

vIR-OLO: an open-source YOLO-based interface for functional- group detection and localization in infrared spectra

Uriel Garcilazo-Cruz

McMaster University

Qichen Nie

McMaster University

Rudra Sondhi

McMaster University

Rodrigo A. Vargas-Hernández

`vargashr@mcmaster.ca`

McMaster University

Research Article

Keywords:

Posted Date: September 25th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10732656/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

vIR-OLO: an open-source YOLO-based interface for functional-group detection and localization in infrared spectra

Uriel Garcilazo-Cruz,^{a,b,†} Qichen Nie^a, Rudra Sondhi^a, and Rodrigo A. Vargas-Hernández^{a,b,c,†}

Accurate identification of functional groups from infrared (IR) spectra is a fundamental task in molecular characterization. Recent data-driven approaches have demonstrated strong performance in functional group recognition; however, most methods formulate the problem as a classification task and do not explicitly identify the spectral regions responsible for their predictions. This limits their utility for annotation workflows, model verification, and human-guided interpretation. Here, we introduce vIR-OLO, an open-source framework that reformulates functional group identification in IR spectroscopy as an object detection problem using the YOLO architecture. Rather than predicting only the presence of functional groups, vIR-OLO localizes characteristic spectral features through bounding-box annotations directly on IR spectra. We developed and benchmarked three pre-trained models based on alternative functional group sets containing 10, 12, and 13 classes, respectively. Across all configurations, the models achieved strong incidence-based detection performance (micro-averaged F1 scores exceeding 0.84) while maintaining accurate localization of matched spectral regions (mean Intersection over Union greater than 0.80). vIR-OLO is available at <https://github.com/ChemAI-Lab/vIR-OLO>, and the pre-trained models are available at <https://huggingface.co/ChemAI-Lab>.

Scientific Contribution Statement: vIR-OLO is the first open-source YOLO-based framework that reframes functional-group identification in IR spectra as an object-detection task. vIR-OLO not only predicts functional group presence but explicitly localizes characteristic spectral regions through editable bounding-box annotations directly on the spectra. By benchmarking three pre-trained models across different prior approaches, vIR-OLO and its GUI framework advance the field by providing an easily deployable, human-in-the-loop tool that enables reproducible evaluation and community-driven model development.

1 Introduction

Molecule elucidation is the process of identifying molecular structures from spectroscopic data. The identification of chemical compounds is critical in experimental pipelines for drug discovery^{1–3}, biomaterials^{4,5}, conservationism^{6,7}, and materials development⁸, to name a few. Molecule elucidation heavily relies on the analysis of spectra generated by quantifying the absorption, emission, or scattering of electromagnetic radiation by molecules. The use of infrared (IR) light to generate spectra is one of the most popular methodologies due to its affordability and readability.

This technique is especially powerful when combined with other forms of spectral analysis, such as proton (H NMR) and carbon-13 (C NMR) nuclear magnetic resonance (NMR)^{9,10} or mass spectrometry.

The characterization of functional groups in IR spectroscopy requires manual identification of the shape and location of the absorption peaks^{11–18}. This form of generalization requires extensive training and constant cross-referencing with databases, which results in a laborious endeavor that depends on technical expertise and can lead to errors when dealing with unusual functional groups or a lack of practice¹⁹. In recent years, advances in machine learning (ML) and artificial intelligence (AI) have automated the identification of functional group peaks, reducing the manual burden but not eliminating the need for expert review.^{8,11,19–22} The use of vision models - particularly convolutional neural networks (CNNs) - to IR spectra has primarily focused on classification tasks, or label prediction, where the model predicts the existence of a set of chemical identities, in this case functional groups, in an IR spectrum^{8,21,23–26}. These approaches could automate functional group prediction without human intervention, but they do not inherently produce explicit, editable peak boundaries; localization must be inferred via post-hoc “interpretability” methods (e.g., saliency maps), which are not designed as direct annotations. This lack of access limits the experimental analyst’s ability to verify the models’ predictions. A currently unexplored approach in the analysis of IR spectra involves the use of vision models dedicated to *object detection*, where the model is trained to localize annotated spectral regions associated with functional group absorptions in an input image, and is also capable of encoding the incidence of functional groups in a format that is directly available for human corroboration.

This hybrid approach could result in a more integrative form of AI-assisted validation for identifying functional groups in IR-style diagrams. The most commonly used models for object detection target a combination of classification and bounding box regression objectives, including approaches such as Region-based CNNs (R-CNN)^{27–29} and You Only Look Once (YOLO)³⁰. The YOLO family of object detection models is particularly well known in the field of computer vision and has received significant attention for its speed and accuracy^{30,31}. The YOLO model works by dividing the input image into a grid and predicting bounding boxes and class probabilities for each grid cell in a single forward pass through the network, further improving over the spatial capabilities of conventional CNNs by dividing the image into regions of interest. YOLO and YOLO-related architectures, including R-CNNs, have been resourcefully integrated into pipelines related to applied chemistry. Examples include identification of individual fruits as a preprocessing step for chemometric analysis³², peak detection in reversed-phase liquid chromatography (RPLC)³³,

^a Department of Chemistry and Chemical Biology, McMaster University, Hamilton, ON, Canada

^b School of Computational Science and Engineering, McMaster University, Hamilton, ON, Canada

^c Brockhouse Institute for Materials Research, McMaster University, Hamilton, ON, Canada

[†] Corresponding author: garcilau@mcmaster.ca, vargashr@mcmaster.ca

liquid chromatography–high-resolution mass spectrometry (LC-MS)^{34,35}, multiple reaction monitoring (MRM)³⁶, NMR^{37,38}, and Gas Chromatography with Mass Spectrometry (GC-MS)³⁹. Recent IR-based functional-group classifiers, including Punjabi *et al.*, have demonstrated that neural networks can accurately predict molecular-level functional-group presence from spectra⁴⁰. However, these approaches generally return label sets rather than editable spectral annotations. In contrast, vIR-OLO is designed for localization-aware, human-editable annotation: each prediction is returned as a functional-group-specific bounding box that can be inspected, corrected, and reused to refine the model.

In this work, we trained a YOLO architecture on IR spectra to identify and delimit functional group peaks. We developed vIR-OLO complemented by an open-source Python-based Graphical User Interface (GUI) software that builds on the Ultralytics YOLO architecture⁴¹ to simplify the training and deployment of object detection models for IR. The program provides a user-friendly interface to the Ultralytics Python package⁴¹, making it easier for users to adapt YOLO-based models without requiring coding expertise. As part of this work, we also developed and benchmarked the performance of three versions of pretrained YOLO models specifically developed for IR, capable of detecting and localizing among 10, 12, and 13 functional groups. These pre-trained models are available at <https://huggingface.co/ChemAI-Lab>, and the vIR-OLO codebase is openly accessible at <https://github.com/ChemAI-Lab/vir-olo>. This enables immediate use for functional group identification and supports further fine-tuning with user-provided annotations, accelerating the development of spectrum-based applications.

