Application of Deep Convolutional Neural Networks and IR Spectroscopy for the Detection of Drugs and Toxins

Gokul Mohanraj, Gagan Jain, Pratyush Agarwal, Vaibhavkumar Patel

Abstract: *This paper explores the use of deep learning architectures to identify and categorize infrared spectral data with the objective of classifying drugs and toxins with a high level of accuracy. The model proposed uses a custom convolutional neural network to learn the spectrum of 192 drugs and 207 toxins. Variations in the architecture and number of blocks were iterated to find the best possible fit. A real-time implementation of such a model faces a lot of issues such as noise from different sources, spectral magnitude off-setting, and wavelength rotation. This paper aims to tackle some of these problems. Another common issue is the use of extensive pre-processing which makes it difficult to automate the entire process. We have aimed to side-step this issue with the architecture proposed. The focus is on 2 applications - detection of drugs and toxins. The data sets used are from different sources, each with its own noise factor and sampling rate. Some of the traditional models like Principal Component Analysis (PCA) and Support Vector Machines (SVM) were also tested on the datasets. The model works with minimal input data of two spectra (and three augmentations of the same) to learn the features and classifies the data from a source independent of the input. The proposed model showed a significant improvement in accuracy when compared to the other models currently in use, achieving an overall accuracy of 96.55\%. The model proposed performs extremely well with a minimal sampling rate and shows no loss in accuracy of classification even with an increase in the number of classes. The research conducted has the scope of being extended to the identification of counterfeit drugs which is a growing cause for concern. Another application could be in the detection of the presence of harmful toxins.*

Keywords: Deep CNNs, Drugs and toxins detection, IR Spectroscopy, Spectral classification.

I. INTRODUCTION

Deep Learning and the use of predictive technology are completely transforming the way we process information. With the vast amount of data generated every second, the use of deep learning algorithms has seen a significant rise in the last few decades. Speech recognition, audio signal processing, and computer vision [1], [2] are some of the areas that have seen growing interest. Machine learning provides exceptionally good performance in areas where classical methods either fail to converge or prove to be computationally inefficient. Among the vast possible applications, one of the areas with a lot of ongoing research is Infrared (IR) spectroscopy. IR Spectroscopy is primarily used for the qualitative and quantitative estimation of organic compounds. Fourier Transform Infrared Spectroscopy (FTIR), which measures the absorbance/transmittance of the substance in the range of 4000 to 400cm^{-1} (wavenumber), has now made it possible to generate an IR spectrum of compounds with relative ease. Conventional methods to analyze such spectra, while accurate, are extremely time-consuming. Deep Learning has the scope to reduce the analysis time exponentially while maintaining the accuracy of results. Here we have focused mainly on two kinds of datasets, the toxins, and the drugs. Detection of counterfeit and substandard drugs is the need of the hour with an ever-increasing list of fake drugs and medicines on the market. Most of these drugs cannot be distinguished by their physical attributes like appearance, color and/or odor. Malaria, pneumonia, and the common illnesses affect more than 200,000 children every year, especially in poor countries where substandard and fake drugs run rampant. Another growing cause for concern in the agricultural sector is the increased use of pesticides and fertilizers. In the last few decades, crops are showing an increased presence of harmful toxins. Examination of the chemical composition of these substances, while accurate, is time consuming and requires equipment that is not readily available. Hence, there is a need for effective identification of the presence of these harmful compounds. IR spectrometers measure either of two values - absorbance or percentage of transmittance. Since the datasets we have used are from various sources, accordingly we have both kinds of data. We have applied the relation derived from Beer Lambert's Law for homogeneous samples which effectively neglects the reflectance. Hence the model is capable of handling both kinds of data by converting absorbance to transmittance [3] using the relation:

$$A = 2 - \log_{10} T$$

where A = absorbance and T = Percentage Transmission

Application of Deep Convolutional Neural Networks and IR Spectroscopy for the Detection of Drugs and Toxins

II. DATA COLLECTION AND PROCESSING

A. Dataset

A lot of previous research has used data from their own lab facilities and equipment for both train and test samples. As a result, the training and testing data used are quite similar.

