfer learning are much better than those not. Visualization of high-level feature maps of models shows that meaningful concepts are easier to recognize in models with transfer learning than those without.

The purpose of this paper is to present a general machinery data exploring framework, focusing on alleviating the imbalanced data problem in machine health area. The remainder of this paper describes our method and experiments in detail. It first reviews related works in Section II. Section III presents the problem formulation and our approach. Section IV presents experiments and Section V gives a conclusion.

II. Literfature Review

A. Deep Learning in Machine Health Area

Conventional multilayer perceptron (MLP) has been applied in field of machine health area for many years [5], but deep learning techniques have just recently been applied to a large number of machine health monitoring problems. Stacked auto-encoder (SAE)-based models [6], Deep Belief Network (DBN)-based models [7], CNN-based models [8]

Manuscript received July 7, 2018; revised September 6, 2018.

Qingwei Guo, Huazhong Yang, Yongpan Liu are with Tsinghua University, China (e-mail: gqw15@mails.tsinghua.edu.cn, yanghz@tsinghua.edu.cn, ypliu@tsinghua.edu.cn).

Yoshinori Miyamae, Zhongjun Wang, Koji Taniuchi are with ROHM Semiconductor, Japan (e-mail: Yoshinori.Miyamae@dsn.rohm.co.jp, zhongjun.wang@res.rohmchina.com.cn, koji.taniuchi@dsn.rohm.co.jp).

doi: 10.18178/ijmlc.2018.8.5.722

and Recurrent Neural Network (RNN)-based models [9] were applied in fault diagnosis or machine health state prediction tasks. We distinguish ourselves from these literature by transferring visual information to help extracting high level concepts for multi-sensor fusion and focusing on imbalanced data problem. Similar works like [10] pre-processed raw vibration signals to generate 2D images, and employed histogram of original gradients (HOG) descriptor to extract features. Different from handcrafted HOG descriptor, low-level filters extracted from trained CNN model are natural detectors of fine-grain visual element. And [8] utilized a 2D-CNN model for four categories rotating machinery conditions recognition, whose input is DFT of two accelerometer signals.

B. Data Imbalance

Issue with the imbalanced learning problem is the ability of imbalanced data to significantly compromise the performance of most standard learning algorithms. Different effective strategies [11] were raised to tackle the imbalanced learning problem. [12] adopted resampling methods to handle imbalanced data distribution for regression. [13] applied cost-sensitive approach to deal with multi-class learning and [14] adopted boosted SVM with active learning strategy for binary imbalanced problem. However, according to [15], [16], [17], degree of imbalance is not the only factor that hinder learning. Data set complexity is the primary determining factor of classification deterioration, which, in turn, is amplified by the addition of a relative imbalance. The core of imbalanced learning problem lies on the fact that we know little about the minority. Data imbalance affecting the performance of machine learning algorithms is similar to size of objects affecting our observation. Before the emergence of microscope, we know little about micro objects. We believe low-level filters learned by CNNs can play a role of microscope, helping deconstruct source data and boosting the performance of machine learning algorithm on minority cases.

III. Problem Formulation and Methods

A. Datasets and Metrics

CMAPSS datasets [18], which we explore in this paper, are run-to-failure datasets from a turbofan engine simulation model. It consists of multiple multivariate time series, arranged in an R-by-26 matrix, and R is the number of data points. Each data point is a snapshot of data taken during a single operational cycle and can be represented as $[x_1, x_2, x_3, ..., x_{26}]$. x_1 represents the engine number and x_2 represents the operational cycle number. x_3, x_4, x_5 are three operation settings which have a substantial effect on engine performance. $x_6 \sim x_{26}$ represent values from 21 different sensors. Data points can be divided into six operation conditions by the three operation setting values. Table I summarizes the four sub-datasets of CMAPSS datasets. The objective is to predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate. We apply standard fully supervised CNN models for implementation throughout the paper and the tunable parameters of all three models are chosen using 5-fold cross-validation procedure based on the training set only.

For this dataset, a piece-wise linear degradation ground truth is favored in previous research, which limits the maximum value of the RUL function and we set the maximum RUL to be 200 for the reason that the average run length of data is 209.

