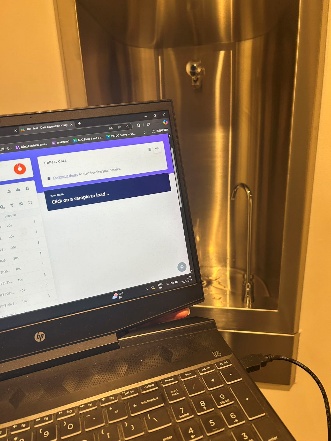
Week1

数据集收集。

Week2

使用test 3已有数据集。

tes6.49已处理是我个人收集的数据集。

Week 1

Data Collection.

Week 2

Used the test3 dataset.

The processed dataset tes6.49 is my personal collection.

About test3 attempt:

The model built following the tutorial was not accurate; the misclassification rate for “faucet” was high. I already tried the default classification; from now on I’ll switch entirely to transfer learning.

However, the tutorial suggests using 30 epochs, which I think is not appropriate. I’ll try to find the best epoch value.

(Training logs and metrics from epoch 1 to 100 are omitted in this summary translation for brevity, but I have them if you need full logs translated.)

Although accuracy kept improving, the validation accuracy plateaued after epoch 89. Therefore, I believe the best epoch is 89. For the sake of performance and training time, I will use 80 epochs moving forward.

Even with that, the model output remains inaccurate. Points below 0.5 are all misclassifications. Unfortunately, I lost the model trained with 30 epochs, so I can’t compare. Right now, I believe the dataset is the issue.

Changing to 80 epochs did not help either.

Regardless of whether it's noise or water flow, the feature points are too clustered—robustness is inherently poor.

Modeling test results are good when using the test set from the same dataset, but poor when using my own recordings. It might be that my validation set was too large (set to 50, default is 20). I reverted this in the next version to test.

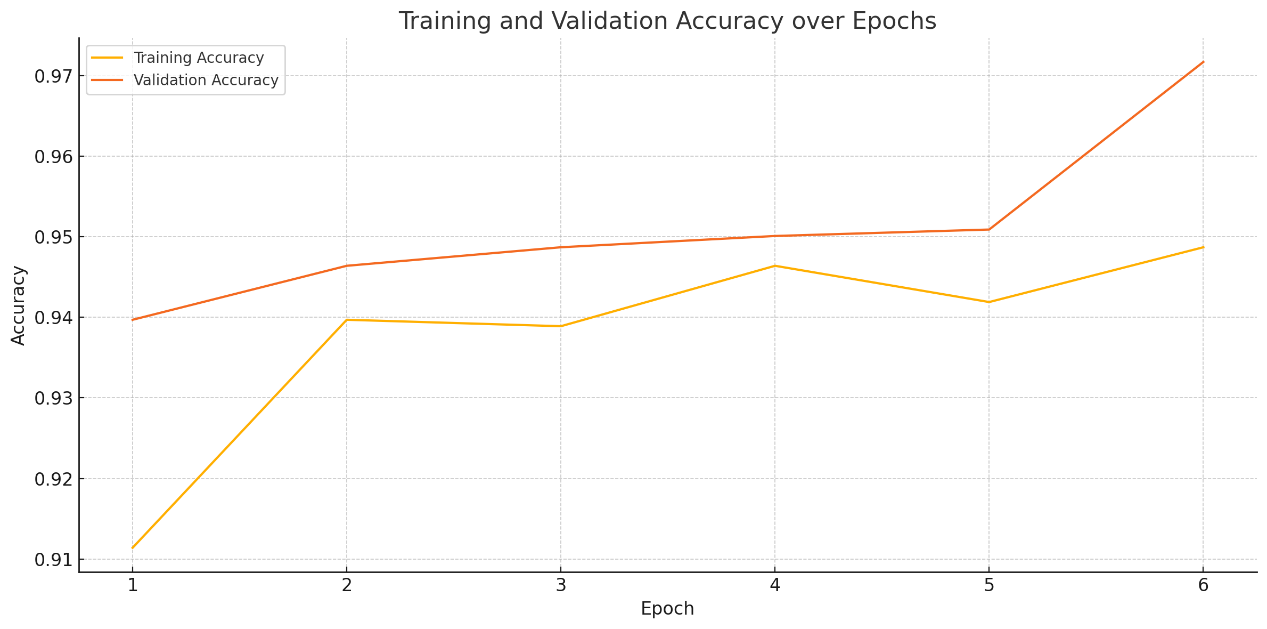
Still not working. It’s almost certainly a dataset problem.

# 关于test 3的尝试：

**The model based on the steps in the tutorial is not accurate and the probability of misidentification as faucet is high. I have tried the classification with default settings and will be using transfer learning all next.**

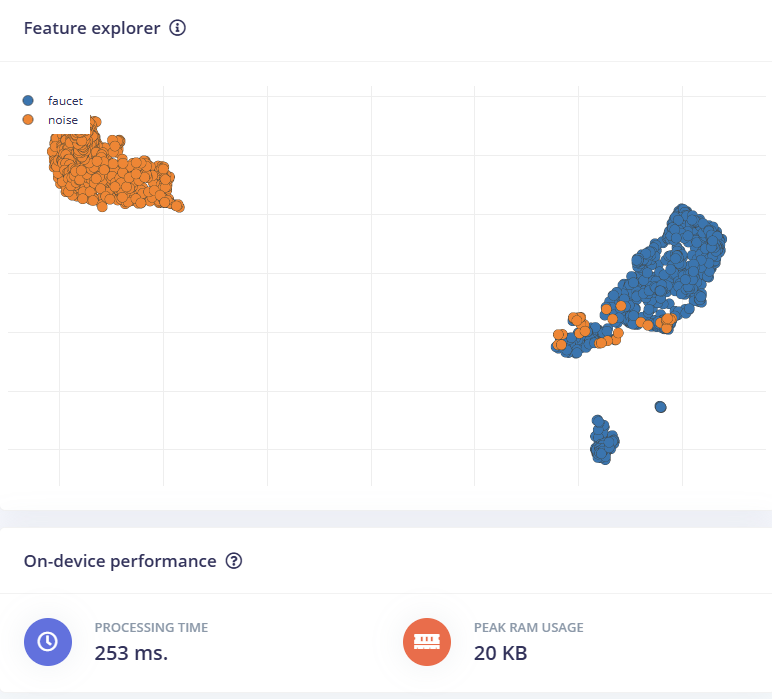
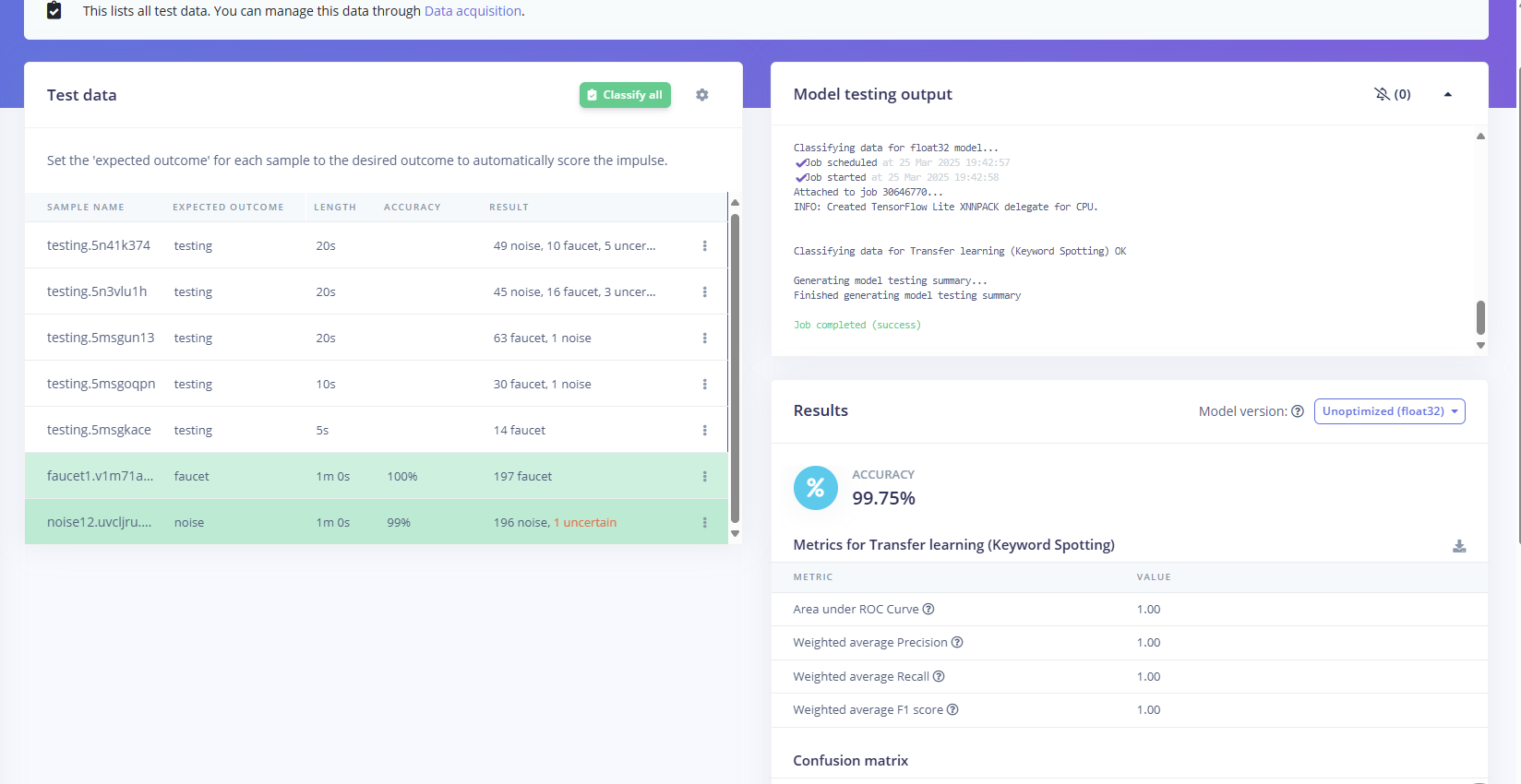
**But the tutorial has an epoch of 30, which I don't think is right, I will try to find the best epoch.**