Here, we have used data from three different sources, resulting in several differences in the data that includes sampling rate, and study domain. Our methods use the infrared spectrum of drugs and toxins collected from the publicly available repositories: SpectraBase [4] and National Institute of Standards and Technology (NIST), United States [5] for training. For testing the model, we have used the data collected from the Spectral Database for Organic Compounds (SDBS) by the National Institute of Advanced Industrial Science and Technology (AIST), Japan [6]. To digitize the spectra images, Web Plot Digitizer [7] is used. The process is automated using PyAutoGui-python module to avoid human error in axis calibration for different plots having same axis. For uniform spacing between data points, X Step with interpolation algorithm is chosen. For spectra images of SDBS dataset [6] having non-uniform X axis, digitization is carried out in parts. For the spectra of NIST dataset [5] in JCAMP-DX format, data points are extracted using python module-jcamp. While machine learning algorithms have demonstrated to be an essential approach to classify based on IR spectra in real time, it requires heavy pre-processing [14] of data. This is true, regardless of whether we use peak-detection or multivariate methods [15], [16] operating on whole spectra are taken in as input. The standard procedure to classify spectral data includes feature selection, smoothing and baseline correction followed by dimensionality reduction techniques like principal component analysis (PCA) [17]. A variety of automated baseline correction methods [18] were proposed like asymmetric least squares, penalized least squares and the rolling ball which require little or no human intervention. A detailed review of all baseline correction methods can be found at Schulze et al [19].
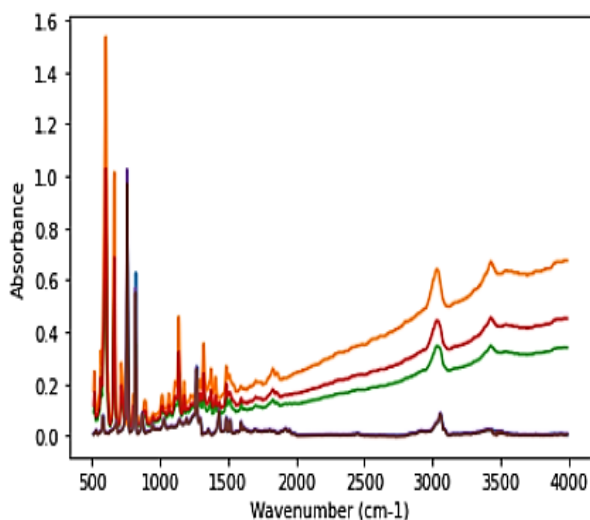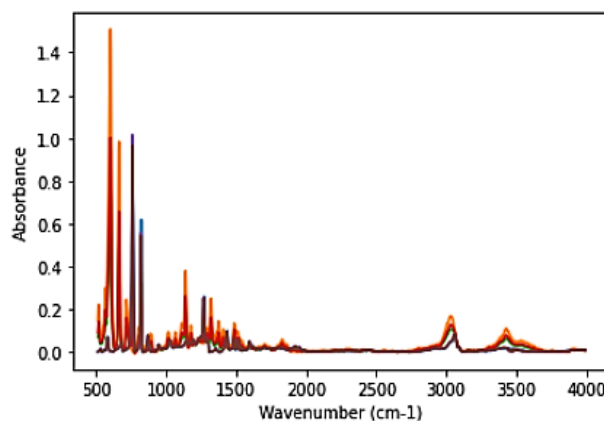

Fig (1). Spectra without Baseline correction

Fig (2). Spectra with baseline correction

III. METHODS

Inspired by a lot of previous research works, we initially used Principal Component Analysis (PCA) [13], [17] to classify the spectra. However, the results suggested that the method did not generalize well for many classes with limited training data. Another popular method used commonly for classification is Support Vector Machine (SVM). So next we tried an implementation of the same on our data. The results were quite much better than PCA but still had scope for improvement. So, we turned to Deep Learning based methods and implemented a one-dimensional Convolutional Neural Network for the task. One of the biggest challenges when it comes to implementing CNN networks, is that they are extremely data hungry. A generic deep learning model has million to tens of millions of parameters to train. Hence, the number of training samples can be exponentially large. To generate such a large set of data, many popular data augmentation techniques are used. Cropping, rotating, shearing, and zooming in/out are the most used techniques when it comes to image data. In case of numerical data with non-categorical variables, the popular data augmentation techniques include linear combinations and adding noise from a distribution (mostly Gaussian). Here, we have adopted linear combinations between the NIST dataset and the Bio-Rad dataset in three ratios 1:1, 1:2 and 2:1 (chosen at random). Since our end goal was to be able to detect these spectra in real time without any pre-processing, the raw spectral data was directly fed into the CNN model without any prior feature engineering. Additionally, to check the impact of number of features on model accuracy, we trained the drugs with 716 features while the toxins were trained on 1740 features, all in the range from $500cm^{-1}$ to $4000cm^{-1}$.

