DOI 10.5162/17dss2024/P34

Optimisation of Convolutional Neural Networks for MOS Gas Sensors

Joshua Petry, David Schu, Till Mertin & Dennis Arendes

Lab for Measurement Technology, Saarland University, Saarbruecken, Germany

Contact: d.arendes@lmt.uni-saarland.de

Introduction

Metal oxide semiconductor (MOS) gas sensors are widely used in applications fields ranging from indoor air quality (IAQ) monitoring and control [1] to fire detection [2]. These sensors offer a low-cost in situ alternative to gas chromatography [3]. Temperature-cycled operating (TCO) modes are proven to enhance selectivity as well as sensitivity [4]. However, MOS sensors still require calibration against a known reference, which is often provided by a gas mixing apparatus [5]. Using the measured conductivity of the sensors exposed to a unique gas mixture (UGM), machine learning (ML) algorithms can be trained to classify or quantify gases.
In this work, we optimized the convolutional neural network (CNN) architecture from [5] and [6], hereinafter referred to as the "reference network" with the goal of further improving quantification performance by reducing the RMSE and minimizing the number of learnable parameters. Reducing the complexity of CNNs offers several advantages, such as shorter training times and more efficient deployment on hardware with limited computational resources. The optimized network was then compared and evaluated against the reference network for this specific quantification task.

Functionality of CNNs

The structure and functionality of a CNN can be divided into four main parts: data input, feature extraction and dimensionality reduction, regression or classification, and output of results.
Like any artificial neural network, a CNN consists of an input layer and an output layer. The input layer receives data and passes it to subsequent layers. Convolutional Units handle feature extraction and dimensionality reduction. These features are then processed by fully connected layers, which perform either regression or classification. The final results are output in the output layer.
A convolutional unit typically includes a convolutional layer with an activation function and, optionally, a pooling layer. Filters are applied to the input data, and the resulting values form a new matrix called a feature map. This process helps reduce data dimensionality, depending on the filter size and stride.
Nonlinear activation functions, such as ReLU, are used to enable the network to learn complex patterns. Pooling layers further reduce the dimensions by applying a sliding window over the feature maps, returning for example the maximum (max-pooling) or the average value (average-pooling) from each step. This reduction lowers computational complexity and helps prevent overfitting.
The extracted features are interpreted by fully connected layers, which make final decisions for classification or predictions for regression. In a classification task, the output layer contains multiple neurons, each representing a class. For regression tasks, a single neuron is used to predict a continuous value.

Dataset

The dataset used in this study consists of measurements obtained from a SGP40 sensor developed by Sensirion (Stäfa, Switzerland). The sensor operates in a temperature-cycled mode, where all four subsensors, each featuring a different sensitive layer, are sampled. During each cycle, the sensor's temperature varies between 100 °C and 400 °C, producing 1440 sampled conductance values per subsensor. This results in a measurement matrix with dimensions of 4 by 1440 for each cycle, as can be seen in figure 2. The dataset consists of 10304 measurement cycles, with a gas mixing apparatus providing various gas concentrations to the sensor. Different unique gas mixtures were used across these cycles, with some UGMs sampled multiple times. This results in a total number of UGMs that is lower than the number of measurement cycles. Table 1 lists the gases used along with their minimum and maximum concentrations.
Acetone was initially selected as the target gas due to its distinctive characteristics in the collected data**.** To ensure consistency across tests, a standardized data split was applied, dividing the dataset into 70% for training, 10% for hyperparameter tuning, and 20% for network validation.

Tab. 1: Gases and their concentration ranges in the dataset.

