Journal of Chromatography A 1672 (2022) 463005

Contents lists available at ScienceDirect

Journal of Chromatography A

journal homepage: www.elsevier.com/locate/chroma

Convolutional neural network for automated peak detection in reversed-phase liquid chromatography

Alexander Kensert[a,b], Emery Bosten[a,d], Gilles Collaerts[a,b], Kyriakos Efthymiadis[a,c], Peter Van Broeck[d], Gert Desmet[b], Deirdre Cabooter[a,*]

[a] Department for Pharmaceutical and Pharmacological Sciences, Pharmaceutical Analysis, University of Leuven (KU Leuven), Herestraat 49, Leuven 3000, Belgium
[b] Department of Chemical Engineering, Vrije Universiteit Brussel, Pleinlaan 2, Brussel 1050, Belgium
[c] Department of Computer Science, Artificial Inte lligence Lab, Vrije Universiteit Brussel, Pleinlaan 9, Brussel 1050, Belgium
[d] Department of Pharmaceutical Development and Manufacturing Sciences, Janssen Pharmaceutica, Turnhoutseweg 30, Beerse 2340, Belgium

ARTICLE INFO

Article history:
Received 21 December 2021
Revised 23 March 2022
Accepted 27 March 2022
Available online 31 March 2022

Keywords:
Machine learning
Convolutional neural networks
Peak finding
Method development

ABSTRACT

Although commercially available software provides options for automatic peak detection, visual inspection and manual corrections are often needed. Peak detection algorithms commonly employed require carefully written rules and thresholds to increase true positive rates and decrease false positive rates. In this study, a deep learning model, specifically, a convolutional neural network (CNN), was implemented to perform automatic peak detection in reversed-phase liquid chromatography (RPLC). The model inputs a whole chromatogram and outputs predicted locations, probabilities, and areas of the peaks. The obtained results on a simulated validation set demonstrated that the model performed well (ROC-AUC of 0.996), and comparably or better than a derivative-based approach using the Savitzky-Golay algorithm for detecting peaks on experimental chromatograms (8.6% increase in true positives). In addition, predicted peak probabilities (typically between 0.5 and 1.0 for true positives) gave an indication of how confident the CNN model was in the peaks detected. The CNN model was trained entirely on simulated chromatograms (a training set of 1,000,000 chromatograms), and thus no effort had to be put into collecting and labeling chromatograms. A potential major drawback of this approach, namely training a CNN model on simulated chromatograms, is the risk of not capturing the actual "chromatogram space" well enough that is needed to perform accurate peak detection in real chromatograms.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Although commercially available software packages nowadays provide the option of performing peak detection and peak integration in an automated way, a visual inspection and manual correction of the obtained integration is in many cases still required. Instead of focusing on problem solving, the chromatographer is hence forced to allocate substantial time and energy to the manual labeling and integration of peaks in chromatograms. It is therefore highly desirable to improve the automation of this process by implementing novel algorithms and computer programs to perform automatic peak detection and integration. However, chromatograms are noisy, both in terms of short-term noise (e.g., white noise) and long-term noise (e.g., baseline drift) (see Felinger [1] for more details on this topic), which makes automatic peak detection and integration a non-trivial task.

Peak detection algorithms aim at identifying the location of peaks in a chromatogram and are thus beneficial in chromatographic data processing. Traditionally, one of the following two approaches is employed for peak detection: derivative-based or curve-fitting approaches. In derivative-based approaches, such as the algorithm developed by Vivó-Truyols et al. [2], derivatives of the chromatographic signal are computed, thereby increasing the initial variation in the signal and thus enhancing the shifts in the initial signal. The identification of local minima in the second derivative of the signal then provides the location of the peaks. However, a major drawback of such derivative-based methods is that they are extremely sensitive to noise or other artefacts and therefore necessitate considerable pre-processing steps, such as chromatogram smoothening and baseline correction, to denoise the original signal [3]. Curve-fitting (or matched filtering) approaches, such as the methods from De Weljer et al. [4–6], attempt to describe the original signal by fitting peak models to the chro-

* Corresponding author.
E-mail address: deirdre.cabooter@kuleuven.be (D. Cabooter).

https://doi.org/10.1016/j.chroma.2022.463005
0021-9673/© 2022 Elsevier B.V. All rights reserved.

matogram. Key to these methods is the appropriate choice of the peak model describing the peak shapes. Often Gaussian curves are assumed, which are, however, rarely appropriate to describe real peaks. On the other hand, the use of more flexible models can lead to the occurrence of false positive or false negative results [7]. A second major drawback is that the number of peaks which need to be fitted (number of compounds in the sample) is often difficult to determine beforehand, resulting in difficulties in the deconvolution of overlapping peaks [3,8].

Recently, deep learning models have been introduced and studied for improved peak detection, classification and integration. For instance, Risum and Boro [9] employed a convolutional neural network (CNN) to classify PARAFAC2 (Parallel factor analysis 2) resolved intervals of chromatograms extracted from GC–MS data into four potential classes: peaks, baselines, shoulder peaks or other. Melnikov et al. [10] employed two separate CNNs to (1) classify intervals (of a given chromatogram) into three potential classes (peak, noise or uncertain) and (2) perform peak integration. To train their CNNs, 4000 segments (or "regions of interest" (ROIs)), proposed by an external algorithm similar to centWave [11], were manually annotated and processed. Gloaguen et al. [12] employed a CNN to classify peaks based on their quality – namely, high quality, acceptable quality, or poor quality (noise). Like Risum et al. and Melnikov et al., their models operated on short intervals of a given chromatogram, which had been proposed by standard preprocessing tools such as centWave.

In contrast to the deep learning models covered in the previous section – in which ROIs where proposed by the use of external tools and the resulting intervals were classified into one of several classes (and in the case of Melnikov et al., also segmented into peak areas etc.) – in this work, a CNN model was developed to perform peak detection on whole chromatograms in an end-to-end fashion, with little to no dependence on third-party tools and/or additional algorithms which themselves might be in need of hyperparameter tuning. Specifically, the CNN implementation of this study was based on the You Only Look Once (YOLO) approach [13]. In brief, YOLO is a computationally fast approach to detect objects in natural images, by inputting a whole image (e.g., a photograph of a human and a dog) and predicting so-called bounding boxes and associated class probabilities (for the human and the dog). In the case of object detection in (natural) images, the bounding boxes are two dimensional; in contrast, the bounding boxes of a chromatogram would be one-dimensional (e.g., an x-coordinate and a width). The data required to develop and train the model in this study were entirely synthetic (generated from an in-house chromatogram simulator), which meant no additional time and resources were needed to collect and extract the required data. Importantly, the purpose of the model developed in this study was not to classify intervals of a chromatogram as containing a peak or not, but to detect and locate peaks in a whole chromatogram. Predicted peak probability values and peak areas were also provided.

2. Materials and method

2.1. Model implementation

2.1.1. Label encoding

One of the main focuses of this study was to decide how to label the chromatograms. In order to train the CNN model to detect peaks desirably, it was imperative to supply the model with meaningful/informative labels to associate with the inputs (the raw chromatograms), to allow it to map the inputted chromatograms to some desired outputs (predictions). Using the YOLO implementation as a guideline, the chromatogram was divided into 256 segments (for the sake of labeling), where each segment was associated with a peak/no-peak label (further referred to as true peak probability, though always taking a value of either 1 or 0), based on whether the apex of a peak was situated within the segment. If a peak existed within a given segment, two more labels were associated with the segment: the location of the peak (relative to that segment, taking a value between 0 and 1; further referred to as true peak location) and the total area of the peak (for which portions of the area may lie outside the segment; further referred to as true peak area) (see Fig. 1). Note, as defining exact bounding boxes (x-coordinates and widths) of the peaks were not of interest, only peak locations (x-coordinates), associated peak probabilities and peak area predictions were defined. Thus, in contrast to the original YOLO approach, no bounding boxes were defined.