A. Principal Component Analysis (PCA)

This is the basic technique used to visualize and analyze data. The data for drugs had 716 features for each drug while that of toxins had 1740 features. PCA was used on the dataset extracted from Bio-Rad dataset. PCA works by projecting the data onto a new coordinate system by trying to maximize the variance along each axis.

The data was normalized and then the covariance matrix was computed. The eigenvectors of the Covariance Matrix were considered in the decreasing order of magnitude of corresponding eigenvalues as the axis of the new coordinate system. The number of components used were varied till about 98% of the variance was captured. As seen in the figure below, this was obtained taking 60 components for toxins and 30 components for drugs. The variance obtained on the dataset versus the number of components used for both the drugs as well as toxins are shown in the figures below. After applying PCA on the dataset, the scores for each drug were stored. The principal features were extracted, and these were used for testing (identification of drugs and toxins). The testing was performed on the SDBS dataset. For the testing process, the spectra to be identified was projected along the principal components and the scores were found out.
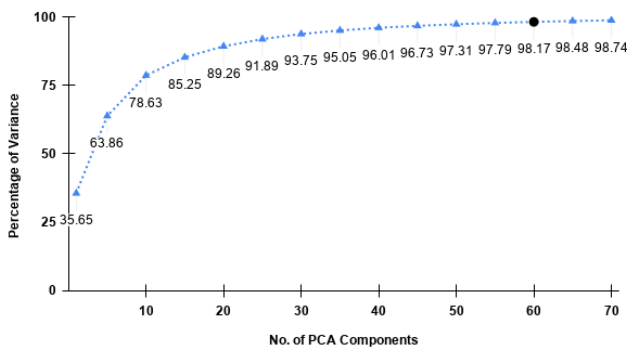

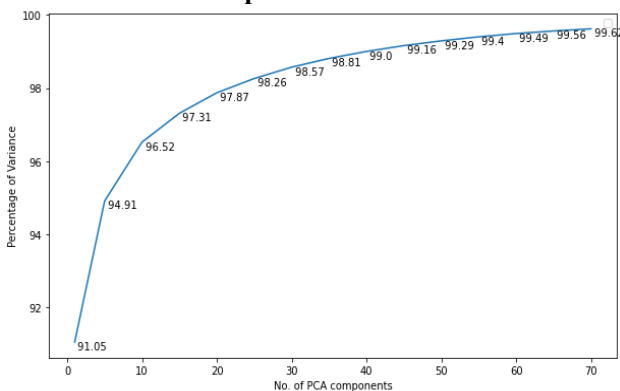**Fig (3). Variance captured versus No. of PCA. components for toxins.**

Fig (4). Variance captured versus No. of PCA. components for drugs.

Two vector norms were evaluated on the testing data, the L1 norm (or the Manhattan Norm) and the L2 norm (or the Euclidean Norm) and the predicted class of the drug was the label with the least norm. As seen from the plots below, the accuracy dropped almost exponentially as the number of classes of each category were increased. Hence, PCA proved to be inadequate for the task. The norm was evaluated according to the equation.

