analytical chemistry

pubs.acs.org/ac Article

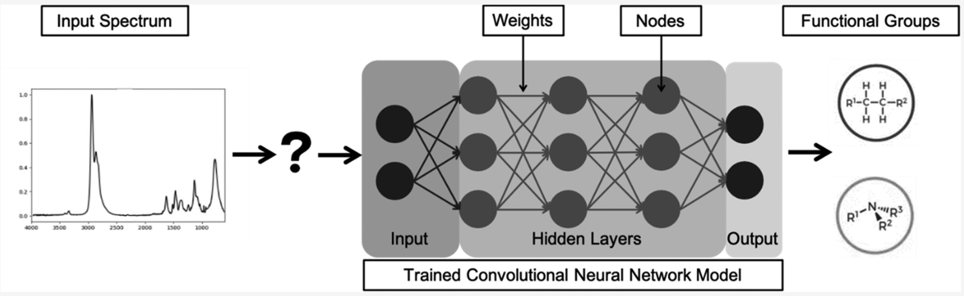
Functional Group Identification for FTIR Spectra Using Image-Based Machine Learning Models

Abigail A. Enders, Nicole M. North, Chase M. Fensore, Juan Velez-Alvarez, and Heather C. Allen*

Cite This: *Anal. Chem.* 2021, 93, 9711−9718

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information



ABSTRACT: Fourier transform infrared spectroscopy (FTIR) is a ubiquitous spectroscopic technique. Spectral interpretation is a time-consuming process, but it yields important information about functional groups present in compounds and in complex substances. We develop a generalizable model via a machine learning (ML) algorithm using convolutional neural networks (CNNs) to identify the presence of functional groups in gas-phase FTIR spectra. The ML models reduce the amount of time required to analyze functional groups and facilitate interpretation of FTIR spectra. Through web scraping, we acquire intensity-frequency data from 8728 gas-phase organic molecules within the NIST spectral database and transform the data into spectral images. We successfully train models for 15 of the most common organic functional groups, which we then determine via identification from previously untrained spectra. These models serve to expand the application of FTIR measurements for facile analysis of organic samples. Our approach was done such that we have broad functional group models that infer in tandem to provide full interpretation of a spectrum. We present the first implementation of ML using image-based CNNs for predicting functional groups from a spectroscopic method.

■ INTRODUCTION

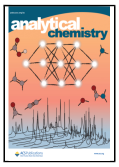
The anthropogenic impact on the climate and environment has prompted the analysis and detection of pollutants or contaminants with Fourier transform infrared spectroscopy (FTIR), such as microplastics in waters[1,2] and table salts,[3] nitrates from agricultural fertilizers in soil,[4−6] and polyaromatic hydrocarbons in the ocean's surface.[7,8] The diversity of the chemical composition of the pollutants and the central fundamental technique of FTIR underscores the importance of a computational method for improved throughput of spectral analysis. The bottleneck is most frequently the assignment of peaks to relevant functional groups.[9,10]

Functional groups describe and define the physical and chemical properties of compounds.[11,12] Identification of many organic groups is accomplished via FTIR due to the associated unique vibrational frequencies.[13,14] Large numbers of spectra are time consuming to analyze and require expert chemist analysis to determine present composition. This limits the application of FTIR spectral techniques as a sampling method for functional group elucidation. There is thus an unexplored, yet applicable field of FTIR spectral interpretation through statistical methods. Progress toward machine learning (ML) methods for environmental pollutant analysis has been explored for specific, targeted applications.[9,15−17] Generalizable functional group ML models would increase the utility of FTIR sample screening in environmental and other chemistry applications.[18,19]

In this study, we investigate the implementation of convolutional neural networks (CNNs)[20] to identify functional groups present in FTIR spectra. By limiting spectral preprocessing, we explore a minimalistic approach to allow the network to learn spectral patterns for successful recognition of the 15 most common organic functional groups (Table 1).

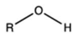
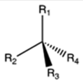
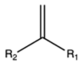
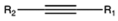
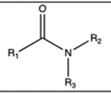
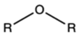
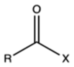
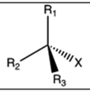
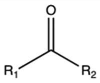
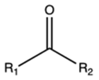
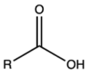
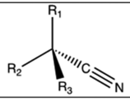
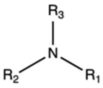
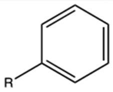
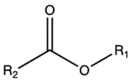
ML serves to address a need for quick identification of spectral components.[21] To date, the use of a CNN to broadly classify functional groups has not been reported. CNNs work by having layers of nodes called neurons; these neurons can be

Received: February 25, 2021
Accepted: June 18, 2021
Published: June 30, 2021



https://doi.org/10.1021/acs.analchem.1c00867

Table 1. Functional Groups for Which Successful Models Were Trained

alcohol	alkane	alkene	alkyne	amide
				
ether	acyl halide	alkyl halide	methyl	ketone
				
carboxylic acid	nitrile	amine	aromatic	ester
				

trained on data to identify spectral components that were observed in the training data in new spectra. The algorithm works to minimize a loss function; this is done by comparing answers given by the CNN to the true answers from a training data set. The difference between the reported and the true presence of a group constitutes the loss function. The training data set is a randomly segmented subset of spectral images that the CNN uses to learn and adjust neuron weights.