Gas	Minimum in ppb	Maximum in ppb
Acetaldehyde	0	985.04
Acetic Acid	0	889.84
Acetone	0	983.74
Carbon Monoxide	100.68	1998.7
Ethanol	0	988.37
Ethyl Acetate	0	888.03
Formaldehyde	0	572.54
Hydrogen	402.89	1998.8
Isopropanol	0	992.11
Limonene	0	296.94
n-Hexan	0	943.11
Toluene	0	1937.7
Water	25.092	74.941

Initial research

1. Sensor fusion

As mentioned earlier, the dataset includes four conductance readings for all four subsensors included in the SGP40 sensor package. Each subsensor features a different active layer, introducing the possibility of different sensor responses per layer in the same gas environment. Figure 1 depicts the conductivity graphs of all four subsensors of the SGP40 sensor during one measurement cycle, highlighting the differences in the conductance profile.

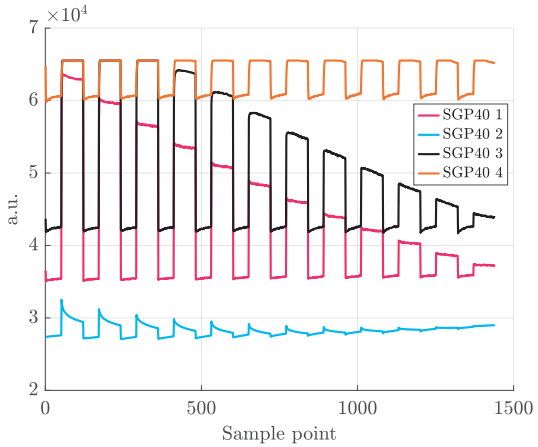

Fig. 1: Varying behaviours of subsensor coatings for the same UGM during temperature-cycled measurements.

By employing multidimensional convolution kernels, which convolve across multiple subsensors measurements, additional information can be extracted from patterns embedded in the combined data - patterns that would not be discernible from individual sensor signals. This approach is exemplified in figure 2, where a 2x4 filter kernel (in red) is used to convolve across two subsensor measurements, while a 1x4 filter kernel is applied for convolutions within each subsensor's measurements.

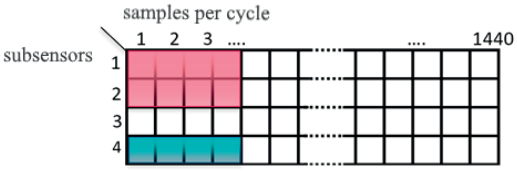

Fig. 2: Different kernel sizes for the convolution layers.

The reference network does not feature these types of multidimensional convolutions (marked in red) and only employs the convolutions along one sensor (marked in blue).

2. Pooling

Pooling is a key component in many popular CNN architectures, such as AlexNet [6], VGGNet [7] or GoogleNet [8]. It is primarily used for dimensionality reduction of extracted features, helping to address the "curse of dimensionality" and reduce the risk of overfitting, which is a common challenge in machine learning algorithms.

In a pooling layer, dimensionality reduction is achieved by reducing the size of the feature maps produced by convolutional layers, while preserving as much information as possible [9]. This is typically done by selecting either the maximum (maximum-pooling) or the average value (average-pooling) within a filter window, often referred to as a mini-batch**.** The size of the filter window and the step size (stride) by which the window moves across the feature map are key parameters that influence the extent of dimensionality reduction.

New network structure

The proposed network structure incorporates two parallel branches: one branch performs convolution on the data from each individual subsensor (single branch), while the other processes the combined data from all four subsensors (multi branch). This parallel structure enables the network to extract features both from individual sensor signals and from the collective subsensor data, thus capitalizing on the benefits of sensor fusion. The subsequent perceptron, consisting of multiple fully connected layers, is then responsible for selecting the most relevant features for quantification.