Finished training



Although accuracy is still improving, val accuracy hasn't improved since epoch 89 All I think the best epoch is 89, and I set all subsequent epochs to 80, given performance and length of time

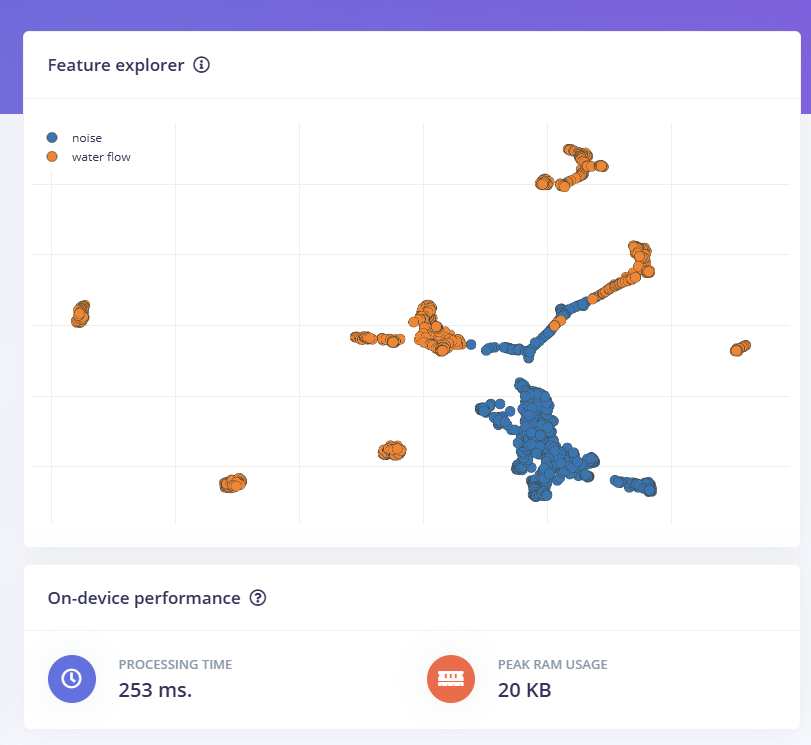
Even so, the output model is still completely inaccurate, and any points below 0.5 in the above graph are misclassified. Unfortunately, I lost the model for the one with an epoch of 30, so I can't compare. But I currently think it's the dataset.

Changing to 80 epochs didn't optimise it   
After all, the characteristic points of both noise and water flow sound are too concentrated, and robustness is bound to be poor.

The modelling test has excellent results for test data in the same dataset, but my own recordings are poor. It could also be that validationset is too big, I hit 50 and the default is 20.I changed it back in the next version, try it.

It still doesn't work. It's almost certainly the dataset.

# tes6.49 Attempts processed:

  
The overlap of data points between the two categories is very significant

There is no clear boundary to distinguish water flow from noise

There are a lot of orange points (water flow sound) “inserted” in the blue area (background noise), and vice versa Attempts have been made to process:

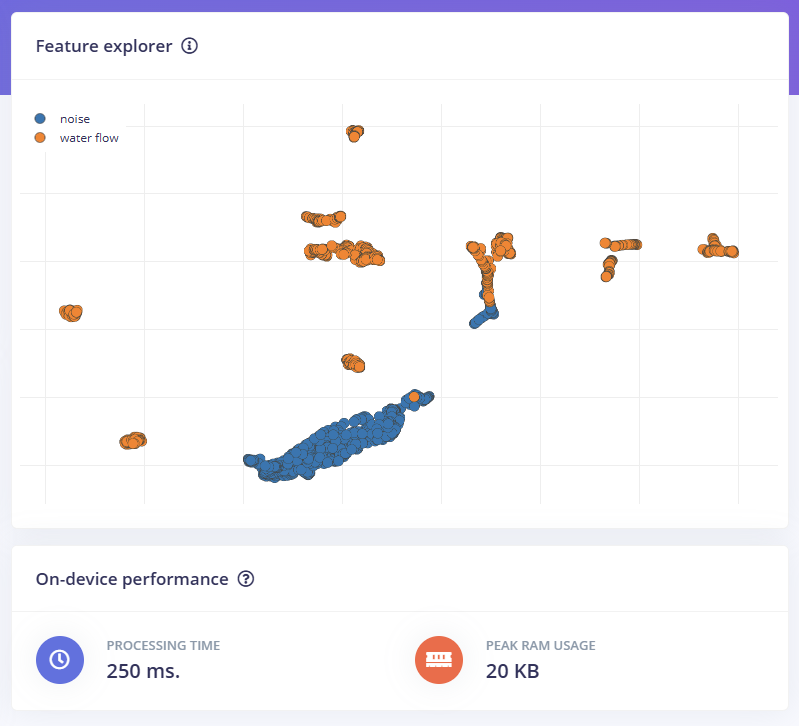
|  |  |
| --- | --- |
| Frame length | 0.02 → 0.03 or 0.04（增加每帧时间） |

|  |  |
| --- | --- |
| Filter number | 40 → 60（更细的频率划分） |

|  |  |
| --- | --- |
| FFT length | 256 → 512（更高频率分辨率） |

|  |  |
| --- | --- |
| Low frequency | 100Hz（包含低频水声） |

|  |  |
| --- | --- |
| High frequency | 8000Hz |



🟠 water flow and 🔵 noise Some distinct clusters are now starting to emerge

Blue noise is very tightly clustered, suggesting relatively uniform features (easy to learn for the model)

Orange water flow is slightly more spread out, but also has clear sub-clusters

The “blending area” between the two classes is significantly reduced

Improvements:

1 Try to group water flow sounds into several categories (e.g. fast flow / slow flow) by “flow rate”, which may help the model to distinguish finer features and possibly reduce intra-class variability. Improvements: 1. Try categorising water flow sounds by “flow rate” (e.g. fast flow / slow flow), which may help the model discriminate finer features and reduce intra-class variation.

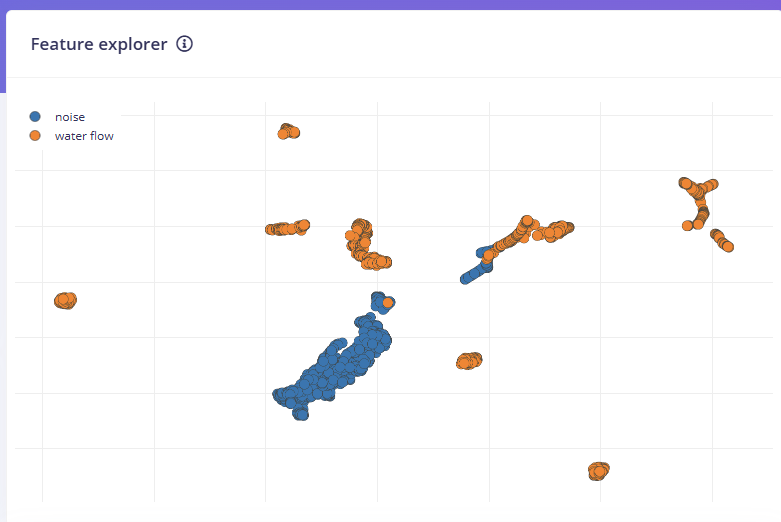
2 The current noise performance is excellent, but you can record more “complex” ambient sounds (people talking, toilets flushing, electrical appliances running) to test the model's immunity to interference.

3 Parameters to try

|  |  |
| --- | --- |
| Filter number | 40 → 60 |

|  |  |
| --- | --- |
| FFT length | 256 → 512 |

|  |  |
| --- | --- |
| Frame length | 0.02 → 0.03 |



I would also like to include data on the shower's use in the dataset, but there is a lot of water spray from the shower, and I am concerned that it will damage the circuitry. I may not be able to obtain this part of the data until I have completed the waterproofing design.

I am still optimising the dataset, and the kitchen in the house is not yet in the dataset, but it could be used as a recording for the TESTING SET.

The dataset is optimized and trained for the first time using CLASSIFICATION with the following Input configuration

→ Reshape (60 columns)

→ Conv1D (8 filters)

→ Dropout

→ Conv1D (16 filters)

→ Dropout

→ Conv1D (32 filters)