CNNs expand upon artificial neural networks (ANNs) by using mathematical convolutions to provide convolved data to the following neuron. Each neuron has a receptive field for which it convolves the information, similar to how a human brain has regions of neurons designated for processing specific information.[22] CNNs significantly reduce the number of neurons per pixel that a traditional feed-forward network requires to capture the complexity of an image. Thus, CNNs are a sophisticated solution to the alternative complex network required to machine learn images by capturing the spatial and temporal uniqueness of images. FTIR spectra offer unique "images" to evaluate using the sophisticated ML advancements.

We probe the effectiveness of image recognition ML as a facile solution to FTIR spectral interpretation. The information contained in a spectrum is most often presented to a chemist as a two-dimensional (2D) image; therefore, it is desirable to develop models that learn via similar spectral visualization.[23] Determining functional groups present in spectra requires analysis of both peak location on the frequency axis and shape; training models of spectra as images allows for an elegant approach that utilizes the totality of information obtained from a spectrum. Previous implementations of FTIR ML for functional group identification have limited,[24] averaged,[25] and segmented[24,26] spectral data to reduce information used during training. The computational resources available today make this an unnecessary and limiting feature. We include all available spectral data from 4000 to 600 cm^{-1} to reduce any biases in the learning process.

Current methods for spectral processing and interpretation are limited to library searching software[27] and highly specific questions using implementations of ML including support vector machines,[9,28] k-nearest neighbors,[29,30] and principal component analysis (PCA),[9,28,29] or factor analysis.[31] Library searching methods require a pre-existing and transferable database for searching spectra. The initial creation of libraries requires an intensive endeavor for collecting a large enough spectral repository. Once implemented, libraries cannot extrapolate beyond the spectra included in the software. The size of libraries is not of significant concern for storage, but it is a cumbersome feature for application compatibility and relative use-to-memory consumption. ML does not require the transfer of training data to the user and can predict beyond the data used for training. The use of ML to resolve challenging implementations of FTIR spectra (e.g., extremely large data sets, continuous analysis) has become of interest as increased processing power makes it possible to train and infer (interpret unknown spectra) with complex algorithms.[32−35] However, these highly specific models are only applicable in the setting in which they are developed because the training is completed on a narrow range of examples. To increase the amount of available training spectra or improve further calculations, ML algorithms in tandem with molecular dynamics have been explored.[35−37]

Previous applications[38−40] of ML have employed data preprocessing prior to training with unsupervised ML methods, such as PCA,[28] which reduces the information in the training data. Spectral preprocessing is becoming an unnecessary component with the advances in ML. Data dimensionality reduction limits the transferability of the final model to broader applications. Deep learning results in feature extraction within the model before and during learning. ML requires any feature extraction (e.g., selecting peaks of interest) to be completed by a user (manually or automatically) before training the algorithm. Results from recent studies identify little variability in prediction success between preprocessed data (e.g., removing wavenumber regions, derivative spectra, and components resulting from PCA) and raw data pipelined to sophisticated ML methods.[41,42] A recent application of ML successfully implemented broader methods for functional group analysis; however, the authors utilize a multilayer perceptron ML method with an autoencoder and train using two sources of data: FTIR and MS spectra.[43] Implementing methods such as selecting spectral regions of interest[24] can result in learning becoming memorization by the model; rather than making a generalizable algorithm that can be inferred on novel spectra, the model overfits the training data. An overfit model does well on spectra it has seen before but performs poorly on new data. Showing select data based on human evaluation increases the time required by an expert and introduces bias. While there are regions of relative disinterest to the chemist, it is not sufficient to ignore them in training. The absence of a peak is equally as informative as the presence of another.

In our work, we create separate functional group models that are executed simultaneously resulting in complete analysis of FTIR spectra. We obtain spectra from the NIST Chemistry

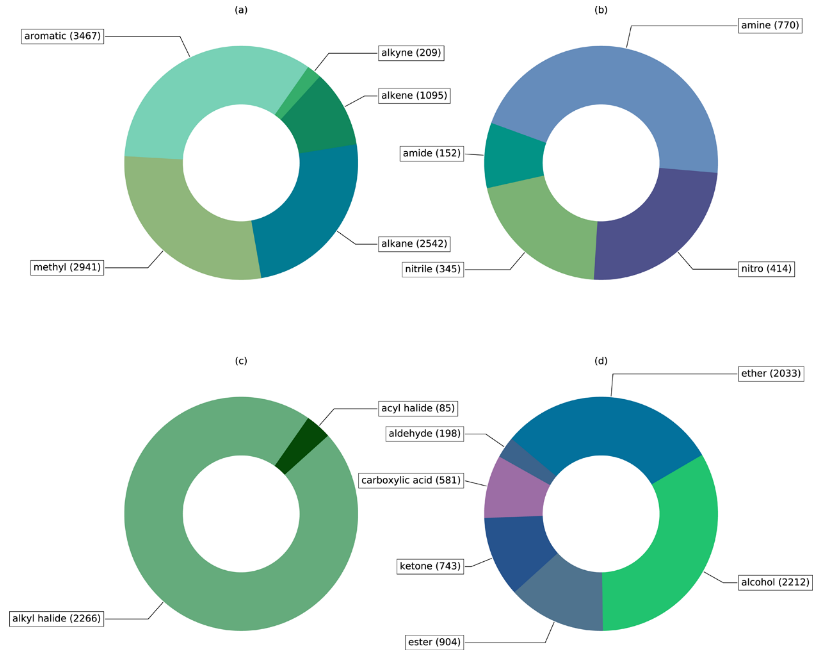


Figure 1. Number of spectra used to train each functional group model: (a) carbon-containing, (b) nitrogen-containing, (c) halide-containing, and (d) oxygen-containing. The number of images is equivalent for the positive and negative cases used in training and testing.