To evaluate the performance of different models on the data-sets, we use the Scoring Function (SF), which is officially suggested by providers of CMAPSS datasets. SF is defined as:

$$SF = \begin{cases} \sum_{i=1}^{N} \left(e^{-\frac{h_i}{13}} - 1 \right) & h_i < 0 \\ \sum_{i=1}^{N} \left(e^{\frac{h_i}{10}} - 1 \right) & h_i \geq 0 \end{cases} \quad (1)$$

N is the number of engines in test set, and $h = (RUL_{predicted} - RUL_{true})$. The SF penalizes late predictions (too late to perform maintenance) more than early predictions (no big harms although it could waste maintenance resources).

TABLE I: Description of the Four Turbofan Degradation Datasets Available from NASA Repository [19]

Datasets	Fault modes	Conditions	Train Units	Test Units
FD001	1	1	100	100
FD002	1	6	260	259
FD003	2	1	100	100
FD004	2	6	249	248

B. Sensor Signal to Image Transformation

The main idea of this paper is to reorganize the serial sensor data into image form and to utilize visual element detectors to help explore complex information hidden in sensor data. Transformation from sensor data to images is flexible depending on the nature of source data as long as sensors relationships can be encoded into local relationships on the transformed image. Since we intend to get rid of the restriction of domain knowledge, we just put sensors data in a zigzag fashion without considering physical meaning of sensors and use a trick of copy to shorten the distance of different sensors.

As Fig. 1(a) shows, sensors data can be transformed into a 120-by-120 gray image. Because the kernel size of the first convolution layer of our CNN is 11, we make each sensor value occupy 10-by-10 pixels. The transformed image would be cropped and rescaled into a 227-by-227 RGB channel image before sent to our CNN.

C. Network Architecture

As Fig. 1(b) shows, detailed information of Senvis-Net architecture can be checked in the table embedded in the figure. Senvis-Net is composed by following components:

1) *Convolutional layer:* convolutional layers apply a convolution operation to the input feature maps $\mathbf{V}$ as:

$$Z_{j,k}^{i} = conv(\mathbf{K}, \mathbf{V}, s)_{j,k}^{i} \quad (2)$$

$$= \sum_{l,m,n} \left[V_{(j-1)*s+m,(k-1)*s+n}^{l} * K_{m,n}^{i,l} \right] \quad (3)$$

where $Z_{j,k}^{i}$ the activated neuron in the i^{th} output feature map. $K_{m,n}^{i,l}$ represents the parameter in the (m,n) position of the filter kernel connecting the l^{th} input and the i^{th} output.

$V_{m,n}^{l}$ is the convolved input neuron and s denotes the stride s steps should be moved when $\mathbf{K}$ convolves $\mathbf{V}$.

2) *ReLU layer:* ReLU refers to the Rectifier Unit. Mathematically, it is described as:

$$U_{j,k}^{i} = \max(0, Z_{j,k}^{i}) \quad (4)$$

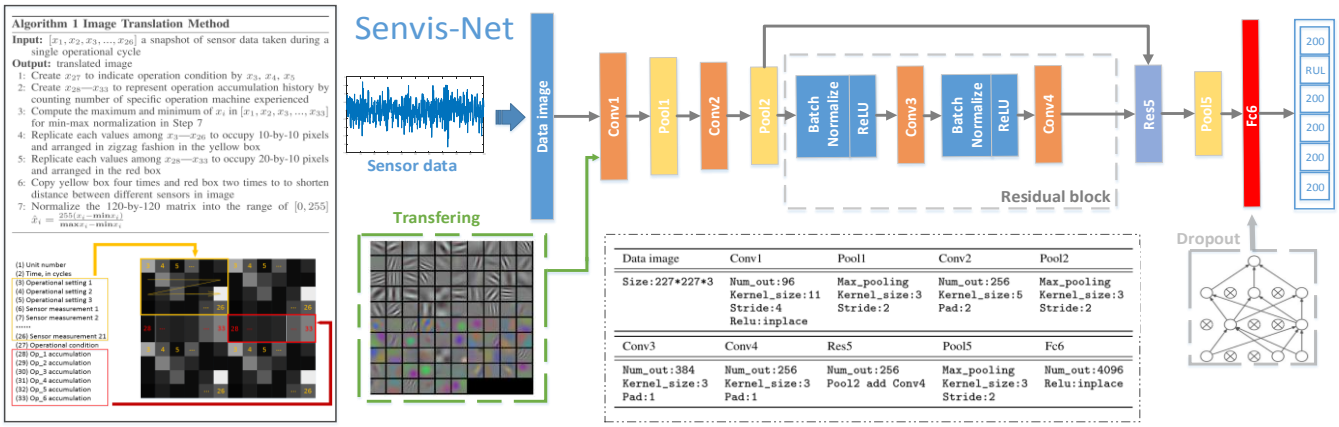
3) *Pooling layer:* pooling layer just picks the largest element from a sliding window that traverses over the entire matrix.

