Strategy：Use IR sensor to predict whether the pan will be burnt in the future

Scene side view

Single point IR sensor

Area of temperature collection

pan

wall

stove

880mm

320mm

wall

Single point IR sensor

Scene front view

Temperature collection area

pan

Stove L

Stove R

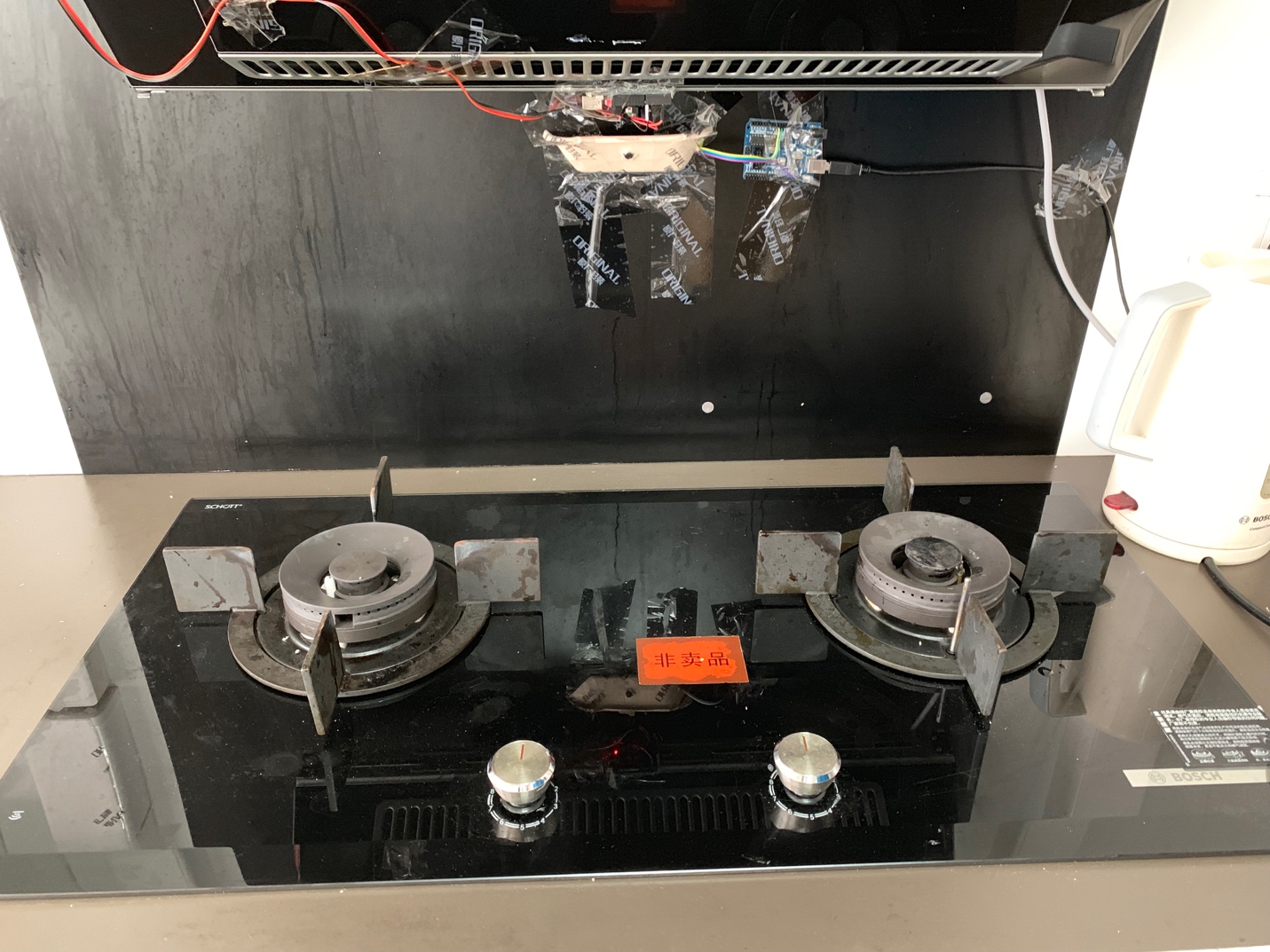
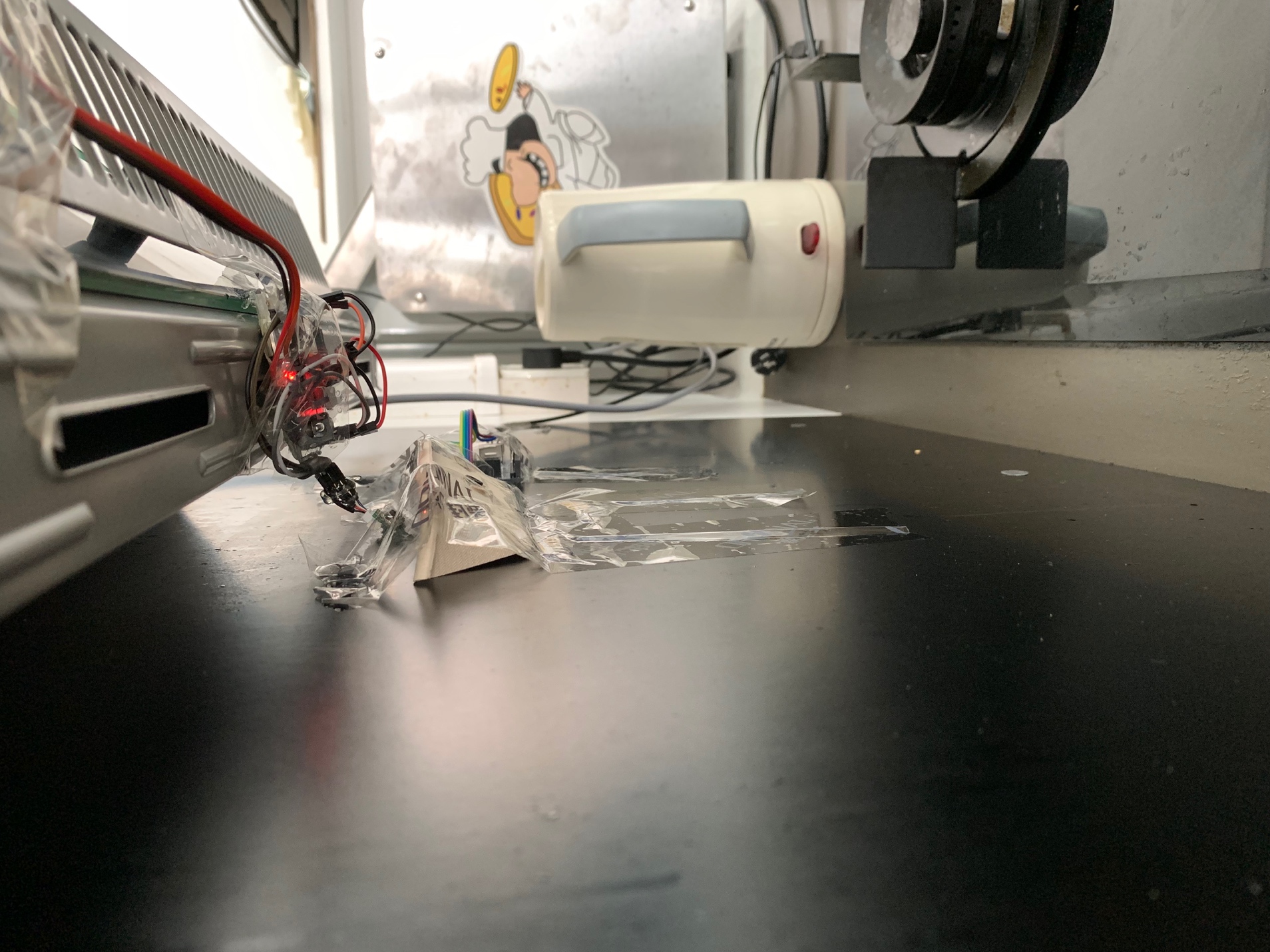
Scene top view