Webbook; peaks are not labeled in this data set. The use of individual functional group models presents a robust approach to establish a broad but precise computational analysis of spectra. Training a model for each functional group improves the overall accuracy attainable because each model is focused on a binary question: is this functional group present? The training of individual models does not impede the speed of spectrum analysis achieved and results are provided succinctly. By approaching the classification of spectra via the proposed method, we reduce the likelihood that the model learns a connection between functional groups that is not chemically relevant. In other words, one present functional group does not indicate another group's presence or absence. Individually trained models reduce the potential for this and improve the overall accuracy by posing a simplified question. Here, we develop effective and accurate FTIR ML models that apply to broader questions, limit spectral preprocessing, and provide the entire spectrum to the algorithm.

■ METHODS

Python Scripts. All Python scripts can be accessed from our repository at this address: https://github.com/Ohio-State-Allen-Lab/FTIRMachineLearning. The FTIR spectra are the property of NIST and can be accessed through their website. The implementation of Inception V3[44] is modified for our use, and the original source is linked on our repository with the published modified version. The computational procedure is described in detail in the Supporting Information and is documented in each Python script.

Spectra Collection. Data was obtained from the National Institute for Science and Technology Chemistry WebBook via a web scraping implementation in Selenium using the CAS number identifier from the official list of compounds in the WebBook.[45] When a compound had an FTIR spectrum, the file, in jcamp-dx format, was retrieved and stored with the CAS number as the filename. A total of 8728 spectra from pure compounds in gas phase were obtained (Figure 1). Each spectrum's InChI key was saved in a collective text file.

Data Preprocessing. Only spectra in absorbance and wavenumbers were used for training models. Each spectrum was evaluated to ensure it was in absorbance and wavenumbers via a Python script. Files in transmission or wavelength were relocated to a distinct directory to preserve all spectra obtained from web scraping. Files in the correct mode were converted from jcamp-dx to csv. Once converted, each spectrum was normalized so that the maximum peak height was 1. Normalized spectra were saved as jpg images.

Labeling. Functional groups were identified via the InChI key. Using SMARTS functional group identifiers, each spectrum's key was parsed to return binary indicators. Present functional groups are labeled as "1" and absent as "0". Results were saved in one spreadsheet with CAS numbers as spectrum and file identifiers. Spectra were copied into directories based on the presence or absence of a functional group. This method allows one compound with multiple functional groups present to be copied into the directory for each group. Each of the 17 functional groups had two directories: positive and negative cases. Positive cases include the functional group and negative cases do not contain the group. Randomly, 10 photos, 5 from positive and negative, for each functional group were reserved for validation. Then, the directory containing more instances for a given group was reduced randomly until both directories contained the same number of spectra.

Machine Learning. A convolutional neural network (CNN) for image recognition was employed. A unique model was trained for each functional group containing two classes. The functional groups and the number of images in the positive cases are presented in the Supporting Information. The architecture, Inception V3, was accessed from the available models on the Google TensorFlow library. Each model was trained for 10 000 epochs at a learning rate of 0.01, using an initialized version of Inception V3 and training the last layer of the model graph (Figure 2).

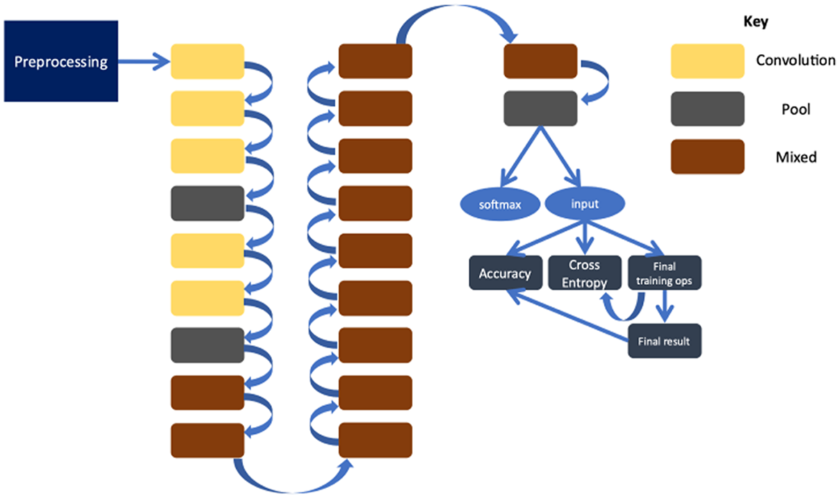


Figure 2. General summary of the Inception V3 architecture. Additional details are provided in the Supporting Information, including a summary of the model preprocessing parameters.