2.1.2. Convolutional neural network

In this work, a one-dimensional CNN model based on YOLO [13] was implemented with TensorFlow [14] (version 2.4) for improved peak detection in chromatograms (see Fig. 2 for a schematic illustration). In brief, an input chromatogram of dimension 8192×1 was compressed into a dimension of 256×3, via a number of so-called convolutional blocks. The output (of dimensions 256×3) yielded, for each segment, a 3-tuple prediction consisting of a predicted peak probability, location, and area. The convolutional block consisted of (1) a convolutional layer, convolving F numbers of filters (of dimension 9×1 or 1×1) with the input; (2) a batch normalization layer (re-centering and re-scaling the input); (3) a rectified linear activation unit (ReLU; a simple non-linear transformation); and for the three initial blocks (4) a max pooling layer, which down-sampled the input by a factor of 4 or 2. The convolutional layer of the final block applied 3 filters (of dimension 1×1) to obtain an output (predictions) with the required dimension. As the predicted location and probability of a peak were within a range of 0 and 1, they were finally passed through a sigmoidal function (not illustrated in Fig. 2).

Before the chromatograms were inputted to the CNN, they were subjected to preprocessing. Each chromatogram was normalized to have a maximum intensity (UV-absorption) at 1, resulting in rescaled peak heights and peak areas. The normalization was performed to make the absolute (unnormalized) intensity irrelevant for the CNN (as it will always be maximum 1). Importantly, the preprocessing procedure could be trivially reversed for predictions later, to obtain the appropriate peak area predictions. The peak probabilities and locations were unaffected by the normalization. Additionally, for real chromatograms, to match the defined input dimension of the CNN model, they were subject to linear interpolation.

2.2. Simulated chromatograms

A chromatogram simulator based on Kensert et al. [16] was implemented to generate a great diversity of chromatograms approximating the "chromatogram-space" encountered in real-life applications. A major difference between the simulator of this study and the simulator of Kensert et al. was the time dependent peak width; i.e., the width of the peaks in this study both depended on time and a random variable (and not only the latter). As the aim was not only to distinguish peaks in partial overlaps (discernible apices/peaks), but also complete (or close to complete) overlaps (further referred to as unresolved peaks, where only a single apex/peak can be discerned), the width of a given single discernible peak was the main deciding factor if one or more peaks were predicted. Importantly, although difficult to implement, this feature (the capability of distinguishing unresolved peaks) is arguably one advantage of the CNN model over traditional approaches (like the SG method). However, this feature also comes with potential drawbacks; for instance, training the CNN model to predict multiple peaks if a given single discernable peak is wider

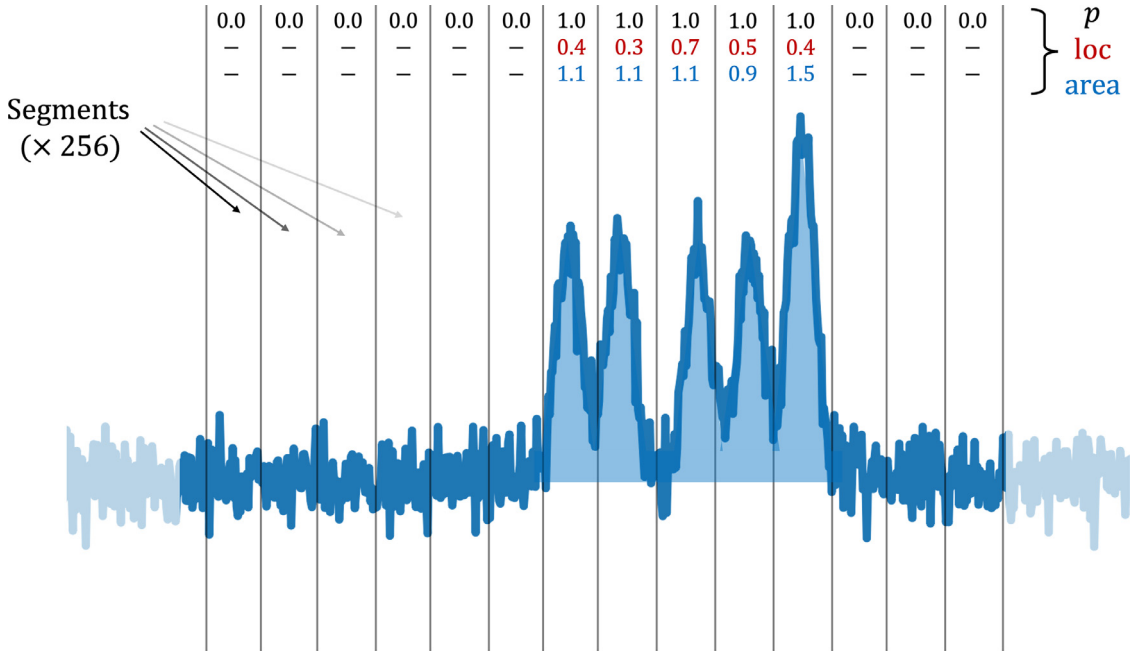

Fig. 1. Labeling of a chromatogram. First, a given chromatogram is divided into 256 segments (note that the segments in the illustration are significantly wider compared to the actual segments); second, peak probability (p), relative location of the peak (i.e., relative to the segment, loc), and area of the peak (area) are assigned to each segment. Peak location and area are only assigned if a peak exists in a given segment, i.e., if its apex is contained within the segment. Note, the chromatogram in this figure is a simple illustration for the purpose of highlighting the labeling procedure. This chromatogram does not reflect a realistic case.

than neighboring peaks, will likewise predict multiple peaks when an actual single peak is unusually wide (resulting in increased false positive rate).

Specifically, for each chromatogram, the simulator generated 10 to 100 peaks from a discrete log-uniform distribution, which were randomly distributed over the given chromatogram (namely, locations were sampled from a uniform distribution). The resolution (number of data points) of the chromatogram was set to 8192. Although irrelevant for the CNN, each data point of the chromatogram was linked to a time point to make it convenient to deal with peak standard deviations, asymmetries, and areas. Thus, time was defined to range from 0 to 1 (in 8192 steps). The peak shapes were modeled as modified Gaussian curves:

$$\text{peak} = A \cdot \exp\left(-\frac{1}{2}\left(\frac{x - loc}{s_o + s_1(x - loc)}\right)^2\right) \quad (1)$$

with a peak standard deviation (S_0), peak asymmetry (S_1) and peak amplitude (A) sampled from random variables $U \sim (0.001, 0.002)$, $U \sim (0.00, 0.30)$ and $U \sim (x, 10x)$, respectively. x could take any value as the intensity was normalized later. Subsequently, s_0 was modified by $(N \sim (0, s_0/10)$ and $f(s_0, loc)$ a linear function incrementing s_0 as a function of its location; A was modified inversely proportional to the incrementation of s_0. These modifications were applied to better represent real chromatograms – where peaks tend to increase in width and decrease in height with time – allowing the CNN model to potentially distinguish unresolved peaks.

To simulate noise, a random variable $\widetilde{SNR}$ was sampled from a random log-uniform distribution $U_{\log} \sim (3, 300)$. Subsequently, each data point of the chromatogram was subjected to Gaussian noise with an SNR approximately at $\widetilde{SNR}$. Thus, the chromatograms simulated, both for training and validation, had an SNR between approximately 3 (corresponding to the detection limit) and 300.

Due to the spatial invariance of the CNN, resulting from the shared weights and them being locally convolved with the input (a dot product between the filters and the input), the CNN will tend to exploit local connectivity rather than global connectivity. In other words, a given segment of the chromatogram will be largely unaffected by other segments far away spatially. This implies that simulated chromatograms do not need to be realistic from a global perspective, but more so from a local perspective. Although the simulated chromatograms arguably were both realistic from a global perspective as well as a local perspective, it was more important that the chromatograms were realistic from a local perspective.