The number of convolutional units in each branch is defined as two independent hyperparameters (“# single conv.” and “# multi conv.”). Each convolutional unit consists of a convolutional layer followed

by batch normalization, ReLU activation, and optionally a maximum-pooling layer. Both the kernel sizes and strides of the convolutional layers are treated as hyperparameters. To increase network width in deeper layers, we set the number of filters in the first convolutional unit in each branch to 8, doubling the number of filters in each subsequent unit in both branches.
In the branch where convolutions are performed on data from each individual subsensor, convolution kernels of size [1, “kernel-single”] are used with corresponding strides of [1, “stride-single”]. In the other branch, where convolutions are applied to the combined data from all four subsensors, kernel sizes of [4, “kernel-multi”] are used with strides set to [4, “stride-multi”]. Similarly, for the pooling layers, the window sizes are set to [“single-pool-size”, “single-pool-size”] and [“multi-pool-size”, “multi-pool-size”] with a fixed stride of one in both branches. These parameters remain consistent across all convolutional units within each branch but are optimized separately for each branch.
For the perceptron, the number of neurons (“# neurons”) and fully connected layers (“# fully connected”) are also treated as hyperparameters.
Table 2 provides the parameter ranges for the hyperparameter search, while figure 3 illustrates the structure of the proposed network.

Tab. 2: Hyperparameter ranges used in hyperparameter search.

Hyperparameter	Minimum	Maximum
stride-single	3	40
stride-multi	3	40
kernel-single	4	100
kernel-multi	4	100
# single conv.	1	5
# multi conv.	1	5
single-pool-size	2	4
multi-pool-size	2	4
# fully connected	1	3
# neurons	100	450

Evaluation of the new network structure

The network architecture is defined by the parallelization described earlier. The width and depth of the network, which determine its overall complexity, are optimized through a hyperparameter search for the parameters mentioned previously. Additionally, we evaluated the network with and without pooling, using maximum-pooling, as it is the most commonly applied pooling method in literature. The RMSE and number of learnables were used as metrics to evaluate the networks.
To assess the performance of the proposed network, we conducted a hyperparameter search with 150 iterations. Using the best-found hyperparameters, the model was then trained for 20 epochs, and the network achieving the lowest RMSE for quantifying a target gas within a gas mixture was selected. This process was repeated five times to ensure reproducibility. The same is done for the reference network. Ultimately, the RMSE and number of learnables of both networks are compared.
Initially, the networks were tested on acetone. Acetone was selected as the target gas due to its strong sensor response and its well-documented characteristics in the collected data. Furthermore, previous studies have shown that CNNs provide good interpretability for this target gas. Comparison tests were later conducted on formaldehyde and hydrogen.

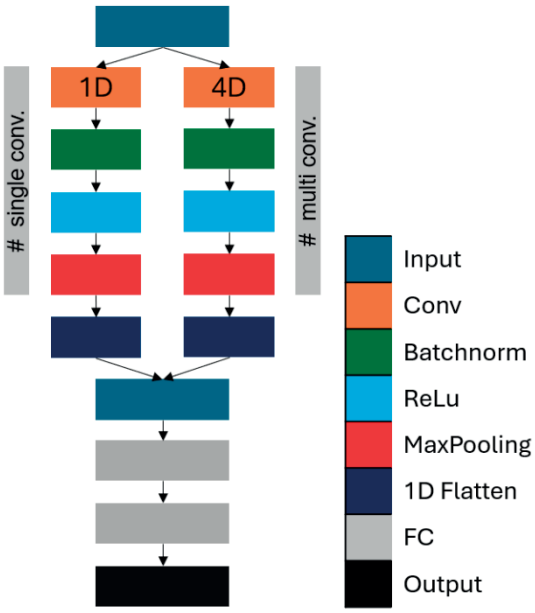

Fig. 3: Newly proposed network structure.

Results and Discussion

1. Improved optimiser
The first observation was that the above-mentioned procedure resulted in significant differences in network complexity, despite yielding similar RMSE values, as can be seen in table 3.
This discrepancy is due to the hyperparameter search, being conducted via Bayesian optimisation, which focused solely on minimizing RMSE without taking the number of learnable parameters into account.

Tab. 3: Number of learnables and RMSE of both the reference and new network for acetone using Bayesian optimisation for the hyperparameter search.