4) *Batch normalization layer:* the Batch normalization is a whitening operation used to reduce internal covariate shift in neural networks, which would benefit the whole model training process [20].

5) *Fully connection layer:* fully connection layers connect every neuron in one layer to every neuron in another layer:

$$O_j = \sum_i \left[I_i^{flatten} * F_{i,j} \right] \quad (5)$$

where $I_i^{flatten}$ is the i^{th} neuron of the flatten version of input feature maps. $F_{i,j}$ is the weight connecting the i^{th} input neuron and the j^{th} output neuron and O_j is the output neuron.



(a) (b)

Fig. 1. **(a) Image translation (b) Network Network architecture:** Conv indictates convolutional layer. Pool indicates pooling layer. *Fc* indicates fully connection layer. Num_out indicates number of output feature maps. Zero padding is used when necessary. Stride of one and no padding are used by default. The concept of residual block follows [23].

A skip connection is introduced to add feature maps after Pool2 to ones after Conv4 in Res5 layer, which enables a clean information path when the loss back propagates efficiently to affect weights in Conv1 and Conv2, reducing the gradient vanishing problem when training our CNN. Dropout [21] (dropping out connection randomly during training process) is employed to Fc6 to reduce overfitting.

6) *Output layer (fully connection):* with each data point as input, Senvis-Net is expected to output a 1-by-6 matrix. The ground truth RUL is arranged in one of six locations according to which operation condition the data point belongs to, while maximum RUL of 200 is arranged in other locations. For example, if a data point belongs to the 2^{nd} operation condition, then the output should be $[200, RUL, 200, 200, 200, 200]$. It means that Senvis-Net are expected to complete a classification function and a regression function at the same time. The Mean Square Error (MSE) between prediction and ground truth is chosen to be calculated for updating the whole network because it is easy to optimize. It is necessary to emphasized that MSE is just chosen to be the loss function when optimize our model, not used to evaluate performance of different models in the following experiments.

D. Transfer Visual Element Detectors

Filters of the first convolutional layer in Alex-Net [22], trained on ImageNet dataset, have been found out to be visual element detectors for lines, dots and other visual element as Fig. 1 shows in the green box. Since configuration of Conv1 in our CNNs is the same to Alex-Net, we can selectively initialize filters with those detectors.

Now that those transformed images carry all the information we hope to learn, an understanding in the structure of these carriers should help CNNs learn in a more reasonable way. Let us denote Conv1 as $\mathbf{Z^1} = conv(\mathbf{K^1}, \mathbf{V}, s)$ and before training our model, $\mathbf{K^1}$ can be chosen in two ways: 1) initialized by random numbers $\mathbf{K^1_{rand}}$ or 2) initialized by those visual element detector $\mathbf{K^1_{tran}}$. Without loss of generality, forward propagation of the first two convolution layers can be denoted as:

$$Z_1^1 = \mathrm{Re}\,LU(V_1 \otimes K_1^1 + V_2 \otimes K_2^1) \quad (6)$$

$$Z_2^1 = \mathrm{Re}\,LU(V_1 \otimes K_3^1 + V_2 \otimes K_4^1) \quad (7)$$

$$Z_1^2 = \mathrm{Re}\,LU(Z_1^1 \otimes K_1^2 + Z_2^1 \otimes K_2^2) \quad (8)$$

where Z_i^j is the i^{th} output activated neurons in the j^{th} layer, K_i^j is the i^{th} kernel in the j^{th} layer, V_i is the i^{th} image and $\otimes$ denotes matrix multiplication. For sake of simplicity, we assume items in the brackets of ReLU operation is always positive. Then the backpropagation for updating (K_1^2, K_2^2) would be:

$$\Delta Z_1^2 = \partial Loss / Z_1^2 \tag{9}$$

$$\Delta K_1^2 = Z_1^{1T} \otimes \Delta Z_1^2 \tag{10}$$

$$\Delta K_2^2 = Z_2^{1T} \otimes \Delta Z_1^2 \tag{11}$$

Let us analyze what would happen in the very beginning of the training process.