2.3. Experimental chromatograms

2.3.1. Chemicals

The compounds 2-hydroxyacetophenone, 2-naphthol, 2-nitrophenol, anisole, benzyl alcohol, bromobenzene, butylparaben, caffeine, ethylbenzene, ethylparaben, eugenol, iodobenzene, isopropyl benzoate, m-cresol, m-methylacetophenone, o-methylacetophenone, p-methylacetophenone, propiophenone, propylparaben and valerophenone were obtained from Sigma-Aldrich (Diegem, Belgium); 2,3-dihydroxynaphthalene, 2,6-dimethylaniline, 2-hydroxybenzyl alcohol, 3-nitrophenol, 4-hydroxypropiophenone, benzhydrol, indole, methylparaben, N,N-dimethylaniline, naphthalene, N-methylaniline, phenol, phthalide, p-xylene, resorcinol, o-xylene, theophylline and acetonitrile (ACN, HPLC) from Acros (Geel, Belgium); 1,2-phenylenediamine, 1-naphthol, 2,7-dihydroxynaphthalene, 3,5-dimethylaniline, 4-methoxyacetanilide, benzophenone, furfuryl alcohol and o-cresol were from Janssen (Geel, Belgium); methyl nicotinate, N-methylphenylacetamide, phenobarbital, phenyl salicylate and salicyl amide from Federa (Brussels, Belgium); 2-methylresorcin, 7-hydroxycoumarine, carvacrol, ethyl benzoate and m-aminophenol were from Fluka (Munich, Germany); 2-aminophenol, acetophenone, nitrobenzene and thymol were from Riedel-de Haën (Seelze, Germany); 2,5-dimethylaniline, 4-methoxyphenol and methyl benzoate were from Aldrich (St. Gallen, Switzerland); 1-naphthylamine, m-xylene and o-dianisidine were from Merck (Overijse, Belgium); paracetamol was from Alpha Pharma (Roeselare, Belgium); acetanilide was from J&K (Lommel, Belgium) and

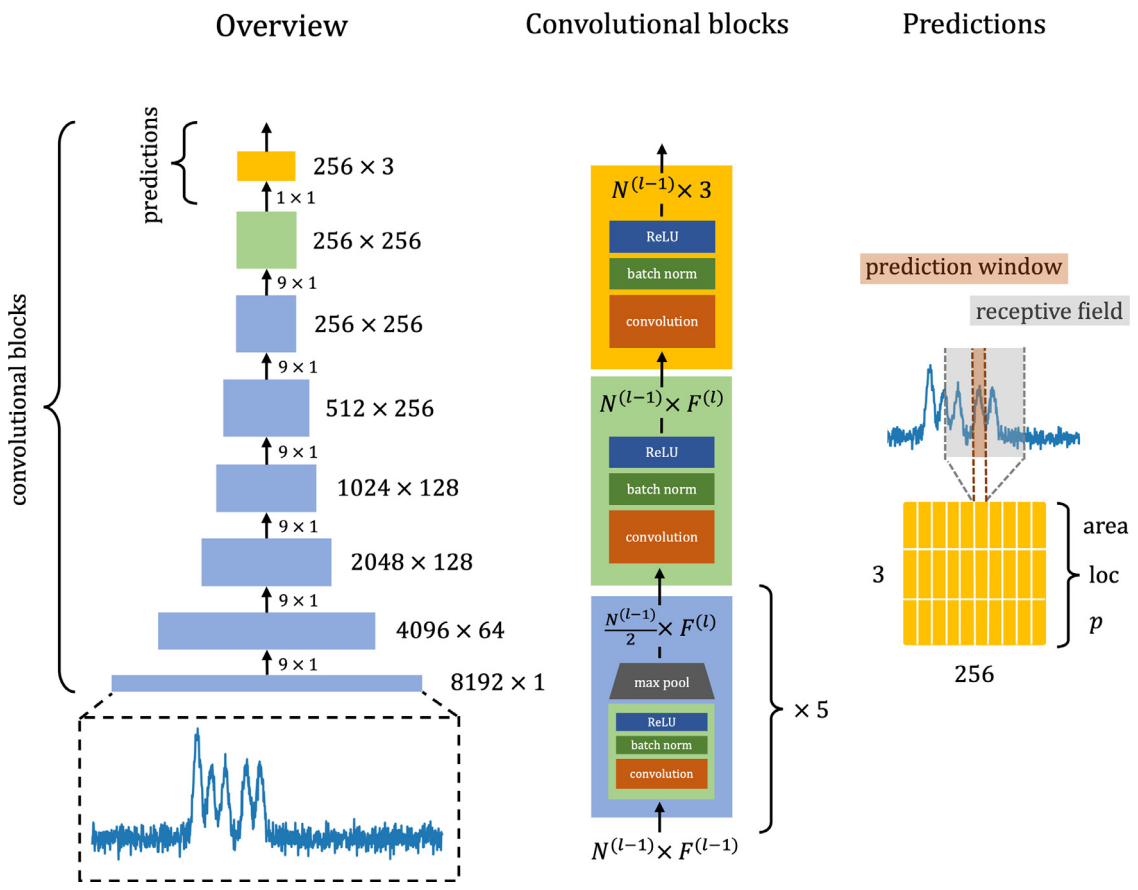

Fig. 2. The architecture of the CNN. The CNN inputs a raw chromatogram of 8192 (x 1) data points, and forward propagates it through the convolutional blocks to output a 256 × 3 array with predictions. The four building blocks of a convolutional block are a convolutional layer (convolution), a rectified linear unit activation (ReLU), a batch normalization layer (batch norm), and (for the blue block) a max pooling layer (max pool). The last block (in yellow) yields the predictions; namely, a peak probability (p), relative peak location (loc), and peak area (area), corresponding to each segment. The prediction window (in shaded brown) illustrates the association between the predictions and inputted chromatogram. The receptive field (in shaded gray) illustrates how each prediction only has access to certain information of the input – i.e., the region of information that produces the corresponding prediction. Note that both the prediction window and the receptive field are simplistic illustrations and do not reflect the actual prediction window/receptive field. $N^{(l)}$and $F^{(l)}$ denote the height and width of the data array for layer l, respectively (e.g., $N^{(0)}$ and $F^{(0)}$ are 8192 and 1, respectively, and $N^{(5)}$ and $F^{(5)}$ are 256 and 256, respectively). The dimension indicated next to the blocks denote the dimension of the data. The dimensions indicated next to the arrows between the convolutional blocks denote the filter size.

chlorobenzene from Panreac (Darmstadt, Germany). Water was produced in the lab using a Milli-Q gradient purification system (Millipore, Bedford, MA, USA).

2.3.2. Apparatus

Chromatograms were recorded on an Infinity 1290 series UHPLC from Agilent Technologies (Waldbronn, Germany), consisting of following modules: a quaternary pump (G4204A), autosampler (G4226A), thermostatted column compartment (G1316C) operated at 30.0 °C and diode array detector (G4212A) operated at a wavelength of 210 nm and 240 nm. Data acquisition was done using OpenLab CDS software (Agilent Technologies).

2.3.3. Generation of experimental chromatograms

Experimental chromatograms were obtained for five different mixtures. To unambiguously know the exact location of all analytes in the mixtures, each analyte was also analyzed individually under the same conditions as for the mixture.

Mixture 1 consisted of 16 analytes (in order of elution): methylparaben, ethylparaben, m-cresol, 2-naphthol, eugenol, p-methylacetophenone, m-methylacetophenone, propiophenone, benzophenone, chlorobenzene, bromobenzene, naphthalene, iodobenzene, o-xylene, ethylbenzene and m-xylene. All analytes had a concentration of 5 µg/mL and were dissolved in 50/50 (v:v) ACN/water. Mixture 1 was analyzed on a Stable Bond C18 column (4.6 × 250 mm, 5 µm) from Agilent Technologies at a flow rate of 1.50 mL/min and using an injection volume of 0.5 µL. The mobile phase consisted of 75/25 (v:v) ACN/water (isocratic elution). The chromatograms were recorded using an acquisition rate of 160 Hz for the mixture and 10 Hz for the individual compounds [16].