→ Dropout

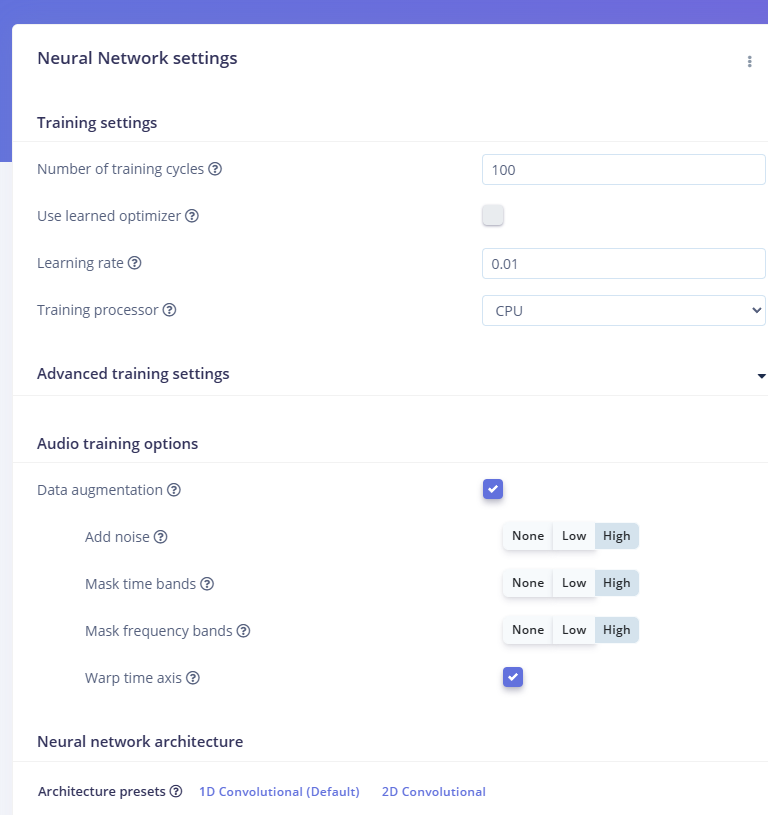
→ Flatten

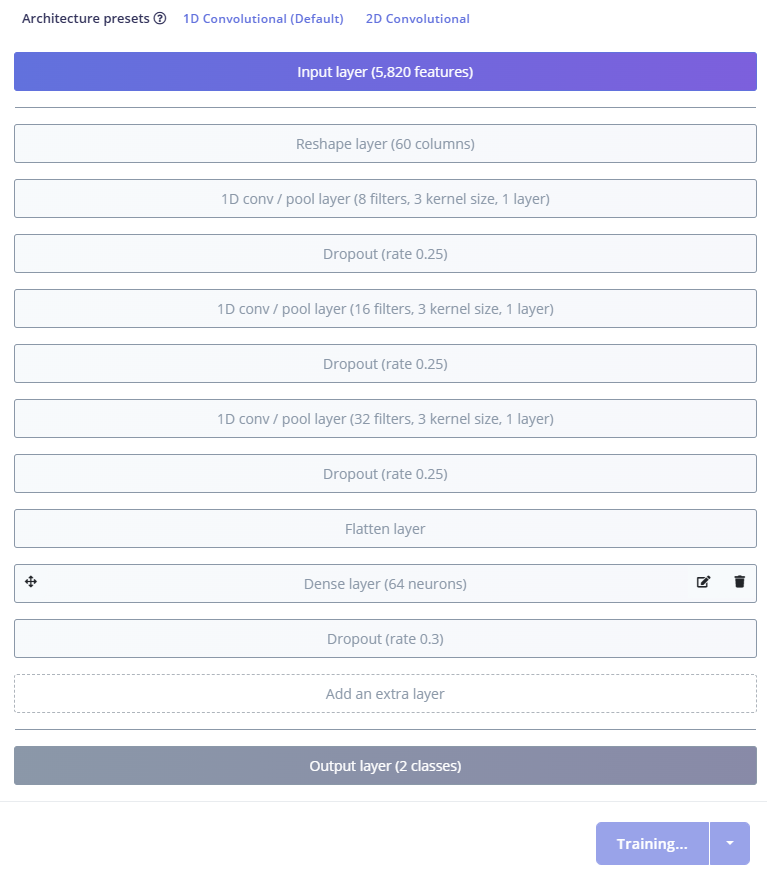
→ Dense Layer（64 units， relu

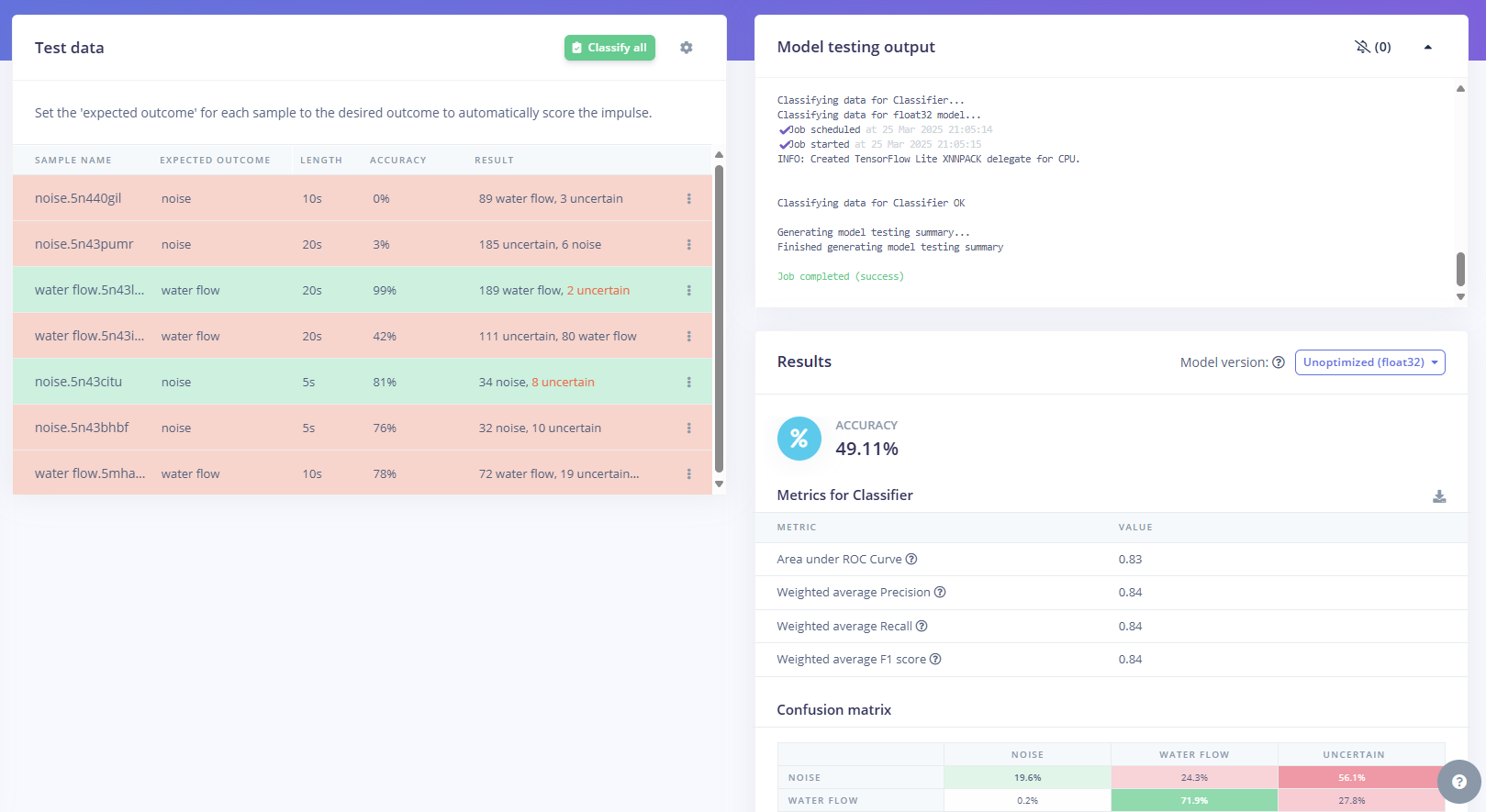
→ Dropout（ 0.3）

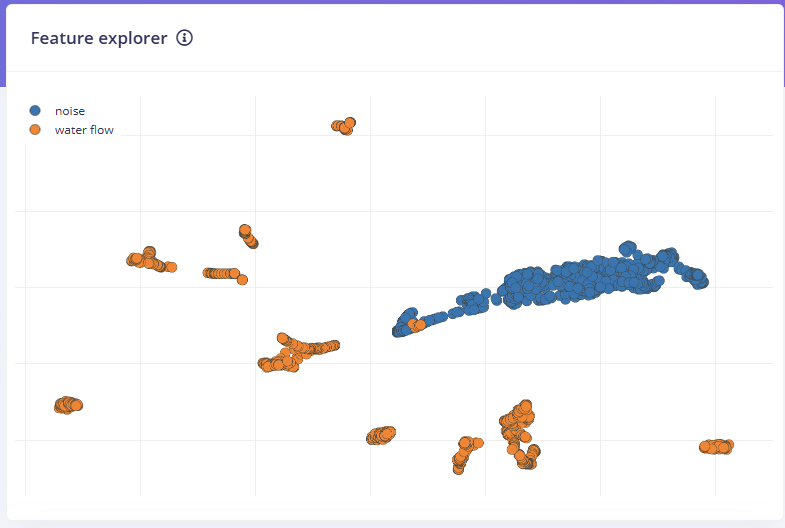
→ Output (2 classes, softmax)