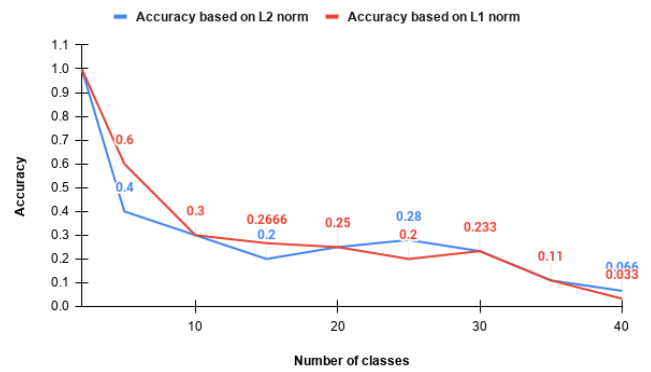$$L_n\ norm: \left(\sum_{i=1}^{m} |y_i - y_{pi}|^n\right)^{1/n}$$

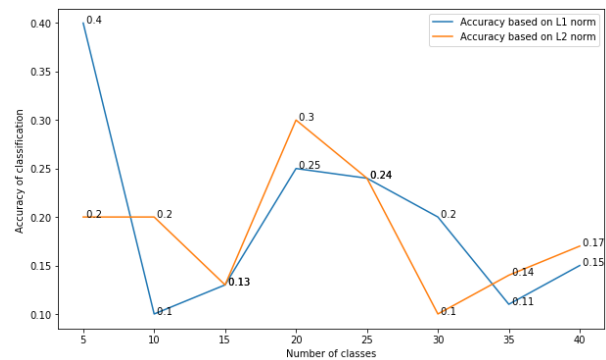**Fig (5). Accuracy on L1, L2 norm versus number of classes for toxins.**

Fig (6). Accuracy on L1, L2 norm versus number of classes for drugs.

B. Support Vector Machines (SVM)

Support Vector Machine [20], [11] is the most popular algorithm for classification of numerical data. The algorithm revolves around finding an optimal hyper-plane to separate the various classes. However, often, the data is not linearly separable. Thus, we applied Kernel transformations [21] to deal with non-linearity while saving up on computation. The four major kernels [22] in use for SVM are.

$$Linear: k(x_1, x_2) = x_1 . x_2$$
$$Polynomial: k(x_1, x_2) = (\gamma x_1 . x_2 + c)^d$$
$$Radial\ Basis: k(x_1, x_2) = \exp\left(-\gamma \|x_1 - x_2\|^2\right)$$
$$Sigmoid: k(x_1, x_2) = \tanh(\gamma x_1 . x_2 + c)$$

Kernel SVM models have plenty of hyper parameters to tune [24] to arrive at the optimal result. The kernels may be Linear, Radial Basis Function, Polynomial or Sigmoid. In case of Polynomials, there are two parameter the degree (d) and the off-set or bias term c. For Radial Basis Kernels, there is a γ which is nothing but $0.5\sigma^2$. Additionally, SVM has another hyper-parameter, the C factor which is like the regularization hyper-parameter. To arrive at the best fit, each of these parameters need to be tuned in tandem. So, we opted for a Grid Search [25] CV to find the optimum model. The C parameters were varied in the range (1,1000) on a logarithmic scale. The γ-factor was varied from $(1,1e^{-5})$. Both Radial Basis and Linear Kernels were experimented with.

The best fit appeared to be for a Linear Kernel with a C value of 1 for toxins and Linear Kernel with a C value of 1 for drugs.
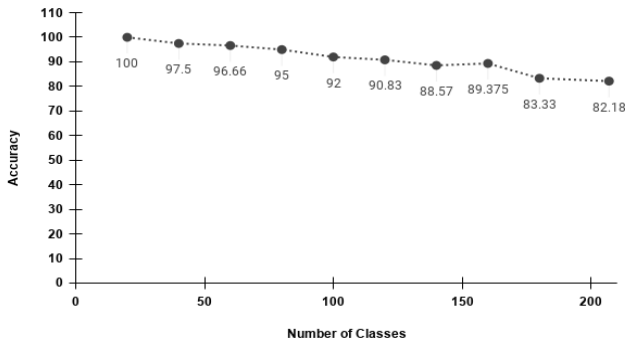

Fig (7). Accuracy of SVM versus number of classes for toxins.
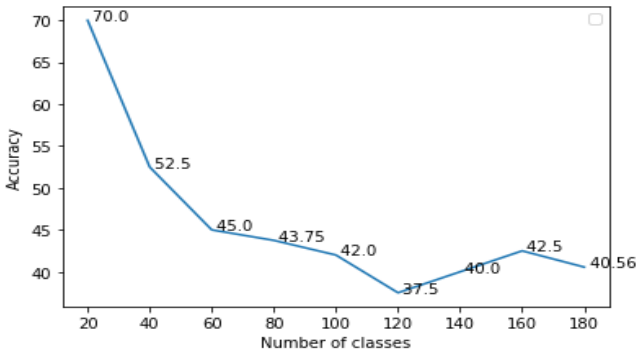

Fig (8). Accuracy of SVM versus number of classes for drugs.