Area of temperature collection

pan

Single point IR sensor

Stove R

****

IR sensor

IR sensor

IR sensor

Stove R

Stove L

**Main process summary:**

1. Infrared sensors start the main process from the time of firing and return single point temperature data with time stamp (time starting from 1)
2. When the timestamp reaches 180, pass 121-180 data into the algorithm module
3. Algorithm module returns the calculated results to the main process (return value is 0 or 1)
4. The main process alarms according to the return value of the algorithm module (the return value is 0, which means that dry burning will not happen in the next 10 seconds; Represents dry burning in the next 10 seconds.)

**About algorithm module:**

Deep learning is used to classify scenes, predict future temperatures and determine whether dry burning will occur in the future



In the class

* **model\_case\_classification** model is used to distinguish usage scenarios (with and without pot lid);
* **case** is the mark of the scene (initial value is -1, 0 is with lid, 1 is without lid);
* **model\_prediction** model is used to generate temperature data in the next 10 seconds;
* **model\_classification\_lid** model is used to distinguish dry burning in the next 10 seconds in the case of pot lid;
* **model\_classification\_nolid** model is used to distinguish dry heat free for the next 10 seconds in a no-lid scenario;
* **model\_classification** serves as a reference to the lid or no lid model

The following figure shows the entire algorithm flow:

self.case == -1 ?

self.case = chooseCase()

Input 60s’ temperature data as **α**

Generate the next 10s’ temperature data as **β**

**True**

**False**

Discriminate the event as **χ** according to classifier

**χ** as output

1

2

3

4

5

6

1. The module receives 60 seconds of temperature data from the main process α
2. Determine whether the member property case of the module class itself is -1 (initial value)
3. If case is -1, the **chooseCase(**) function is called and the return value of the function is assigned to case

* **chooseCase()** receives the temperature data α, calls the model\_case\_classification model to calculate a value of 0 or 1 and returns this value

1. **model\_prediction** model receives temperature data α，and calculates the temperature data in the next 10 seconds β
2. Select a model from **model\_classification\_lid** and **model\_classification\_nolid** according to the current value of the case, and assign the reference to **model\_classification**. Splice α and β to form a set of 70-second temperature data and calculate with **model\_classification** to get χ (value is 0 or 1)

* select **model\_classification\_lid** if the case is 0; For 1, select **model\_classification\_nolid**

1. Return the value of χ to the main process as the prediction result: 0 means that dry burning will not occur in the next 10 seconds; 1 means dry burning in the next 10 seconds

**Deep learning models for use:**

model\_prediction:

The training set is extracted from the temperature data .CSV file by a 70\*1 sliding window. In each window, the first 60\*1 data is regarded as the X set, and the last 10\*1 data is regarded as the y set. The same is true for test sets. Data shall be normalized by min max scalar

The model structure is

* The input layer receives an array of (1, 60)
* The hidden layer is

1. Long short-term memory neural network of 40 units and dropout of 0.1

* Output layer

1. 10 units fully connected layer, output an array of (10,)

Compile model

* Mean squared error as the loss function
* Adam as the optimizer
* Metric use mse

Why LSTM neural network

The unique gate control mechanism of LSTM, especially the forgetting gate, can achieve the function of "forgetting" by reducing the weight of elements with little influence in the input data. At the same time, since RNN algorithm has the feature of global processing, it is possible to identify the property that each group of data in the batch is correlated with each other (because of the time-dependence between the data). In addition, according to a large number of existing experiments, it is proved that LSTM has a better effect in processing data with time series properties (including generation and discrimination).

model\_case\_classification:

The data set selects only temperature data. CSV file timestamp 111-190 data, from which a 60\*1 sliding window selects the training X set data. The y set is the label matrix composed of the use scenarios (0 or 1) corresponding to each window, and the y set can be generated by the to\_categorical() function in the keras framework. The same is true for test sets. Data shall be normalized by min max scalar

The model structure is

* The input layer receives an array of (60, 1)
* The hidden layer is

1. One dimensional convolutional neural network with 128 units, filter size 2, convolution stride size 1, output and input length the same, activation function using ReLU, using randomly distributed values as the initial value of the weight matrix
2. One dimensional convolutional neural network with 128 units, filter size 2, convolution stride size 1, output and input length the same, activation function using ReLU, using randomly distributed values as the initial value of the weight matrix
3. The pooling layer uses the maximum one-dimensional pooling scheme, with the pooling size of 2
4. One dimensional convolutional neural network with 328 units, filter size 3, convolution stride size 1, output and input length the same, activation function using ReLU, using randomly distributed values as the initial value of the weight matrix
5. One dimensional convolutional neural network with 328 units, filter size 3, convolution stride size 1, output and input length the same, activation function using ReLU, using randomly distributed values as the initial value of the weight matrix
6. The pooling layer uses the maximum one-dimensional pooling scheme, with the pooling size of 2
7. The flatten layer
8. Fully connected layer with 512 units and tanh as activation function
9. 0.2 Dropout layer
10. Fully connected layer with 512 units and ReLU as activation function
11. 0.2 Dropout layer

* Output layer

1. Fully connected layer with 2 units, activation function using SoftMax, output (2,) probability array

Compile model

* categorical cross entropy as the loss function
* Adam as the optimizer
* Metric uses Accuracy

Why choose SoftMax output

Although there are only two kinds of classification problems in user scenarios considered at present (with and without lid), the possibility of adding other scenarios in the future cannot be ruled out, so multi-classification problems will be faced in the future. For multi-classification problems, the loss function needs to be categorical cross entropy, and the output layer needs SoftMax.

