DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

Feature fusion algorithm based on modular scalable integrated sensor behavior recognition

Feature Fusionsalgorithmus basierend auf modularer skalierbarer integrierter Sensorverhaltenserkennung

Fuyin Wei
Fei Xiang
Bohao Chu
Bernd Noche

Department of Transport Systems and Logistics, Faculty of Engineering, Division of Mechanical and Process Engineering, University of Duisburg-Essen

Based on the behavior recognition model of Convolutional Neural Network, we developed a modular scalable integrated (MSI) sensor system together with a signal feature fusion algorithm. The integrated sensor system can obtain high-quality signals without having to be embedded in the body of the object and has good modular scalability and timeliness. The feature fusion algorithm improves the recognition accuracy as well as the robustness of the model.

[Keywords: Convolutional Neural Network, modular scalable integrated sensor, feature fusion, accuracy, robustness]

Basierend auf dem Verhaltenserkennungsmodell des Convolutional Neural Network haben wir ein modulares skalierbares integriertes (MSI) Sensorsystem zusammen mit einem Signal-Feature-Fusion-Algorithmus entwickelt. Das integrierte Sensorsystem kann hochqualitative Signale erhalten, ohne in den Körper des Objekts eingebettet werden zu müssen, und weist eine gute modulare Skalierbarkeit und Aktualität auf. Der Feature-Fusion-Algorithmus verbessert die Erkennungsgenauigkeit sowie die Robustheit des Modells.

[Schlüsselwörter: Convolutional Neural Network, modularer skalierbarer integrierter Sensor, Feature-Fusion, Genauigkeit, Robustheit]

1 INTRODUCTION AND MOTIVATION

Humans have an intuitive ability to perceive changes in our surroundings, or changes in a particular object, through sight, hearing, smelling, and touching. At first, we are unable to recognize these changes, which we learn later in life to acquire this decision-making ability. This ability can then be used to record and respond to changes in the environment [Pew74]. For example, we can calculate the output of a machine tool through observation or perceive the sound of a mechanical arm collision by hearing, to make a response that is conducive to safety. Therefore, first of all we need a stand-alone device to simulate the five human senses that can accurately pick up perceptible signals from the surrounding environment. Then we need a model that can simulate the human brain to be able to process and analyze the collected signals as well as make recognition and classification. During the signal acquisition process, large number of signals data are generated.

For classification and recognition problems, obtaining correct data is essential. One of the traditional methods is to deploy a single dedicated sensor with a certain perception ability [GK03] in an appropriate location to get this specific type of data. If multiple different types of data are required, additional sensors need to be deployed, such as BOSCH BME280 sensor [BOSCH20], TDK MPU6500 sensor [TDK20], which have different sensing capabilities. It must be admitted that the data obtained by this method is effective and accurate, but it also carries a significant negativity:

a) If clean and effective data is required to meet special needs, the sensor must be placed in a suitable location, usually as close as possible to the position of the machine movement. This could mean that the machine needed to be modified.

b) The sensor network needs to be designed to realize the communication between different sensors.

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

c) Combined with the two aspects mentioned above and the sensor itself, the whole system is usually very costly.

Another option is to integrate sensors with different perception capabilities to form a general-purpose sensor cluster [NK15]. It only needs one MCU to control these sensors together. For example, Sensor Tag from Texas Instruments (TI) [TI18] it can be attached to objects to obtain different types of raw data then transmitted via Bluetooth. But its sensor types and data sampling frequency are limited, which cannot always meet the requirements of multiple scenarios perfectly. At the same time, the data must be uploaded via Bluetooth before it can be processed, which is not very convenient for scenes with high real-time performance requirements, such as collision detection for robotic arms. Laput et al. proposed the Synthetic Sensors [LZH17] solution, which integrated several kinds of sensors and transmitted the data through WIFI. Synthetic Sensors face a defined field rather than a specific object, which can reduce the task of adapting to different scenarios. Synthetic sensors, for example, can be located directly on a kitchen to monitor and identify the behavioral events of all objects in the kitchen. However, only a few sensors can achieve versatility in this system, and the effectiveness of data collected by most sensors will decrease as the distance between the object and the Synthetic Sensors increases. It can only effectively monitor and identify objects in its vicinity. Also, it cannot handle these scenes with high real-time performance requirements.

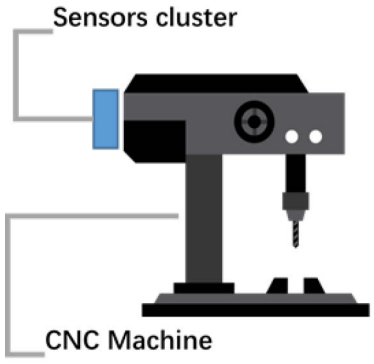


Figure 1. *Illustration of a scenario where our MSI sensors cluster works on a CNC machine to monitor and recognize its various behavior events*

As previously reviewed, we need to solve the problem of versatility when facing various behavior under different scenarios, as well as improving the effectiveness of the data sampling. Therefore, based on benchmark research above, we adopt a modular design idea, which allows easy integration of specific sensor clusters to meet actual recognition needs of various behaviors under different scenes.

2 Related Work

2.1 Feature Extraction

In the behavior recognition system, the accuracy of the recognition result is mainly determined by three aspects: the representativeness of the feature description for the recognition object, the appropriateness of the selection for the classifier model, and the training of the recognition model.

Feature extraction as a representative feature description technology for selecting classification and recognition objects greatly influences the final classification and recognition performance.

Feature extraction is a method of signal processing to extract information from raw sensor data that is representative of the various behaviors of the testing object. As a necessary module to link sensor data acquisition and recognition classification, feature extraction can maintain good recognition accuracy while reducing the dimensionality of processing data.

In order to extract representative feature values in real applications without compromising classification and recognition efficiency, the following requirements must be met in the feature extraction process:

- The feature values should represent as much information as possible so that the recognition model can distinguish between the classification objects.
- Direct correlation between characteristics should be as few as possible to minimize repetition and dimensionality of the data.
- The complexity of the feature extraction algorithm should be as low as possible to reduce processing time and improve efficiency

In the field of behavior recognition research based on multiple modular sensor system (e.g., acceleration, magnetic field, sound, infrared, distance, etc.), there are three common types of feature extraction methods: ***time domain*** feature extraction, ***frequency domain*** feature extraction, and ***time-frequency domain*** feature extraction.

- The ***time domain*** is the domain in which the real world exists. People are generally accustomed to analyze things in chronological order. Time domain features are a statistical extraction method for processing signals of a time series
- Extraction of ***frequency domain*** features takes data features from the signal with frequency as the coordinate axis. In the frequency domain, any waveform can be considered as a synthesis of a sine wave. The Fourier transform can represent a function that meets certain conditions as a sinusoidal function

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

or a combination of its integrals. However, the Fourier transform requires a large amount of calculation and cannot handle real-time problems in time, researchers usually use the improved fast Fourier transform (FFT) to analyze the sine wave component of the signal.