The paper is structured as follows. Section 2 describes the details of the proposed vision models, training process, and benchmarks. Section 2.4 describes the specification and deployment details of vIR-OLO that is developed in this work. We also present some quality metrics in a comparative context to evaluate the capacity of our three models. Section 3 consolidates the results and metrics obtained during training and evaluation of each model. Finally, in Section 4 we discuss our findings.

2 Methodology

2.1 Data

We manually selected and downloaded 2,200 IR spectra from the Spectral Database for Organic Compounds (SDBS)⁴², filtering for spectra containing at least one of the 12 functional groups targeted in this study: aromatics, alcohols, amines, esters, alkene, carboxylic acids (carb. acids), ketones, phenol, nitriles, amides, aldehydes, and alkynes, totaling 5,635 bounding boxes (detailed bounding box counts per functional group can be found in the Supplemental Information (SI) Table S1). As all spectra were selected to contain at least one target functional group, this benchmark does not assess performance on spectra lacking all target classes. Future validation should include negative and out-of-domain spectra to quantify false-positive behavior under broader deployment conditions. Each spectrum was manually annotated using an earlier version of the vIR-OLO GUI annotator, developed in Python using PyQt5⁴³, enabling efficient navigation of spectra

through suggested functional groups. The implementation builds upon our previously developed GUI annotator, LivePyxel⁴⁴.

Annotations were stored in .txt format using the relative coordinates and dimensions of each bounding box. The annotation process was iterative, with each entry verified by at least two annotators. Ambiguous or uncertain functional group assignments were cross-validated against the original molecular structure provided by SDBS. Finally, the dataset was split into training, validation and test datasets using an 80/10/10 split, and before any augmentation to avoid data leakage. All augmented samples were generated exclusively from the training set. Further details are provided in the SI: Section S1.

2.1.1 Data Augmentation

To account for variability in bounding box placement during manual annotation, each spectrum in the training set was augmented by generating five additional versions with perturbed annotations. These perturbations were applied only to the bounding box coordinates and dimensions, leaving the underlying spectrum unchanged.

For positional variation, the center coordinates of each bounding box were displaced by adding Gaussian noise with a standard deviation equal to one-third of a maximum offset of 10% over the original box’s width, applied in both the horizontal and vertical directions. The width and height of each bounding box were independently scaled by random factors uniformly sampled from 1.0 to 1.25, allowing up to a 25% increase in box size. These transformations emulate inter-annotator variability in defining spectral peak boundaries while ensuring that most augmented boxes remain within the image frame.

Importantly, this augmentation strategy does not introduce new spectral information; instead, it improves robustness to annotation noise and variability in bounding-box definitions. The resulting training dataset comprised 10,560 image/annotation pairs. The validation and test datasets both comprised 220 pairs. Further details are provided in the SI: Section S1.

As an additional robustness analysis, we also evaluated the sensitivity of model performance to dataset partitioning using eight independent random 70/15/15 train/validation/test splits without explicit external augmentation in the training dataset. This repeated-split experiment was designed as a conservative complement to the augmented validation benchmark rather than as a replacement for the main evaluation; details are provided in the SI: Section S6.

2.2 Model Training

A YOLO-based architecture was selected to align with the object detection formulation of our task, where the goal is to localize and classify spectral features within IR spectra. As a foundational model for real-time object detection, YOLO provides a natural framework for predicting bounding boxes and associated functional group labels, fitting directly into the proposed annotation pipeline. In addition, the relatively limited size of the available training dataset motivated the use of a convolutional detection model over transformer-based alternatives. While transformer architectures have shown strong performance in vision tasks, they

require substantially larger datasets to effectively learn attention patterns and reach optimal capacity^{45,46}. In this setting, YOLO offers a more data-efficient and stable training regime, providing reliable predictions without the need for large-scale pretraining. Furthermore, YOLO achieves a favorable balance between prediction speed and detection accuracy, enabling rapid inference on standard desktop hardware. This is particularly important for the human-in-the-loop annotation framework considered in this work, where low-latency predictions are essential for iterative user interaction and feedback. The Ultralytics Python package⁴¹ was used for training and deployment due to its flexibility and ease of integration. Specifically, the *YOLO11l* (large) variant was selected to balance model capacity and computational cost. The model was implemented using a custom Ultralytics environment.

For all models, we used a batch size of 64 and enabled early stopping, letting the Ultralytics package select the optimization algorithm based on training duration (stochastic gradient descent with momentum and Nesterov acceleration for long runs, and AdamW for shorter runs). The training objective is a multi-component loss: bounding box, regression loss, classification loss, and distribution focal loss (dfl). All simulations were performed on an NVIDIA A100SXM4 GPU with 40 GB VRAM. See SI: Section S4 and S5.3 for further training details.

To evaluate changes in the model’s capacity to identify different functional groups, we created three versions of the **vIR-OLO** model. The first model contained the original 12 functional groups (**vIR-OLO_12-FG**), and for the second model (**vIR-OLO_10-FG**), we merged ketones, aldehydes, and esters into a single label “*carbonyl subgroups*” using the ambiguity in their absorption frequencies as rationale. In both the **vIR-OLO_12-FG** and **vIR-OLO_10-FG** models, several functional groups, including amines, alkenes, alkynes, amides, and carboxylic acids, often exhibit two or more peaks that are clearly distinguished by large wavenumber differences. Such spatial differences are also illustrated in Fig. 1A and B, where the normalized coordinates of these functional groups are clustered near the lower and upper quartiles of the box plots, and relatively few data points fall within the interquartile range. To address this polarized, bimodal distribution, we subdivided the peaks of amines, alkenes, and alkynes according to their relative horizontal position and assigned suffixes “L” (left) or “R” (right). The amides label was retained as a single class in the 13FG ontology under the name *amides-R*, rather than being split into paired subclasses. This refinement introduced the third model (**vIR-OLO_13-FG**), enabling more precise handling of multi-region functional groups (Fig. 1C); further details in the SI: Section S2. Throughout the manuscript, we will refer to these models as **10FG**, **12FG**, and **13FG**, respectively, for simplicity.

