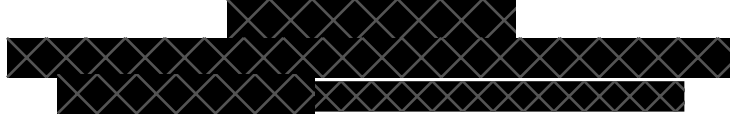


Mars Spectrometry 2: Gas Chromatography - Second place solution



Abstract—The Mars Spectrometry 2: Gas Chromatography challenge was sponsored by NASA and run on the DrivenData competition platform in 2022. This report describes the solution which achieved the second-best score on the competition’s test dataset. The solution utilized two-dimensional, image-like representations of the competition’s chromatography data samples. A number of different Convolutional Neural Network models were trained and ensembled for the final submission.

I. CHALLENGE SUMMARY

The goal of the *Mars Spectrometry 2: Gas Chromatography* challenge [1] was to develop a model, which could automatically process supplied gas chromatography-mass spectrometry (GCMS) data files. The challenge was set up as a supervised multi-label classification machine learning problem, where for each *training* sample data file, nine binary nonexclusive *target* labels were provided. The challenge participants were required to predict the target labels for a test dataset for which the ground-truth labels were not supplied. Submitted test predictions were evaluated by multilabel aggregated *log loss* score (a lower value is better), which penalized confident but incorrect predictions.

The first place solution achieved 0.1443, this second place solution had 0.1485 (+2.9% higher relative to the first place), closely followed by the third 0.1497 (+3.7%). The corresponding three best scores from the first Mars Spectrometry (Mars-1) competition [2] were: 0.0920 (1st place), 0.1160 (+26%) and 0.1189 (+29%), where the same *log loss* metric was used.

Two observations could be made by comparing the Mars-1 and Mars-2 final leaderboards. First, the top three solutions of Mars-2 are comparable within 4% of the used performance metric, while the first-place solution of Mars-1 had a 26% improvement

gap compared to the second-best solution. This superiority gap motivated me to focus on applying the first-place solution of Mars-1 [3] to the Mars-2 challenge.

The second observation was due to the 56% deterioration of the winning Mars-2 score (0.1443) compared to the corresponding Mars-1 score (0.0920), indicating that Mars-2 was a “harder” challenge to solve.

II. SOLUTION DEVELOPMENT

In Mars-2, each training and test data file was a CSV file containing three columns: *time* (in minutes), *mass* (mass-to-charge ration, m/z) and *intensity* of detected ions per second in arbitrary but relative (within one sample) units. The main critical difference of Mars-2 samples (compared to Mars-1) was the absence of temperature values. As per the competition’s problem description, time values did not have any spectrometry significance by themselves and were only proxies for missing temperature values. Furthermore, the only thing we could assume was that sample temperature increased with time, “*but the temperature ramp is not exactly known nor the same across samples*” [1].

This solution evolved from the published first-place solution of the predecessor Mars-1 competition [2], *Mars Spectrometry: Detect Evidence for Past Habitability* [3]. The CSV sample files from the training dataset were converted to 2D images and used for training (or fine-tuning) Deep Learning models such as Convolutional Neural Networks (CNNs). The ImageNet [4] pre-trained, PyTorch-based [5] backbones and models were used from the timm [6] python package.

Working within the limits of not knowing the spectrometry-relevant temperatures, time values

were scaled in all samples to the $[0,1]$ range and then binned to a configurable number of time slots N_t , where $N_t = 192$ was used in the final submission. When training a CNN, the time dimension was randomly batch-wise resized on GPU within the 128-256 range of values. Then, at the inference phase, the corresponding TTA (test-time-augmentation) was done by averaging time sizes (5 steps of 32, centred at 192). Such time resizing during training and testing (via TTA) assisted in capturing some temperature-related data features. However, samples with "slower" ramping up to "lower" (than the training mean) temperatures as well as samples with "faster" ramping up to "higher" temperatures would be unlikely to yield correct predictions within this solution. It is possible that the large variations in the ramping rates together with the large variations in the actual final temperatures were the main reason why Mars-2 leaderboard scores were much worse than the corresponding Mars-1 values. Hence:

Actionable recommendation: *Even if the exact time-temperature ramp functions are not known, the availability of start and end temperature values (per sample) should greatly improve this solution.*

For example, if the maximum temperature of one sample T_i is double the temperature of another sample T_j , $T_i = 2 * T_j$, then the i 'th sample time values should be binned into a proportionally larger number of time slots. Furthermore, the variations in the ramping functions could be modelled by random time warping, for example via

$$f(t) = t^\alpha, \quad 0.5 \leq \alpha \leq 2, \quad 0 \leq t \leq 1, \quad (1)$$

which was also attempted in this solution but was not fully explored to verify its utility.