	Reference Network		New Network	
Run #	Best RMSE	Learn-ables	Best RMSE	Learn-ables
1	11,91	24,2 M	10,95	70,8 M
2	11,3	13,5 M	10,2	63,6 M
3	11,17	12,7 M	10,11	152,1 M
4	10,78	14,3 M	10,42	71,7 M
5	11,26	14,1 M	10,64	157,1 M

To achieve lower network complexity with reduced variation while maintaining comparable RMSE values, we adjusted the optimiser used in the hyperparameter search. This was done by introducing a penalty for networks with great complexities. Specifically, the optimiser's cost function was adjusted to include the number of learnables, multiplied by a weighting factor and added to the total RMSE sum. A weight of 100000^{-1} proved to be effective in producing networks with a consistently smaller number of learnables compared to the original optimiser.

Tab. 4: Number of learnables of the reference and new network for acetone using the improved optimisation for hyperparameter search.

	Number of learnables	
Run #	Reference Network	New Network
1	572.3k	531,7k
2	591.5k	177,2k
3	456.4k	257,5k
4	543.3k	161,7k
5	571.5k	266,5k

For the reference network, the original optimiser resulted in network sizes ranging from over 10 million to 24 million learnable parameters for acetone (see Table 3). However, with the new optimiser, network sizes were reduced to under 600,000 learnable parameters (see Table 4). In the new network structure, the number of learnable parameters was further decreased, from over 150 million to as few as 161.7k, demonstrating the new optimiser's effectiveness in significantly reducing network size. Table 5 compares the RMSE values of both networks trained with the old and new optimiser. It can be observed that, when using the new optimiser, both networks achieve comparable RMSE values. With the old optimiser, the new network slightly outperforms the reference network, although the difference between the two never exceeds 2 ppb.

Tab. 5: RMSE of the reference and new network for acetone using the old and improved optimiser for hyperparameter search.

	Best RMSE			
Run #	Ref. Net. old opti.	Ref. Net. new opti.	New Net. old opti.	New Net. new opti.
1	11,91	12,46	10,95	12,31
2	11,3	13,83	10,2	11,26
3	11,17	12,97	10,11	13,22
4	10,78	12,90	10,42	14,38
5	11,26	12,97	10,64	12,29

2. Impact of using pooling layers

Next, we analyzed the impact of pooling layers on the network performance. Networks with maximum-pooling consistently showed higher RMSE values and required more learnable parameters compared to those without pooling, even when the optimiser did not focus on minimizing the number of learnables during hyperparameter optimization. Tables 6 and 7 present the average RMSE and number of learnable parameters for the new network, with pooling enabled and disabled, respectively. Due to the slightly higher RMSE network sizes observed with pooling, it will not be used in subsequent steps.

Tab. 6: Number of learnables and mean RMSE for the new network with pooling, trained on acetone.

Run #	Learnables	RMSE
1	172.8 k	15.8
2	178.4 k	18.1
3	291.3 k	25.0
4	349.7 k	14.4
5	182.6 k	17.0

Tab. 7: Number of learnables and mean RMSE for the new network without pooling, trained on acetone.

Run #	Learnables	RMSE
1	531,7k	22.1
2	177,2k	12.0
3	257,5k	19.0
4	161,7k	15.7
5	266,5k	13.1

3. Extraction of global features

A review of the selected hyperparameters for the new network reveals that the runs with the lowest RMSE values utilized relatively large convolutional kernel sizes and high strides across both branches,

as shown in table 8 for the best-performing network based on the new network structure used to quantify acetone.

The kernels are relatively large compared to a single temperature cycle in the dataset, indicating that the extracted features focus more on the overall pattern rather than small segments of a cycle. These larger filter kernels and higher strides reduce the emphasis on fine-grained, localized details, instead capturing broader, more generalized patterns across the input. This ability to extract more global features appears to be advantageous for the quantification task, resulting in improved overall performance.