The trained models were evaluated on the 220 data points comprising the test dataset (see the SI: Tables S2, S3, and S4 for detailed functional group counts in the test dataset for each of the three models). For each IR spectra image, bounding box predictions were generated with a confidence threshold of 0.25, ensuring detection of low-confidence candidates. Ground-truth labels were loaded from corresponding annotation files in YOLO format (normalized center coordinates, width, and height). Pre-

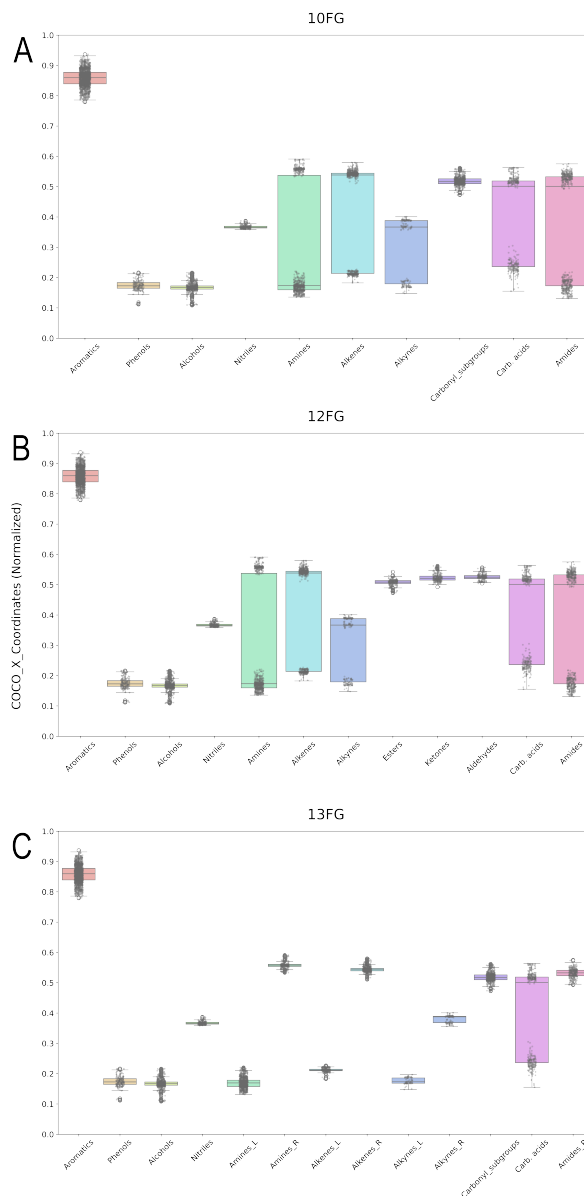


Fig. 1 Distribution of the normalized box x coordinates among functional groups for models (A) 10FG, (B) 12FG, and (C) 13FG.

dicted and ground-truth boxes were matched per class via optimal (Hungarian) assignment maximizing total IoU; assigned pairs with $\text{IoU} \geq 0.7$ were counted as true positives. Unmatched predictions were classified as false positives (FP), while unmatched ground-truth boxes were classified as false negatives (FN). True positives (TP) were defined as successfully matched prediction-ground-truth pairs.

2.3 Performance Metrics

The YOLO architecture jointly optimizes the localization of bounding boxes (localization) and the prediction of object class labels (incidence) within an image. We computed two complementary detection metrics.

Localization-based metrics: To assess the accuracy of the predicted bounding box locations, we used the Intersection over

Union (IoU), which quantifies the spatial agreement between a predicted bounding box and its corresponding ground-truth box by measuring the ratio of their area of overlap to the area of their union. For evaluation, we used an IoU threshold of 0.7 to define a correctly matched detection. This choice enforced a relatively strict localization requirement, distinguishing tightly aligned predictions from loosely overlapping detections. Once a model was trained, IoU was computed for each functional group across the validation dataset to characterize its localization performance. The Hungarian algorithm⁴⁷ was employed to optimally match predicted bounding boxes to ground-truth annotations for each class independently, ensuring a one-to-one correspondence and maximizing overall localization quality. A prediction was classified as a true positive (TP) only when the matched box exceeded the IoU threshold (0.7); matches below the threshold and unmatched predictions were classified as false positives (FP), while unmatched ground-truth boxes were classified as false negatives (FN). We then evaluated localization-aware F1 scores using TP, FP, and FN to establish the optimal spatial correspondence between predictions and ground-truth annotations. This type of localization-aware metric provides a more comprehensive evaluation of model performance, as they penalize both missed detections and spatially misaligned predictions.

Incidence-based metrics: In addition, we complemented our findings with a simpler version of the F1 score to evaluate the capacity of the model to detect the presence of functional groups, regardless of their bounding boxes. For each functional group, we counted the number of predicted boxes and ground-truth boxes per image, computing precision and recall as the proportion of correctly identified class instances regardless of spatial accuracy.

To contextualize the incidence-based performance of vIR-OLO, we also compared the **12FG** model against representative IR-based functional group classification models reported in the literature. This comparison was restricted to presence-based metrics because most prior models do not produce bounding box annotations. The comparison is therefore intended as a contextual benchmark rather than a strict head-to-head evaluation, as datasets, label spaces, and model objectives differ across studies; see Table 1.

Finally, to assess co-occurrence and co-prediction patterns among functional groups, we computed pairwise Matthews Correlation Coefficients (MCCs) from binary functional group presence vectors for both ground-truth annotations and model predictions. MCC values range from -1 to 1 and were used to assess whether label associations present in the annotated data were also retained in model predictions.

2.4 Graphical User Interface and Finetuning

Any of the three vIR-OLO models can be deployed via a graphical user interface that allows annotating new spectra either by loading one of the three pre-trained models for prediction or by manually labeling data for subsequent fine-tuning. Beyond prediction, vIR-OLO streamlines dataset preparation by automatically restructuring user-provided folders of spectra into the YOLO format. It also provides an accessible interface for training cus-

tom YOLO models, supporting a broad range of spectrum-based and chemistry-related applications.

The GUI is organized into three main panels (Fig. 2).

- **Panel A** provides tools for image navigation and model inference. **Subpanel A.1** enables navigation through the dataset via image indexing or name-based search, as well as sequential browsing (Fig. 2-A.1). The model information subpanel (Fig. 2-A.2) displays details of the chosen vision model, while **Subpanel A.3** (Fig. 2-A.3) allows users to execute predictions using a selected pre-trained model. Predictions are automatically stored in the `/labels/train` directory.
- **Panel B** corresponds to the label manager, which allows users to add, remove, or modify label identities and propagate these changes across all existing annotations (Fig. 2-B).
- **Panel C** contains the annotation canvas, where users (or the detection model) can create, edit, and delete bounding boxes (Fig. 2-C.2–C.4). It also includes quick access to the project image directory (Fig. 2-C.1).