SVM showed remarkable accuracy of 95% for toxins till about 60 classes after which we began to lose accuracy. For 207 classes, the accuracy dropped to 82.18%. This was a significant improvement over PCA but there was still quite a scope for improvement. For drugs however, even though the accuracy was better than PCA, but low in general, so there was large scope of improvement.

C. Custom Convolutional Neural Network

Convolutional Neural Networks are well known for their capability of capturing and identifying important features from the data. The limited accuracy of Support Vector Machines (SVMs) for drug and toxin classification task led us to the One-Dimensional Convolutional Neural Networks (1D-CNN) [26] Approach, which is commonly used for time series analysis and as a pre-processing technique in Recurrent Neural Networks (RNN) for down sampling. Convolutional layers work by extracting hidden patterns in the input data. Each kernel works as a feature identifier, filtering out where the feature exists in the input data. Max-Pooling Layers extract the most prominent of these features [27] by taking the max in the local region. Activation Layers provide the needed non-linearity to the model. Finally, dense fully connected layers which follow these convolutional blocks act as classifiers and use the newly engineered flattened feature vectors as input for this purpose.

Training Overview

The infrared spectra of drugs and toxins were treated as a fixed domain sequence of values. Data augmentation techniques mentioned earlier were used to increase the size of the dataset. The number of classes were set to 207 for toxins and 192 for drugs. However, in principle the model

architecture can take in a lot more classes with negligible loss in accuracy. The entire model was split into several Convolutional Blocks (CB). The convolutional blocks are arranged sequentially to finally extract the feature vector used for classification. Each Convolutional Block consists of a 1D-CNN with layer. The output is normalized [28] and passed through a Leaky ReLU activation layer. Finally, the outputs are pooled using Max Pooling (chosen over Averaged Pooling for better model performance) [29]. These operations can be summarized by the following mathematical equations.

$$\text{Convolution: } y^j \rightarrow f(b^j + \sum_i k^{ij} * x^i)$$

$$\text{Batch Normalization: } y^j \rightarrow \left(\frac{x^j - \mu}{\sqrt{\sigma^2 + \epsilon}}\right) + \beta$$

$$\text{where } \mu = \frac{1}{m}\sum_{i=1}^{m} x^i \text{ and } \sigma^2 = \frac{1}{m}\sum_{i=1}^{m} m(x^i - \mu)^2$$

$$\text{Leaky ReLU: } f(x) = \begin{cases} x, & if \ x > 0 \\ ax, & otherwise \end{cases}$$

The model was trained using the categorical cross entropy [30] function. To back propagate and update the weights after each iteration we used the Adamax optimizer [31] with a learning rate of $1e^{-2}$. Adamax is an adaptation of the popular Adam [32] algorithm which uses Batch Gradient Descent. The model was iterated over 60 epochs for both drugs and toxins with a batch size of 32. Finally, we used connected layers and the soft-max activation for the final classification step. The complete model architecture entailing all the layers is shown in the figure below (Fig (9)).

$$\text{Categorical cross entropy: } -\sum_{i=1}^{m}(y_i . \log(y_i^1))$$

$$\text{softmax: } \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, K = number \ of \ classes$$

Once the basic setup was ready, the method underwent a lot of experimentation. The number of convolutional blocks can be linked to the ability of the model to identify more complex features in the data. This might sometimes also lead to problems like over-fitting on the training data, wherein the model starts to focus on the redundant features of the data with more priority because of which, the model does not generalize well for the unseen data. We finally settled for 3 blocks since that performed the most accurately on the unseen testing data.