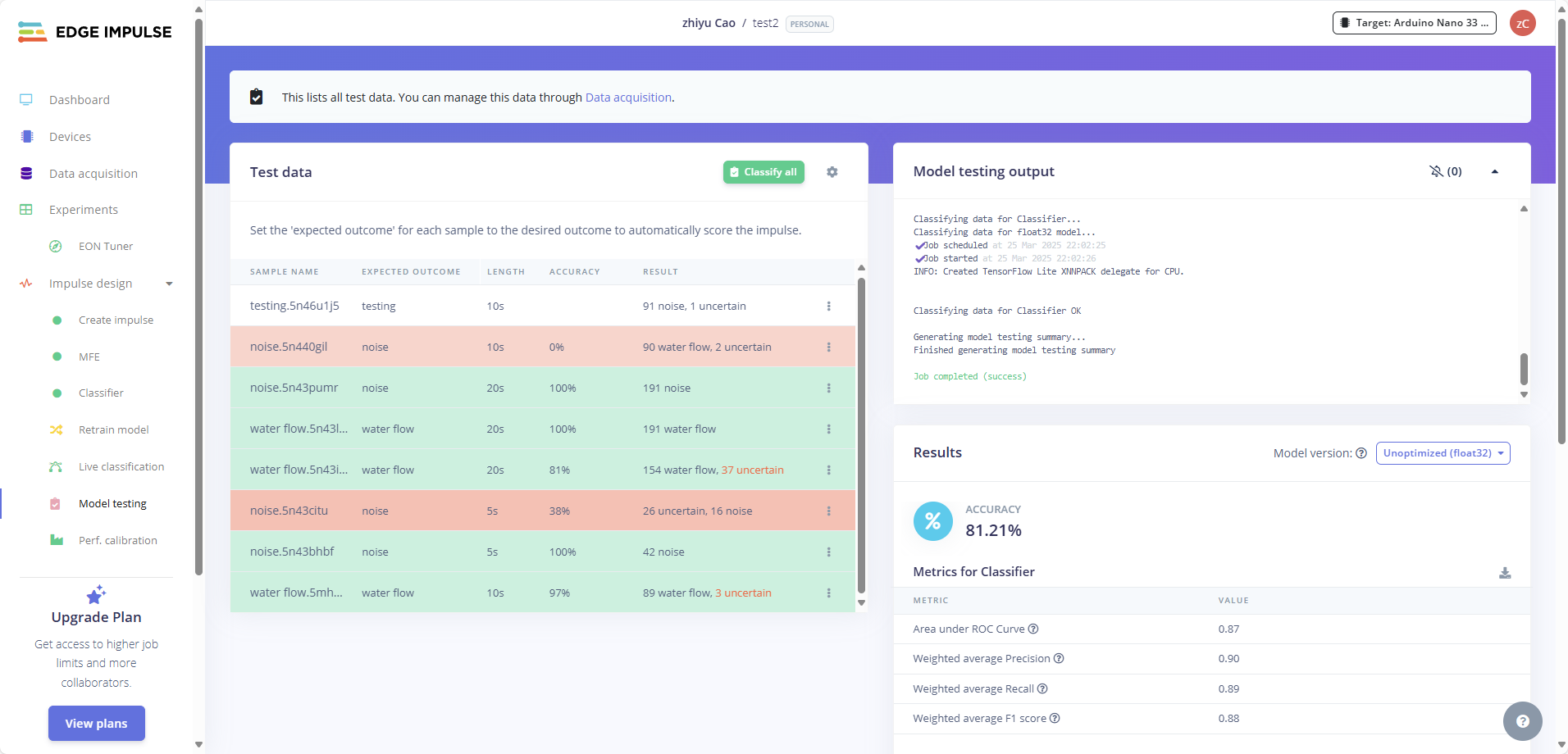
MFE。





Finished training  
I need to do the same thing first and use 100epoch to find out what the best epoch is. Also save this model first and test its correctness.

Well there is absolutely no need to keep this model, forget it. The errors were more in the noise of the pinching bottles and the quiet room with a very strong floor noise. I think it's the bottom noise so I changed the low frequency to 300 and changed the Noise floor (dB) to -60.  
It feels ok. Changed epoch to 50 to try

The accuracy rate of the test has significantly improved! I'm a genius! However, it can be seen that the problem of excessively high background noise has still not been solved.

After the training is completed, at the bottom of the "Model testing" page or the "NN Classifier" page:

Click "View training performance"

The loss and accuracy of each epoch can be seen

The Edge Impulse will automatically save the best-performing model (usually the round with the highest val\_accuracy).

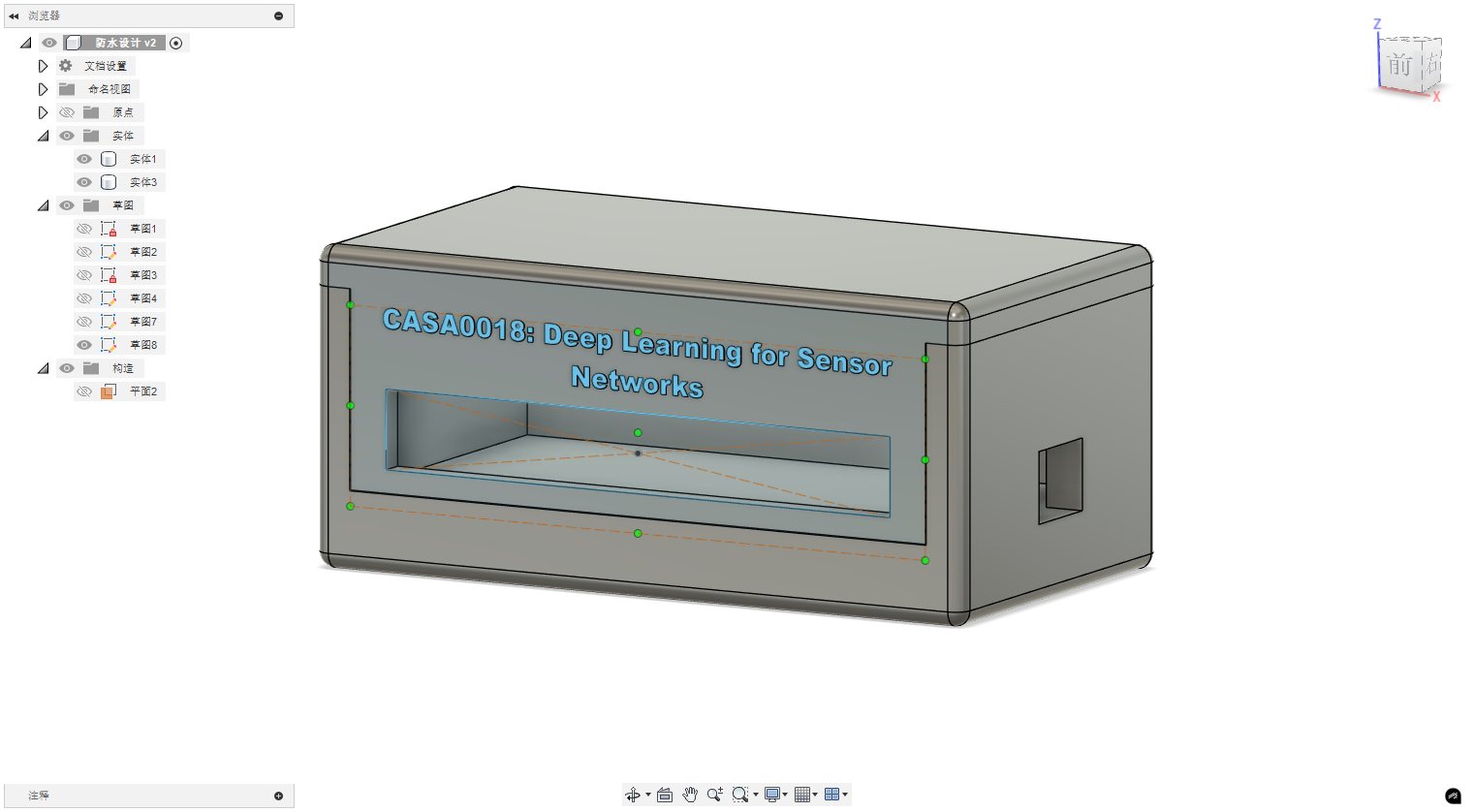
Click on "Use this model" at the bottom and this will be regarded as your current "best model".

I can't find this page. Just ask

It seems that nobody knows. I'll check it next week.

# Week 3

I have started to design the waterproof treatment and it must be done before I return to my country. I will give myself a vacation next week because I'm going back to my country.

The design is done. There is a 3mm slot left in the middle. I want to use it to insert a transparent baffle for waterproofing. As for the waterproofing at the wiring, I want to try the design of the glove closure. Use rubber bands and plastic to make an elastic closure.

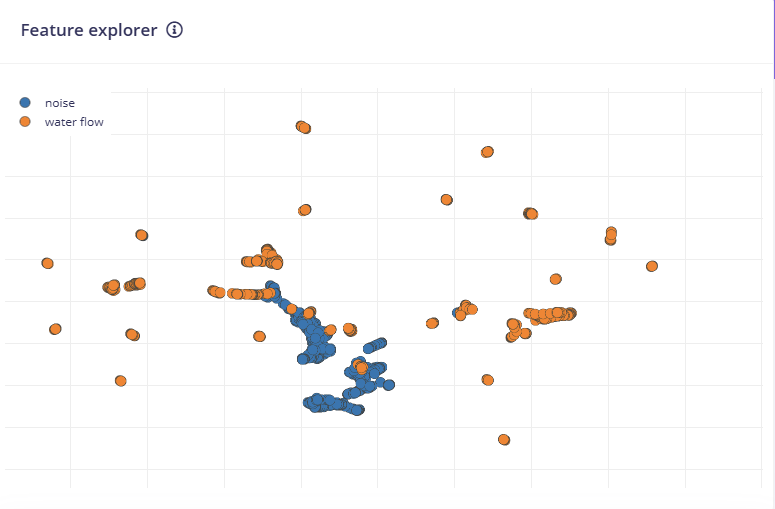
# Week4

Collect data and supplement it to test2. The newly bought Arduino Nano 33 BLE has arrived. Start testing the water resistance and collecting data at the same time. Now I can no longer avoid water vapor and water mist when receiving data.

This week, I'm in China and have been busy with family matters. I can't find enough time for experiments. It's all fragmented time, and I need to be careful to check for any mistakes.

# week5

During my stay in China, I expanded the dataset to a data volume of 16 minutes.