Mixture 2 consisted of 17 analytes (in order of elution): benzhydrol, butylparaben, N,N-dimethylaniline, ethyl benzoate, carvacrol, thymol, chlorobenzene, benzophenone, bromobenzene, isopropyl benzoate, valerophenone, iodobenzene, o-xylene, ethylbenzene, m-xylene, p-xylene and phenyl salicylate. All analytes had a concentration of 20 µg/mL and were dissolved in 40/60 (v:v) ACN/water. Mixture 2 was analyzed on a Zorbax Extend-C18 column (2.1 × 50 mm, 1.8 µm) from Agilent Technologies at a flow rate of 0.50 mL/min and using an injection volume of 1.0 µL. The mobile phase consisted of 40/60 (v:v) ACN/water (isocratic elution). The chromatograms were recorded using an acquisition rate of 160 Hz for the mixture and 10 Hz for the individual compounds.

Mixture 3 consisted of 35 analytes (in order of elution): 1,2-phenylenediamine, 2-methylresorcin, 2,5-dimethylaniline, salicyl amide, 7-hydroxycoumarine, 2-hydroxyacetophenone, acetanilide, 4-methoxyphenol, 2,7-dihydroxynaphthalene, phthalide, N-methylphenylacetamide, phenol, phenobarbital, 4-hydroxypropiophenone, 3-nitrophenol, m-cresol, 2,3-dihydroxynaphthalene, o-cresol, acetophenone, ethylparaben, o-dianisidine, N-methylaniline, 2-nitrophenol, nitrobenzene,

indole, 1-naphthylamine, 2-naphthol, p-methylacetophenone, o-methylacetophenone, methyl benzoate, propylparaben, anisole, propiophenone, 1-naphthol and eugenol. All analytes had a concentration of 20 µg/mL and were dissolved in 30/70 (v:v) ACN/water. Mixture 3 was analyzed on a Zorbax Extend-C18 column (2.1 × 50 mm, 1.8 µm) from Agilent Technologies at a flow rate of 0.50 mL/min and using an injection volume of 1.0 µL. The mobile phase consisted of 30/70 (v:v) ACN/water (isocratic elution). Chromatograms for the mixture and the individual compounds were recorded at an acquisition rate of 40 Hz at two different wavelengths: 210 nm and 240 nm.

Mixture 4 consisted of 34 analytes (in order of elution): m-aminophenol, 1,2-phenylenediamine, 2-methylresorcin, 2,5-dimethylaniline, salicyl amide, methyl nicotinate, 2-hydroxyacetophenone, acetanilide, benzyl alcohol, 4-methoxyphenol, 2,7-dihydroxynaphthalene, phthalide, N-methylphenylacetamide, phenobarbital, 4-hydroxypropiophenone, 3-nitrophenol, 2,3-dihydroxynaphthalene, o-cresol, ethylparaben, o-dianisidine, N-methylaniline, 2,5-dimethylaniline, 3,5-dimethylaniline, 2,6-dimethylaniline, nitrobenzene, indole, p-methylacetophenone, m-methylacetophenone, o-methylacetophenone, propylparaben, anisole, propiophenone, 1-naphthol and eugenol. All analytes had a concentration of 20 µg/mL and were dissolved in 30/70 (v:v) ACN/water. Mixture 4 was analyzed on a Zorbax Extend-C18 column (2.1 × 50 mm, 1.8 µm) from Agilent Technologies at a flow rate of 0.50 mL/min and using an injection volume of 1.0 µL. The mobile phase consisted of 30/70 (v:v) ACN/water (isocratic elution). Chromatograms for the mixture and the individual compounds were recorded at an acquisition rate of 40 Hz.

Mixture 5 consisted of 43 analytes (in order of elution): theophylline, paracetamol, caffeine, resorcinol, furfuryl alcohol, 2-hydroxybenzyl alcohol, 2-methylresorcin, m-aminophenol, salicylamide, 4-methoxyacetanilide, 2-hydroxyacetophenone, 7-hydroxycoumarine, acetanilide, benzyl alcohol, methylnicotinate, 4-methoxyphenol, 2-aminophenol, 2,7-dihydroxynaphthalene, 1,2-phenylenediamine, phenobarbital, phenol, phthalide, 4-hydroxypropiophenone, N-methylphenylacetamide, 3-nitrophenol, m-cresol, o-cresol, ethylparaben, acetophenone, 2-nitrophenol, 2,6-dimethylaniline, 2-naphthol, propylparaben, indole, 2,5-dimethylaniline, 1-naphthylamine, p-methylacetophenone, 1-naphthol, eugenol, m-methylacetophenone, o-methylacetophenone, 3,5-dimethylaniline and benzhydrol. All analytes had a concentration of 5 µg/mL and were dissolved in 30/70 (v:v) ACN/water. Mixture 5 was analyzed on a Stable Bond C18 column (4.6 × 250 mm, 5 µm) from Agilent Technologies at a flow rate of 1.00 mL/min and using an injection volume of 1.0 µL in gradient mode. The starting composition of the mobile phase was 30/70 (v:v) ACN/water and increased to a final composition of 90/10 (v:v) ACN/water in 24 min. The chromatograms were recorded at an acquisition rate of 10 Hz. An injectionless run was performed under the same conditions to obtain the baseline.

2.4. Training and evaluation of the CNN on simulated chromatograms

To train the CNN model, 1,000,000 simulated chromatograms were generated and used as input (in batches of 32) for the CNN model. The input was divided into batches for improved training, via stochastic gradient descent (specifically, Adam) [17]. The CNN model was trained to minimize three losses simultaneously: (1) the negative log-likelihood loss between true peak probabilities and predicted peak probabilities, (2) the negative log-likelihood loss between the true peak locations and predicted peak locations, and (3) the mean relative error (MRE) loss between the true peak areas and predicted peak areas. The negative log-likelihood loss (NLL) was defined as follows, for a given single example:

$$\mathrm{Loss}_{\mathrm{NLL}}(y, \hat{y}) = -y \cdot \ln(\hat{y}) + (1 - y) \cdot \ln(1 - \hat{y}) \quad (2)$$

and the (M)RE as follows:

$$\mathrm{Loss}_{\mathrm{(M)RE}}(y, \hat{y}) = \frac{|y - \hat{y}|}{y} \quad (3)$$

where y denotes the ground truth label and $\hat{y}$ the prediction. Note, if the true peak probability was 0 (i.e., no peak), loss (2) and (3) were ignored. The backpropagation (i.e., the actual training of the model) was performed via the Adam optimizer [18] with a starting learning rate at 0.001 – which decayed over the course of the training until reaching 0.00001.

To evaluate the performance of the trained CNN on simulated chromatograms, a validation set containing 10,000 simulated chromatograms was used to compute the area under the receiver operating characteristic curve (known as ROC-AUC), MRE and mean absolute error (MAE) for true vs. predicted peak probabilities, true vs. predicted peak areas and true vs. predicted locations, respectively. ROC describes the diagnostic ability of binary classifiers (e.g., peak/no-peak), where it evaluates the true positive rate against the false positive rate at different thresholds (from 0 to 1, in small steps). The thresholds (e.g., $t = 0.5$) denote if a given probability p (of a peak) should be predicted as a peak or no peak ($(p \geq t \rightarrow 1.0) \wedge (p < t \rightarrow 0.0)$). The AUC gives the area under the ROC curve (resulting from the incremental thresholding), taking a value between 0 and 1; where 1.0 indicates a perfect binary classifier and 0.5 a classifier no better than randomly guessing. In regard to finding the best threshold, different thresholds (further referred to as prediction thresholds) were evaluated on the validation set. Specifically, prediction thresholds between 0.05 and 0.95 were evaluated in steps of 0.01, via accuracy metrics.