In the transfer situation, K_i^1 is initialized by $\mathbf{K}_{\mathbf{tran}}^{\mathbf{1}}$. Assume (K_1^1, K_2^1) be detectors for vertical and horizontal lines and (K_3^1, K_4^1) be detectors for curves. Considering the characteristics of the transformed images, in forward propagation, Z_1^1 would be strongly activated and Z_2^1 would be weakly activated. It would lead to $\left\|\Delta K_1^2\right\|_2 \gg \left\|\Delta K_2^2\right\|_2$ during the next round of backward propagation, which indicates just a little portion of K_i^2 become important in future training process.

In the non-transfer situation, K_i^1 is initialized by $\mathbf{K}_{\mathbf{rand}}^{\mathbf{1}}$, which would lead Z_i^1 to a random output. Then the backward propagatted gradient ΔK_i^2 would also be random.

As a result, we may expect K_i^2 learned in the transfer situation be sparser and more meaningful while K_i^2 learned in the non-transfer situation be in less control due to random information flow of gradients. In another word, by transfer learning we make the learning of low level concepts more reasonable and it should help neural network make an efficient gradient feedback to learn information about minority class.

IV. Experiments

A. Overall Performance of Senvis-Net

The task of our CNNs is to estimate RUL of engines in test sets. The metric to evaluate the bias from ground truth is official Scoring Function (SF). We first examines the overall performance of our Senvis-Net (T) with transfer learning on the four sub-datasets.

We use mini-batch stochastic gradient descent (SGD) optimizer and step learning policy with base learning rate of 5E-6, batch sizes of 4 and momentum of 0.9 throughout our experiments. We initialized all the weight to zero mean Gaussian noise with a standard deviation of 0.01.

Fig. 2 shows some examples of estimation in all four sub-datasets by Senvis-Net (T). It can be seen that the CNN is able to learn the degradation process. And as it estimates RUL when end of life, it seems more accurate than when initial. It is consistent to the intuition that life time of machines gradually falls into specific facts from random events as they degrade.

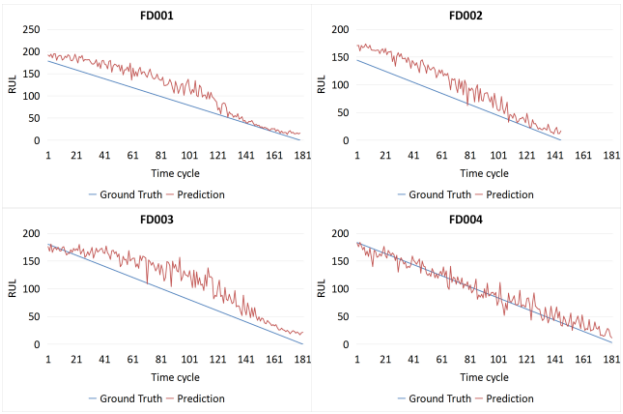


Fig. 2. Sample prediction.

Generally speaking, network of a larger size has a larger capacity. Capacity of a model is its ability to fit a wide variety of functions. Models with low capacity may struggle to fit the train set. Models with high capacity can overfit training data by memorizing properties of the train set that do not preserve in the test set. Since CMAPSS datasets contain data of different complexity, we also tests shadowNet (T), which removes the residual block of Senvis-Net(T), making it only contains two convolutional layers.

Table II records the best results we can get from the two models we trained. Comparison of network depth shows that shadowNet (T) only gets a pleasing result on FD001, which is the simplest dataset, while obvious deterioration can be seen on FD002, FD003, FD004. For example, Senvis-Net (T) gets 2.71E+04 of SF while shadowNet (T) gets 4.17E+05 on FD004 test set, which is most complex sub-dataset. Remember the smaller SF means more accurate our model predicts RUL. It suggests that more complex the data is, higher capacity of network is demanded.