When a user creates a new project, vIR-OLO automatically restructures the selected folder into a format compatible with YOLO training workflows. This removes the need for manual dataset organization and provides a smooth starting point for training new models directly within the GUI. Such automation is particularly valuable for researchers who may not be comfortable managing file systems or programming AI workflows; see Fig. SI: 5 for further details on the final structure of the data handled by the software vIR-OLO.

vIR-OLO works by handling images and annotations while respecting the underlying YOLO formatting, enabling the training of this architecture. This means the user can train their own models using the Python Ultralytics package. When a new project is created, a `dataset.yaml` is also generated, which contains information on the classes to which the expected labels belong. This file is dynamically updated to reflect the order and names provided by the user in the label management panel (Fig. 2B). When the user is ready to train their own model, they can use the `train.py` script located in the project’s GitHub repository at `./train`. This script will traverse the folder structure, evaluate the quality of the dataset, and ensure every image has an annotation, in which case the program takes 20% of the images and labels from the training folders into the validation folders before starting training. The training process relies on a pre-existing version of the YOLO model, which is then fine-tuned with the provided annotations and the new `dataset.yaml` file. Once the model is trained, a new folder containing the model’s weights and other statistics provided by the Ultralytics package during training will be created. While this protocol ensures users can train their models for specific applications, it is also possible to reinsert and load a custom-trained model back into vIR-OLO by selecting a folder containing the weights with `.pt` extension and a file `dataset.yaml`. This functionality is illustrated in the online tutorials and examples provided with the software on the GitHub repository, <https://github.com/ChemAI-Lab/vIR-OLO>, ensuring the training process can be reintegrated into a custom

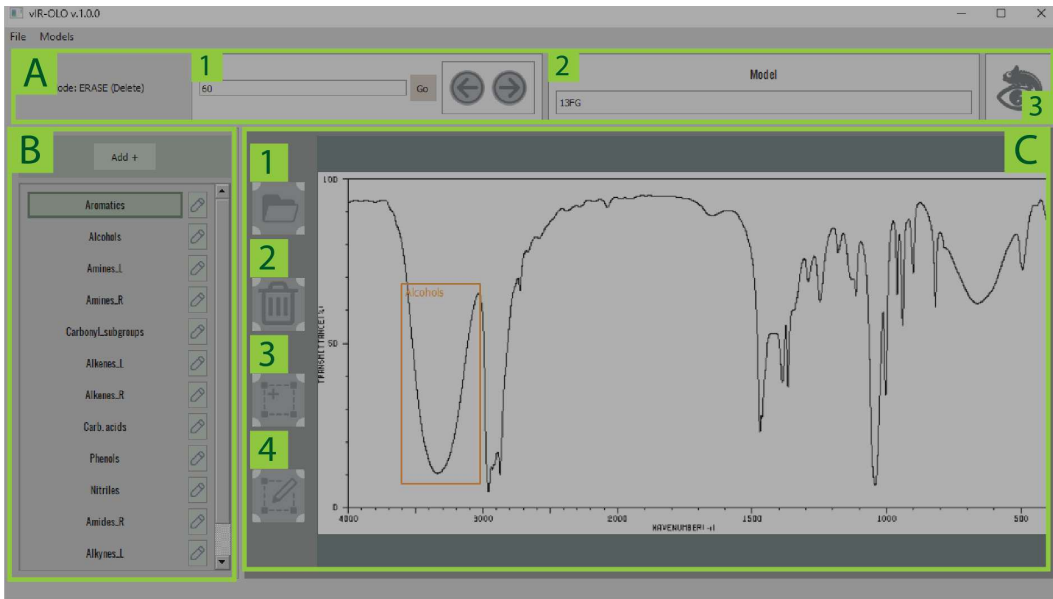


Fig. 2 Overview of the vIR-OLO graphical user interface (GUI) with its main components labeled. (A) Model control panel, where users can select and load pretrained YOLO architectures, monitor model status, and navigate between images. (B) Label management panel for selecting or adding functional group categories to annotate. (C) Central canvas displaying IR spectra and predicted or manually drawn bounding boxes.

application that end users can use to further augment their data with new annotations assisted by predictions, serving as an application for inexperienced annotators or a fully automated tool. We aim to expand the functionality of the vIR-OLO software to fully automate this process in a future release.

3 Results

3.1 Models performance

Here, we list the individual performance for the three variants of vIR-OLO. For additional information, please refer to the SI: Section S4.

Model 12FG: Incidence-based scores: Evaluation on the 220 test samples yielded a micro-averaged precision of 0.8279, recall of 0.8623, and F1 score of 0.8448. Macro-averaged precision, recall, and F1 were 0.7930, 0.8640, and 0.8212, respectively, with an exact match ratio of 0.6818. These results indicate strong performance in detecting the presence of functional groups across images, with high per-class F1 scores for several categories, including aromatics, alkynes, and carboxylic acids (Fig. 3). **Localization-based scores:** To assess spatial localization, bounding box predictions were evaluated using an IoU threshold of 0.7. The model achieved a localization precision of 0.6458, recall of 0.6683, F1 score of 0.6569, and mean IoU of 0.8046 (Fig. 4), indicating accurate spatial alignment for matched detections. Per-class localization F1 scores ranged from 0.471 for ketones to 0.842 for esters. Localization performance remained moderate for some functional groups (i.e. near 0.8), particularly those with limited training data or more complex spectral features.

Model 10FG: Incidence-based scores: Evaluation on the test dataset yielded a micro-averaged precision of 0.8403, recall of 0.8403, and incidence-based F1 score of 0.8403. Macro-averaged

precision, recall, and F1 were 0.8188, 0.8303, and 0.8208, respectively, with an exact match ratio of 0.7182. Per-class F1 scores ranged from 0.69 for amides to 0.94 for aromatics, with the merged *carbonyl subgroups* class achieving an F1 score of 0.80 (Fig. 3). **Localization-based scores:** Localization performance was evaluated using an IoU threshold of 0.7. The model achieved a precision of 0.6539, recall of 0.6517, localization F1 score of 0.6528, and mean IoU of 0.8096 (Fig. 4). These results indicate accurate spatial localization of matched detections, although recall remains moderate for certain classes, particularly those with limited representation or more complex spectral features.

Model 13FG: Incidence-based scores: Evaluation on the test dataset yielded a micro-averaged precision of 0.8571, recall of 0.8533, and incidence-based F1 score of 0.8552. Macro-averaged precision, recall, and F1 were 0.8611, 0.8255, and 0.8373, respectively, with an exact match ratio of 0.6727. Per-class F1 scores ranged from 0.69 for *amides-R* to 0.94 for aromatics, with the *carbonyl subgroups* class achieving an F1 score of 0.80 (Fig. 3). **Localization-based scores:** Localization performance was evaluated using an IoU threshold of 0.7. The model achieved a precision of 0.6634, recall of 0.6733, localization F1 score of 0.6683, and mean IoU of 0.8064 (Fig. 4). While overall localization performance remained comparable to the other models, the introduction of position-dependent labels revealed variable behavior across functional groups. The *carbonyl subgroups* class maintained broadly consistent performance relative to the 10FG model. *Alkynes-R* improved relative to the unsplit alkyne class, whereas *alkynes-L* remained lower. Separating amines led to asymmetric localization performance, with stronger results for *amines-R* than for *amines-L*, while subdividing alkenes produced mixed results, with *alkenes-L* exceeding and *alkenes-R* falling below the original unsplit label.