Tab. 8: Hyperparameters for the best-performing network based on the new network structure, trained on acetone using then new optimiser.

Hyperparameter	Value
stride-single	43
stride-multi	92
kernel-single	53
kernel-multi	6
Neurons fully connect	152
# Single Conv.	2
# Multi Conv.	2
# Fully connected	3
Initial learning rate	9.97e-04

4. Network performances for different target gases

As outlined above, comparison tests were conducted on formaldehyde and hydrogen. Figures 3 and 4 present a performance comparison of both networks, using the new optimiser across different target gases. The figures display the average RMSE and number of learnable parameters for the top five networks from each architecture (new and reference networks). Results indicate that both networks achieve comparable RMSE values across all three gases, with the smallest RMSE difference observed for acetone at 0.31 ppb and the largest for hydrogen at 5.55 ppb.

For acetone and formaldehyde, the new network is smaller than the reference network, while for hydrogen, the reference network has a more compact size. The smallest difference in network size is observed for formaldehyde at 250.98k, and the largest for hydrogen at 567.86k.

These findings suggest that, although the new network architecture does not improve RMSE over the reference network, it can yield smaller network sizes, particularly for acetone and formaldehyde but result in larger network sizes for hydrogen.

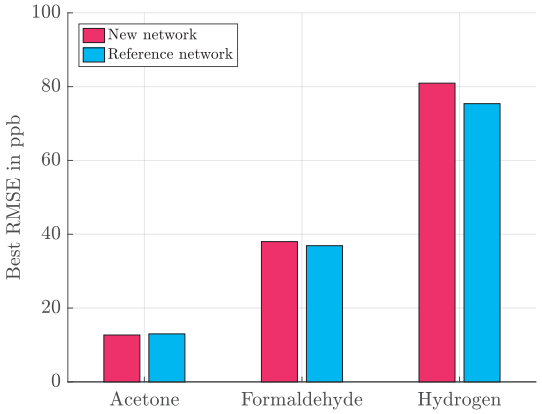

Fig. 4: Average RMSE of the top five networks for new and reference architectures across different target gases using the new optimiser.

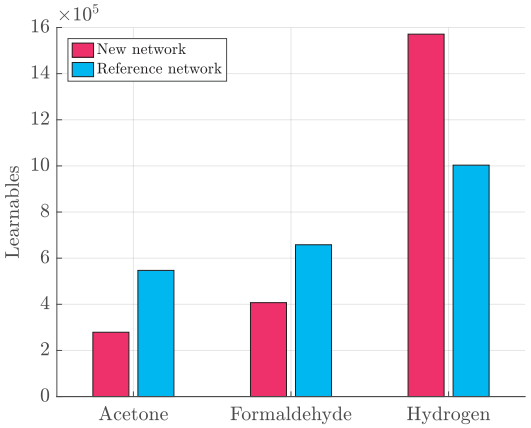

Fig. 5: Average number of learnable parameters for the top five networks in new and reference architectures across different target gases using the new optimiser

Conclusion and outlook

This work investigated and evaluated the use of pooling and sensor fusion for quantifying target gas concentrations in a UGM using a CNN. Both methods were incorporated into a new network design, along with a network size optimization strategy. The results demonstrate a substantial reduction in network size with minimal variation in RMSE across multiple gas types when using an optimiser for hyperparameter search that considers the number of learnable parameters in the network. This outcome is particularly advantageous for implementing neural network-based gas detection and quantification on resource-constrained platforms, such as microcontrollers. Additionally, these results highlight the importance of tracking and feeding back network size to the optimiser.