In addition to time domain and frequency domain features extraction, there is another ***time-frequency domain*** feature extraction which is also often used in signal analysis. Short-time Fourier transform (STFT) is a kind of deformation of Fourier transform, also called windowed Fourier transform or time-dependent Fourier transform, which is used to determine the sinusoidal frequency and phase of a partial part of a signal that varies with time. In fact, the process of calculating the Short Time Fourier Transform (STFT) is to divide the long-time signal into several shorter equal length signals and then calculate the Fourier transform of each shorter segment separately. STFT is usually used to depict the changes in the frequency and time domains, and it is one of the most important tools in time-frequency analysis [Hec95] [HDW11] [Moo17].

2.2 Recognition classification algorithms

The choice of recognition classification algorithm is very important. In behavior recognition classification, researchers usually use statistical pattern recognition methods, which requires a known data sample and sample corresponding labels to learn a classification function or construct a classification model. This classification model is what we often referred to as a classifier. After getting the behavior recognition classifier, we can make the final classification prediction of the behavior samples in the behavior recognition classification system.

In the field of sensor-based behavior recognition research, the most commonly used statistical classification methods are NAVIE Bayes, K-Nearest Neighbour (KNN), Decision Tree, Support Vector Machine (SVM). Compared with that, deep learning is currently a more effective and widely applied method in feature recognition. Among them, the convolutional neural network (CNN) has shown good performance in feature extraction.

CNN is powerful in classification because it can learn feature representation from a large number of samples, and the whole network expresses the mapping relationship between original data and their class features [CGJ18].

The convolutional neural network is responsible for receiving the detected data, and the training results of each round are passed backwards to the whole network structure parameters, through the training set as well as the validation set. It consists of a convolutional layer, a pooling layer, a fully connected layer. The convolutional layer sequentially extracts image features through convolution kernel and image convolution filtering. The result of the convolution is passed through the activation function to form the feature map of this layer, which contains not only the feature values but also the relative position information. The pooling layer compresses the feature maps from the convolutional layer to simplify the complexity of network calculation. The compression of the features leads to further extraction of the main features and removal of redundant features [AMA17] [ZJD17].

The CNN recognition model used in this study is a 5-layer convolutional neural network to process our feature fusion map. It contains two convolutional layers, one pooling layer and two fully connected layers. The details of the processing method of MSI sensor system will be described in Chapter 4.

3 Design of modular Scalable Integrated (MSI) Sensor System

The overall design of MSI sensor system consists of the following three layers, which is illustrated in figure 2:

- Physical Layer
- Processing Layer
- Application Layer

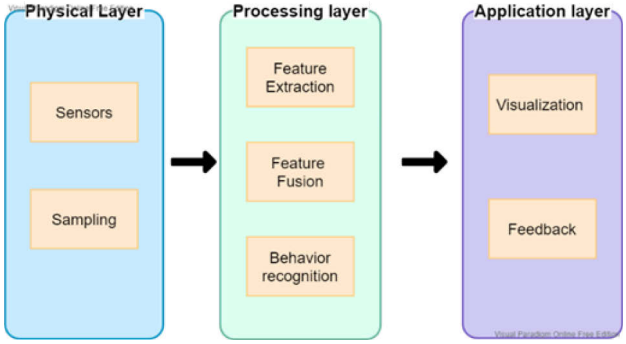


Figure 2. *Three layers of the whole system design*

The whole system adopts a modular design scheme. The sensors in the physical layer are easily replaceable. A stack fusion algorithm is used in the processing layer [CZL04]. The visualization of the application layer can also be adjusted according to the increase or decrease of the sensors.

The whole MSI sensor system can be adjusted according to the observed scenario and the requirements of the measured object.

3.1 Sensor Board hardware Design

At first, to sample different sensor signals, we implemented the physical layer by stacking commercial sensors directly bought in the market. During the

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

experiment, we often get interference due to the instability of DuPont line connection when we collect signals.

In addition, simple stacking leads to the entire device's size being too large (showed in Figure 3). It causes some difficulties for experimenters to embed the device at the proper measurement position of the test object to collect clean signals.

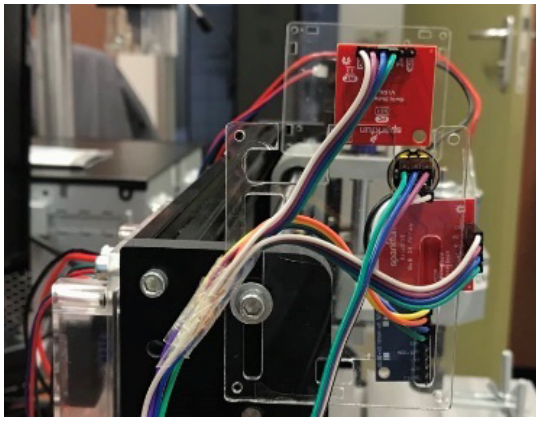


Figure 3. *The first generation of hardware design, sensor stacking mode through DuPont line*

3.1.1 Sensor Board Design

In order to solve the above problems, we have developed an integrated sensor board. It has a size of 40mm * 40mm * 1.6mm, which is relatively small and can be embedded well with correct measurement position of the test object. A customized data cable is used to replace multiple DuPont-cables to enhance the stability and purity for the collecting sensor signals. Due to circuit integration, the power consumption of the whole sensor board has been reduced from 300mW to 200mW.

One of the main targets of our work is that the modular scalable integrated sensor system we developed can be applied to multiple scenarios. Therefore, the MSI sensor board currently integrates five kinds of sensors, which can collect 9 different sensor signals (see Table 1).

For various scenario requirements, the sensor system could be easily scalable due to its modular design concept.

3.1.2 Sensor board layout

The integrated sensor board is divided into two sides: Top-Layer and Bottom-Layer. Normally Top-Layer is directed towards the sampled object. Therefore, most of sensors are placed on the Top Layer to achieve better signal quality.

Hence the sampling channel of the microphone sensor is located on the back of the chip, we place it on the Bottom Layer, which makes the sampling channel face the sampled object.

ID	Name	Sensor	Fun.	Freq
A	IR	AMG8833	1	10Hz
B	Microphone	INMP441	1	16KHz
C	Temp/Hum/Bar	BME280	3	10Hz
D	Laser	VL53L1X	1	10Hz
E	Acc/Mag/Gro	MPU9250	3	1KHz

Table 1. *Hardware list of our current MSI sensor system, which includes 5 kinds of sensors. 9 different types of sensor data can be sampled from the environment*

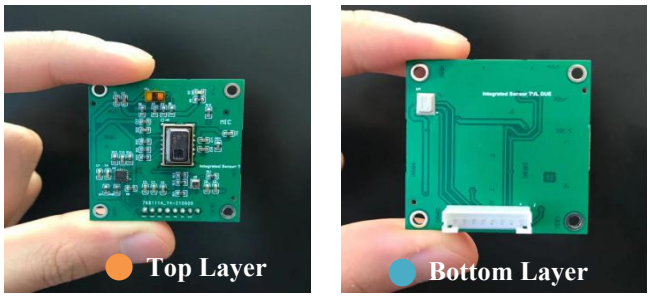


Figure 4. *Top-Layer (Left) and Bottom-Layer (Right) of our modular scalable integrated sensor board*

While designing the layout of each sensor board, we need to ensure that our MSI sensor board could cope with most scenarios and collect signal data with high precision and good quality.