TABLE II: SF for Deep and Shallow Models on Cmapss Datasets

Datasets	FD001	FD002	FD003	FD004
Senvis-Net (T)	1.47E+03	1.06E+04	5.27E+03	2.71E+04
shallowNet (T)	1.46E+03	2.07E+04	6.36E+03	4.17E+05

B. Performance on Imbalance Cases

In order to evaluate performance of our CNNs on imbalanced dataset, we intentionally create two imbalanced cases. In Case-1, we use all the training data from FD004 as training set, and the trained model is used to predict the RUL of test sets from the four datasets. Since FD004 dataset incorporates 2 fault modes and 6 operational conditions, we believe somewhat imbalanced information of the other three datasets also exist in FD004 training dataset. In Case-2, we use all 100 units in FD001 training data and only 10 units in FD004 training data as training set. The reason of combining FD001 and FD004 is that FD001 is the simplest set while FD004 is the most complex set, and large number of concepts such as different operating conditions do not exist in FD001. In another word, FD001 ~ FD003 are treated as minority class in Case-1 while FD004 is treated as minority class in Case-2. To make a fair comparison and demonstrate the ability of different models in digging minority class information, we define an I-indicator by dividing the minority class performance by the majority class performance of models

respectively ($I_{minor} = SF_{minor} / SF_{major}$).

We experiment two models, a transfer-version Senvis-Net (T) with transfer learning and a scratch-version Senvis-Net (S) without transfer learning, and compare them with a CNN architecture designed by [24] to learn directly from sensor data, which is denoted as A*STAR-CNN. Performance on minority class and learned high-level concept are both in our consideration when evaluating these models.

TABLE III: SF FOR VARIOUS MODELS ON IMBALANCE CASES

Datasets	FD001	FD002	FD003	FD004
		Case-1		
A*STAR-CNN	4.55E+11	6.88E+04	5.32E+11	6.35E+04
Senvis-Net (S)	1.60E+05	1.83E+04	4.89E+05	2.79E+04
Senvis-Net (T)	1.74E+04	1.42E+04	1.56E+05	2.71E+04
		Case-2		
A*STAR-CNN	3.79E+03	n.a.	n.a.	2.88E+09
Senvis-Net (S)	1.94E+03	n.a.	n.a.	2.86E+06
Senvis-Net (T)	1.94E+03	n.a.	n.a.	8.56E+04

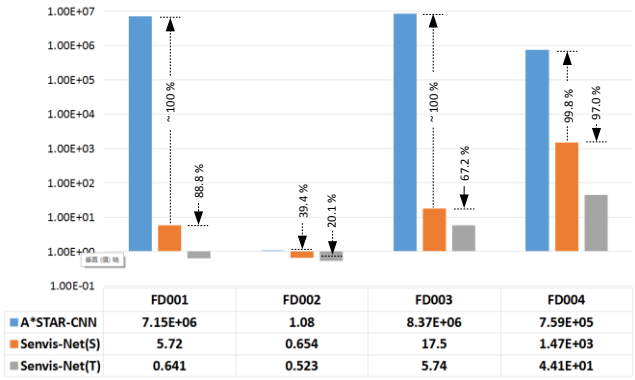


Fig. 3. I-indicator of SF for minority class.

From Table III, it can be seen that the three models work very well on majority class, but differ a lot when inferring minority class. For example in Case-1, A*STAR-CNN behaves terribly on minority classes like FD001 (4.55E+11 of SF) and FD003 (5.32E+11 of SF), which means that it almost learns nothing about FD001 and FD003 from training data. More apparently shown in Fig. 3, our proposed model outperforms A*STAR-CNN, and we think that is because we make our model structure adapted to the characteristic of the CMAPSS datasets, for instance the output layer of Senvis-Net.

Transfer learning boosts the performance of our network further on minority cases. Since the only difference between FD002 and FD004 is number of fault modes involved, it seems that FD002 should be regarded as majority class even though we treat it as minority class in problem setup in this paper. In Case-1, as imbalance severity in FD002 is not as serious as FD001 and FD003, transferring visual element detectors only brings 20.1% performance on FD002, while it can bring 88.8% on FD001 and 67.2% on FD003. In Case-2, as imbalance ratio reaches roughly 10:1 between FD001 and FD004, Senvis-Net (T) can get 97.0% performance benefit compare to Senvis-Net (S). Especially we notice in Case-2, Senvis-Net (T) acquire 8.56E+04 of SF on FD004, almost catching up with 2.71E+04 of SF when Senvis-Net (T) is trained by all 249 train units of FD004.

C. High-Level Concept Analysis

In order to understand what the networks have learned, we visualize feature maps [25] after each Conv layer in the three models. Since less meaningful information can be observed from feature maps of A*STAR-CNN, we just present feature maps of Senvis-Net (T) and Senvis-Net (S).