IV. RESULTS

The model was trained for 60 epochs and achieved a test accuracy of 96.55% with a top 3 accuracy of 97.13% and a top 5 accuracy of 97.7% for toxins and a test accuracy of 95.46% with a top 3 accuracy of 96.68% and a top 5 accuracy of 96.88% for drugs, which is certainly a significant improvement over SVMs. The model achieved 99.6% accuracy on the training dataset implying that the model had a near perfect fit on the entire input data. The complete results for the accuracy are summarized in the tables below.

Table (1). The trends in the top 1,3 and 5 accuracies for toxins versus number of convolutional blocks (CB)

CNN Blocks	Accuracy (Top1)	Accuracy (Top 3)	Accuracy (Top 5)

1	90.8	91.95	93.1
2	92.53	93.68	94.25
3	96.55	97.7	97.7
4	95.98	96.55	96.55
5	96.55	97.13	97.13
6	94.83	95.4	96.55

Table (1). The trends in the top 1,3 and 5 accuracies for drugs versus number of convolutional blocks (CB)

CNN Blocks	Accuracy (Top1)	Accuracy (Top 3)	Accuracy (Top 5)
1	89.53	90.46	92.43
2	91.98	92.74	94.12
3	95.46	96.68	96.88
4	94.62	95.23	95.23
5	93.24	94.26	94.34
6	93.12	93.86	94.06

```
Layer (type)                  Output Shape          Param #
=================================================================
conv1d_1 (Conv1D)             (None, 686, 16)       352
_____
bn1 (BatchNormalization)      (None, 686, 16)       64
_____
leaky_re_lu_1 (LeakyReLU)     (None, 686, 16)       0
_____
max_pooling1d_1 (MaxPooling1  (None, 343, 16)       0
_____
conv1d_2 (Conv1D)             (None, 333, 32)       5664
_____
bn2 (BatchNormalization)      (None, 333, 32)       128
_____
leaky_re_lu_2 (LeakyReLU)     (None, 333, 32)       0
_____
max_pooling1d_2 (MaxPooling1  (None, 166, 32)       0
_____
conv1d_3 (Conv1D)             (None, 162, 64)       10304
_____
bn3 (BatchNormalization)      (None, 162, 64)       256
_____
leaky_re_lu_3 (LeakyReLU)     (None, 162, 64)       0
_____
max_pooling1d_3 (MaxPooling1  (None, 81, 64)        0
_____
flatten_1 (Flatten)           (None, 5184)          0
_____
dense_1 (Dense)               (None, 2048)          10618880
_____
bn7 (BatchNormalization)      (None, 2048)          8192
_____
activation_1 (Activation)     (None, 2048)          0
_____
dropout_1 (Dropout)           (None, 2048)          0
_____
dense_2 (Dense)               (None, 192)           393408
_____
bn8 (BatchNormalization)      (None, 192)           768
_____
activation_2 (Activation)     (None, 192)           0
=================================================================
Total params: 11,038,016
Trainable params: 11,033,312
Non-trainable params: 4,704
```

Fig (9). Complete model architecture

V. CONCLUSION AND FUTURE WORKS

One of the significant achievements of this model is that many of these compounds are very similar in their IR spectra in the required range. The variation is only the position of certain functional groups. The Convolutional Block efficiently captures the hidden patterns and variations in the data. A model wise comparison in performance can be seen in the table below. Some significant take-aways from the results obtained.

1) The increasing number of convolutional blocks implies an increased ability to extract hidden patterns from the input data. However, beyond a threshold (dependent on the nature of the dataset), the model tends to overfit and does not generalize well.

2) An increase in the number of features or wavelengths assessed provides only a marginal increase in the accuracy (about 1%). The drugs which used 716 features gave a top accuracy of 95.5% whereas the toxins using 1740 features gave a top accuracy of 96.55%.

3) Since the sources are different for both train and test, a dip in accuracy is generally expected due to changing noise factors and sampling rate. However, that is not the case and the model is proven to be extremely robust.

4) The model requires just two input spectra (can be absorbance or transmittance) to learn the class's distinguishing features and classify. This is a huge leap in the domain of CNNs which are extremely data dependent.