2.4.1. Finetuning of the hyperparameters

In order to obtain a well-performing CNN model, a number of hyperparameters were finetuned on the validation set:

1. Down-sampling of the data: max pooling, average pooling, and a convolutional layer (with a stride of >1) were tested, for which the first (max pooling) was selected for the final model.
2. Size (magnitude) of the down-sampling: pooling sizes (or strides) of 2 and 4 were tested; for which 2 was selected. Selecting a greater pooling size increased the magnitude of the down-sampling, which resulted in the need of fewer convolutional blocks to reach the required dimension (e.g., 256 × 3). This both reduced computational complexity and run-time. However, having a greater pooling size also worsened the performance on the validation set.
3. Number of convolutional layers in the convolutional blocks: 1, 2 and 3 layers were tested, for which the 1-layer approach was selected for the final model. Using fewer convolutional layers reduced computational complexity and run-time. Although this may come at the cost of reduced performance, no significant reduction in performance was observed.
4. Number of segments: 128, 256, 512 and 1024 were tested, for which 256 was selected for the final model. The aim was to find a balance between few peak collisions and computational complexity. Having 128 segments resulted in too many peak collisions, while 512 and 1024 segments resulted in difficulties training the model. Note that peak collisions refer to a situation in which two (or more) peaks are located in the same segment. As there can only be one label per segment, only one of the peaks can be labelled. For the simulated chromatograms, all but one peak (the one for which the label is maintained) will be removed.
5. Input dimension: 4096, 8192 and 16,384 were tested, for which 8192 was selected for the final model. Having 8192 data points

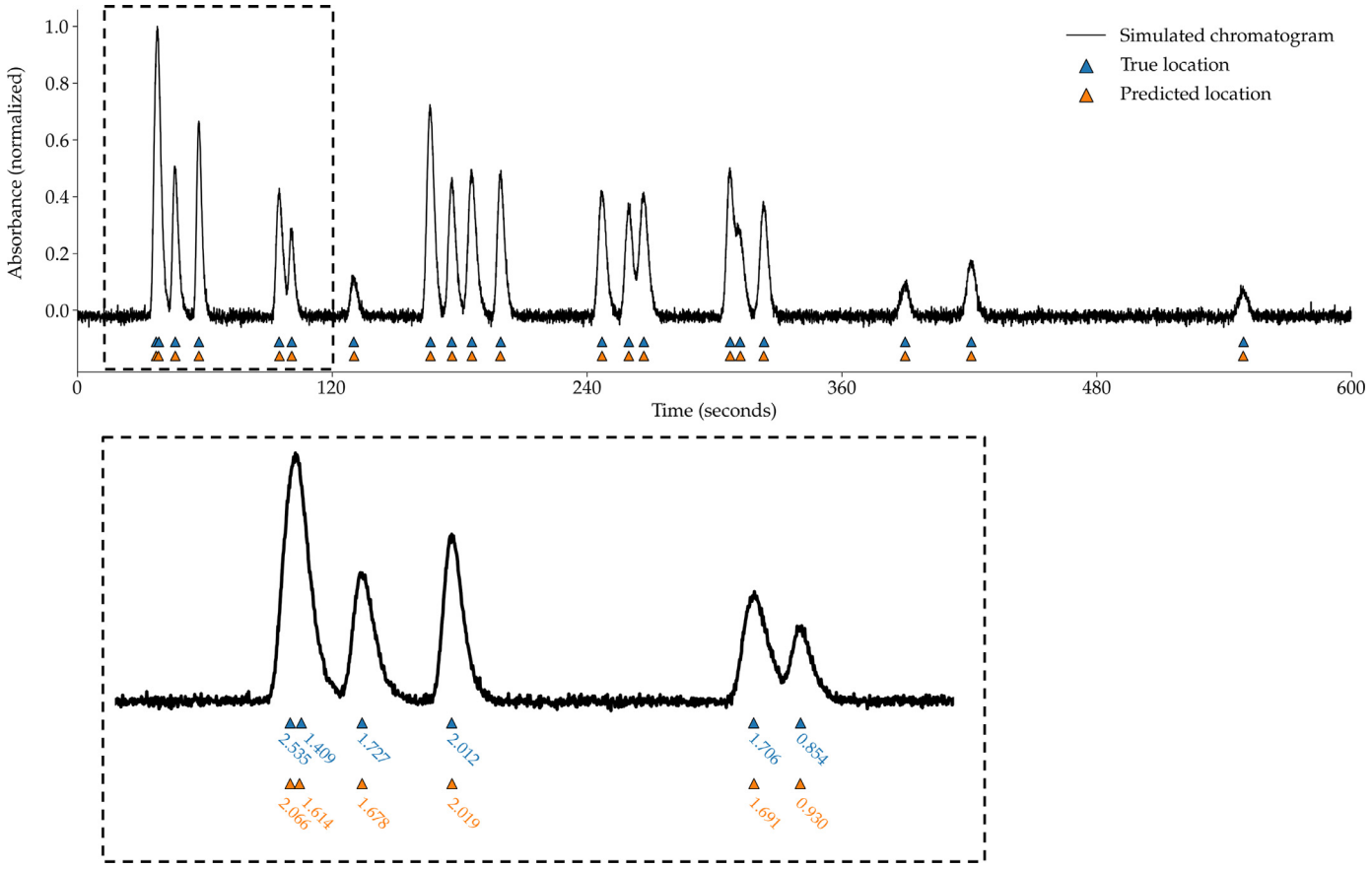

Fig. 3. Predicted peaks on a simulated chromatogram. The top figure illustrates the full chromatogram and the bottom figure a zoomed-in area from the top chromatogram (indicated by the dashed rectangle). Orange triangles denote predicted locations of peaks and orange values denote the predicted area of the corresponding peaks. The blue triangles and values below the peaks denote the ground truth location and area of the peaks, respectively. Note, the original time range of 0 and 1 (unitless) was transformed into a time range of 0 and 600 ("seconds").

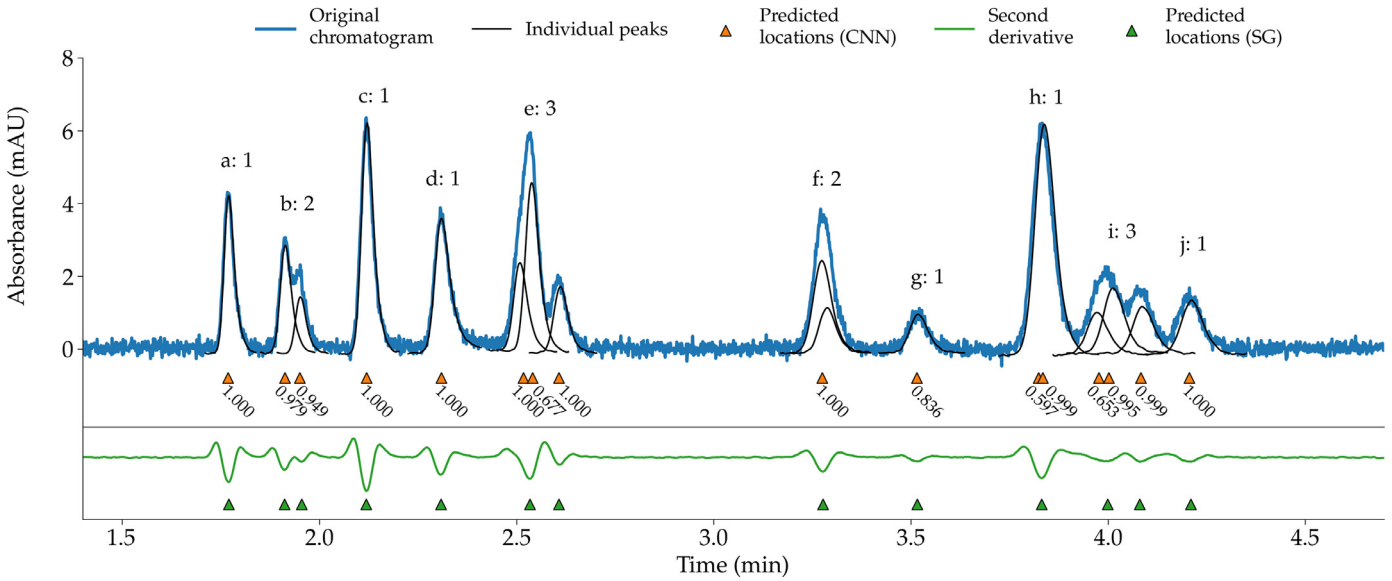

Fig. 4. Peak detection performed on a real chromatogram (Mixture 1; mobile phase (isocratic)–75/25 (v:v) ACN/water; 210 nm) by the CNN model and the SG method. The original chromatogram (96,000 data points; recorded at an acquisition rate of 160 Hz) and the underlying peaks (recorded individually at an acquisition rate of 10 Hz) are visualized in blue and black, respectively. The orange and green triangles indicate the predicted peak locations by the CNN model and SG method, respectively. The black text below the orange triangles indicates the peak probability generated by the CNN model. The prediction threshold is set to 0.5 (hence peak probabilities below 0.5 are not visualized). The green line indicates the second-order derivative computed by the SG method (window length = 601; polynomial order = 2), from which the peak locations were derived. The numbers above the peak clusters indicate the number of underlying peaks.