1) Feature study between transfer and non-transfer

As Fig. 4 shows, all feature maps are presented when Senvis-Net (S) and Senvis-Net (T) process transformed image data in different convolutional layers. A little square indicates a feature map, for example Conv1 outputs 96 feature maps and Conv3 outputs 384 feature maps. By comparing the difference between the "scratch" model (S) and the "transfer" model (T), we can get an insight on the question why "transfer" model (T) performs better than "scratch" model (S). In Conv2, feature maps in the T model appear much sparser than the S model, which is consistent with the theoretical derivation above. It can be found that redundant feature maps have been created in Conv3 and Conv4 of the S model. Especially feature maps in Conv4 of the S model almost can be divided into two groups by human eyes. That means in the S model, filters of similar function have been copied in Conv4 of the S model during training process. On the other hand, feature maps of the T model in Conv3 and Conv4 keep diverse, and high-level concepts can be easier to recognize in Conv4 of the T model, like the 71st feature map in Conv4, which we would explain later.

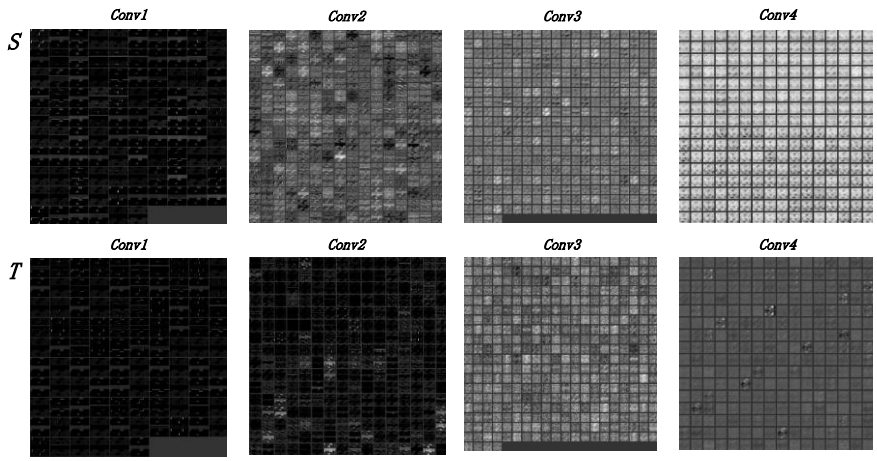
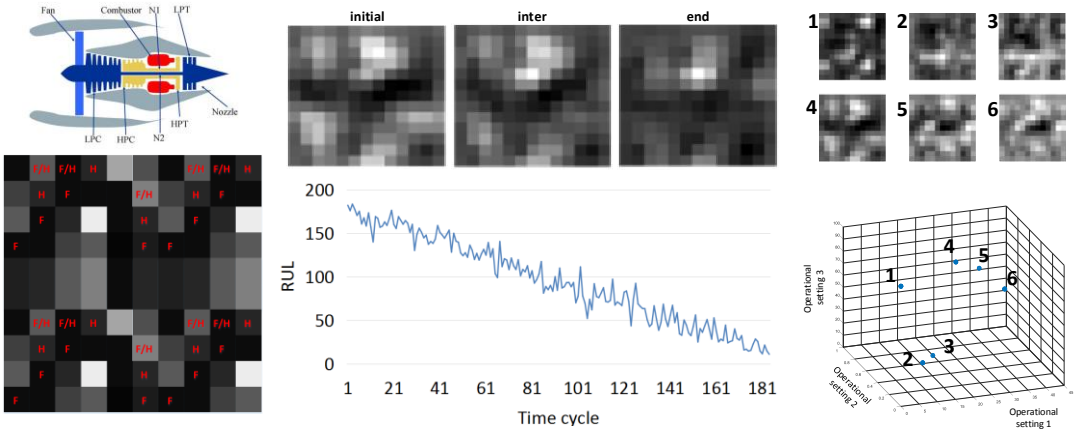


Fig. 4. **Feature maps when Senvis-Net (S) and Senvis-Net (T) process transformed image data:** Upside row indicates "scratch" model (S), while downside row indicates "transfer" model (T).



(a) (b) (c)

Fig. 5. **Feature visualization:** (a) Faults-related sensor distribution on images. (b) Feature maps indicating the engines are at the initial, intermediate and end of run. (c) Similarity of six exclusive patterns under six different operating condition is consistent to their positions in feature space.