Since the new network varied in performance across different gases compared to the reference, it

is likely that a single network structure will not be optimal for all gases. Instead, a tailored evaluation and selection of an appropriate network structure may be necessary for each specific application. Future research could explore testing and evaluating other optimiser configurations and weight settings, as well as deeper evaluation on pooling methods and parameters. Future research could explore whether incorporating sensor fusion would be beneficial in other problem settings. This could include simulating the failure of individual sensors or assessing the network's performance under domain shifts. Another area of investigation could focus on visualizing and understanding the dimensions and strides of convolutional kernels to gain deeper insights into why larger values yield better results compared to smaller ones, especially in relation to the duration of a single temperature cycle in the measurements. Such an analysis could clarify the underlying mechanisms driving performance differences between different kernel sizes and potentially guide further improvements to the network structure.

The primary factor behind the reduction in learnable parameters was the enhanced Bayesian optimization process. In the future, alternative optimization strategies, such as Adam optimisation, could be explored to further improve the network's accuracy and efficiency.

Bibliography

[1] Y. Robin, J. Amann, T. Baur, P. Goodarzi, C. Schultealbert, T. Schneider and A. Schütze, "High-performance VOC quantification for IAQ monitoring using advanced sensor systems and deep learning", *Atmospehre,* vol. 12, no. 11, 2021.

[2] P. Vorwerk, J. Kelleter, S. Müller and U. Krause, "Classification in Early Fire Detection Using Transfer Learning Based on Multi-Sensor Nodes", in *Proceedings*, 2024.

[3] J. Amann, T. Baur, C. Schultealbert und A. Schütze, „Bewertung der Innenraumluftqualität über VOC-Messungen mit Halbleitergassensoren - Kalibrierung, Feldtest, Validierung“, *Technisches Messen : tm,* Bd. 88, Nr. 1, 2021.

[4] A. Schütze, T. Baur, M. Leidinger, W. Reimringer, R. Jung, T. Conrad and T. Sauerwald, "Highly sensitive and selective VOC sensor systems based on semiconductor gas sensors: how to?", *Environments,* vol. 4, p. 20, 2017.

[5] Y. Robin, P. Goodarzi, T. Baur, C. Schultealbert, A. Schütze and T. Schneider, "Machine Learning based calibration time reduction for gas sensors in temperature cycled operation", in *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Glasgow, 2021.

[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM,* vol. 60, no. 6, pp. 84-90, 2017.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556,* 2014.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, 2015.

[9] M. Suna, Z. Song, X. Jiang, J. Pan and P. Yanwei, "Learning Pooling for Convolutional Neural Network", *Neurocomputing,* vol. 224, pp. 96-104, 2017.

[10] D. Arendes, J. Amann, C. Tessier, O. Brieger, A. Schütze und C. Burgard, „Qualification and optimisation of a gas mixing apparatus for complex trace gas mixtures“, *Technisches Messen : tm,* Bd. 90, Nr. 12, 2023.

[11] Y. Robin, J. Amann, P. Goodarzi, T. Baur, C. Schultealbert, T. Schneider und A. Schütze, „Überwachung der Luftqualität in Innenräumen mittels komplexer Sensorsysteme und Deep Learning Ansätzen“, in *15. Dresdner Sensor-Symposium*, 2021.

[12] D. Scherer, A. Müller und S. Behnke, „Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition“, in *International conference on artificial neural networks*, Berlin, Heidelberg, Springer Berlin , 2010, pp. 92-101.

[13] Y. Robin, J. Amann, P. Goodarzi, A. Schütze und C. Bur, „Transfer Learning to Significantly Reduce the Calibration Time of MOS Gas Sensors“, in *IEEE International Symposium on Olfaction and Electronic Nose (ISOEN)*, 2022.

[14] Y. LeCun, L. Bottou, Y. Bengio und P. Haffner, "Gradient-Based Learning Applied to Document Recognition", LeNet: Proceedings of the IEEE 86.11, 1998.

Acknowledgment of work

The authors have contributed equally to this work. This paper is a conclusion of a mandatory graded seminar work for the lecture “Multisensor Signal Processing” at the *Lab for Measurement* Technology at the *Universität des Saarlandes* of the three authors supervised by Dennis Arendes.