Inception V3 employs several techniques that ultimately lead to an increase in accuracy in the final model results. The model begins with preprocessing the input spectra, which includes decoding and reducing spectra to 299 × 299 × 3 pixels. The RMSProp optimizer results in the greatest accuracy. The specific equations describing the available optimizers are provided (Supporting Information). As a postoptimization step, exponential moving average is employed (Supporting Information). Batch normalization aids in reducing the time to convergence and occurs after convolutions in Inception V3; specific details, including the equations used, are provided (Supporting Information). Additionally, learning rate adaptation is utilized to efficiently train the algorithms. Using gradual learning rate ramp-up, the initial learning rate for the model is ~10% of the defined rate; after initializing, the rate is linearly increased until the slope of the decay rate intersects with the theoretically defined exponential decay rate. More specific details are included in the Supporting Information.

Models are trained and validated using an 80/20 split. For example, if 100 spectra are provided, 80 are used for training and 20 are used for validation. The preprocessing methods include distortion of the spectra such that after each backpropagation, the 80 spectra are altered and are not identical between iterations. Validation spectra are used to guide backpropagation but do not affect the training directly. The final train and validation accuracies presented are from the described source. After the models converge, the reserved 10 spectra are used to test the models further. These spectra are not used to adjust the model and present a more accurate representation of the model accuracy beyond the current data set.

Parameters are initialized to reduce the time and computational power required to train a custom model. Models converge within the 10 000 training steps, and early termination of training is not employed. It took 5 h to train the 15 models. Classification of an unknown spectrum requires 1 min.

Accuracy and Loss. Accuracy and cross entropy (loss) for both training and test models were obtained and saved as csv files. The final accuracies and entropies for training and test results from each model are investigated to identify any anomalies.

Classification of Validation Data. When spectra were classified, the models were called upon to infer (determine the functional groups present) and a final result was provided. Each functional group model was trained and validated separately, and the predictions were not used in conjunction to attempt ensemble classification. Models were evaluated independently of each other and from the embedded validation methods to further analyze prediction accuracy. The 10 reserved spectra were analyzed via the respective models they were withheld from to examine the learning quality of the algorithm. Confusion matrices[46] were used to represent the true and predicted functional group for the 17 models.

RESULTS AND DISCUSSION

Using ~9000 gas-phase FTIR spectra, we train 17 functional group models using image-based ML. Our methods result in 15 effectively identified functional groups (Table 2). Each

Table 2. Final Accuracy and Cross Entropy for Train and Validation of Each Functional Group Model Are Presented in Order of Increasing Number of Training Images

	accuracy		cross entropy	
	train (%)	validation (%)	train	validation
acyl halide	100	98	0.025347	0.143665
amide	100	70	0.060037	0.900095
aldehyde	100	80	0.04735	0.385665
alkyne	99	80	0.079596	0.332745
nitrile	97	65	0.17019	0.668148
nitro	98	89	0.12624	0.668148
carboxylic acid	98	98	0.070173	0.076216
ketone	93	76	0.228837	0.501178
amine	93	80	0.24136	0.494815
ester	97	83	0.111057	0.323917
alkene	85	68	0.407803	0.743543
ether	89	81	0.27397	0.443644
alcohol	90	84	0.236544	0.330591
alkyl halide	85	73	0.350733	0.531302
alkane	85	90	0.327755	0.265718
methyl	81	84	0.384048	0.358021
aromatic	92	89	0.199645	0.259044

model is trained independently. When a molecule has more than one functional group present, the models for the relevant functional group will identify the presence from the spectrum, resulting in a complete analysis for the 15 trained functional groups. Accuracy and cross entropy results from the last step of training are reported for the train and validation process. The two functional group models that underperform are aldehyde and nitrile based on model prediction of untrained spectra (as seen in the Supporting Information). We define underperforming as misidentifying more than 60% of test cases. The training accuracy is a measure of how well the model classifies the training data, which is used to train the network. A higher training accuracy indicates that the model is learning the training spectra. Validation accuracy expresses the ability of the model to generalize to untrained spectra, which is determined by the number of correctly classified validation spectra. Thus, it is more meaningful to have a higher validation accuracy, albeit not a requirement for a successful inferencing model. Cross entropy is the loss function used to evaluate the final model and is defined as the logarithm of the likelihood of a correct assignment. Smaller cross entropy values indicate that a model is well trained. We observe that cross entropy for training is less than validation. Models are more likely to correctly infer spectra that have been used to train and adjust weights, in comparison to the validation spectra.

A confusion matrix for each model was created by using spectra that have been withheld from training and testing data (as seen in the Supporting Information). A confusion matrix compares model assignments to the actual identities of the samples; it shows correct assignments along the trace of a matrix and false assignments off of the trace. Four models have perfect confusion matrices from the classification of 10 withheld images, five containing and not containing functional group spectra examples. The presence or absence of carboxylic acid, aromatic, methyl, and ester functional groups is correctly identified in the withheld spectra (Figure 3).