2) *Feature representing operation*

We try to analyze those high-level features from view of nature in sensor data. As Fig. 5(a) shows, we check physical meaning of 21 sensors from [26] and find out sensors appear to be related to the two faults: Fan Degradation (F) and HPC Degradation (H), and mark them out on the image we composed.

We present the 71st feature map after Res5 Layer of Senvis-Net (T) in Fig. 5(b) and Fig. 5(c). The network seems to treat downside related sensor areas of images for measurement of RUL, while activation of those areas fade as engines degrade. Exclusive patterns with respect of different conditions can be seen in the overall feature maps. We compare those patterns among 6 conditions and find out that similarity between them is consistent to their positions in feature space. For example, patterns under 4^{th}, 5^{th} and 6^{th} condition have some kind of similarity while they are adjacent in the feature space.

V. Conclusion

In this paper, we propose a general framework to explore machinery data, trying to leverage visual element detectors learned by a pre-trained CNN to tackle challenges faced in machine health area. By organizing sensors data into image form, and utilizing transfer knowledge from computer vision area, we successfully explore complicated indication (including degradation process and operation condition about target engines) from different sensors with little domain knowledge. Our experiments show that transferring low-level visual element information can help extracting more meaningful features in sensor data, which may help us understand behaviors of sensor signal from target machines. We also prove that transferring visual element detectors does help alleviate imbalance data dilemma in machine health area, while it can bring obvious performance benefits of our deep learning model when lack of data in minority classes.

In our future works, firstly we would add Recurrent Neural Network models, which can learn the temporal dynamics of the serial sensor data, into our framework to treat sensors data as form of video, instead of static images in this paper and it can be expected to bring improvement on our framework. Secondly, we would explore how strategy of transforming from sensors data to image affects the performance of our models. Lastly, we may try to figure out more information reflected by those sensors signal.

Our work not only helps two major challenges in area of machine health management, but also builds a bridge between machinery application and computer vision application. Employing more and more state-of-the-art deep learning technologies in computer vision area can be expected to have a huge impact on real-world machinery applications.

Acknowledgment

This work was supported in part by NSFC Grant 61674094 and Beijing Innovation Center for Future Chip. We would like to thank the support from ROHM Semiconductor.

References

[1] C.-K. Shie, C.-H. Chuang, C.-N. Chou, M.-H. Wu, and E. Y. Chang, "Transfer representation learning for medical image analysis," in *Proc. 2015 37th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 711–714.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255, 2009.

[3] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. European Conference on Computer Vision*, Springer, 2014, pp. 818–833.

[4] A. Saxena and K. Goebel, "Phm08 challenge data set. nasa ames prognostics data repository," *Moffett Field*, CA, 2008.

[5] B Samanta and K. R. Al-Balushi, "Artificial neural network based fault diagnostics of rolling element bearings using time-domain features," *Mechanical Systems and signal Processing*, vol. 17, no. 2, pp. 317–328, 2003.

[6] L. Guo, H. L. Gao, H. F. Huang, X. He, and S. C. Li, "Multifeatures fusion and nonlinear dimension reduction for intelligent bearing condition monitoring," *Shock and Vibration*, 2016.

[7] S. Y. Shao, W. J. Sun, P. Wang, R. X. Gao, and R. Q. Yan, "Learning features from vibration signals for induction motor fault diagnosis," in *Proc. International Symposium on Flexible Automation (ISFA)*, 2016, pp. 71–76.

[8] O. Janssens, V. Slavkovikj *et al.*, "Convolutional neural network based fault detection for rotating machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.

[9] R. Zhao, J. J. Wang, R. Q. Yan, and K. Z. Mao, "Machine health monitoring with lstm networks," in *Proc. International Conference on Sensing Technology*, 2016, pp. 1–6.

[10] H. Oh, B. C. Jeon, J. H. Jung, and B. D. Youn, "Smart diagnosis of journal bearing rotor systems: Unsupervised feature extraction scheme by deep learning," 2016.