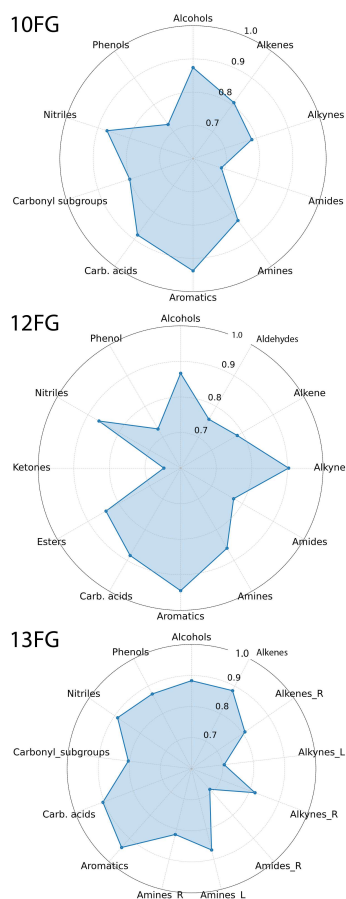


Fig. 3 Comparison of per-class F1 scores between 10FG, 12FG and 13FG YOLO-based models. See the text for further details.

3.2 Co-prediction patterns

Pairwise MCC analysis revealed moderate functional group associations that were present in both ground-truth annotations and model predictions (SI: Fig. S10). In the 10FG and 12FG label schemes, phenols co-occurred with aromatics and alcohols in the ground-truth, with MCC values of 0.45 and 0.53, respectively. The phenol-aromatic association was reproduced in the predictions, with MCC values of 0.37 for 12FG and 0.39 for 10FG. In the 13FG model, the dominant association patterns observed in the ground-truth were retained in the predictions, albeit with lower MCC values. These results indicate that the models capture some label co-prediction structure, but the pairwise associations should not be interpreted as deterministic rules.

4 Discussion

4.1 Localization-aware versus presence-based detection performance

A clear distinction emerged between detecting the presence of functional groups and precise bounding-box localization. Presence-based F1 scores were consistently high for all models, with micro-averaged F1 scores of 0.8448 for 12FG, 0.8403 for 10FG, and 0.8552 for 13FG. Thus, when evaluated only as a per-label incidence task, the 13FG model performed best, suggesting that finer label granularity can help the model capture dis-

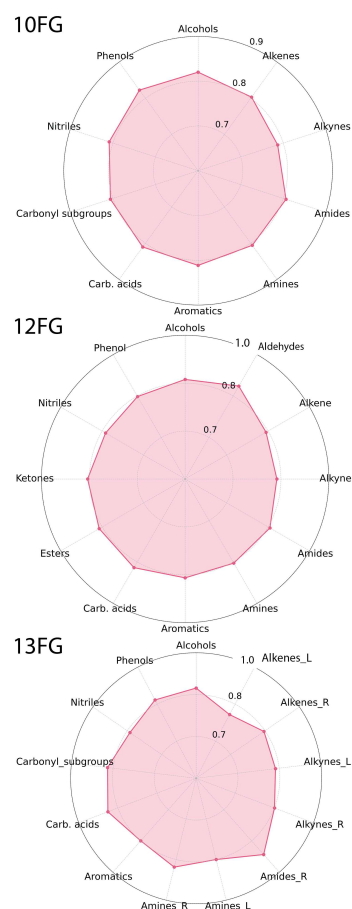


Fig. 4 Comparison of mean Intersection over Union (mIoU) scores between models 10FG, 12FG, and 13FG. See the text for further details.

tinct spectral patterns. However, this improvement should not be interpreted as a universal advantage of greater granularity: exact-match performance was highest for the 10FG model, and localization-aware F1 scores remained comparable between 10FG and 13FG.

In contrast, localization-aware evaluation using Hungarian-matched bounding boxes and an IoU threshold of 0.7 revealed lower F1 scores of 0.6569 for 12FG, 0.6528 for 10FG, and 0.6683 for 13FG, despite high mean IoU values exceeding 0.80. This discrepancy highlights an important pattern: when a detection is made, the predicted bounding boxes are generally well-aligned with the annotated regions, but a considerable fraction of ground-truth boxes remain undetected or are detected with insufficient confidence to meet the strict localization criterion. The high mean IoU, therefore, reflects accurate spatial alignment for true positives, while the moderate localization-aware F1 scores primarily indicate challenges in recall rather than localization precision.

The choice of an IoU threshold of 0.7 imposes a relatively strict definition of a correct detection. While such a threshold may underestimate usable detections in practical applications, it provides a conservative evaluation of localization accuracy under the scoring protocol used here. This level of penalization toward false positives is desirable from an application standpoint. The combination of high mean IoU and strong presence-based F1 scores sug-

Table 1 Contextual comparison of functional group (FG) identification performance across representative IR-based machine learning approaches. All entries report macro-averaged F1 over functional groups, together with the exact match ratio (EMR, perfect prediction of all functional groups for a molecule) where the original work provides one. Works reporting only molecule-level, weighted-average, or accuracy-based metrics are omitted rather than compared on non-equivalent terms. This table is intended to establish an expected performance range rather than a strict head-to-head benchmark, as datasets and label spaces differ, and vIR-OLO alone is additionally evaluated on the localization of each functional group.

Work	Input	Dataset	# FG	Task	EMR / MPR	Reported F1
Jung <i>et al.</i> ⁸	1D IR spectra	NIST + SDBS	37	Multi-label FG classification	0.72	0.85
Lee <i>et al.</i> ⁴⁸	1D IR spectra	NIST	17	Multi-label FG classification	—	0.88
j-IR-vis ²⁶	IR images	NIST + SDBS	10	Multi-label FG classification	0.76	0.84
Kong ⁴⁹	1D IR spectra	NIST	21	Multi-label FG classification	—	0.917
vIR-OLO-10FG	IR images	SDBS	10	FG detection + set inference	0.718	0.821
vIR-OLO-12FG	IR images	SDBS	12	FG detection + set inference	0.682	0.821
vIR-OLO-13FG	IR images	SDBS	13	FG detection + set inference	0.673	0.837

gests that the models are well-suited for assisted annotation and exploratory analysis, especially when supervised by a human annotator. The limited size of the manually annotated dataset likely contributes to these recall limitations, particularly for underrepresented functional groups, despite extensive augmentation.