The number of instances of each functional group occurring in the spectra varies significantly, with aromatic-containing spectra occurring most frequently with 3467 images. In contrast, acyl halide has 85 spectra for training and testing the model. We explored the relationship between the number of images and the cross entropy and accuracy for training and testing results (Figure 4). Training accuracy decreases with increasing number of spectra (Table 3 and Supporting Information). However, the final accuracy, determined by evaluating the unknown spectra for functional group identification, is not correlated to the number of images used for training. Our results indicate that the total number of training spectra does not affect the final performance of the models. The scattered, nonuniformity exhibited in Figure 4a,b depicts the deviation from a linear relationship between the number of spectra and accuracy and cross entropy for validation, confirming that the number of images is not influencing the performance of the models. Training accuracy provides insight into how well the model has learned the training images for a functional group model. Counterintuitively, few spectra being trained for a functional group will result in a higher training accuracy because the model trains on the same spectra more frequently. This model memorizes or overfits functional groups resulting in a model incapable of extrapolating to new spectra.

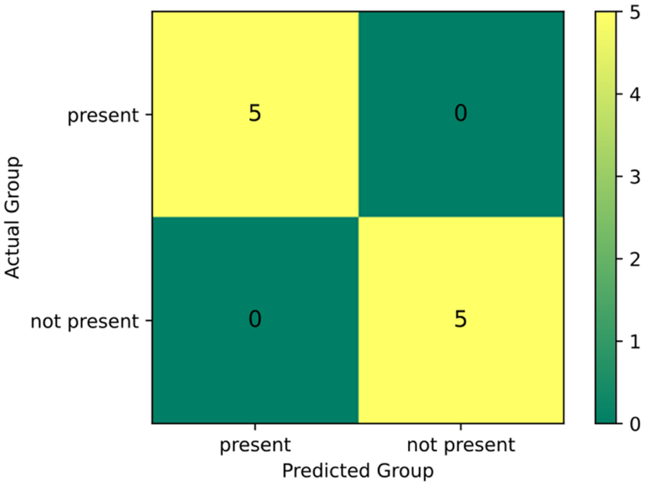


Figure 3. Confusion matrix is identical for carboxylic acid, aromatic, methyl, and ether functional group models.

However, from our results, the challenges of limited training spectra do not result in less accurate models. We confirmed this by investigating the relationship of the number of images per class as a function of validation accuracy and cross entropy (Table 2). Models that have more spectra to train on have lower overall training accuracy but still perform well when analyzing unknown spectra. To investigate the linear correlation between the number of spectra used for training and the final accuracy and cross entropy, the Pearson's correlation coefficient is used (Figure 3). More linearly correlated relationships have a coefficient closer to one, where positive coefficients indicate a positive correlation and negative coefficients indicate a negative correlation. The coefficient for training accuracy and number of training spectra indicates that they are indirectly correlated, whereas the coefficient for training cross entropy and number of training spectra is positive or positively correlated. However, the models with less training spectra show no correlation between the final accuracy and ability to classify unknown spectra. Furthermore, both validation accuracy and cross entropy do not have significant linear correlation. While training results display correlation with the number of images, the validation data indicates that models are successful with a range of number of training spectra.

We can determine some of the underlying shortcomings in the model, from both spectroscopic and computational perspectives, by investigating two functional groups: aldehyde and carboxylic acid. The model results for aldehyde are promising for the training data but do not perform as effectively in validation and testing (Table 2 and Supporting Information). The confusion matrix for carboxylic acid describes how well the model performs on spectra that have not been used for training or validation. We observe that the IR mode frequencies for the carboxylic acid and aldehyde affect the performance of the model, in addition to the number of spectral examples available for training and validation.

Aldehyde C−H stretching frequency (2830−2695 cm^{-1}) is commonly overlapping in organic spectra with other C−H bonds because it is a weaker mode (Table 3). The carbonyl stretch is also frequently unresolved in compounds that contain multiple oxygen atoms. The C−H bending mode is often weak, in addition to being in the fingerprint region, which is a challenge to interpret due to the complexity. With these stretching and bending modes considered, it is