Among the five sensors selected for our MSI sensor system, the microphone sensor, laser distance sensor, and IR thermophilic array sensor have strong directivity. Therefore, their position in the layout must be prioritized to ensure their optimal detection range.

The final schematic diagram and layout of MSI sensor board is shown in Figure 5.

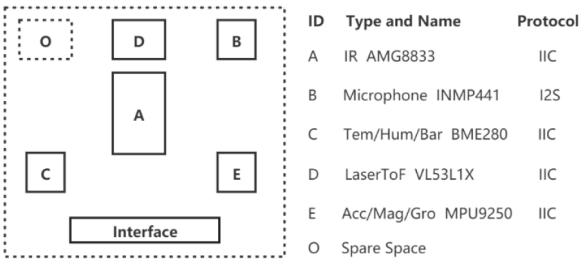


Figure 5. *Schematic diagram and layout of our MSI sensor board*

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

3.2 SOFTWARE DESIGN

The software design of the scalable integrated sensor system is divided into the following three parts, namely the sampling program running on the physical layer, the recognition program running on the processing layer, and the visualization program running on the application layer, as shown in Figure 6.

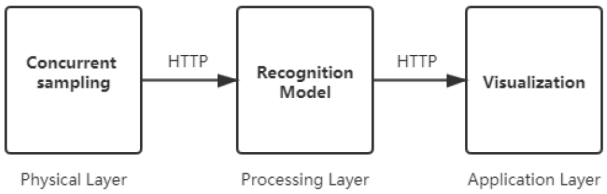


Figure 6. *Software design of our MSI sensor system*

(a) SAMPLING PROGRAM ON PHYSICAL LAYER

At the physical layer, a Raspberry Pi 4B module is used as controller [VBLS20], which drives and communicates with five sensors through a customized data-bus interface.

After sensor initialization, the sampling program achieves parallel sampling of multiple sensors through multi-threaded synchronization [WBKW07]. Once the sampling is completed, the controller packs the data into Json format and sends them to the processing layer through HTTP transmission protocol [Wan11]. When the data package is successfully sent, the program will go back to repeat the parallel sampling process.

The whole workflow of sampling program is illustrated in Figure 7. The customized data-bus interface includes IIC [Lee09] communication protocol and I²S [Woo16] communication protocol, as shown in Figure 8.

In our design, four sensors (IR Thermophile array sensor, microphone sensor, temperature sensor and laser distance sensor) share an IIC data bus. And the 9-axis sensor (Acc. / Mag. / Gro.) used an independent I²S data bus.

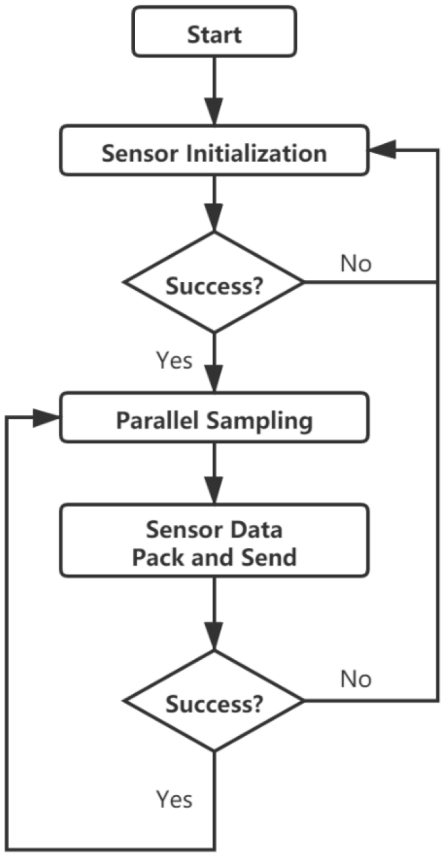


Figure 7. *Flow chat of sampling program running on the physical layer*

3.2.1 RECOGNITION PROGRAM ON PROCESSING LAYER

Once the recognition program of the processing layer starts, it is set into the data monitoring mode.

When the json data [PRSUV16] packet is received from the physical layer, the data is preprocessed and fed to the model for behavior identification.

After recognition, the program sends the original data and the recognition result to the application layer and then it is set back to the data monitoring mode.