这是现在训练集的 Feature explorer，我复查了在中国整理的数据，Label没有问题。实时证明在套上外壳后，板载的麦克风收音效果出现了变化。为了优化音频切片效果我对MFE 参数进行修改试图优化

1.缩短 Frame Length为0.02

更短的帧长有助于捕捉突发的水声信号变化； Jurafsky & Martin (2021) - Speech and Language Processing

水流声具有高频短时成分，缩短帧长可以提高时间分辨率。 <https://arxiv.org/abs/1807.00129> Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks

2. 适当提升 Filter Number（80）

提高 Mel 频带数，可以让模型更细致地区分水声的频谱形态。 [Sainath et al., 2015] - Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks (ICASSP)

https://arxiv.org/abs/1706.02291 Sound Event Detection Using Spatial Features and Convolutional Recurrent Neural Network

3. 扩大频率范围（尝试 Low = 100, High = 8000）

有些细微的高频“哗哗声”或管道漏水声可能超出 8kHz； <https://arxiv.org/abs/1609.04243> [Choi et al., 2017] - Convolutional recurrent neural networks for music classification

增大频率范围可能增强“水声”检测能力，尤其是在静音环境中。 <https://ieeexplore.ieee.org/document/7952132> [Hershey et al., 2017] - CNN architectures for large-scale audio classification (Google AudioSet)（ Convolutional Neural Networks (CNNs) have proven very effective in image classification and show promise for audio. We use various CNN architectures to classify the soundtracks of a dataset of 70M training videos (5.24 million hours) with 30,871 video-level labels. We examine fully connected Deep Neural Networks (DNNs), AlexNet [1], VGG [2], Inception [3], and ResNet [4]. We investigate varying the size of both training set and label vocabulary, finding that analogs of the CNNs used in image classification do well on our audio classification task, and larger training and label sets help up to a point. A model using embeddings from these classifiers does much better than raw features on the Audio Set [5] Acoustic Event Detection (AED) classification task. ）

4. FFT Length 可以保持 512

如果你将频率范围扩大，可以尝试 FFT length = 1024 以提升频率精度，但代价是计算量提升。 FFT 长度越大，频率分辨率越高（Δf = fs/N）；

如果要精细分辨 200Hz vs 300Hz 的水流频段差异，512 点可能不够；

1024 提供更细粒度，但会增加运算复杂度（重要在 Edge 上部署时权衡）。

<https://www.karolpiczak.com/papers/Piczak2015-ESC-Dataset.pdf> [Piczak, 2015] - ESC: Dataset for Environmental Sound Classification

5. 数据增强：

启用：

Time shift（声音偏移）

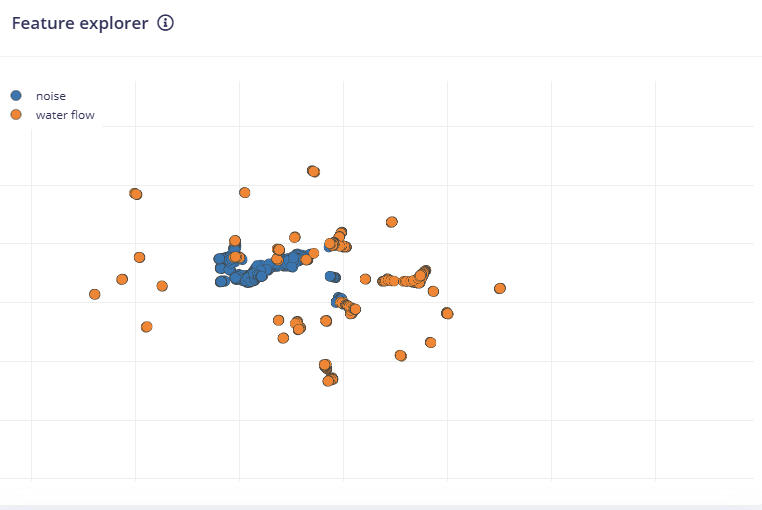
Add noise（白噪声注入）

Time masking / Frequency masking（防过拟合）

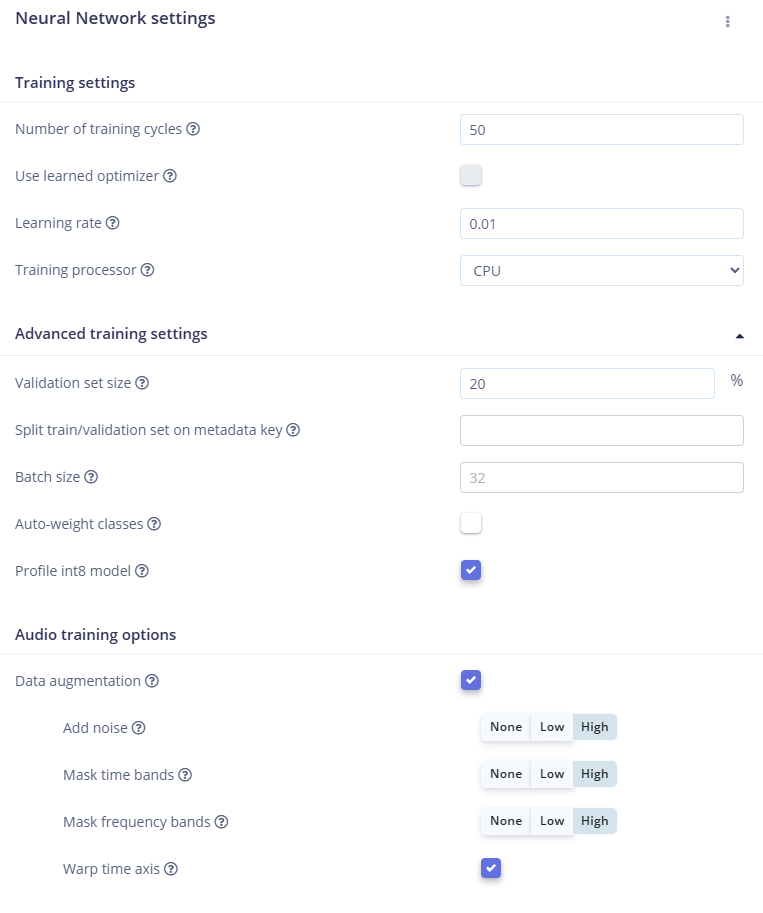
可以让模型更鲁棒地识别不同时段的水流特征。

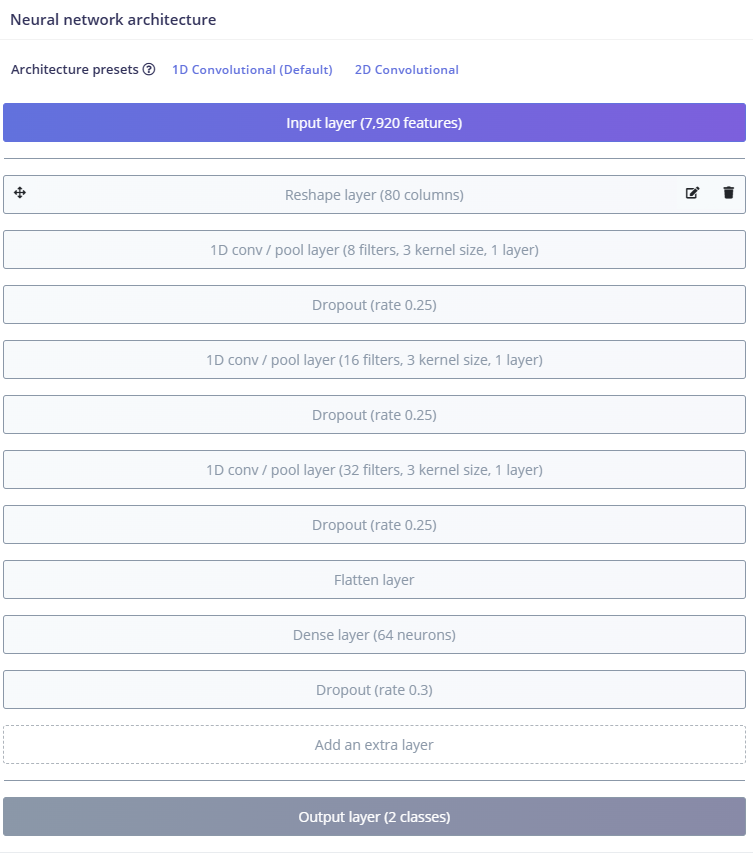
[Park et al., 2019] - SpecAugment: A simple data augmentation method for automatic speech recognition（Google Brain）

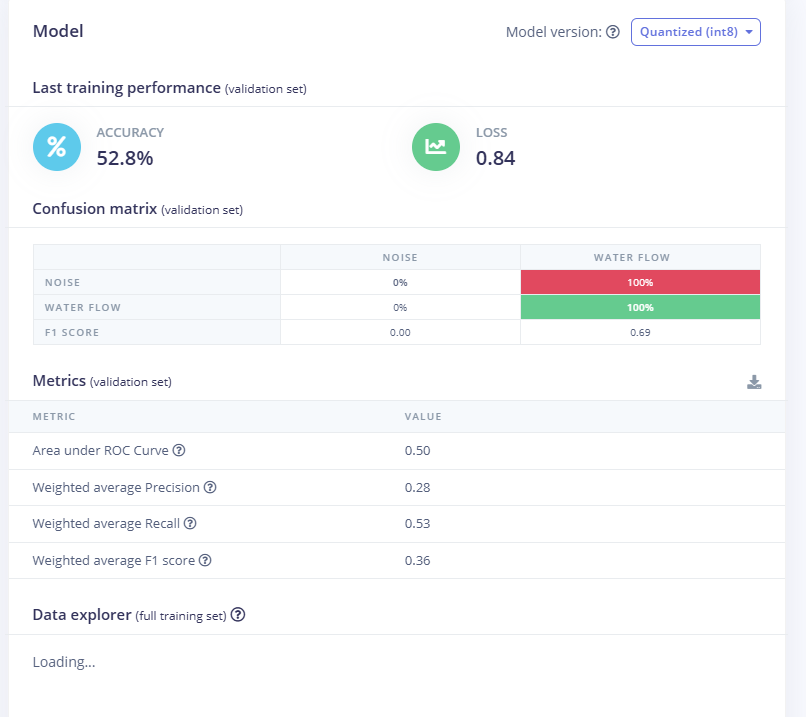
https://ieeexplore.ieee.org/document/7829341 [Salamon & Bello, 2017] - Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification



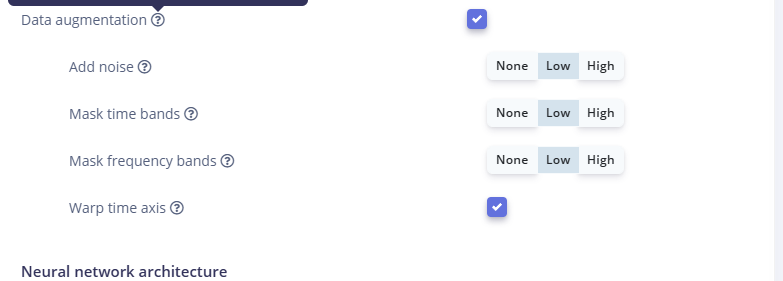
Now the Feature explorer has changed like this. I'm not sure if such feature points have been optimized. I plan to conduct training directly and then test.