model\_classification\_lid & model\_classification\_nolid:

The data set selects temperature data. CSV data starting from timestamp 319 is intended to balance the samples without dry burning with those with dry burning. The X set is an array of (70,), which is composed of an array of (60,) and an array of (10,). The array is obtained from the data set by the window of size (60, 1). (10,) array is generated according to the array (60,) by model\_prediction. The y set is the batch size () tag, which is obtained from the tag column in the data set through the window of (70, 1). Finally, the same batch sizes of X and y should be guaranteed. If there are more than X, the redundant parts should be deleted. Data need not be normalized in advance

The model structure is

* The input layer

1. The 328 units embedded layer takes the (70,) array and converts it to a fixed-size dense vector of up to 1400 vocabulary sizes

* The hidden layer

1. One-dimensional convolutional neural network with 328 units, filter size of 5, convolution stride size of 1, output and input length of the same length, activation function using ReLU, using randomly distributed values as the initial value of the weight matrix
2. The pooling layer uses the maximum one-dimensional pooling scheme with the pooling size of 2
3. One-dimensional convolutional neural network for 328 units, filter size is 10, convolution stride size is 1, output is as long as input, activation function USES ReLU, and randomly distributed values are used as the initial value of the weight matrix
4. The pooling layer uses the maximum one-dimensional pooling scheme, with the pooling size of 10
5. 0.2 Dropout layer
6. The long short-term memory neural network of 128 units and the internal dropout of 0.2 and the recurrent dropout of 0.2

* The output layer

1. 1-unit full connection layer. The activation function uses Sigmoid to output (1,) probability value

Compile model

* binary cross entropy as the loss function
* Adam as the optimizer
* Metric uses Accuracy

Why choose the convolution kernel with larger size

Changing the size of the convolution kernel directly affects the visual field of the kernel. Larger kernel can perceive a wider range. As some training data have the characteristics of fluctuation disorder in time series, large kernel is adopted to make the output of convolution tend to reflect the overall situation rather than local jitter, so as to improve the overall training effect and convergence speed.

Why choose a larger pooling size

Under the premise of using a large convolution kernel, the expanded pool size improves the generality of the model and significantly optimizes the test results.

Why it has a recurrent dropout in LSTM

Repeated testing has shown that increasing loop dropout by 0.2 improves the generality of the model.

**Training and testing data:**

There are 19 groups of effective data in dry burning experiment, including 9 groups of dry burning with lid and 10 groups of dry burning without lid.

1. Training and testing **model\_prediction**

* 8 groups of data of dry burning with lid and 8 groups of data of dry burning without lid were used for training
* Testing set is the remaining 3 groups of data

1. Training and testing **model\_case\_classification**

* 7 groups of data of dry burning with lid and 8 groups of data of dry burning without lid were used for training
* Testing set is the remaining 4 groups of data

1. Training and testing **model\_classification\_lid**

* 8 groups of data of dry burning with lid were used for training
* Testing set is the remaining 1 group of data of dry burning with lid

1. Training and testing **model\_classification\_nolid**

* 9 groups of data of dry burning without lid were used for training
* Testing set is the remaining 1 group of data of dry burning without lid

**About test results:**

1. model\_prediction

The mean squared error is about 3.4\*10^(-4).

1. model\_case\_classification

Accuracy 1.0

1. model\_classification\_lid

The accuracy was about 95.7%

Marco-F1 value is about 0.947

1. model\_classification\_nolid

The accuracy was about 95.5%

Marco-F1 value is about 0.956

**The direction of subsequent tuning:**

1. Expand the data volume. The larger the data volume, the better the effect will be
2. Use multi-point IR sensor to accurately measure the overall temperature of the pan, reducing the influence of the stove temperature on the data
3. Control variables are required for subsequent experiments, such as redo experiments after the stove, pan or lid is completely cooled
4. Add an appropriate filter on raw data to smooth the temperature change curve to reduce the outliers collected by sensors and the impact of violent data fluctuations
5. After adding the filter with good effect, we can try to add a new dimension as the training object, such as the two-point slope with a certain step length
6. When selecting data to train **model\_classification\_lid** and **model\_classification\_nolid**, balanced sampling is adopted to ensure the balance of sample quantity and to ensure the coverage of most data. Data can be sampled according to gaussian distribution.
7. When data is normalized according to min max scalar, the minimum and maximum values take the global minimum and maximum values of all data
8. Try to modify the hyperparameters of the hidden layer or deepen the network to achieve better test results
9. When evaluating the model, cross testing method can be selected to make the results more reliable