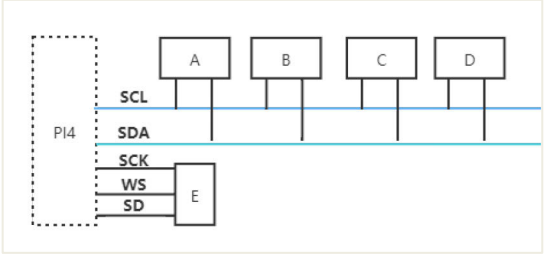


Figure 8. *Communication bus connection status*

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

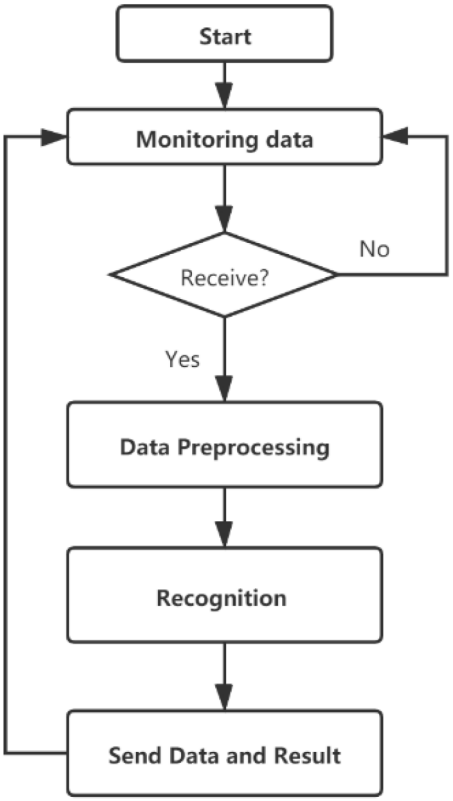


Figure 9. *Flow chat of recognition program running on the processing layer*

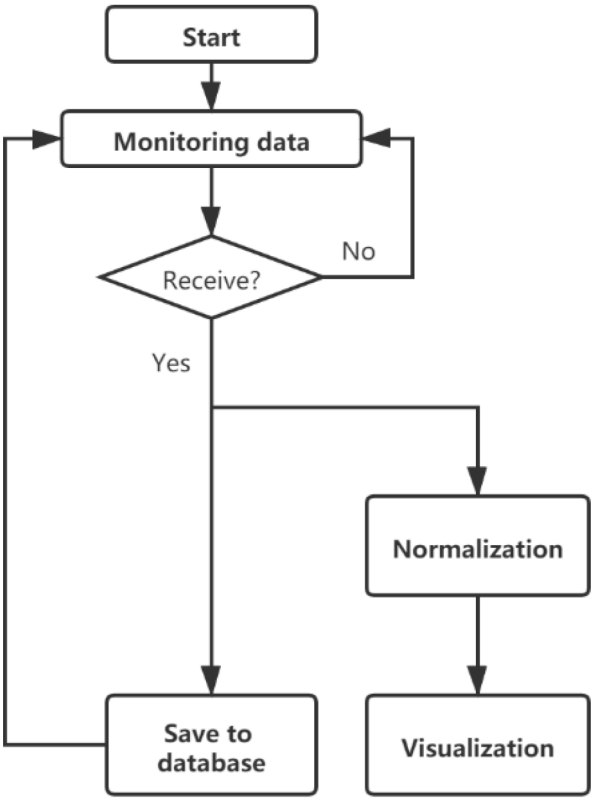
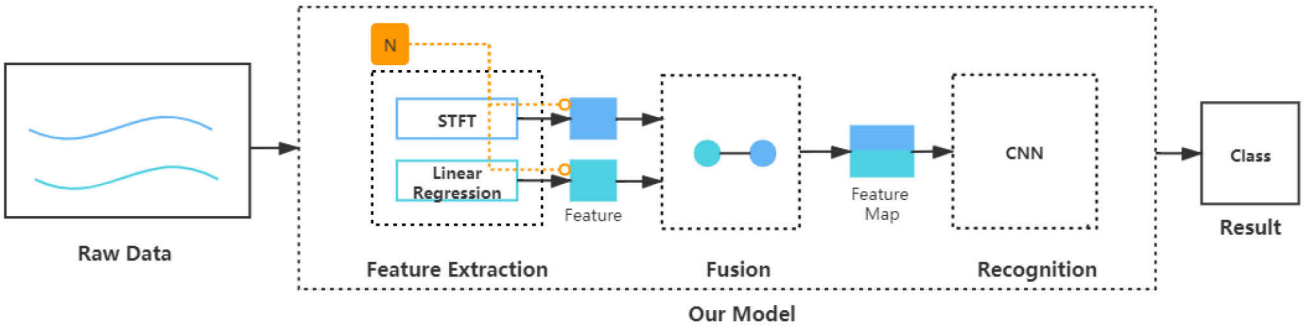


Figure 10. *Flow chat of visualization program running on the application layer*



3.2.2 VISUALIZATION PROGRAM ON APPLICATION LAYER

When received the feature data and recognition result, the visualization program running on application layer normalizes the feature data from each sensor while storing them into the database.

The visualization program performs an average compression on high-frequency feature data, like audio feature data, and a repeated expansion on low-frequency feature data, such as acceleration data. So that each feature data can be visualized at a synchronous rate.

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

4 PROCESSING METHOD OF MSI SENSOR SYSTEM

This chapter will describe the processing method of MSI sensor system in detail. It includes the process of signal feature extraction, feature fusion, CNN model design and model evaluation (Figure 11).

4.1 SYMBOL DEFINITION

In this paper, we mark the original signal collected at a certain moment as $\mathbf{X} = \mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_n}$. Where **n** represents different types of signal, also represents the number of sensors in the MSI sensor system.

We denote the recognizable behavior categories in the environment as $\mathbf{Y} = \mathbf{Y_1}, \mathbf{Y_2}, ..., \mathbf{Y_m}$. Where **m** represents the maximum number of identifications.

In a real-time prediction model, the prediction time interval is a very important parameter.

Normally, if the prediction time interval is long enough, high-quality original signals can be obtained, while the fault tolerance and noise interference also can be improved. However, the time interval can also not be too long, thus the meaning of real-time prediction will be lost. Therefore, a reasonable setting of time interval parameter T needs to be evaluated.

The input original raw signal at a certain moment is noted as $\mathbf{x}^{\mathrm{T}} = (\boldsymbol{x}_1^{t+T}, \boldsymbol{x}_2^{t+T}, ..., \boldsymbol{x}_n^{t+T})$, in which t represents the recognition start time. We denote the category predicted by our model as: $\widehat{\mathbf{Y}} = \widehat{\mathbf{Y}}_1, \widehat{\mathbf{Y}}_2, ..., \widehat{\mathbf{Y}}_m$.

4.2 FEATURE EXTRACTION

In the encoder network, according to the different signals obtained by different sensors, the corresponding feature extraction algorithm is selected to process data of different frequencies.

Once we have the spectrogram, we need to reinforce the features of the signal. Therefore, we perform a binary logarithm on the spectrogram, which enhances the features.

The high-frequency feature extraction formula is as follows.

$$f_i^{t+T} = \cup_{n=1}^{N} \boldsymbol{log} \int_{-\infty}^{\infty} x_i^t(\tau n) h(\tau - t) e^{-j2\pi f \tau} d_\tau \qquad [1]$$

Where N is the number of sliding steps. The general calculation is

$$\mathrm{N} = x_i^{t+T} / \tau \qquad [2]$$

$x_i^t(\tau n)$ is one section of the window-sized original signal, and $h(\tau - t)$ is the window function.

For high-frequency time-domain signal data, short-time fast Fourier transform (STFT) is used to extract the feature in the frequency domain [Hec95] [HDW11] [Moo17].

For low-frequency time-domain signal data, SVM linear regression is used to analyze its feature [VM02] [CZH01].

Encoder Network converts the input multi-dimensional original signal $\mathbf{x}^{\mathrm{T}} = (\boldsymbol{x}_1^{t+T}, \boldsymbol{x}_2^{t+T}, ..., \boldsymbol{x}_n^{t+T})$ into feature data $F^{\mathrm{t}} = (f_1^{t+T}, f_2^{t+T}, ..., f_n^{t+T})$.

4.2.1 HIGH FREQUENCY SIGNAL FEATURE EXTRACTION

Signals base on the high frequency time domain, such as sound signals, are intuitively difficult to determine its pattern. However, any periodic signal can be expressed as a series of linear combinations of sine and cosine signals.

Fourier Transform can help us find these components, and give each component the frequency, amplitude and phase. If only use the Fast Fourier Transform, the frequency distribution of a section of data can be gotten. But losing the time domain information leads to lose the frequency distribution changes over time. Therefore, we choose the short-time fast Fourier transform method (STFT).

STFT divides a continuous signal into frames by adding frame windows. Then perform Fourier Transform on each frame and stack the results of each frame along another dimension to obtain a two-dimensional signal into a image form.

If our original signal is sound signal, then the two-dimensional signal obtained by STFT expansion is the so-called spectrogram.

Figure 12 shows the original microphone signal we collected when a CNC machine was rotating. The sampling time is 1 second, sampling frequency is 16KHz, bit depth is 8bit and the size of this signal is 16K bytes.

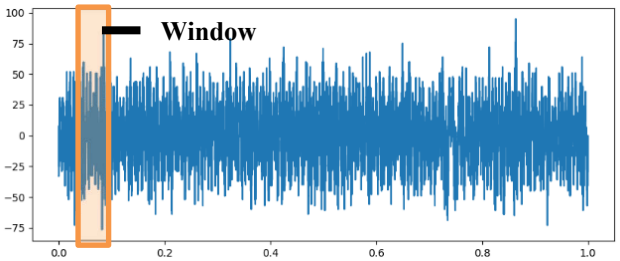


Figure 12. *The raw microphone signal generated when the CNC machine rotates. The orange part is the Hamming window*

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

A 256 bytes Hamming window is added to divide the original signal. Moving step length of dividing window is 224 bytes. That means the overlap size between each window is 32 bytes. And a fast Fourier transform is performed in each Hamming window to get a continuous spectrogram with time domain information, as shown in Figure 13.