A. Two-dimensional representations

Dot-scatter plot in Fig. 1 displays the raw intensity values for $m/z = 18$ in one of the samples. The red line is the same values averaged into 192 equally spaced time slots. Note that the raw values and the 192-bin smoothed values are very similar.

Fig. 2 displays the actual 2D representation of the same sample for all mass and time values, where 256 mass bins (image rows) and 192 time slots (image columns) were used. Mass (i.e. m/z) values were rounded to the nearest integer values retaining only the first 256 values. The positional encoding

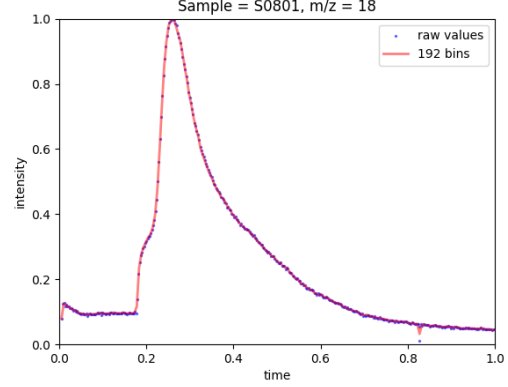


Fig. 1. Raw intensity values for sample S0801 at $m/z=18$

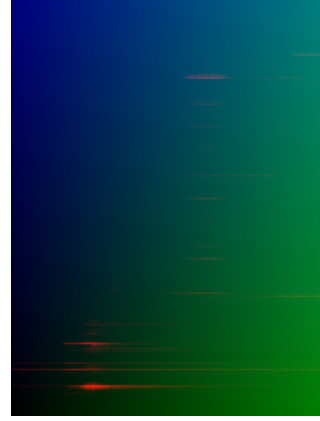


Fig. 2. Sample S0801 converted to 2D representation and saved into red channel, where y-axis is mass rows ($0 \leq m \leq 255$) and x-axis is time columns ($0 \leq t \leq 191$). The green and blue channels are loaded with $t/191$ and $m/255$ as per [3], correspondingly.

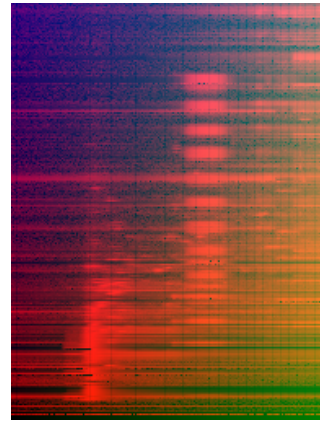


Fig. 3. Same as in Fig. 2 but divided by maximum column values (mass-normalization).

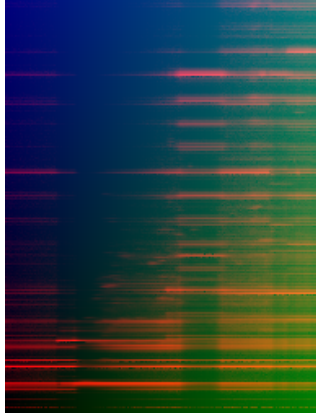


Fig. 4. Same as in Fig. 2 but divided by maximum row values (time-normalization).

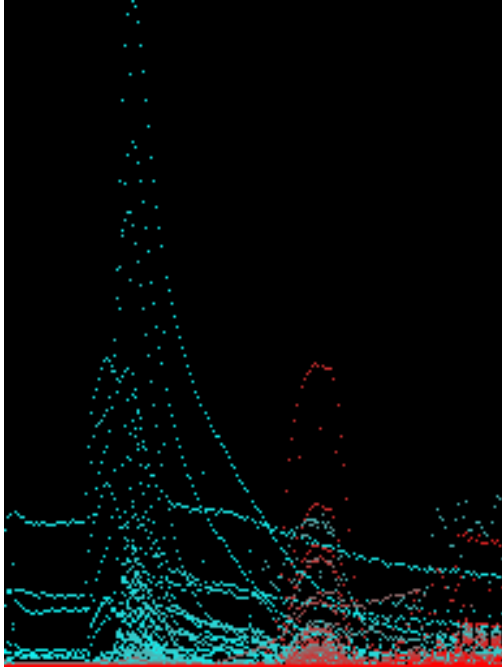


Fig. 5. Same as in Fig. 2 but with swapped mass and intensity values, and without the mass and time positional channel encodings. The red color component was proportional to mass, where a mass value of 255 corresponded to the maximum color red value of 255 (in uint8 RGB color encoding).