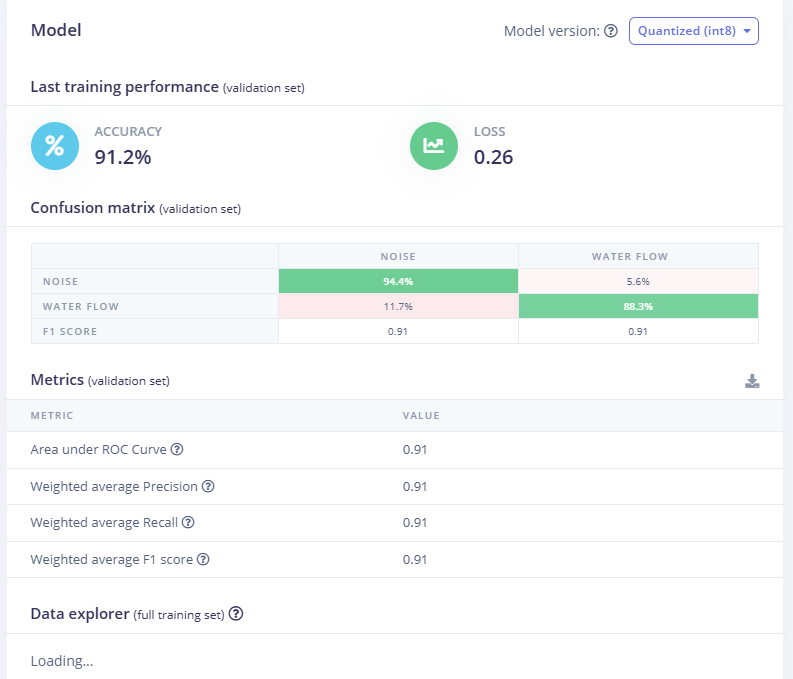




c The part of classifier will not be changed for now. Start trainin、The accuracy rate is very low. It's very bad. I think it might be that the data enhancement Settings are too strong → disrupting the characteristics of the water flow sound

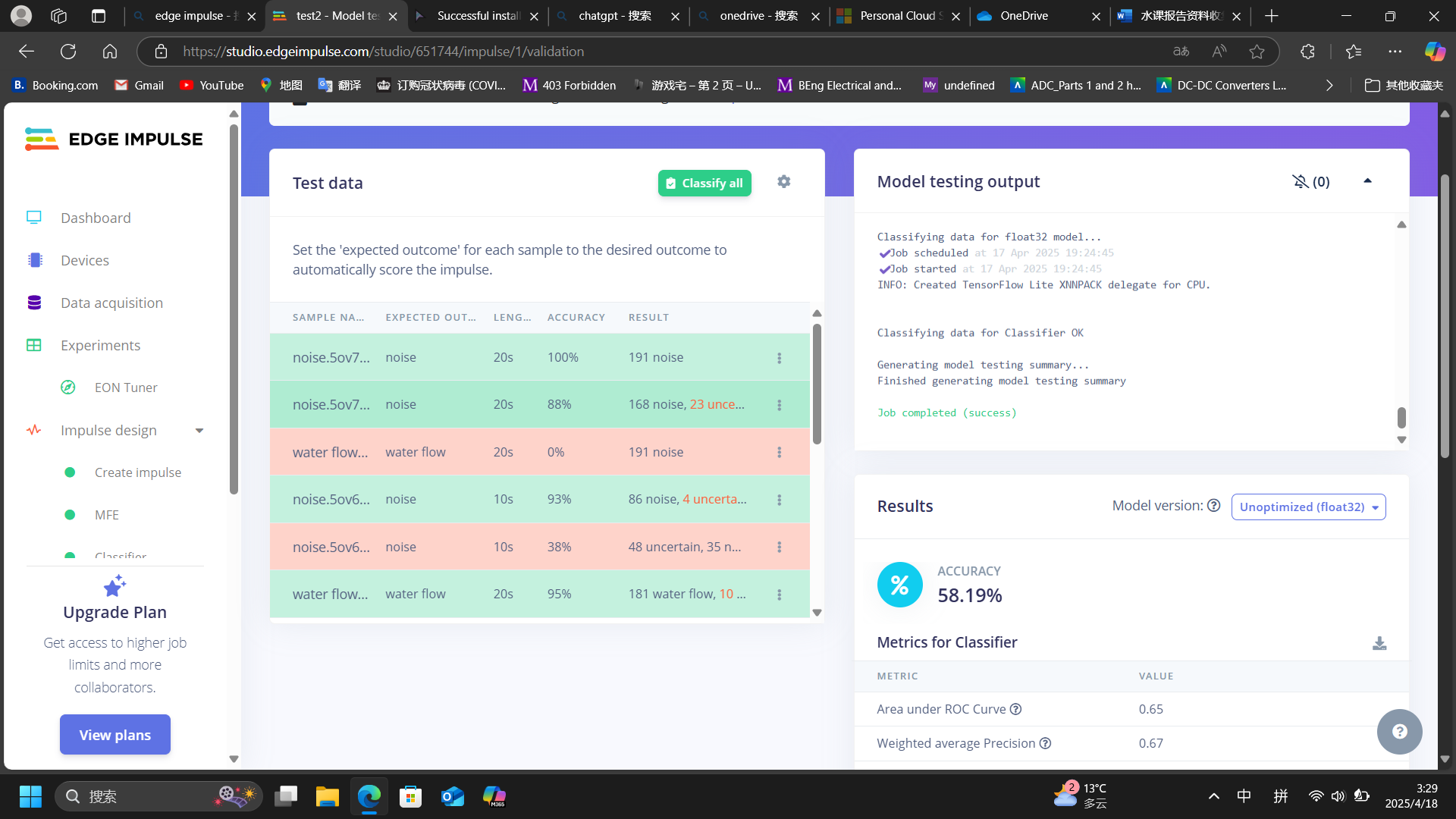
When high-level noise, time masking, frequency masking, and time distortion are enabled, for tasks with weak differences like "water flow sound vs. noise", excessive enhancement may make it difficult for the model to learn its original features. So make the following changes to the Settings