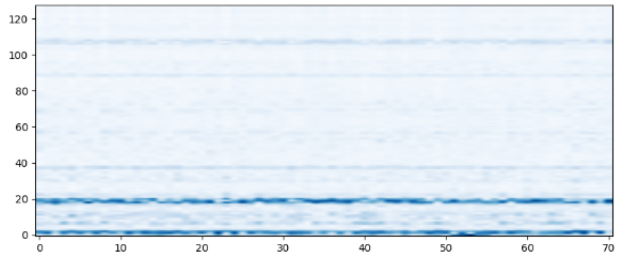


Figure 13. *Continuous spectrogram with time domain information*

After the frequency spectrum is obtained, a binary logarithm is performed to get enhanced features that are easier to be identified. The enhanced feature map is shown in Figure 14.

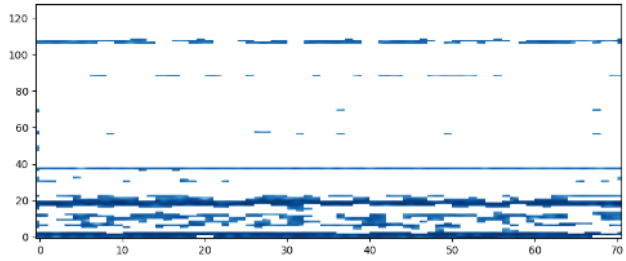


Figure 14. *Continuous spectrogram with enhanced features*

4.2.2 LOW FREQUENCY SIGNAL FEATURE EXTRACTION

Linear regression is a statistical analysis method that uses regression analysis in mathematical statistics to determine the quantitative relationship between two or more variables.

What we try to analyze is the relationship between environmental signals and time. To find a straight line or a plane through the linear regression model, which can fit the relationship between environmental signals and time well, as shown in Figure 15.

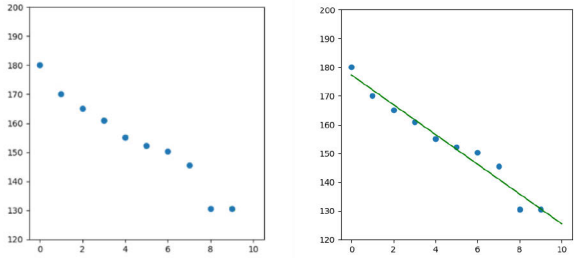


Figure 15. *Schematic diagram of linear feature extraction for low-frequency signal*

In the left diagram in Figure 15 we use time data as the X axis and sampling data as the Y axis to establish a coordinate system to visualize the sampled data.

We use a hyperplane [3] to standardize this relationship, where θ^{T} represents the change trend of environmental signals, and θ_0 represents the base value.

$$y = \theta^T x + \theta_0 \qquad [3]$$

In this paper, linear regression is used to fit the hyperplane, so that the sum of the distances from each signal point to the hyperplane in a time unit is the smallest. When the hyperplane is obtained, the feature data of our low-frequency signal can be expressed as:

$$f_i^{t+T} = [\theta^{\mathrm{T}}, \theta_0] \qquad [4]$$

4.3 FEATURE FUSION

In the Feature Fusion stage, each original signal will be feature extracted to get the correspondingf_i^{t+T}. Which means, the features of each sensor after feature extraction will be fused into a Feature Map, as shown in Figure 16.

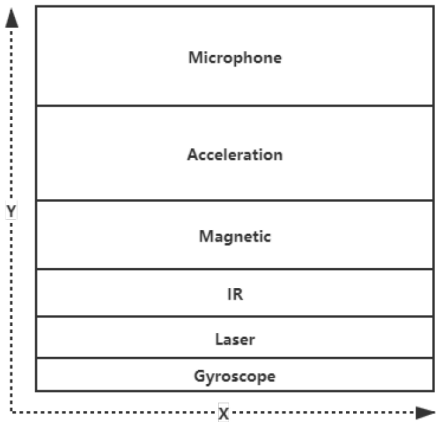


Figure 16. *Feature fusion of different signals*

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

Since the original signals from different environments have features of different sizes after feature extraction, we must map these features again before feature fusion. For high-dimensional feature data, we will reduce its dimensionality, and for low-dimensional features, we will expand it and format the data into a Xi x Yi tensor, and make them have the same size in the X-dimension, so we can follow Fusion in Y dimension.

A multilayer perceptron [MBPM09] is used to implement the feature remapping. The remapping formula is shown in [5], where W_f is the parameter of the perceptron, f_i^{t+T} is the feature data, and noise is random standard Gaussian white noise.

$$f_i^{t+T\prime} = \mathrm{MLP}\,(f_i^{t+T} + \mu * noise;\ W_f) \qquad [5]$$

To reduce the overfitting problem of the model and enhance the anti-interference robustness of the entire model, a certain amount of noise would be added to the fusion feature map.

To control the influence of noise on the model, we have set a noise weight μ. By adjusting it, different iterative models will be generated. We can evaluate them to determine the optimal noise weight.

After the feature mapping is completed, different features have the same size in the X dimension, so we merge the different features along the Y dimension, the concatenate arrays formula is as follows [6].

$$F^t = \mathrm{Cat}\,(f_1^{t+T\prime}, f_2^{t+T\prime}, f_3^{t+T\prime}, ..., f_n^{t+T\prime}) \qquad [6]$$

4.4 CONVOLUTIONAL NEURAL NETWORK MODEL DESIGN

In the recognition network, we use the convolutional neural network model to recognize our processed feature map F^t.

A 5-layer convolutional neural network is designed to process our feature map F^t, which contains two convolutional layers, a pooling layer and two fully connected layers.

The kernel size of the first convolution is 3*3, and there are 20 kernels in total.

The kernel size of the second convolution is 3*3, and 50 kernels in total

The pooling kernel size is 2*2.

The output size of the first fully connected layer is 128.

The output of the second fully connected layer is the needed recognition classes.

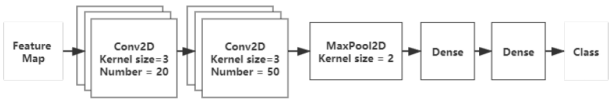


Figure 17. *Convolutional neural network architecture design*

The specifications of F^t obtained from different original environmental signals through feature extraction may be inconsistent. In order to avoid the problem of feature disappearance or getting excessive feature during fusion, we need to normalize F^t before training and testing. We apply the mean normalization method; the formula is as follows:

The feature map $F^{t\prime}$ output by our convolutional neural network belongs to the probability of each category, so we choose the cross-entropy loss function, the formula is as follows:

$$l = -\sum_{i=1}^{n} p(x_i) log(q(x_i)) \qquad [8]$$

In which, $p(x_i)$ is the real label and $q(x_i)$ represents the probability of distribution for our prediction.

4.5 EVALUATION INDICATORS AND METHODS

During the CNN model training iteration process, a common evaluation index in classification problems called F1-Score is used to evaluate our model [Sas07] [Pow11].

Firstly, some indicators need to be introduced.

TP: True Positives, which represents the number of samples that are actually positive and judged as positive by the classifier.

FP: False Positives, which represents the number of samples that are actually negative and judged as positive by the classifier.

FN: False Negatives, which represents the number of samples that are actually positive but are judged as negative by the classifier.