The repeated random-split analysis in the Supplemental Information supports this interpretation under an unaugmented setting (SI: Section S6). At the model level, the held-out micro-F1 scores across eight random train/validation/test partitions remained close to — and in fact marginally above — the localization-aware F1 scores from the augmented benchmark: 10FG changed from 0.6528 to 0.680 ± 0.022 , 12FG from 0.6569 to 0.674 ± 0.022 , and 13FG from 0.6683 to 0.688 ± 0.032 . This agreement suggests that the overall localization trends are not simply artifacts of a single split. The two regimes nonetheless reach comparable F1 by different routes: the unaugmented models achieve higher recall (0.787–0.803 versus 0.652–0.673) at lower precision (0.591–0.603 versus 0.646–0.663). At the per-functional-group level, narrow and sharply localized absorptions benefited most from the unaugmented regime, with nitriles improving by 0.19 to 0.28 across the three models, whereas under-represented or spectrally ambiguous labels degraded: phenols decreased from an augmented-benchmark localization F1 of 0.850 to 0.622 ± 0.063 in 13FG and from 0.727 to 0.618 ± 0.050 in 10FG, and carboxylic acids in 13FG showed marked split-level variability (0.723 ± 0.107). The consistently high mean IoU values in the repeated-split analysis (0.907–0.913) further suggest that, when detections are successfully matched, the predicted regions are generally well aligned with the annotations. Thus, the dominant source of error appears to be class-specific false positives, together with missed detections in the least-represented classes, rather than poor spatial overlap among matched boxes.

4.2 Effect of carbonyl-label consolidation

Collapsing carbonyl-containing functional groups into a single class in the 10FG model improved presence-based classification performance compared to the 12FG model while preserving similar localization-aware metrics. This result suggests that the features associated with carbonyl peaks are readily learned by the model. However, the finer-grained discrimination among esters, aldehydes, and ketones introduces ambiguity that is difficult to resolve using IR spectra alone within the dataset size available in this study.

The 12FG model offers greater chemical specificity but exhibits increased susceptibility to FPs and FNs within the carbonyl subclasses, likely reflecting intrinsic spectral overlap rather than model inadequacy. In contrast, the 10FG model mitigates this ambiguity by aligning the labeling scheme more closely with the observability of functional groups, improving robustness at the cost of reduced granularity. Functional groups such as alcohols, aromatics, and amines showed consistently strong presence-based performance across both models, indicating that their characteristic spectral features are distinct enough to support reliable incidence prediction.

From a practical perspective, the 10FG model may therefore be preferable for automated or semi-automated annotation workflows, where minimizing misclassifications due to overlapping spectral signatures is critical. The 12FG model, while more detailed, may be better suited for expert-guided interpretation, downstream analyses that incorporate complementary structural information beyond IR spectra, or workflows where additional data can be collected to refine subclass-specific predictions.

4.3 Functional group relationships and label granularity

The 13FG model tested whether position-dependent labels could improve the handling of functional groups that occupy multiple spectral regions. This refinement produced mixed results. For example, *alkynes-L* improved relative to the unsplit alkyne class, whereas *alkynes-R* remained weaker; amines showed asymmetric behavior, with stronger performance for *amines-R* than *amines-L*; and the subdivision of alkenes reduced F1 scores relative to the original alkene label. These results suggest that label splitting can help when the resulting subclasses correspond to sufficiently distinct and well-represented spectral patterns, but it can also amplify data imbalance or annotation ambiguity.

The representative spectra in Fig. 5 illustrate this behavior qualitatively: the same spectrum can lead to different label assignments depending on whether the model uses collapsed, original, or position-dependent functional group labels. These examples highlight both the benefit of explicit localization and the tradeoff introduced by finer label granularity. The bounding boxes provide qualitative evidence for a predicted assignment, but they also reveal cases where related spectral regions or overlapping functional group patterns can produce different predictions across model configurations.

Phenols are a useful example of chemically coupled labels. Be-

cause phenolic compounds contain both aromatic structure and O–H character, their annotations can co-occur with aromatic and alcohol-associated spectral regions. The MCC analysis supports this relationship at the label level, showing moderate associations between phenol and aromatics or alcohols (SI: Fig. S10). Although this correlation is not proof that spatial overlap causes misclassifications, it suggests that phenol prediction may be influenced by chemically related spectral context, which may help explain why phenol incidence improves in the 13FG model while phenol localization remains more challenging than alcohol localization.

4.4 Contextual comparison with prior IR models

To contextualize **vIR-OLO_12-FG**, we compared its validation-set functional group predictions with representative IR-based multi-label classification approaches, including j-IR-vis²⁶. These prior models address the related task of predicting which functional groups are present in an IR spectrum. By contrast, **vIR-OLO** is designed for localization-aware functional-group detection: it predicts not only the molecular-level label set, but also functional-group-specific spectral regions that can be visually inspected, edited in the GUI, and reused for downstream model refinement. The validation dataset was evaluated using custom scripting to produce multi-hot label vectors for each spectrum. Precision, recall, and F1 scores were computed using both micro and macro-averaging, along with the exact match ratio (EMR), defined as the fraction of spectra for which the predicted set of functional groups exactly matched the ground-truth; see Table 1. Using this architecture, 12FG achieved a micro-averaged precision of 0.8463, recall of 0.8987, and F1 score of 0.8717, while macro-averaged metrics yielded a precision of 0.8379, recall of 0.8843, and F1 score of 0.8577. The EMR was 0.7390, indicating that 73.90% of spectra were assigned a perfectly correct set of functional groups. Given the class imbalance inherent to IR datasets and the limited sample size of each functional group, macro-averaged F1 is taken as the primary summary metric for functional group identification performance.

4.5 Software utility and human-in-the-loop annotation

Beyond the pretrained detectors, **vIR-OLO** is a human-in-the-loop annotation environment in which automation is treated as the outcome of iterative model refinement guided by experts, rather than as a one-step replacement for spectral interpretation. In many IR classification workflows, spectra are transformed into numerical representations, and the model returns functional group labels without editable visual evidence. By contrast, **vIR-OLO** keeps the spectrum in a human-readable visual space: predictions are displayed as bounding boxes on the same canvas used for manual annotation, allowing users to confirm, reject, or correct each assignment.

This workflow gives researchers direct control over the annotations used for model refinement. Corrected predictions can be reused to fine-tune custom models, meaning the resulting detector can progressively reflect its users’ annotation policy and domain-specific judgment. Because the software combines an-

notation, prediction, dataset organization, and model reloading within a single open-source interface, it lowers the barrier to building specialized IR detection models. The use of YOLO-based architectures further makes this workflow computationally accessible on commercial hardware, supporting a practical path from assisted annotation toward increasingly automated functional group detection.

The same workflow also creates a path toward community-based model sharing. Once users fine-tune models for specialized spectral domains, those models can be reloaded locally in **vIR-OLO** or shared through public repositories such as Hugging Face. In principle, this could support a distributed ecosystem in which research groups contribute and reuse models trained on different chemical spaces, gradually expanding the practical coverage of AI-assisted IR annotation while preserving transparent, visually inspectable predictions.