Table 1
Summary of the peak detection results obtained on the experimental chromatograms. *Total* indicates the actual total number of peaks present in the chromatograms.

		True positives		False positives		False negatives	
	Total	CNN	SG	CNN	SG	CNN	SG
Mixture 1 (Fig. 4)	16	15	13	1	0	1	3
Mixture 2 (Fig. 5)	17	16	13	1	2	1	4
Mixture 3 (210 nm) (Fig. 6)	35	30	26	5	1	5	9
Mixture 4 (Fig. S1)	34	32	27	1	1	2	7
Mixture 5 (Fig. S2)	43	38	36	0	0	5	7
Mixture 3 (240 nm) (Fig. S3)	35	26	24	2	1	9	11
Sum	180	151	139	10	5	23	41

was enough to not lose too much information, but at the same time reduced the computational complexity, which caused difficulties in training the model.

The optimizer, learning rate and loss functions were not thoroughly experimented with. This was not needed, because the Adam optimizer is usually a good (often default) alternative for training CNN models. A starting learning rate of 0.001 is often a default learning rate for the Adam optimizer. The three loss functions were natural choices; as peak probability and location both took values between 0 and 1, the negative log-likelihood loss functions were suitable; and as the area took a value between approximately 0 and 2, MRE was suitable. Notably, for the peak area, the mean squared error and mean absolute error loss functions were also evaluated, without any obvious improvements.

2.5. Evaluation on experimental chromatograms

To evaluate the performance of the CNN, it was compared to a peak detection methodology which used the well-established Savitzky-Golay (SG) filter to obtain second order derivatives in combination with local minima searches on the computed derivatives (from here on referred to as *SG method*). As the development of a sufficiently robust alternative approach to perform accurate automatic peak detection for tens of thousands of chromatograms is highly non-trivial (as the parameters need to be optimized for every chromatogram separately to be fairly compared to the CNN model), it was decided to make the comparison between the CNN and the SG method on a limited number of real chromatograms only. The real chromatograms therefore served as the test set of this study, even though they only constituted a small set (see the "Experimental chromatograms" section for more details on these chromatograms).

The SG filter, as well as the local minima search, were implemented with SciPy [15] (version 1.4.1; scipy.signal.savgol_filter and scipy.signal.find_peaks, respectively). In brief, the SG filter moved a polynomial filter (of a given window length) along the chromatogram in small steps. From the polynomial filter – which at each step, fits a polynomial function to the segment of the chromatogram (defined by the window length) – second order derivatives could be calculated. These derivatives were then directly used for the local minima search, indicating peak apices.

The SG filter had two hyperparameters: window length and polynomial order – both of which were adjusted (manually) until good values were obtained (i.e. values that resulted in a derivative that was smooth enough to avoid false positives while capturing most true positives). As for the local minima search, there were mainly three hyperparameters: height, width and distance (for detailed information on these hyperparameters, see SciPy's documentation). In short, these hyperparameters helped ignoring local minima that were insignificant (i.e. did not correspond to peak apices). Similar to the SG filter, these hyperparameters were manually adjusted until good values were obtained, as it was highly non-trivial to automate this.

3. Results and discussion

3.1. Quantitative evaluation: performance of CNN on simulated chromatograms

Fig. 3 illustrates the peak localization and peak area predictions on a noisy simulated chromatogram from the validation set. As can be observed from the blue (true value) and orange triangles (predicted value), most peaks were correctly detected, including unresolved peaks (indicated by more than one triangle under a single discernable peak) which should be hard to detect even for a trained chromatographer. The accuracy of the area predictions varied depending on whether the peaks were overlapping or not. For peaks that were not overlapping (isolated peaks), the relative percentage error was low (<5%), while for overlapping peaks a somewhat higher error (5%–20%) was obtained.

By analyzing all 10,000 (simulated) validation chromatograms, it was observed that the peak probability predictions of the CNN resulted in a ROC-AUC at 0.996, indicating a high true positive rate and a very low false positive rate. Furthermore, the MRE between true peak areas and predicted peak areas was calculated at 0.1445; and the MAE between true peak locations and predicted peak locations was calculated at 0.062 (note that the errors are relative to the segment and are unitless in the range of 0 and 1). The errors (for the peak area and location predictions) were only calculated from peaks predicted in the correct segments (the segments containing true peaks; with a prediction threshold set to 0.5). Based on Fig. 3, and also indicated by the peak area prediction error (0.1445), it was observed that overlapping peaks had high peak area prediction errors – likely due to the great uncertainty in how much each peak contributed to the cluster of peaks. Finally, the optimal prediction threshold (based on accuracy metrics) for predicting a peak or no peak was 0.50 (an accuracy score of 0.9790). In comparison, a prediction threshold of 0.45 and 0.55 gave accuracy scores of 0.9788 and 0.9789, respectively.

3.2. Qualitative evaluation: performance of CNN and SG on experimental chromatograms

In order to evaluate the generalization of the CNN to real chromatograms, six experimental chromatograms were considered (see "Experimental chromatograms" section). Additionally, to get a sense of how well the CNN performed, it was compared to the SG method. For the evaluation of the CNN model, peak probabilities above a threshold of 0.5 were considered (see Table 1 for a summary of the results). Fig. 4 illustrates the peak detection capabilities of the CNN model and the SG method on the chromatogram obtained for Mixture 1, with a low SNR (around 10). The CNN model managed to distinguish 2 out of 3 unresolved peaks (each