TN: True Negatives, which represents the number of samples that are actually negative and judged as negative by the classifier.

- Precision

  The precision rate refers to the proportion of the true class in the sample predicted to be the positive class.

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

For multi-class problems, precision value of each category is obtained, and the minimum value is taken among all. The formula is as follows:

$$MinPrecision = \min_{i=1\ldots n} \frac{TP}{TP + FP}$$

- Recall

  The recall rate refers to the proportion of all positive classes that are predicted to be positive.

  For multi-class problems, recall value of each category is obtained, and the minimum value is taken among all. The formula is as follows

$$MinRecall = \min_{i=1\ldots n} \frac{TP}{TP + FN}$$

- F1-Score

  F1 Score is an indicator used in statistics to measure the accuracy of a binary classification model. It takes both precision rate and recall rate of the classification model into account, and thus can be regarded as a harmonic average of the model.

  When evaluating a multi-classification model, the minimum F1-Score for each category should be calculated.

$$F1Score = \frac{2 * MinPrecision * MinRecall}{MinPrecision + MinRecall}$$

The process of training model iteration is divided into multiple stages.

At the beginning of each stage, 80 seconds of raw data for each behavior are sampled. The newly sampled data would be used to test and evaluate the training model from the previous stage.

If the evaluation result does not meet the set threshold, all previously sampled behavior data would be mixed and retrained. When a new model is generated, we will enter the next stage. After completing multiple iteration stages, the model that meets the set threshold is the final completed one.

5 EXPERIMENT

As described in chapter 1, our MSI sensor system is a kind of general perception sensor system and is expected to be applied in a variety of scenarios.

In this chapter, an industrial scenario is simulated. Through our MSI sensor, a CNC machine is monitored to get identified its behavior.

In this scenario, nine machine behaviors can be recognized, including 6 basic behaviors and 3 multi-behaviors. The recognition frequency is 1Hz.

5.1 FEATURE EXTRACTION IMPLEMENTATION

After each behavior is sampled separately, as described in section 4.2, different feature extraction algorithms are used to process sensor sampling data of different frequencies.

Figure 18 and Figure 19 are the original sampling data and feature extraction diagrams of five behaviors from high-frequency sensors (microphone sensor and X-axis acceleration sensor).

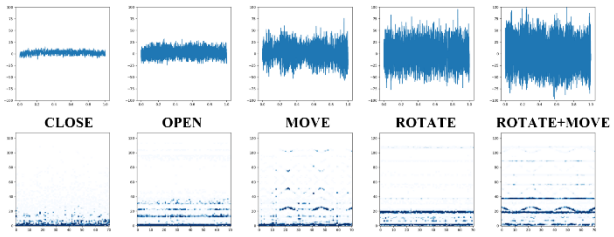


Figure 18. *Raw data (above) and feature data (under) for five different behaviors of CNC machine from Microphone sensor*

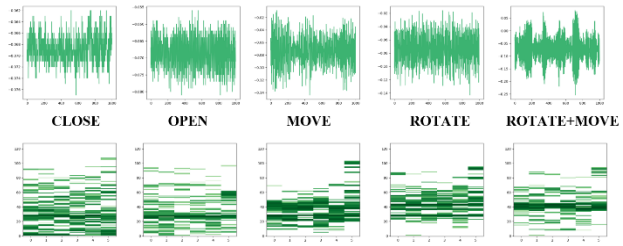


Figure 19. *Raw data (above) and feature data (under) for five different behaviors of CNC machine from ACC X sensor*

Figure 20, Figure 21 and Figure 22 are the original sampling data and feature extraction diagrams of moving behavior from low-frequency sensors (Laser, magnetic and IR sensor).

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

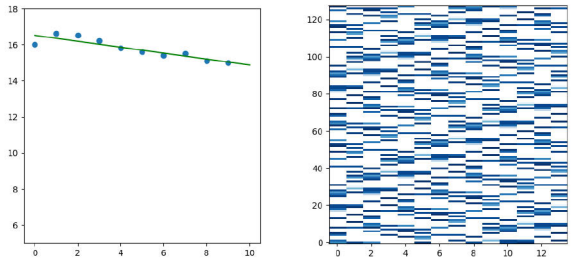


Figure 20. *Raw data (left) and feature data (right) for moving behavior of CNC machine from Laser sensor*

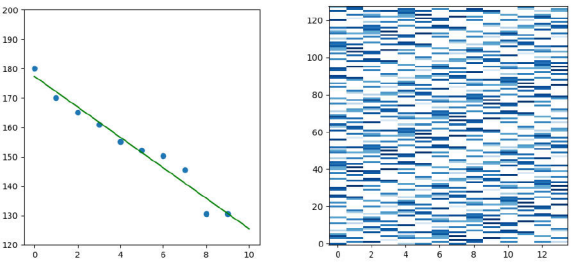


Figure 21. *Raw data (left) and feature data (right) for moving behavior of CNC machine from Magnetic sensor*

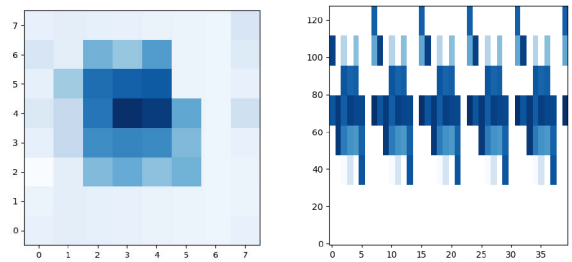
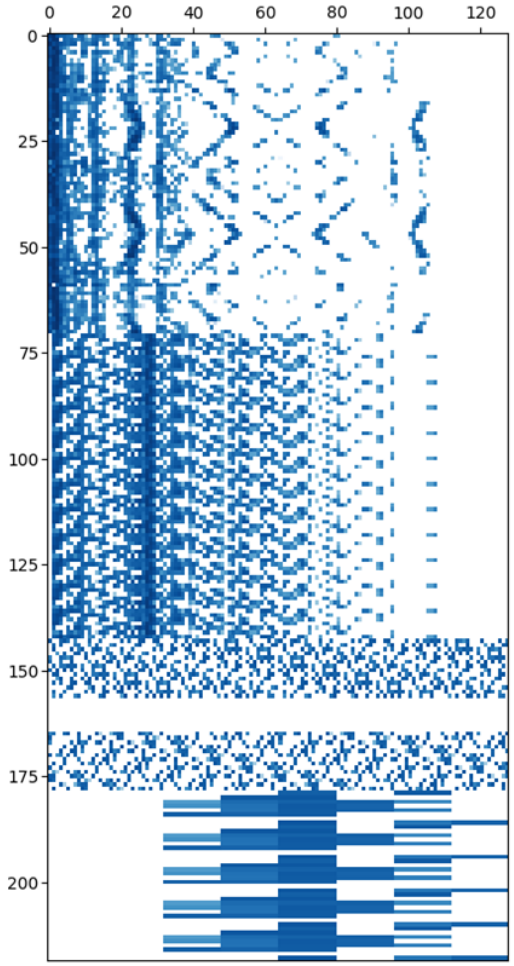


Figure 22. *Raw data (left) and feature data (right) for moving behavior of CNC machine from IR sensor*

After the feature extraction, according to the MLP remapping equation in Chapter 4.3, the features should be remapped to form a feature fusion map.