[11] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Progress in Artificial Intelligence*, pp. 1–12, 2016.
[12] L. Torgo, P. Branco, R. P. Ribeiro, and B. Pfahringer, "Resampling strategies for regression," *Expert Systems*, vol. 32, no. 3, pp. 465–476, 2014.
[13] Z. H. Zhou and X. Y. Liu, "On multi-class cost-sensitive learning," in *Proc. National Conference on Artificial Intelligence*, pp. 567–572, 2010.
[14] M. ZiÄ Żba and J. M. Tomczak, "Boosted SVM with active learning strategy for imbalanced data," *Soft Computing*, vol. 19, no. 12, pp. 3357–3368, 2015.
[15] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.
[16] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.
[17] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.
[18] A Saxena and K Goebel, "Turbofan engine degradation simulation data set," *NASA Ames Prognostics Data Repository*, 2008.
[19] E. Ramasso and A. Saxena, "Performance benchmarking and analysis of prognostic methods for cmapss datasets," *International Journal of Prognostics and Health Management*, vol. 5, no. 2, pp. 1–15, 2014.
[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.
[21] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
[23] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[24] G. S. Babu, P. L. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Proc. International Conference on Database Systems for Advanced Applications*, Springer, 2016, pp. 214–228.
[25] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," arXiv preprint arXiv:1506.06579, 2015.
[26] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. International Conference on Prognostics and Health Management*, 2008, pp. 1–9.

Qingwei Guo received his B.S. degree from School of Electronic Engineering, Beijing University of Posts and Telecommunications in 2012. Currently, he is a master student in the Electronic Engineering Department, Tsinghua University. His research has been focused on deep learning theory and application in prognostics and health management.

Yoshinori Miyamae received his M.S degree from Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, in 2003.

He joined ROHM Semiconductor in 2003, and had 10 year's experience as LSI circuit designer. He had worked for Rohm Semiconducor Taiwan from 2008 to 2011, as Circuit Designer over LCD Driver field. He is the research engineer of Fundamental Research and Development Division from 2015.

Zhongjun Wang received the B.Eng. and M.Sc. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1987 and 1990, respectively, and the M.Eng. and Ph.D. degrees in electrical engineering from the National University of Singapore, Singapore, in 1996 and 2009, respectively.

From 1990 to 1994, he was a lecturer with Shanghai Jiao Tong University. From 1996 to 2004, he was a member of technical staff with the Institute of Microelectronics, Singapore. From 2004 to 2008, he was a principal engineer with the Oki Techno Centre Singapore. From 2008 to 2011, he was a senior technical consultant with the Wipro Techno Centre Singapore. He joined ROHM Semiconductor in 2011, and now is with the Tsinghua-ROHM Joint Research Center, Beijing, China. His research interests include wireless communications, digital signal processing, low power solutions for sensor network and AI, and very-large-scale integration implementation.

Koji Taniuchi received his B.S. from Electronic Engineering Department, Ritsumeikan University in 1992.

He joined ROHM Semiconductor in 1992. He has more than 20 years' experience in LSI circuit design. He had worked for ROHM Semiconductor GmbH from 2010 to 2012 as a deputy director of technical marketing. He is the general manager of Fundamental Research and Development division from 2015.

Huazhong Yang was born in Ziyang, Sichuan Province, P. R. China, on Aug. 18, 1967. He received the B.S. degree in microelectronics in 1989, M.S. and Ph.D. degrees in electronic science and technology in 1993 and 1998, respectively, all from Tsinghua University, Beijing. He joined the Department of Electronic Engineering, Tsinghua University, Beijing, in 1993, where he is a full professor since 1998. Dr. Yang is a specially-appointed professor of the Cheung Kong Scholars Program.

His current interest includes wireless sensor networks, data converters, nonvolatile processors, and energy-harvesting circuits. Dr. Yang has authored and co-authored over 400 technical papers and 100 granted patents.

Yongpan Liu received his B.S., M.S. and Ph.D. degrees from Electronic Engineering Department, Tsinghua University in 1999, 2002 and 2007. He has been a visiting scholar at Penn State University during summer 2014. He is a key member of Tsinghua-Rohm Research Center and Research Center of Future ICs. He is now an associate professor in Dept. of Electronic Engineering Tsinghua University.

His main research interests include nonvolatile computation, low power VLSI design, emerging circuits and systems and design automation. His research is supported by NSFC, 863, 973 Program and Industry Companies such as Huawei, Rohm, Intel and so on. He has published over 60 peer-reviewed conference and journal papers and led over 6 chip design projects for sensing applications, including the first nonvolatile processor (THU1010N) and has received Design Contest Awards from (ISLPED2012, ISLPED2013) and best paper award HPCA2015.