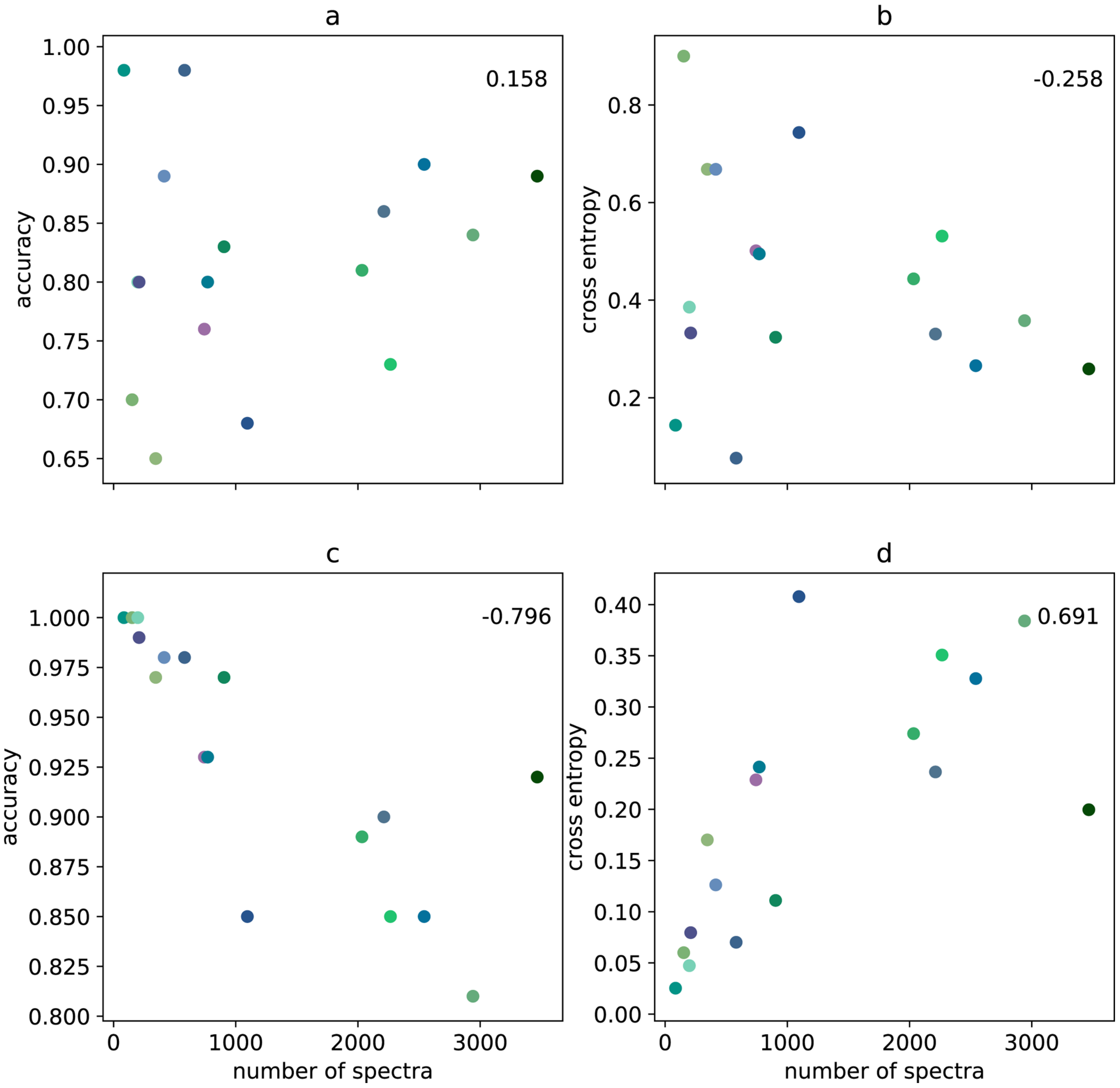


Figure 4. Final train and validation accuracy and cross entropy as a function of the number of spectra used to train each functional group: (a, b) validation results and (c, d) training results. Pearson's correlation coefficients (PCCs) are inset in the plots for final accuracy and cross entropy of training and validation as a function of the number of spectra. The coefficients closer to ±1 indicate that the train accuracy and cross entropy are linearly correlated (negative is inversely correlated and positive is directly correlated) to the number of spectra used in training. Validation accuracy and cross entropy are not linearly correlated to the spectral examples used.

Table 3. Aldehyde and Carboxylic Acid IR Stretching and Bending Mode Frequencies[47]

	mode	frequency (cm^{-1})	appearance
aldehyde	C−H stretch	2830−2695	weak, medium
	C=O stretch	1740−1720	medium, strong
	C−H bend	1390−1380	weak, medium
carboxylic acid	O−H stretch	3300−2500	strong, broad
	C=O stretch	1760	strong
	O−H bend	1440−1395	medium

reasonable to anticipate that an aldehyde functional group is challenging for the model to identify in spectra. In comparison, the carboxylic acid functional groups are always correctly identified in spectra by the model. The model for carboxylic acids is well trained. As observed by the validation accuracy and cross entropy (Table 2), the carboxylic acid model has a more robust transferability to spectra it has never observed. We confirm the effectiveness of the model with a correct assignment of unknown spectra. In totality, there are more carboxylic acid training spectra, and the IR modes are better resolved, especially the strong COO−H stretching, in comparison to aldehydes (Table 3).

We do not specifically probe the temperature and pressure dependences of model success. However, we hypothesize that the models would perform at high accuracy because pressure effects have been shown to have minimal effect on IR response.[48] In general, the IR transition moment strength for hydrocarbon bonds decreases with increasing temperature and this may affect model accuracy.

From our results, we observe that the models are more accurate for functional groups when there are more training spectra examples for the functional group and IR peaks are well resolved. Albeit this is an intuitive result for a trained spectroscopist with respect to accuracy correlating to peak resolution, yet there is no precedent using a machine learning approach.

■ CONCLUSIONS

We present a novel method for FTIR spectral interpretation using CNNs and the NIST database. Fifteen functional group models successfully and effectively classify unknown spectra in a facile method for spectral submission to interpretation. We find that the image recognition features inherent in CNNs are transferable to a chemical-identification application. From our observations, we can conclude that CNNs are effective at identifying spectral features for classification and generalizable models are achievable with ample spectral examples. In future work, optimization for functional group identification with less spectral examples should be investigated to improve accuracy. Further investigation of the models could include determining the ability of the model to predict binary and higher-order mixture compositions as well as shifts along the frequency axis.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.analchem.1c00867.