5.3 MODEL GENERATION AND EVALUATION RESULTS

From Chapter 4.6, process of training model iteration is divided into multiple stages.

Starting from the second stage, the evaluate results of the iterative model generated in the previous stage are as shown in the Table 2 -Table 5 below.

In these tables, A1-A9 refers different behaviors of the CNC machine, in which A1 means shutdown, A2 standby, A3 shift left, A4 shift right, A5 shift bottom, A6 rotation, A7 rotation + left shift, A8 rotation + right shift, A9 rotation + bottom shift.

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

Table 2. *The first iteration evaluation3*

	A1	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*	*A9*
A1	77	3	0	0	0	0	0	0	0
A2	4	75	0	0	0	0	1	0	0
A3	0	1	65	2	5	0	3	1	3
A4	0	2	1	68	0	2	1	4	2
A5	0	1	1	4	60	0	5	3	6
A6	0	3	0	0	0	60	1	1	15
A7	0	0	2	0	1	5	65	2	5
A8	0	0	0	1	2	8	2	58	9
A9	0	0	2	2	5	25	5	6	35

According to the previous formula, the distribution can be calculated: MinPrecision = 0.4666, MinRecall = 0.4375, F1Score=0.45

Table 3. *The second iteration evaluation*

	A1	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*	*A9*
A1	79	1	0	0	0	0	0	0	0
A2	4	76	0	0	0	0	0	0	0
A3	0	1	70	3	0	0	5	1	0
A4	0	2	1	72	0	2	1	4	2
A5	0	1	1	4	79	0	5	3	6
A6	0	3	0	0	0	65	1	1	15
A7	0	0	2	0	1	5	72	2	5
A8	0	0	0	1	2	8	2	75	9
A9	0	0	2	2	5	25	5	6	68

According to the previous formula, the distribution can be calculated: MinPrecision = 0.619, MinRecall = 0.812, F1Score=0.703

Table 4. *The third iteration evaluation*

	A1	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*	*A9*
A1	80	0	0	0	0	0	0	0	0
A2	0	80	0	0	0	0	0	0	0
A3	0	1	73	1	4	0	0	1	0
A4	0	1	1	75	0	2	1	0	0
A5	0	0	0	0	78	0	0	0	2
A6	0	0	0	0	0	69	1	0	10
A7	0	0	0	0	1	1	75	2	1
A8	0	0	0	0	2	0	2	74	2
A9	0	0	4	0	0	6	0	2	68

According to the previous formula, the distribution can be calculated: MinPrecision = 0.819, MinRecall = 0.85, F1Score=0.834

Table 5. *The 4th iteration evaluation*

	A1	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*A8*	*A9*
A1	80	0	0	0	0	0	0	0	0
A2	0	80	0	0	0	0	0	0	0
A3	0	1	76	2	0	0	0	1	0
A4	0	0	3	77	0	0	0	0	0
A5	0	0	0	0	80	0	0	0	0
A6	0	1	0	0	0	76	0	0	3
A7	0	0	2	0	1	0	77	0	0
A8	0	0	0	1	0	0	0	79	0
A9	0	0	0	2	2	5	0	0	71

According to the previous formula, the distribution can be calculated: MinPrecision = 0.959, MinRecall = 0.8875, F1Score=0.922

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

The evaluation results of the four iterations are illustrated in Figure 24.

The evaluation result of the first iteration is poor due to iteration training model underfitting by lack of sampled data. As the number of iterations increases, the evaluation results are getting better.

After four iterations, the evaluation results have reached the threshold value which we have set at 0.9. The model training iteration is completed, and the final model is generated.

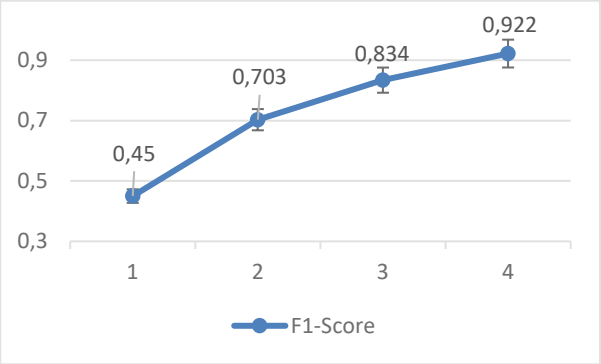


Figure 24. *The model iteratively evaluates the results*

5.4 Model robustness implementation

In order to prevent the model from overfitting, we add Gaussian random white noise to the fusion feature map.

The random noise is sampled from a standard normal distribution (expectation:0, variance:1) and multiplied by different noise weights μ before adding to the fusion feature map. Iterative training with noise-fusion feature maps can improve the anti-interference robustness of the generated model.

We tested the performance of generated model with different noise weight μ, as shown in Figure 25.

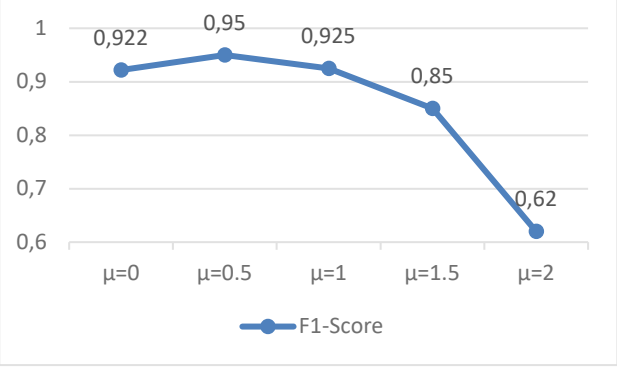


Figure 25. *Model performance under different weight of noises*

After adding 0.5 times the weight of random noise, the evaluation result of the model has been improved. As the noise weight increases, the evaluation results of the generative model begin to deteriorate. Out of this test result, we get an optimal noise weight μ = 0.5

5.5 Visualization presentation

As we mentioned in chapter 3 of our MSI system design, the visualization program could be applied on the application layer to display the final model recognition results. The real-time visualization in Figure 26 shows that the testing CNC-machine first moves to the left, then stops the movement towards left and starts the rotation.

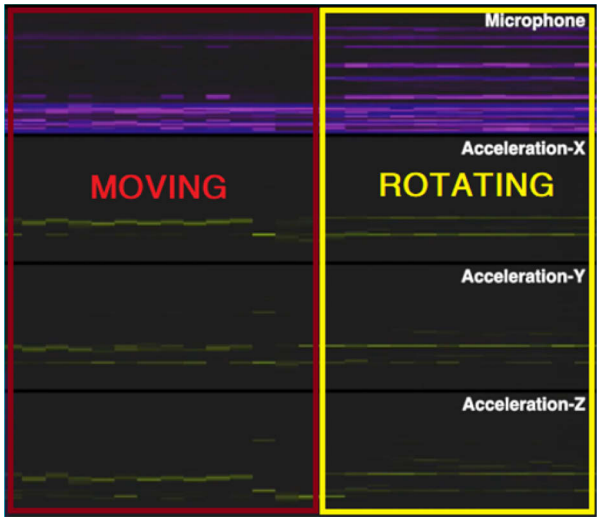


Figure 26. *Visualization of Real-time recognition results of CNC machines*

6 Summary and Outlook

In this paper, we have designed and tested the first version of our MSI system, in which 5 different sensors are integrated and 9 types of sensor data are collected to train the recognition model. The test under a simulated industrial environment with a CNC machine to identify different working status is our first experiment of the MSI sensor system. So far, the evaluation results of recognition are within our expectation.