5 Summary

In this work, we introduced **vIR-OLO**, an open-source framework for functional group detection and localization in IR spectra using YOLO-based object detection models. Unlike conventional IR classification approaches that only predict the presence of functional groups, **vIR-OLO** produces explicit bounding-box annotations that identify the spectral regions associated with each prediction. We developed and evaluated three pretrained models based on alternative functional group ontologies (10FG, 12FG, and 13FG), achieving strong incidence-based detection performance while maintaining accurate localization of matched spectral features.

Beyond model development, we provide a GUI that supports annotation, prediction, dataset management, and retraining of custom YOLO models without requiring extensive programming expertise. By combining object detection with human-in-the-loop annotation, **vIR-OLO** offers a practical workflow for AI-assisted spectral analysis and the creation of new spectroscopic datasets. All software, pretrained models, and training resources are released as open-source tools to support further development of machine-learning approaches for IR spectroscopy.

Author Contributions

UG wrote the code for **vIR-OLO**, QN annotated the dataset, and UG, QN, and RS performed all experiments. RAVH guided the project. All authors wrote and approved the final version of the manuscript.

Conflicts of interest

The authors have no conflicts to disclose.

Data availability

Ultralytics YOLO architecture⁴¹ is licensed under AGPL-3.0. **vIR-OLO** is freely available at <https://github.com/ChemAI-Lab/vIR-OLO> and through PyPi wheels, *pip install virolo*. All trained vision models are available through Hugging Face.

- **vIR-OLO-10FG:** <https://huggingface.co/ChemAI-Lab/vIR-OLO-10FG>

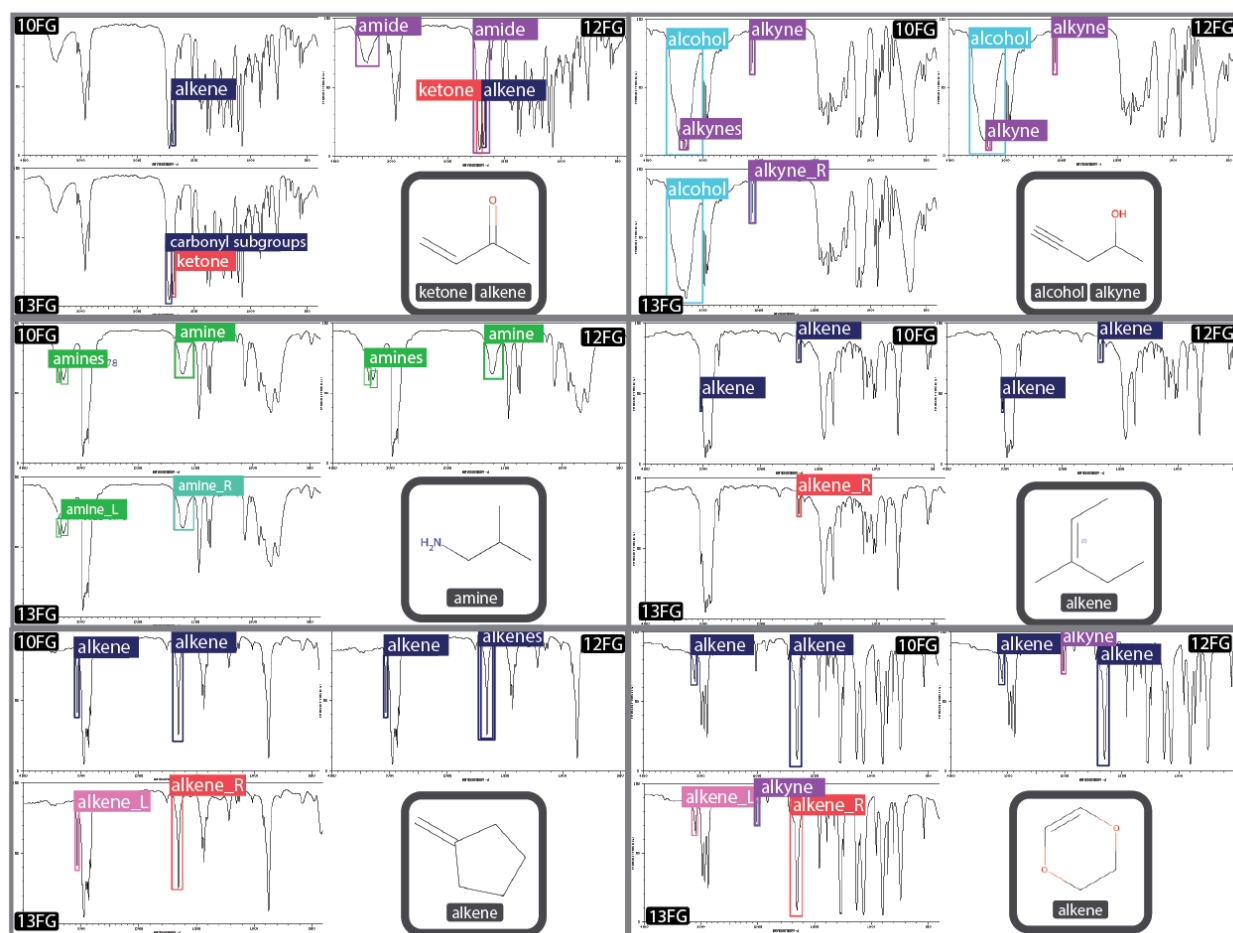


Fig. 5 Comparison of functional group predictions from the 10FG, 12FG, and 13FG models across six representative molecules. Each panel shows predicted bounding boxes overlaid on the IR spectra. Differences in label assignments reflect the varying granularity of each classification scheme, particularly for functional groups that have position-dependent labels (e.g., *amine_L/amine_R*, *alkene_L/alkene_R*).

- vIR-OL0-12FG: <https://huggingface.co/ChemAI-Lab/vIR-OL0-12FG>
- vIR-OL0-13FG: <https://huggingface.co/ChemAI-Lab/vIR-OL0-13FG>

Acknowledgment

RAVH acknowledges the support from the Digital Research Alliance of Canada and NSERC Discovery Grant No. RGPIN-2024-06594.