of mass and time dimensions into separate image channels was adopted from the Mars-1 first-place solution [3].

To reveal feature-rich patterns in the 2D representations, the images could be divided by maximum column (mass-normalization [3], Fig. 3) or row (time-normalization Fig. 4) values. In fact, all the models used in this solution’s final submission

were either mass or time normalized.

All intensity values were converted to a logarithmic scale via either one of the following two ways

$$\log_{10}(1 + 10^3 \times a), \quad 0 \leq a \leq 1, \quad (2)$$

$$\log_{10}(a), \quad 10^{-4} \leq a \leq 1. \quad (3)$$

Another completely different (from the ones considered above, see Figs. 2-4) conversion to a 2D representation (see Fig. 5) was attempted but did not contribute to the final ensembling. In that conversion variation, mass values were encoded as the pixel color values while the intensity values were binned into 256 y-axis values (image rows). It is likely that this approach could be made useful with further development as it represents the data samples from a different point of view.

B. Models and training pipeline

I followed the first-place solution [3] of the previous Mars challenge [2] by converting the mass spectrometry data into 2D images. A total of 13 CNN models and data-processing configurations were ensembled for the final submission. For ensembling, averaging logits (inverted 10^{-4} clipped sigmoids) rather than probabilities improved both validation and test leaderboard results.

I experimented with different conversion configurations (see preceding subsection) and found that the first 256 mass values (y-axis) were sufficient. For the time axis (x-axis), 192 time slots were selected as a reasonable baseline value, where larger values slowed down the CNN training.

In this competition, time was only a proxy for temperature. Therefore, I only explored ideas where exact dependence on time values was not required. That led to two key ideas, which pushed this solution to the winning range. First was, as explained earlier, the random resizing of the time dimension with the corresponding TTA. The second win-contributing idea was creating a custom time-averaged head, where only the time dimension of the CNN backbone 2D features was averaged (rather than both the mass and time dimensions) before the last fully connected liner layer.

For the second (standard model) architecture, a small improvement was gained by using the full pre-trained timm models before the required 9-label linear classifier, where the ImageNet classification

heads were retained rather than just the models' backbones.

An extensive search of timm [6] pre-trained models found HRNet-w64 [7] to be particularly accurate for the considered 2D representations of data. The following other timm's pre-trained backbones were also used: dpn98 and dpn107 [8], regnetx_320 [9] and resnet34.

Small but consistent improvement was gained by encoding the *derivatized* meta-data column as a 2-channel image and adding a trainable conversion layer before a CNN backbone.

While the original Mars-1 solution [3] used noise augmentations extensively, I was not able to achieve any consistent validation/OOF loss improvement by adding noise and/or smoothing/pre-processing original data at the training and/or inference stages. Hence, all training and inference were done with the original data converted to 2D images (as described in the preceding subsection) without any further processing.

All models were trained for 20 epochs with a cosine learning rate schedule and linear 2-epoch warm-up and a base learning rate of 10^{-4} . About half of the ensembled models were trained with mixup [10] (a probability of 0.1).

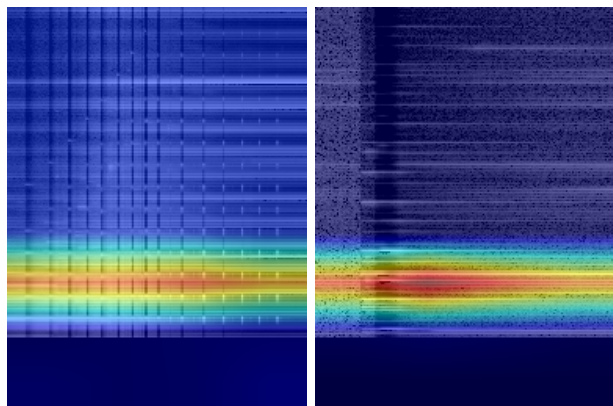


Fig. 6. Grad-CAM++ visualisation of two samples (S0047 and S0078) containing *hydrocarbon* compounds.