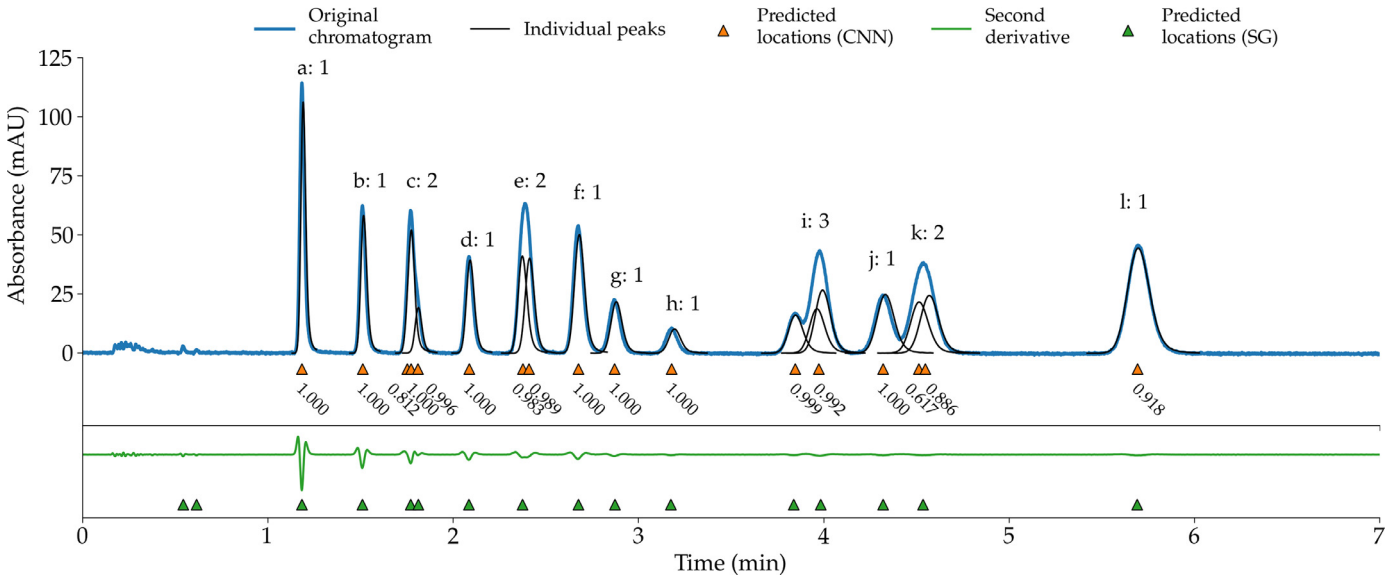

Fig. 5. Peak detection performed on a real chromatogram (Mixture 2; mobile phase (isocratic)–40/60 (v:v) ACN/water; 210 nm) by the CNN model and the SG method. The original chromatogram (96,003 data points; recorded at an acquisition rate of 160 Hz) and the underlying peaks (recorded individually at an acquisition rate of 10 Hz) are visualized in blue and black, respectively. The orange and green triangles indicate the predicted peak locations by the CNN model and SG method, respectively. The black text below the orange triangles indicates the peak probability generated by the CNN model. The prediction threshold is set to 0.5 (hence peak probabilities below 0.5 are not visualized). The green line indicates the second-order derivative computed by the SG method (window length = 241; polynomial order = 2), from which the peak locations were derived. The numbers above the peak clusters indicate the number of underlying peaks.

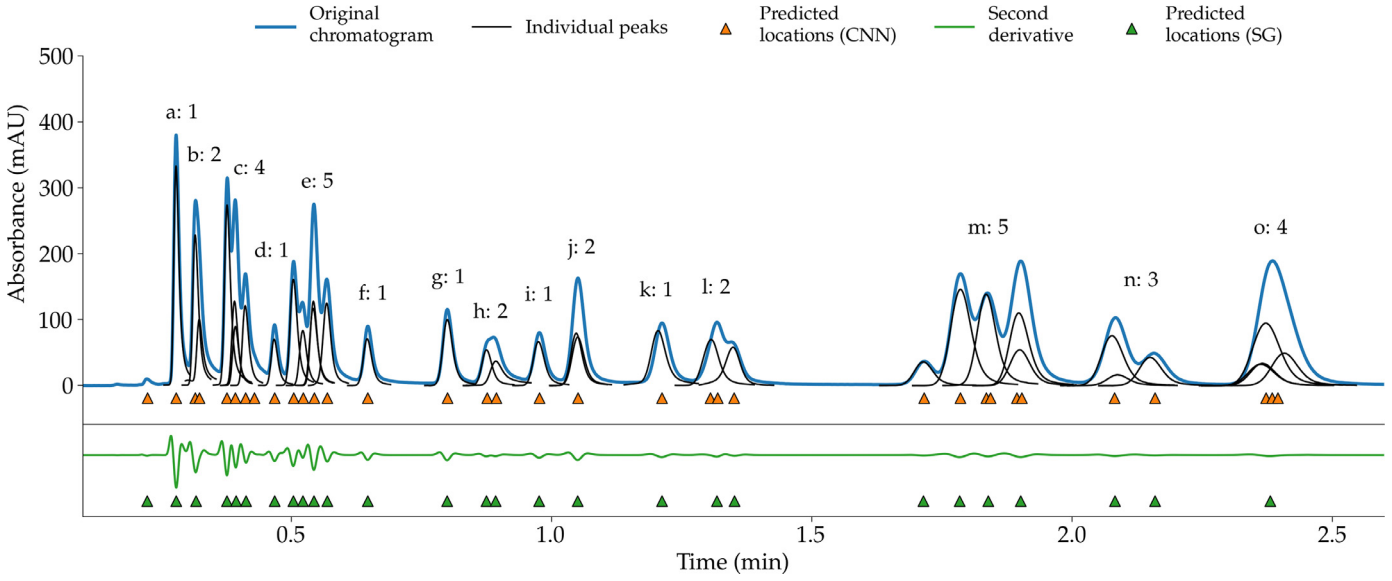

Fig. 6. Peak detection performed on a real chromatogram (Mixture 3; mobile phase (isocratic)–30/70 (v:v) ACN/water; 210 nm) by the CNN model and the SG method. The original chromatogram (7201 data points; recorded at an acquisition rate of 40 Hz) and the underlying peaks (recorded individually at the same acquisition rate) are visualized in blue and black, respectively. The orange and green triangles indicate the predicted peak locations by the CNN model and SG method, respectively. The prediction threshold is set to 0.5 (hence peak probabilities below 0.5 are not visualized). The green line indicates the second-order derivative computed by the SG method (window length = 45; polynomial order = 2), from which the peak locations were derived. The numbers above the peak clusters indicate the number of underlying peaks. For better visualization, probabilities have been omitted from the plot.

hidden within a different discernable peak; see clusters e, f and i), while the SG method could not distinguish any of them. Due to the low SNR, the SG method required aggressive smoothing which caused problems in distinguishing these unresolved peaks based on the second derivative. The peak probabilities (in black text) indicate how certain the CNN was that a peak existed there. Isolated (completely separated) peaks tended to take high probability values (>0.99), while overlapping peaks, not surprisingly, tended to take lower probability values (<0.90). Interestingly, small (but isolated) peaks tended to take lower probability values (see for example peak g). As mentioned previously, a potential drawback of training the CNN model to predict unresolved peaks is that it might predict false positives too; this is for example observed for peak h.

Fig. 5 compares the peak detection capabilities of the CNN model and the SG method on the chromatogram obtained for Mixture 2, with a significantly higher SNR (>20). This chromatogram has four pairs of peaks which overlap in such a way that it is extremely hard to distinguish them via second-order derivatives (see clusters c, e, i and k in the figure). Interestingly, while the SG method struggled in distinguishing these peaks (as expected), the CNN model could distinguish three of them (c, e and k; al-

though in c a false positive was also observed). It seems that the differences in width between the unresolved peaks and the neighboring peaks made it possible for the CNN model to distinguish them (note that peaks that are in the vicinity of each other have similar peak widths).

Similar to the comparisons made in Figs. 4 and 5, further comparisons were made on chromatograms obtained for Mixture 3 and 4, which both comprised significantly more peaks (see Figs. 6 and S1, respectively). Although both methods fell short in distinguishing many unresolved peaks, the CNN model overall performed better than the SG method regarding true positives, but somewhat poorer regarding false positives (see Table 1).

Finally, the CNN model and the SG method were also tested on a chromatogram obtained in a different mode (gradient instead of isocratic mode) and a chromatogram obtained at a different wavelength (240 nm instead of 210 nm) (see Figs. S2 and S3, respectively), to evaluate the capabilities of the CNN model in different modes and at different wavelengths. Note that, before performing the peak detection, the baseline (recorded in a separate experiment) was subtracted from the gradient chromatogram to remove the baseline drift. Even though the CNN model was developed for the purpose of performing peak detection on chromatograms obtained in isocratic mode, the CNN model performed relatively well on these chromatograms, compared to the SG method, by capturing more true positives. Note that several of the peaks of Mixture 3 (Fig. S3) were extremely small at a wavelength of 240 nm, making it close to impossible to detect them.