We adopt a modular design idea for our MSI sensor system at the begin of the development, which allows easy integration of specific sensor clusters to meet actual recognition needs of various behaviors under different scenes. We will continue our tests in the future under different environment and various scenarios, representing daily working and living circumstances.

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

Figure 27. *MSI sensor board applied in various scenarios*

Meanwhile, we will add an edge computing module with certain capabilities to realize edge processing of data, so that our system can deal with scenarios required relatively high real-time performance.

7 ACKNOWLEDGEMENT

Parts of this paper are compiled within the frame-work of the BMBF research project 01IS19068A “KI-LiveS” (KI-Labor für verteilte und eingebettete Sys-teme) and supported by Federal Ministry of Education and Research (BMBF).

LITERATURE

[AMA17] Albawi, Saad; Mohammed, Tarq A.; Al-Zawi, Saad: *Understanding of a convolutional neural network*. 2017 International Conference on Engineering and Technology (ICET), pp. 1-6. 2017.

[BOSCH20] Bosch BME280, *Humidity sensor measuring relative humidity, barometric pressure and ambient temperature.* 2020. https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/

[CGJ18] Chauhan, Rahul; Ghanshala, Kamal Kumar; Joshi, R. C.: *Convolutional Neural Network (CNN) for Image Detection and Recognition.* 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 278-282. 2018.

[CZH01] Chen, Yunqiang; Zhou, Xiang Sean; Huang, T. S.: *One-class SVM for learning in image retrieval.* In Proceedings 2001 International Conference on Image Processing, vol.1. pp. 34-37. 2001.

[CZL04] Chao, Rui; Zhang, Ke; Li, Yan-jun: *An Image Fusion Algorithm Using Wavelet Transform*. In Chinese Journal of Electronics, 32(5): 750-753. 2004.

[GK03] Gharavi, H.; Kumar, S. P.: *Special issue on sensor networks and applications.* In Proceedings of the IEEE, vol. 91, no. 8, pp. 1151-1153, 2003.

[HDW11] Halliday, J. Ross; Dorrell, David G.; Wood, Alan R.: *An application of the Fast Fourier Transform to the short-term prediction of sea wave behaviour*. Renewable Energy, Volume 36, Issue 6, 2011.

[Hec95] Heckbert, Paul: Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm. Notes 3, Computer Graphics 2, pp. 15-463. 1995.

[Lee09] Leens, Frederic: *An introduction to I2C and SPI protocols*. In IEEE Instrumentation & Measurement Magazine, vol. 12, no. 1, pp. 8-13. 2009

[LZH17] Laput, G.; Zhang Y.; Harrison C.: *Synthetic Sensors: Towards General-Purpose Sensing.* In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems Denver, Colorado, USA, S.3986-3999. 2017.

[MBPM09] Marius, Popescu; Balas, Valentina; Perescu-Popescu, Liliana; Mastorakis, Nikos: *Multilayer perceptron and neural networks*. WSEAS Transactions on Circuits and Systems. 8. 2009.

[Moo17] Moorer, James A.: *A note on the implementation of audio processing by short-term fourier transform.* 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 156-159. 2017

[NK15] Nguyen, Thanh-Tung; Koo, Insoo: *Sensor Clustering and Sensing Technology for Optimal Throughput of Sensor-Aided Cognitive Radio Networks Supporting Multiple Licensed Channels.* In: International Journal of Distributed Sensor Networks. 2015.

[Pew74] Pew, Richard W.: Human Perceptual-Motor Performance. Defense Technical Information Center, Michigan Univ Ann Arbor Human Performance Center. 1974.

[Pow11] Powers, David M. W.: *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness &*

DOI: 10.2195/lj_Proc_wei_en_202112_01
URN: urn:nbn:de:0009-14-54471

Correlation. Journal of Machine Learning Technologies. 2 (1). pp. 37–63. 2011.

[PRSUV16] Pezoa, Felipe; Reutter, Juan L.; Suarez, Fernando; Ugarte, Martin; Vrgoc: *Foundations of JSON Schema.* In WWW '16: Proceedings of the 25th International Conference on World Wide Web, pp.263-273. 2016.

[Sas07] Sasaki, Yutaka: *The truth of the F-measure.* 2007. https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/F-measure-YS-26Oct07.pdf

[TDK20] TDK MPU-6500, *Six-Axis (Gyro + Accelerometer) MEMS MotionTracking™ Devices.* 2020. https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6500/

[TI18] Texas Instruments CC2650STK, SimpleLink™ Bluetooth low energy/Multi-standard SensorTag. 2018. https://www.ti.com/tool/CC2650STK

[VBLS20] Vieira, Gustavo; Barbosa, José; Leitão, Paulo; Sakurada, Lucas: *Low-Cost Industrial Controller based on the Raspberry Pi Platform*. 2020 IEEE International Conference on Industrial Technology (ICIT), pp. 292-297. 2020.

[VM02] Vishwanathan, S. V. M.; Murty, M. Narasimha: *SSVM: a simple SVM algorithm.* In Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), pp. 2393-2398 vol.3. 2002.

[Wan11] Wang, Guanhua: *Improving Data Transmission in Web Applications via the Translation between XML and JSON*. In 2011 Third International Conference on Communications and Mobile Computing, pp. 182-185. 2011

[WBKW07] Witkowski, Thomas; Blanc, Nicolas; Kroening, Daniel; Weissenbacher, Georg: *Model checking concurrent linux device drivers*. In ASE '07: Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering, pp. 501–504. 2007.

[Woo16] Wootton, Cliff: *Audio and Inter-IC Sound (I2S).* In: Samsung ARTIK Reference. Apress, Berkeley, CA. 2016.

[ZJD17] Zhou, Fei-Yan; Jin, Lin-Peng; Dong, Jun: *Review of Convolutional Neural Network.* In Chinese Journal of computers, Vol.40, Online Publishing No.7. 2017.

Fuyin Wei, M.Sc., Researcher at the Department of Transport Systems and Logistics (TuL), coordinator Sino-German Department of Centre for Logistics and Traffic (ZLV) at University Duisburg- Essen.
Phone: +49 203 379-7719
E-Mail: fuyin.wei@uni-due.de

Fei Xiang, M.Sc., Researcher at the Department of Transport Systems and Logistics (TuL)
E-Mail: fei.xiang@uni-due.de

Bohao Chu, B.Sc., Researcher Assistant at the Department of Transport Systems and Logistics (TuL)
E-Mail: bohao.chu@stud.uni-due.de

Prof. Dr.-Ing. Bernd Noche is the Chair holder of the Department of Transport Systems and Logistics (TuL), and the borad Chairman of Centre for Logistics and Traffic (ZLV) at University Duisburg-Essen.
Phone: +49 203 379-7050
E-Mail: bernd.noche@uni-due.de