> Detailed computational methods; number of spectra per functional group; confusion matrices; final model train and validation cross entropy and accuracy; machine learning model architecture; optimization techniques; exponential moving average model parameters; batch normalization methods; and learning rate optimization methods (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Heather C. Allen − *Department of Chemistry & Biochemistry, The Ohio State University, Columbus, Ohio 43210, United States;* orcid.org/0000-0003-3120-6784; Phone: +1-614-292-4707; Email: allen.697@osu.edu, allen@chemistry.ohio-state.edu; Fax: +1-614-292-1685

Authors

Abigail A. Enders − *Department of Chemistry & Biochemistry, The Ohio State University, Columbus, Ohio 43210, United States*

Nicole M. North − *Department of Chemistry & Biochemistry, The Ohio State University, Columbus, Ohio 43210, United States*

Chase M. Fensore − *Department of Chemistry & Biochemistry, The Ohio State University, Columbus, Ohio 43210, United States*

Juan Velez-Alvarez − *Department of Chemistry & Biochemistry, The Ohio State University, Columbus, Ohio 43210, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.analchem.1c00867

Author Contributions

A.A.E.: conducted experiments and writing; N.M.N.: contributed to code and editing; C.M.F.: wrote Selenium web scraping script; J.V.-A.: intellectual contributions; and H.C.A.: intellectual contributions.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors acknowledge R. L. Enders for helpful discussions. The authors acknowledge support by NSF through the Center for Aerosol Impacts on Chemistry of the Environment (CAICE), CHE-1801971. Data archiving is underway at the UC San Diego Library Digital Collections for CAICE website: https://doi.org/10.6075/J0RR1WRV. All Python scripts can be accessed from our repository at this address: https://github.com/Ohio-State-Allen-Lab/FTIRMachineLearning.

■ REFERENCES

(1) Yonkos, L. T.; Friedel, E. A.; Perez-Reyes, A. C.; Ghosal, S.; Arthur, C. D. *Environ. Sci. Technol.* **2014**, *48*, 14195−14202.
(2) Song, Y. K.; Hong, S. H.; Jang, M.; Kang, J. H.; Kwon, O. Y.; Han, G. M.; Shim, W. J. *Environ. Sci. Technol.* **2014**, *48*, 9014−9021.
(3) Lee, H. J.; Song, N. S.; Kim, J. S.; Kim, S. K. *J. Hazard. Mater.* **2021**, *402*, No. 123743.
(4) Bogard, J. S.; Johnson, S. A.; Kumar, R.; Cunningham, P. T. *Environ. Sci. Technol.* **1982**, *16*, 136−140.
(5) Anıl, I.; Golcuk, K.; Karaca, F. *Water, Air, Soil Pollut.* **2014**, *225*, No. 1898.
(6) Linker, R.; Shmulevich, I.; Kenny, A.; Shaviv, A. *Chemosphere* **2005**, *61*, 652−658.
(7) Hardy, J. T. *Prog. Oceanogr.* **1982**, *11*, 307−328.
(8) Wurl, O.; Obbard, J. P. *Mar. Pollut. Bull.* **2004**, *48*, 1016−1030.
(9) Michel, A. P. M.; Morrison, A. E.; Preston, V. L.; Marx, C. T.; Colson, B. C.; White, H. K. *Environ. Sci. Technol.* **2020**, *54*, 10630−10637.
(10) Rezania, S.; Park, J.; Md Din, M. F.; Mat Taib, S.; Talaiekhozani, A.; Kumar Yadav, K.; Kamyab, H. *Mar. Pollut. Bull.* **2018**, *133*, 191−208.
(11) Carey, F. A.; Giuliano, R. M. *Organic Chemistry*, 10th ed.; McGraw-Hill Education, 2016.
(12) Carey, F. A.; Sundberg, R. J. *Advanced Organic Chemistry. Part A: Structure and Mechanisms*, 5th ed.; Springer: US, 2007.
(13) Barnes, R. B.; Gore, R. C. *Anal. Chem.* **1949**, *21*, 7−12.
(14) Peck, R. L. *Anal. Chem.* **1950**, *22*, 121−126.
(15) Kedzierski, M.; Falcou-Préfol, M.; Kerros, M. E.; Henry, M.; Pedrotti, M. L.; Bruzaud, S. *Chemosphere* **2019**, *234*, 242−251.
(16) Galimberti, D. R.; Bougueroua, S.; Mahé, J.; Tommasini, M.; Rijs, A. M.; Gaigeot, M. P. *Faraday Discuss.* **2019**, *217*, 67−97.
(17) Bougueroua, S.; Spezia, R.; Pezzotti, S.; Vial, S.; Quessette, F.; Barth, D.; Gaigeot, M. P. *J. Chem. Phys.* **2018**, *149*, No. 184102.
(18) Coe, J. V.; Nystrom, S. V.; Chen, Z.; Li, R.; Verreault, D.; Hitchcock, C. L.; Martin, E. W.; Allen, H. C. *J. Phys. Chem. B* **2015**, *119*, 13079−13092.
(19) Geiger, A.; Cao, Z.; Song, Z.; Ulcickas, J.; Simpson, G. Autonomous Science: Big Data Tools for Small Data Problems in Chemistry. In *Machine Learning in Chemistry: The Impact of Artificial Intelligence*; Royal Society of Chemistry, 2020; Chapter 18, pp 450−487.