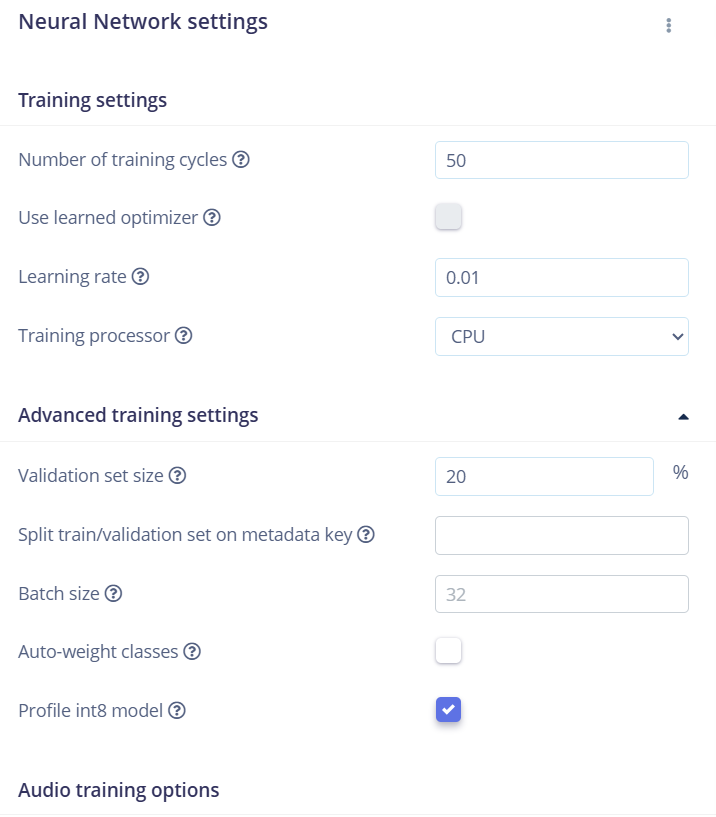
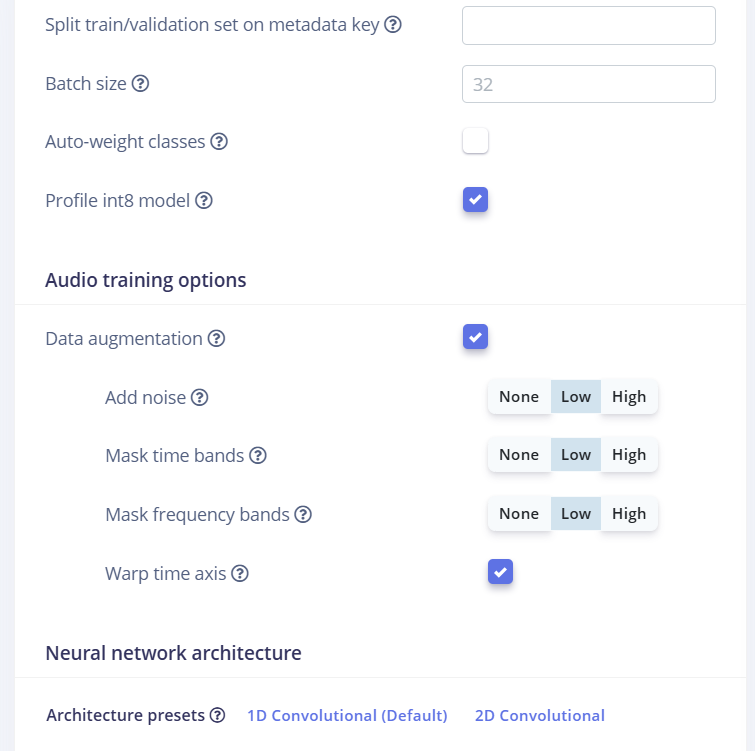
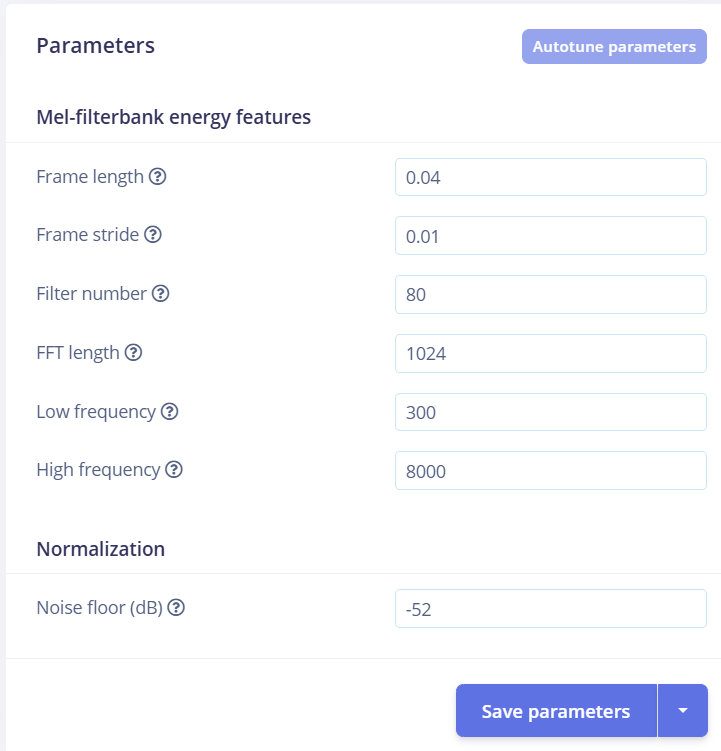
The situation has improved somewhat. However, according to the content in the Training output, the loss starts to increase after epoch15, which puzzles me. Why does overfiting come earlier when the data volume increases?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Epoch** | **Train Accuracy** | **Val Accuracy** | **Val Loss** | **评论** |
| 3 | 93.1% | 70.5% | 0.47 | 起步不错 ✅ |
| 5 | 94.1% | **85.1%** | 0.38 | 初步提升 ✅ |
| 9 | 95.0% | **86.4%** | 0.30 | 很棒，模型开始泛化 ✅ |
| **15** | **95.9%** | **91.4%** | **0.26** | 🎯 **最佳点**：模型泛化能力最强 |
| 27 | 96.1% | 86.9% | 0.30 | 依然优秀 |
| 50 | 96.6% | 70.6% | 0.75 | 已明显过拟合 ⚠️ |

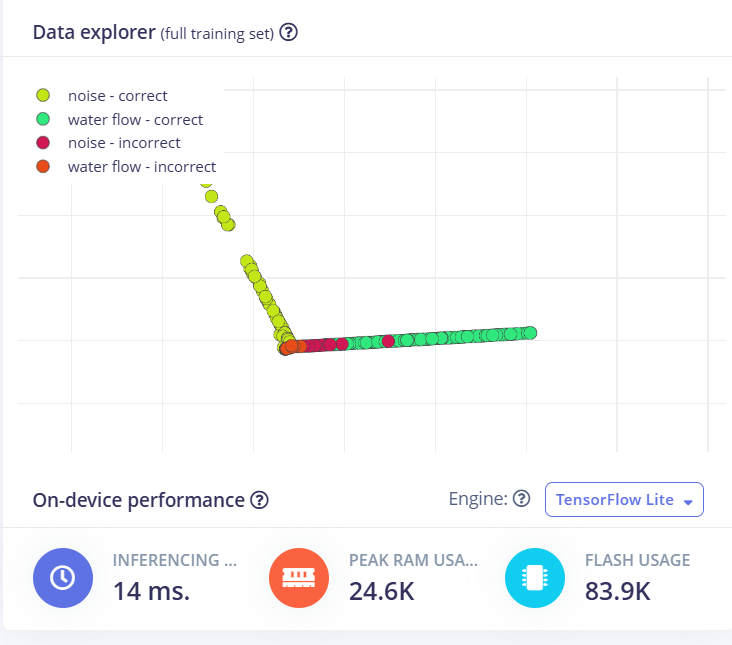




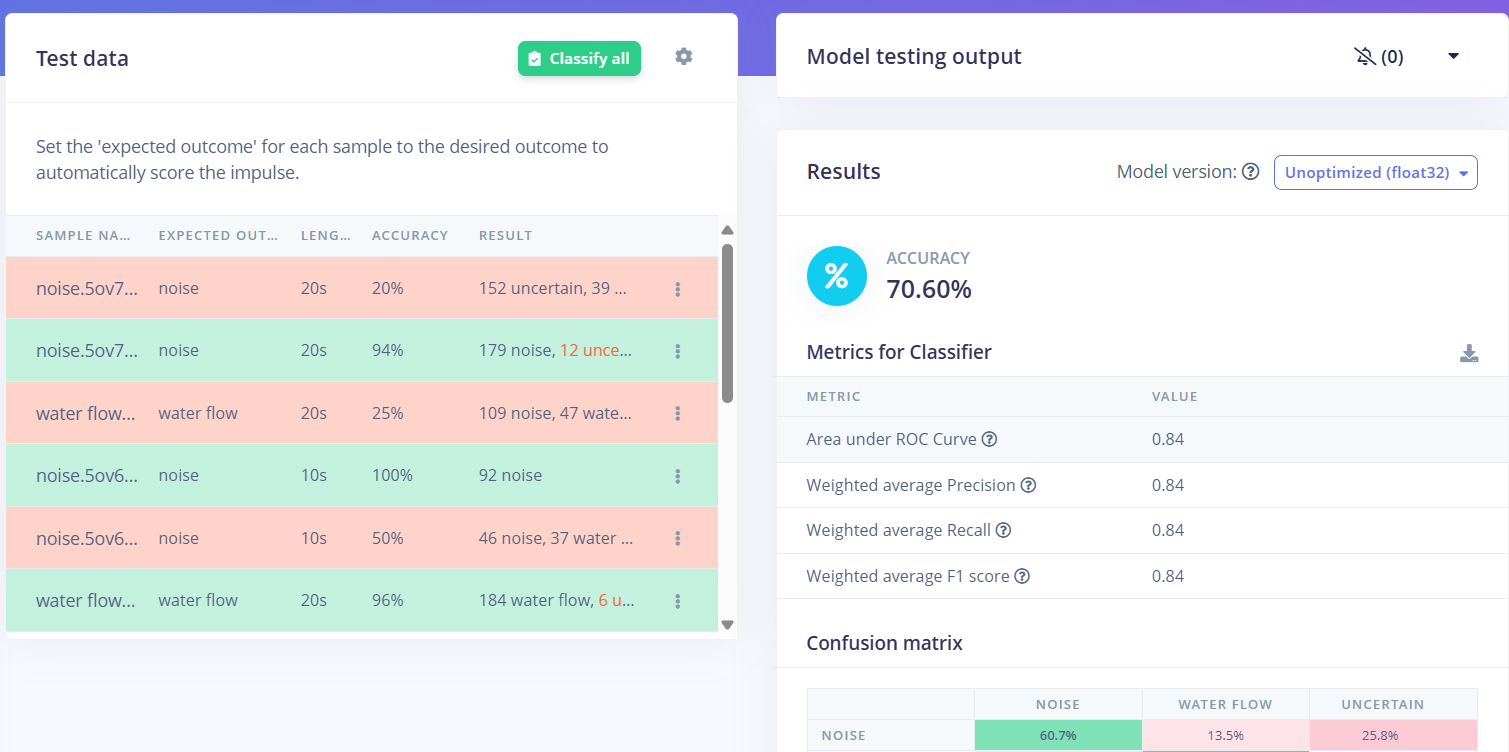
The training results were not satisfactory. I adjusted the Epochs and trained again. If it failed, I changed the Frame length back to 0.04 and slicing the training set once more. Finally, I made the following changes to the MFE page:



The following results were obtained



Its performance in the test set is as follows: A correct rate of 70.60% is acceptable, but it is obviously not as good as the last time. I should not have obtained this result by adding the data set and the test set.



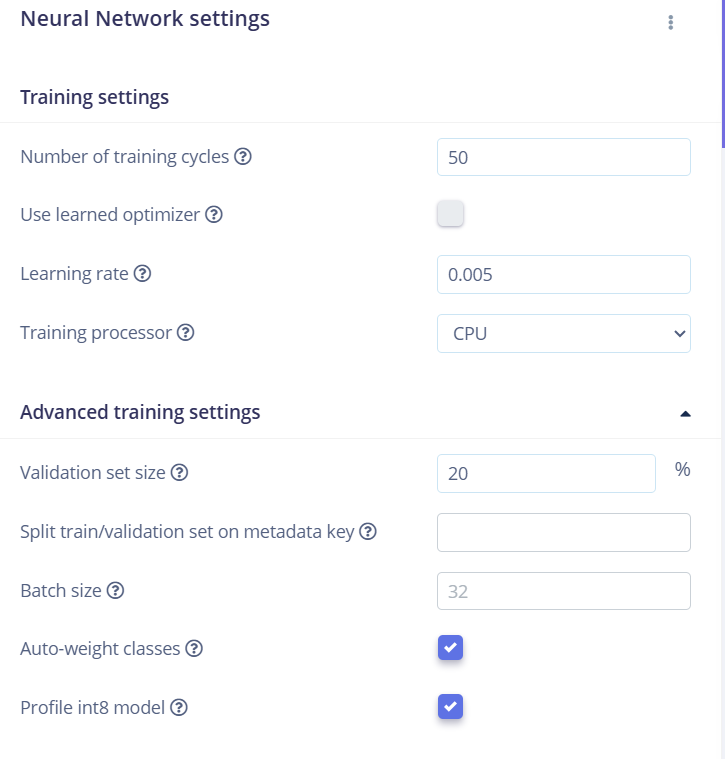
I attempted to avoid the overfitting of the model by reducing the learning rate but increasing the epochs, but edge impulse does not support a training volume exceeding 20 minutes.