References

- 1 Y. Liu, E. P. Romijn, G. Verniest, K. Laukens and T. De Vijlder, *TrAC Trends in Analytical Chemistry*, 2019, **121**, 115686.
- 2 A. S. Edison, M. Colonna, G. J. Gouveia, N. R. Holderman, M. T. Judge, X. Shen and S. Zhang, *Analytical chemistry*, 2020, **93**, 478–499.
- 3 G. Nagana Gowda and D. Raftery, *Analytical chemistry*, 2023, **95**, 83–99.
- 4 Z. Movasaghi, S. Rehman and I. U. Rehman, *Applied Spectroscopy Reviews*, 2008, **43**, 134–179.
- 5 C. Wang, B. Zhang, I. Timári, Á. Somogyi, D.-W. Li, H. E. Adcox, J. S. Gunn, L. Bruschweiler-Li and R. Bruschweiler, *Analytical chemistry*, 2019, **91**, 15686–15693.
- 6 J.-L. Xu, K. V. Thomas, Z. Luo and A. A. Gowen, *TrAC Trends in Analytical Chemistry*, 2019, **119**, 115629.
- 7 A. Yadav, S. Sharma, V. Singh, M. Kapoor and R. Singh, *Frontiers in Analytical Science*, 2024, **4**, 1508509.
- 8 G. Jung, S. G. Jung and J. M. Cole, *Chemical Science*, 2023, **14**, 3600–3609.
- 9 F. Hu, M. S. Chen, G. M. Rotskoff, M. W. Kanan and T. E. Markland, *ACS Central Science*, 2024, **10**, 2162–2170.
- 10 F. Zipoli, M. Alberts and T. Laino, *Scientific Data*, 2025, **12**, 1375.
- 11 A. Mokari, S. Guo and T. Bocklitz, *Molecules*, 2023, **28**, 6886.
- 12 A. Kassem, L. Abbas, O. Coutinho, S. Opara, H. Najaf, D. Kasperek, K. Pokhrel, X. Li and S. Tiquia-Arashiro, *Frontiers in Microbiology*, 2023, **14**, 1342406.
- 13 K. B. Beć, J. Grabska and C. W. Huck, *Analytica chimica acta*, 2020, **1133**, 150–177.
- 14 S. A. Alajaji, R. Sabzian, Y. Wang, A. S. Sultan and R. Wang,

- Cancers*, 2025, **17**, 796.
- 15 C. Delrue, R. Speeckaert, M. Oyaert, T. Kerre, S. Rottey, R. Coopman, W. Huvenne, S. De Bruyne and M. M. Speeckaert, *International Journal of Molecular Sciences*, 2023, **24**, 17007.
 - 16 S. Pasieczna-Patkowska, M. Cichy and J. Flieger, *Molecules*, 2025, **30**, 684.
 - 17 J. Madejová and H. Pálková, *Clays and Clay Minerals*, 2024, **72**, e30.
 - 18 J. Workman Jr, 2024.
 - 19 J. A. Fine, A. A. Rajasekar, K. P. Jethava and G. Chopra, *Chemical science*, 2020, **11**, 4618–4630.
 - 20 A. Jahangiri and V. Orekhov, *Communications Chemistry*, 2024, **7**, 244.
 - 21 A. A. Enders, N. M. North, C. M. Fensore, J. Velez-Alvarez and H. C. Allen, *Analytical Chemistry*, 2021, **93**, 9711–9718.
 - 22 M. Alberts, T. Laino and A. C. Vaucher, *Communications Chemistry*, 2024, **7**, 268.
 - 23 S. Jiang, Z. Xu, M. Kamran, S. Zinchik, S. Paheding, A. G. McDonald, E. Bar-Ziv and V. M. Zavala, *Computers & Chemical Engineering*, 2021, **155**, 107547.
 - 24 X. Yan, H. Fu, S. Zhang and H. Qu, *Journal of Separation Science*, 2020, **43**, 663–670.
 - 25 C. Peng, L. Zhong, L. Gao, L. Li, L. Nie, A. Wu, R. Huang, W. Tian, W. Yin, H. Wang *et al.*, *International Journal of Pharmaceutics*, 2024, **655**, 124001.
 - 26 R. Sondhi, E. Chacko and R. A. Vargas-Hernández, *ChemRxiv*, 2025.
 - 27 R. Girshick, J. Donahue, T. Darrell and J. Malik, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
 - 28 R. Girshick, *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
 - 29 S. Ren, K. He, R. Girshick and J. Sun, *Advances in Neural Information Processing Systems*, 2015.
 - 30 J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
 - 31 J. Redmon and A. Farhadi, *arXiv preprint arXiv:1804.02767*, 2018.
 - 32 J. Xu and P. Mishra, *Analytica Chimica Acta*, 2022, **1202**, 339668.
 - 33 A. Kensert, E. Bosten, G. Collaerts, K. Efthymiadis, P. Van Broeck, G. Desmet and D. Cabooter, *Journal of Chromatography A*, 2022, **1672**, 463005.
 - 34 J. Zeng, H. Wu and M. He, *Analytica Chimica Acta*, 2023, **1238**, 340189.
 - 35 A. D. Melnikov, Y. P. Tsentalovich and V. V. Yanshole, *Analytical chemistry*, 2019, **92**, 588–592.
 - 36 C.-C. Lee, Y.-C. Lin, T. Y. Pan, C. H. Yang, P.-H. Li, S. Y. Chen, J. J. Gao, C. Yang, L. J. Chu, P.-J. Huang *et al.*, *Analytical Chemistry*, 2023, **95**, 15486–15496.
 - 37 H. S. Alghamdi, A. Lisitsa, I. Barsukov and R. Grosman, *ICAART (3)*, 2023, pp. 940–947.
 - 38 P. Klukowski, M. Augoff, M. Zięba, M. Drwal, A. Gonczarek and M. J. Walczak, *Bioinformatics*, 2018, **34**, 2590–2597.
 - 39 A. B. Risum and R. Bro, *Talanta*, 2019, **204**, 255–260.
 - 40 D. Punjabi, Y.-C. Huang, L. Holzhauer, P. Tremouilhac, P. Friederich, N. Jung and S. Bräse, *Journal of Cheminformatics*, 2025, **17**, 1–13.
 - 41 J. Q. Glenn Jocher and C. A., *Ultralytics YOLO11*, 2023, <https://github.com/ultralytics/ultralytics>.
 - 42 N. I. of Advanced Industrial Science and Technology, *Spectral Database for Organic Compounds (SDBS)*, <https://sdb.sdb.aist.go.jp/>, Accessed: 2025-10-12.
 - 43 Riverbank Computing Limited, *PyQt5*, [urlhttps://www.riverbankcomputing.com/software/pyqt/](https://www.riverbankcomputing.com/software/pyqt/), 2025.
 - 44 U. Garcilazo-Cruz, J. O. Okeme and R. A. Vargas-Hernández, *Digital Discovery*, 2026, **5**, 835–843.
 - 45 A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, *arXiv preprint arXiv:2010.11929*, 2020.
 - 46 I. B. Akkaya, S. S. Kathiresan, E. Arani and B. Zonooz, *Pattern Recognition*, 2024, **153**, 110510.
 - 47 H. W. Kuhn, *Naval research logistics quarterly*, 1955, **2**, 83–97.
 - 48 G. Lee, H. Shim, J. Cho and S.-I. Choi, *ACS Omega*, 2025, **10**, 12717–12723.
 - 49 Y. Kong, *ACS Omega*, 2026, **11**, 22369–22378.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [si.pdf](#)