III. INTERPRETABILITY/EXPLAINABILITY

Bonus algorithm explainability awards were run as part of DrivenData's "Where's Whale-do?" competition [11]. One of the winners of that bonus round was the 4th place submission [12], which utilized the Grad-CAM [13] package containing

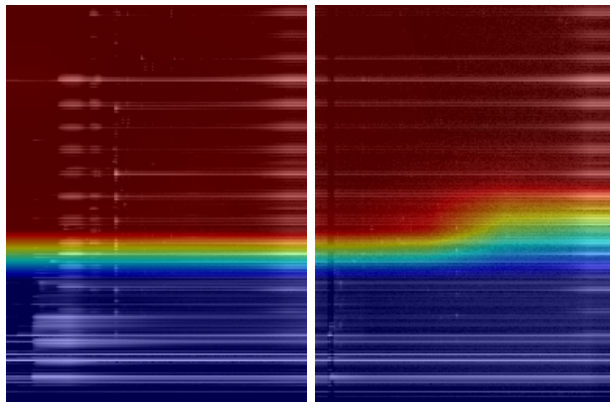


Fig. 7. Same as Fig. 6 but for S0107 and S0305 samples containing *mineral* compounds.

the state-of-the-art methods for *Explainable AI for computer vision*. The Grad-CAM approach of [12] was replicated here to illustrate its feasibility to explain the solution's predictions.

The best out-of-fold (OOF) model of this solution was the custom time-averaged head on top of the head-less HRNet-w64 [7] backbone, where the ImageNet pre-trained HRNet-w64 backbone was available in the timm [6] package. The Grad-CAM++ [14] implementation from [13] was used to calculate the gradients of the backbone's last 2D layer. Fig. 6 shows two OOF samples ground-truth labelled to contain only *hydrocarbon* compounds, where they were correctly and confidently (probability > 0.9) predicted by the model. In all figures except Fig. 5, $m/z = 0$ is the bottom row of pixels and $m/z = 255$ is the top row. Comparison of Fig. 6 and Fig. 7, revealed that the mineral compounds (Fig. 7) were classified by the model due to the ions with $m/z > 100$, while the ions around $m/z = 80$ were the main reason for predicting the hydrocarbon compounds. Also, note that the mineral compounds had nearly uniform activations (red-hue colors) across the time dimension (x-axis), while hydrocarbons were more activated at lower temperatures (smaller time values).

REFERENCES

- [1] DrivenData, "Mars Spectrometry 2: Gas Chromatography," 2022, <https://bit.ly/3hoX4hD>, Last accessed on 8-Nov-2022.
- [2] —, "Mars Spectrometry: Detect Evidence for Past Habitability," 2022, <https://bit.ly/3tbJm4I>, Last accessed on 8-Nov-2022.

- [3] Dmytro Poplavskiy, “The Winning Solution for the Mars Spectrometry: Detect Evidence for Past Habitability Challenge,” 2022, <https://bit.ly/3hoo29m>, Last accessed on 8-Nov-2022.
- [4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.
- [5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <https://bit.ly/3hnKEqq>
- [6] Ross Wightman, “PyTorch Image Models,” 2022, <https://bit.ly/3zVYsiy>, Last accessed on 8-Nov-2022.
- [7] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, “Deep high-resolution representation learning for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2021.
- [8] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, “Dual path networks,” *Advances in neural information processing systems*, vol. 30, 2017.
- [9] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, “Designing network design spaces,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 428–10 436.
- [10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [11] DrivenData, “Where’s Whale-do?” 2022, <https://bit.ly/3DYAS5X>, Last accessed on 10-Nov-2022.
- [12] Raphael Kiminya, “Where’s Whale-do? Explainability Bonus, 4th place,” 2022, <https://bit.ly/3tdRkKB>, Last accessed on 10-Nov-2022.
- [13] J. Gildenblat and contributors, “Pytorch library for cam methods,” 2021, <https://bit.ly/3TnJtVI>, Last accessed on 10-Nov-2022.
- [14] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks,” in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 839–847.