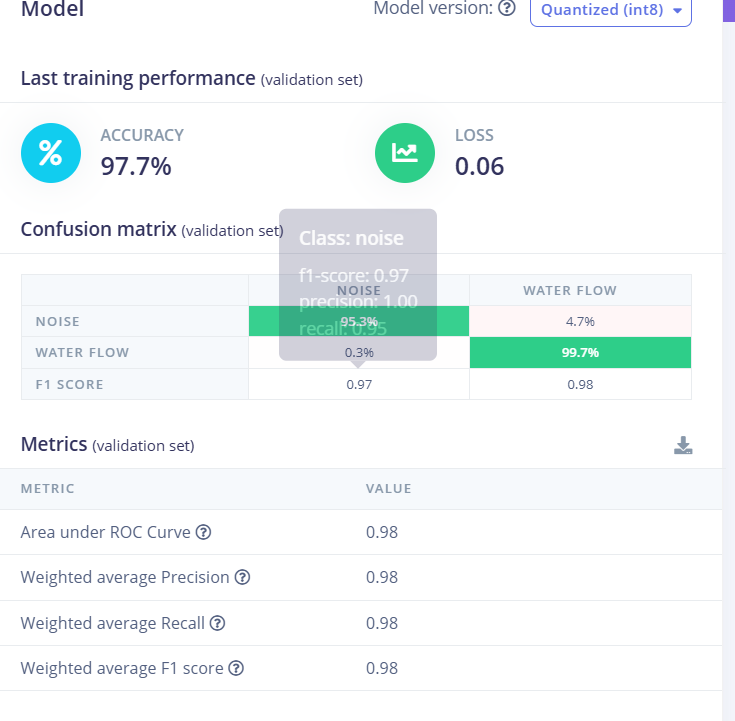
ERR: Estimated training time (26m 47s) is larger than compute time limit (20m 0s).

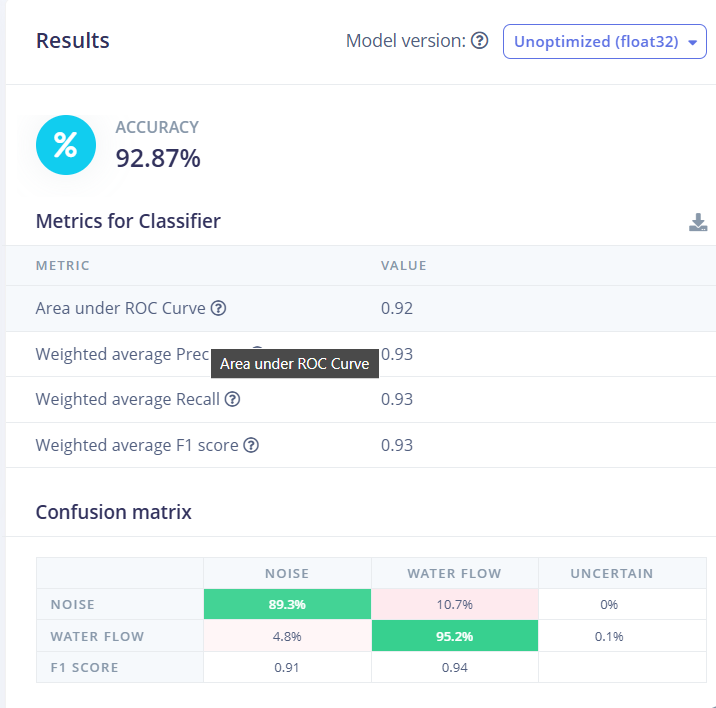
See https://docs.edgeimpulse.com/docs/tips-and-tricks/lower-compute-time on tips to lower your compute time requirements.

Alternatively, the enterprise version of Edge Impulse has no limits, see https://www.edgeimpulse.com/pricing for more information.

Application exited with code 1

Job failed (see above)  
So with limited resources, I can only choose to weaken the reinforcement learning, use the existing training and find the most suitable epochs. By the way, I have changed the window side of the sampling to 2 seconds because I think it is reasonable to use even if it takes 2 seconds to identify the water flow that has not been closed. Considering the model size and slice duration, I have also reduced the FFT Length to 512, mainly to save time. However, at this point, it also led me to make a guess: Could it be that the training set after the data was strengthened is difficult to find features, and the upper limit of the training time of edge impulse makes it impossible for me to continue increasing the epoch to obtain a more complex model in a simple way? So I simply turned off the reinforcement learning and simply used the following data for training:

The following results were obtained ：

And in the test set, its performance was also very perfect

It seems that the Window side can only be 1000. The model I ran in 2000ms cannot run on BLE33. What was used was EI\_CLASSIFIER\_SLICES\_PER\_MODEL\_WINDOW = 4 (each model window is divided into 4 pieces).

The number of samples required for each slice is relatively high. If the board cannot collect an entire slice in time, it will lead to an error! I want to directly modify the code EI\_CLASSIFIER\_SLICES\_PER\_MODEL\_WINDOW = 4 to 1, but it seems that it won't work. The model files of this version are called ei-test2-arduino-1.0.5 and ei-test2-arduino-1.0.6.

ei-test2-arduino-1.0.7 is a peculiar model with a version training set accuracy rate as high as 100%. Detection in the test set found that as long as there is interference, its detection ability will significantly decrease and its robustness is very low.

Considering the problems encountered during deployment, I changed "window size" and "window increase" back to the values in the official tutorial. The ei-test2-arduino-1.0.8.zip file obtained thereby is also a very accurate model, but it still lacks robustness. It can still recognize the sound of rain as the sound of flowing water, and it is also difficult to distinguish the sound of flowing water in the case of a large amount of background noise.

The deployment test of ei-test2-arduino-1.0.7.zip has ended. It can operate in an ideal environment and has a certain anti-interference ability, but it has no detection ability for smaller water flow sounds

noise: 0.99609

water flow: 0.00391

Predictions (DSP: 108 ms., Classification: 39 ms., Anomaly: 0 ms.):

noise: 0.99609

water flow: 0.00000

Predictions (DSP: 107 ms., Classification: 39 ms., Anomaly: 0 ms.):

noise: 0.99609

water flow: 0.00000  
这是部分取样效果，以此推断将启动蜂鸣器的阈值设置为water flow:0.9是合理的。

I have integrated the buzzer code. This file is currently in the uncompressed folder test2\_inferencing1.0.7. The experimental test results show that it works well in a quiet and ideal environment. This instrument can even detect whether there is the sound of water flow at a distance of more than one meter from the water source. Considering the instability of deep learning, I set the threshold for triggering the buzzer very high as: 0.9f is used to avoid false triggering by complex background noise. This makes the detection very unstable if the water flow is weak or there is environmental sound interference. However, this brings a short triggering effect just in the case of weak water flow. But when the water flow is obviously strong, the continuous triggering of the buzzer will change the effect to a sharp and long sound. During the long-term stability test, I also discovered some problems. For example, after a strong water flow continuously triggered the water flow label for several seconds, the output value of the Water Flow label might remain above 0.9, and then suddenly return to normal after a long period of time, or be forcibly powered off by me. It might be that the constantly triggered buzzer serves as background noise, interfering with the model's judgment and causing it to constantly misjudge. There is also the sound of water flow. This indicates that if the model's accuracy cannot be further improved, the design of using the buzzer as a reminder device must be changed. It might be better to send a message to the mobile phone via Bluetooth that the water is not turned off. But for now, the current model works well in a quiet environment or an environment with only human voices.

In conclusion, the current model and deployment rate measurement merely offer a development direction. Based on the existing model and accuracy rate, in order to easily achieve multi-faceted expansion. For instance, instead of using the on-board microphone, an external high-quality microphone can be used to enhance the quality of the received audio. Or, a sliding window and consecutive hit judgment can be added to the code. Only when the "target tag" exceeds the threshold for N consecutive inferences will the buzzer be triggered. However, this undoubtedly increases the difficulty of triggering the buzzer in a small water flow or noisy environment.

**To-do list: I think every qualified model can be trained using both classification and transfer learning methods. However, at least after the model testing accuracy rate is higher than 80% and the best epoch is found, comparisons should be made; otherwise, it makes no sense. Meanwhile, the different layer architectures of classification should also be compared directly.**

**At this stage, I also think that I should set a reasonable model qualification standard. I think the initial qualification should be that the modeling teat accuracy rate is higher than 80%, and the second step is to pass that low-noise noise test. The third step is the long-term actual testing after packaging.**

**Carry out waterproof treatment; otherwise, data in a water mist state cannot be collected. Two hollowed-out boards can be printed with PLA, with cling film sandwiched in the middle, and the connection points can be made using mortise and tenon joints.**

**I should write a program that can easily switch between different model deployments to improve the efficiency of testing and deployment. Otherwise, every time you redeploy, you have to manually delete the library in the libraries, then reload the model and modify the code, which is very troublesome.**

**The deployment of the latest model, transfer learning, failed and it is not feasible.**