(20) Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. In *Rethinking the Inception Architecture for Computer Vision*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

(21) Nalla, R.; Pinge, R.; Narwaria, M.; Chaudhury, B. In *Priority Based Functional Group Identification of Organic Molecules Using Machine Learning*. Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, 2018; pp 201−209.

(22) Yang, J.; Xu, J.; Zhang, X.; Wu, C.; Lin, T.; Ying, Y. *Anal. Chim. Acta* **2019**, *1081*, 6−17.

(23) Kowalski, B. R.; Jurs, P. C.; Isenhour, T. L.; Reilley, C. N. *Anal. Chem.* **1969**, *41*, 1945−1949.

(24) Fessenden, R. J.; Györgyi, L. *J. Chem. Soc., Perkin Trans. 2* **1991**, 1755.

(25) van Est, Q. C.; Schoenmakers, P. J.; Smits, J. R. M.; Nijssen, W. P. M. *Vib. Spectrosc.* **1993**, *4*, 263−272.

(26) Wu, W.; Massart, D. L. *Chemom. Intell. Lab. Syst.* **1996**, *35*, 127−135.

(27) Averill, D. F.; Baird, K. C.; Hopkins, L. L.; Yerkes, M. J. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 133−136.

(28) dos Santos, V. H. J. M.; Do Canto Bruzza, E.; de Lima, J. E.; Lourega, R. V.; Rodrigues, L. F. *Energy Fuels* **2016**, *30*, 4905−4915.

(29) Judge, K.; W Brown, C.; Hamel, L. *Anal. Chem.* **2008**, *80*, 4186−4192.

(30) Renner, G.; Schmidt, T. C.; Schram, J. *Anal. Chem.* **2017**, *89*, 12045−12053.

(31) Golz, E. K.; Vander Griend, D. A. *Anal. Chem.* **2013**, *85*, 1240−1246.

(32) Agbaria, A. H.; Beck Rosen, G.; Lapidot, I.; Rich, D. H.; Huleihel, M.; Mordechai, S.; Salman, A.; Kapelushnik, J. *Anal. Chem.* **2018**, *90*, 7888−7895.

(33) Sharaha, U.; Rodriguez-Diaz, E.; Sagi, O.; Riesenberg, K.; Lapidot, I.; Segal, Y.; Bigio, I. J.; Huleihel, M.; Salman, A. *Anal. Chem.* **2019**, *91*, 2525−2530.

(34) Kedzierski, M.; Falcou-Préfol, M.; Kerros, M. E.; Henry, M.; Pedrotti, M. L.; Bruzaud, S. *Chemosphere* **2019**, *234*, 242−251.

(35) Kananenka, A. A.; Yao, K.; Corcelli, S. A.; Skinner, J. L. *J. Chem. Theory Comput.* **2019**, *15*, 6850−6858.

(36) Morawietz, T.; Urbina, A. S.; Wise, P. K.; Wu, X.; Lu, W.; Ben-Amotz, D.; Markland, T. E. *J. Phys. Chem. Lett.* **2019**, *10*, 6067−6073.

(37) Zhai, Y.; Caruso, A.; Gao, S.; Paesani, F. *J. Chem. Phys.* **2020**, *152*, No. 144103.

(38) Doblies, A.; Boll, B.; Fiedler, B. *Polymers* **2019**, *11*, No. 363.

(39) Lasch, P.; Stämmler, M.; Zhang, M.; Baranska, M.; Bosch, A.; Majzner, K. *Anal. Chem.* **2018**, *90*, 8896−8904.

(40) Lasch, P.; Diem, M.; Hänsch, W.; Naumann, D. *J. Chemom.* **2006**, *20*, 209−220.

(41) Takamura, A.; Halamkova, L.; Ozawa, T.; Lednev, I. K. *Anal. Chem.* **2019**, *91*, 6288−6295.

(42) Malek, S.; Melgani, F.; Bazi, Y. *J. Chemom.* **2018**, *32*, No. e2977.

(43) Fine, J. A.; Rajasekar, A. A.; Jethava, K. P.; Chopra, G. *Chem. Sci.* **2020**, *11*, 4618−4630.

(44) Advanced Guide to Inception v3 on Cloud TPU, 2015. https://cloud.google.com/tpu/docs/inception-v3-advanced.

(45) Linstrom, P. J.; Mallard, W. G. *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*, 2018.

(46) Stehman, S. V. *Remote Sens. Enivorn.* **1997**, *62*, 77−89.

(47) Wade, L.; Simek, J. *Organic Chemistry*, 9th ed.; Pearson, 2016.

(48) Heath, C.; Myers, M.; Pejcic, B. *Anal. Chem.* **2017**, *89*, 13391−13397.