The research presented in this paper opens the possibility to have handheld devices to recognize the presence of drugs and toxins in an instant. It also opens paths for novel methods of instant counterfeit drug detection. Presence of harmful organic toxins in food and other consumables can be identified. New classes of drugs can be appended to the input dataset with ease and the model can classify with no prior pre-processing involved. The model proposed can also be extended to other kinds of IR and UV spectral databases with minimal tweaks in the proposed architecture to account for the varied dataset by the process of transfer learning that has seen numerous applications in the field of object detection.

REFERENCES

1. N. Jmour, S. Zayen, and A. Abdelkrim. Convolutional neural networks for image classification. In 2018 International Conference on Advanced Systems and Electric Technologies (ICASET), pages 397 − −402, 2018.
2. Bin Liu, Ruiming Tang, Yingzhi Chen, Jinkai Yu, Huifeng Guo, and Yuzhou Zhang. Feature generation by convolutional neural network for click-through rate prediction. In The World Wide Web Conference on - WWW 19. ACM Press, 2019.
3. Luciene GonÃ Palmeira Moraes, Renata Sanches Ferreira Rocha, LÃvia Maluf Menegazzo, Eudes Borges de AraÃ, Keizo Yukimito, and JoÃCarlos Silos Moraes. Infrared spectroscopy: a tool for determination of the degree of conversion in dental composites. Journal of Applied Oral Science, 16:145 − 149, 04 2008.
4. Wiley SpectraBase http://spectrabase.com/ (accessed June 20, 2020).
5. P.J. Linstrom and Eds W.G. Mallard., NIST Chemistry WebBook, NIST Standard Reference Database Number 69, National Institute of Standards and Technology, Gaithersburg MD, 20899, https://doi.org/10.18434/T4D303, (retrieved July 19, 2020).
6. SDBSWeb: https://sdbs.db.aist.go.jp (National Institute of Advanced Industrial Science and Technology, June 20, 2020).
7. Ankit Rohatgi. Webplotdigitizer 4. https://automeris.io/webplotdigitizer.
8. Muna Alsallal, Mhd Sharif, Baydaa Al-Ghzawi, and Sabah Al-Mutoki. A machine learning technique to detect counterfeit medicine based on x-ray fluorescence analyser. pages 118–122, 08 2018.
9. Sebastian Berisha, Mahsa Lotfollahi, Jahandar Jahanipour, Ilker Gurcan, Michael Walsh, Rohit Bhargava, Hien Nguyen, and David Mayerich. Deep learning for ftir histology: leveraging spatial and spectral features with convolutional neural networks. The Analyst, 144, 12 2018.

10. Cungui Cheng, Jia Liu, Chang-Jiang Zhang, Miaozhen Cai, Hong Wan, and Wei Xiong. An overview of infrared spectroscopy based on continuous wavelet transform combined with machine learning algorithms: Application to chinese medicines, plant classification, and cancer diagnosis. Applied Spectroscopy Reviews, 45:148–164, 03 2010.

11. Qingbo Li, Wei Wang, Xiaofeng Ling, and Jin Guang Wu. Detection of gastric cancer with fourier transform infrared spectroscopy and support vector machine classification. BioMed Research International, 2013:1–4, 2013.

12. Yunyi Wu and Guanyu Wang. Machine learning based toxicity prediction: From chemical structural description to transcriptome analysis. International Journal of Molecular Sciences, 19(8):2358, aug 2018.

13. Hao Zhang, Haifeng Sun, Ling Wang, Shun Wang, Wei Zhang, and Jiandong Hu. Near infrared spectroscopy based on supervised pattern recognition methods for rapid identification of adulterated edible gelatin. Journal of Spectroscopy, 2018:1–9, dec 2018.

14. N. Kumar, P. C. Panchariya, S. S. Patel, A. H. Kiranmayee, and R. Ranjan. Application of various pre-processing techniques on infrared (ir) spectroscopy data for classification of different ghee samples. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pages 1–6, 2018.

15. Rekha Gautam, Sandeep Vanga, Freek Ariese, and Siva Umapathy. Review of multidimensional data processing approaches for raman and infrared spectroscopy. EPJ Techniques and Instrumentation, 2(1), jun 2015.

16. Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), jul 2019.

17. Ritsuko Hori and Junji Sugiyama. A combined FT-IR microscopy and principal component analysis on softwood cell walls. Carbohydrate Polymers, 52(4):449–453, jun 2003.

18. Yunliang Chen and Liankui Dai. An automated baseline correction method based on iterative morphological operations. Applied Spectroscopy, 72(5):731–739, mar 2018.

19. Georg Schulze, Andrew Jirasek, Marcia Yu, Arnel Lim, Robin Turner, and Michael Blades. Investigation of selected baseline removal techniques as candidates for automated implementation. Applied spectroscopy, 59:545–74, 06 2005.

20. Otkrist Gupta, Anshuman J. Das, Joshua Hellerstein, and Ramesh Raskar. Machine learning approaches for large scale classification of produce. Scientific Reports, 8(1), mar 2018.

21. S. Ghosh, A. Dasgupta, and A. Swetapadma. A study on support vector machine based linear and non-linear pattern classification. In 2019 International Conference on Intelligent Sustainable Systems (ICISS), pages 24–28, 2019.

22. Lijuan Liu, Bo Shen, and Xing Wang. Research on kernel function of support vector machine. In Yueh-Min Huang, Han-Chieh Chao, Der-Jiunn Deng, and James J. (Jong Hyuk) Park, editors, Advanced Technologies, Embedded and Multimedia for Human-centric Computing, pages 827–834, Dordrecht, 2014. Springer Netherlands.

23. Junping Wang, Quanshi Chen, and Yong Chen. RBF kernel-based support vector machine with universal approximation and its application. In Advances in Neural Networks – ISNN 2004, pages 512–517. Springer Berlin Heidelberg, 2004.

24. Matthias Feurer and Frank Hutter. Hyperparameter Optimization, pages 3–33. Springer International Publishing, Cham, 2019.

25. Raji Ghawi and Jürgen Pfeffer. Efficient hyperparameter tuning with grid search for text categorization using kNN approach with BM25 similarity. Open Computer Science, 9(1):160– 180, aug 2019.

26. Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel Inman. 1d convolutional neural networks and applications: A survey, 05 2019.

27. Fen Li, Ming Liu, Yuejin Zhao, Lingqin Kong, Liquan Dong, Xiaohua Liu, and Mei Hui. Feature extraction and classification of heart sound using 1d convolutional neural networks. EURASIP Journal on Advances in Signal Processing, 2019(1), dec 2019.

28. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR, abs/1502.03167, 2015.

29. Víctor Suárez-Paniagua and Isabel Segura-Bedmar. Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction. BMC Bioinformatics, 19(S8), jun 2018.

30. Katarzyna Janocha and Wojciech Czarnecki. On loss functions for deep neural networks in classification. Schedae Informaticae, 25, 02 2017.

31. Eustace Dogo, Oluwatobi Afolabi, Nnamdi Nwulu, Bhekisipho Twala, and Clinton Aigbavboa. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. 12 2018.

32. Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. International Conference on Learning Representations, 12 2014.

AUTHORS PROFILE

Gokul Mohanraj is pursuing a dual degree, B.Tech and M.Tech in the Electrical Engineering department at Indian Institute of Technology, Madras. His primary interest lies in the domain of machine learning, computer vision, deep learning, and image processing.

Gagan Jain is pursuing his majors in Mechanical Engineering with a minor in Computer Science at the Indian Institute of Technology, Bombay. He is a technical team member at SeDriCa, UMIC IITB, working on self-driving cars. His interests include optimal control, machine learning, and their application in robotics.

Pratyush Agarwal is pursuing his B.Tech in the Computer Science department at Indian Institute of Technology, Bombay. His primary interest lies in machine learning, reinforcement learning and formal verification of programs. He is a competitive programmer who has been to the ACM ICPC World Finals, 2019.

Vaibhavkumar Patel is pursuing his dual degree, B.Tech in Mechanical and M.Tech in Data Sciences at Indian Institute of Technology, Madras. His interest lies in Deep Learning and Graph Theory. He has worked on object detection, theft detection and STR Frameworks at Robert Bosch Centre for Data Science and Artificial Intelligence (RBCDSAI).