4. Conclusions

In this study, a deep learning (CNN) model based on YOLO was implemented to accurately detect peaks in noisy and complex RPLC chromatograms. The CNN model was able to accurately predict the precise location and probability of the peaks, and also predicted the area of the peaks relatively well (within 0.15 MRE). The model showed great potential, with a ROC-AUC score at 0.996 for peak detection, and comparable or better peak detection capabilities to the SG method (for the experimental chromatograms of this study). This was attributed to the fact that the CNN could distinguish overlapping peaks based on differences in width between the unresolved peaks and the neighboring peaks. Notably, although this study focused on RPLC chromatograms, with a relatively clean baseline, it is believed that a similar model (merely trained on different training data) can be developed to perform peak detection on more baseline-distorted chromatograms and in other modes, such as hydrophilic interaction liquid chromatography (HILIC) as well.

Furthermore, the CNN model still has plenty of room for further improvements. For instance, more effort could be put into the fine-tuning of the model's hyper-parameters: e.g., modify the input dimension, number of convolutional blocks/layers, number of segments, and loss function. The model could also be improved by splitting up the chromatogram into smaller pieces and perform peak detection on smaller regions of the chromatogram – to allow for more "fine-grained" segments and less computational complexity. Further improvements could also be made by performing test-time augmentation; namely, average predictions of augmented versions of a given original chromatogram. Augmentation here could for example be random horizontal flipping, random Gaussian noise or random shifting. Finally, improvements could potentially also be made by filtering out false positives; namely, applying non-max suppression on the predictions of a given segment of the chromatogram for which it is assumed to have many false positives (e.g., a region of the chromatogram with a single discernable peak but more than five predicted peaks). However, the latter would involve an additional separate step with rules to decide when and how to perform the non-max suppression.

Furthermore, as the model was successfully trained on simulated chromatograms, to allow it to generalize to unseen real chromatograms, no effort was required into gathering and labeling real data (chromatograms) – which saved significant time and resources. The limitation of this approach, however, is that it is restricted to performing peak detection on real chromatograms that lie within the "chromatogram space" (or distribution) of the simulated chromatograms (hence challenging). For instance, the performance of the CNN will decline drastically (to the point where it is no longer applicable to the problem at hand) as soon as peaks are wider than the peaks it was trained on; or if the peak tailing is more pronounced than what was observed in the training data. Notably, this goes for deep learning approaches in general. Concretely, in order to successfully train the model for future applications (i.e. perform peak detection on new experimental chromatograms), the simulator has to be adjusted in such a way that it covers the chromatogram space that is expected in future experiments.

Finally, as the number of segments was set to 256, with only one peak assigned to each segment, the model was, in the worst case, limited to predicting peaks that are at minimum 32 data points apart from each other (corresponding to peak apices that are approximately 2 s apart from each other on a 10 min long chromatogram), and in the best case, at minimum 1 data point apart. In other words, the model can neither predict nor be trained on peaks that would fall inside the same segment.

Availability

Implementations and code used in this study can be found at https://github.com/akensert/deep-learning-peak-detection.

Declaration of Competing Interest

The authors declare that there is no conflict of interest.

CRediT authorship contribution statement

Alexander Kensert: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft. **Emery Bosten:** Data curation, Formal analysis, Investigation, Validation, Writing – review & editing. **Gilles Collaerts:** Data curation, Formal analysis, Investigation, Validation, Writing – review & editing. **Kyriakos Efthymiadis:** Formal analysis, Investigation, Methodology, Validation, Writing – review & editing. **Peter Van Broeck:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing. **Gert Desmet:** Conceptualization, Funding acquisition, Investigation, Methodology, Supervision, Writing – review & editing. **Deirdre Cabooter:** Conceptualization, Funding acquisition, Formal analysis, Methodology, Investigation, Supervision, Writing – review & editing.

Acknowledgments

Alexander Kensert, Gilles Collaerts and Kyriakos Efthymiadis are funded by a joint-initiative of the Research Foundation Flanders (FWO) and the Walloon Fund for Scientific Research (FNRS) (EOS – research project "Chimic" (EOS ID: 30897864)). Emery Bosten is funded by Janssen Pharmaceutica.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.chroma.2022.463005.

References

[1] A. Felinger, Data Analysis and Signal Processing in Chromatography, Data Analysis and Signal Processing in Chromatography, 21, 1st ed., Elsevier Science, 1998.

[2] G. Vivó-Truyols, J.R. Torres-Lapasió, A.M. van Nederkassel, Y. Vander Heyden, D.L. Massart, Automatic program for peak detection and deconvolution of multi-overlapped chromatographic signals–Part I–Peak detection, J. Chromatogr. A 1096 (1) (2005) 133–145, doi:10.1016/j.chroma.2005.03.092.

[3] T.S. Bos, W.C. Knol, S.R.A. Molenaar, L.E. Niezen, P.J. Schoenmakers, G.W. Somsen, B.W.J Pirok, Recent applications of chemometrics in one- and two-dimensional chromatography, J. Sep. Sci. 43 (9–10) (2020) 1678–1727, doi:10.1002/jssc.202000011.

[4] A.P. De Weijer, C.B. Lucasius, L. Buydens, G. Kateman, H.M. Heuvel, H. Mannee, Curve fitting using natural computation , Anal. Chem. doi:10.1021/ac00073a006 (accessed 2021 12 -06).

[5] K.J. Goodman, J.T. Brenna, Curve fitting for restoration of accuracy for overlapping peaks in gas chromatography/combustion isotope ratio mass spectrometry, Anal. Chem. doi:10.1021/ac00080a015 (accessed 2021 -12 -06).

[6] S.N. Chesler, S.P. Cram, Iterative curve fitting of chromatographic peaks, Anal. Chem. doi:10.1021/ac60330a031 (accessed 2021 -12 -06).

[7] J. Listgarten, A. Emili, Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry, Mol. Cell. Proteom. 4 (4) (2005) 419–434, doi:10.1074/mcp.R500005-MCP200.

[8] S. Peters, H.G. Janssen, G. Vivó-Truyols, A new method for the automated selection of the number of components for deconvolving overlapping chromatographic peaks, Anal. Chim. Acta 799 (2013) 29–35, doi:10.1016/j.aca.2013.08.041.

[9] A.B. Risum, R. Bro, Using deep learning to evaluate peaks in chromatographic data, Talanta 204 (2019) 255–260, doi:10.1016/j.talanta.2019.05.053.

[10] A.D. Melnikov, Y.P. Tsentalovich, V.V. Yanshole, Deep learning for the precise peak detection in high-resolution LC–MS data, Anal. Chem. 92 (1) (2020) 588–592, doi:10.1021/acs.analchem.9b04811.

[11] R. Tautenhahn, C. Böttcher, S. Neumann, Highly sensitive feature detection for high resolution LC/MS, BMC Bioinform. 9 (1) (2008) 504, doi:10.1186/1471-2105-9-504.

[12] Y. Gloaguen, J.A. Kirwan, D. Beule, Deep learning assisted peak curation for large scale LC-MS metabolomics | bioRxiv https://www.biorxiv.org/content/10.1101/2020.08.09.242727v1 (accessed 2021 -11 -05).

[13] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once–Unified, real-time object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 779–788, doi:10.1109/CVPR.2016.91.

[14] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, et al. TensorFlow –Large-scale machine learning on heterogeneous distributed systems, 2015.

[15] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0–Fundamental algorithms for scientific computing in python, Nat. Methods 17 (3) (2020) 261–272, doi:10.1038/s41592-019-0686-2.

[16] A. Kensert, G. Collaerts, K. Efthymiadis, P. Van Broeck, G. Desmet, D. Cabooter, Deep convolutional autoencoder for the simultaneous removal of baseline noise and baseline drift in chromatograms, J. Chromatogr. A (1646) 462093 2021, doi:10.1016/j.chroma.2021.462093.

[17] L. Bottou, O. Bousquet, The tradeoffs of large scale learning, in: Proceedings of the 20th International Conference on Neural Information Processing Systems; NIPS'07, Curran Associates Inc., Red Hook, NY, USA, 2007, pp. 161–168.

[18] Kingma, D.; Ba, J.Adam: A method for stochastic optimization, Proceedings of the 3rd International Conference for Learning Representations, San Diego, 2015, doi:10.48550/arXiv.1412.6980.