DI504 Project Report - Predicting Concentration of Crystal Violet molecule from Raman Spectra Using CNNs

İrem Topsakal
Department of Metallurgical and Materials Engineering
Middle East Technical University
Ankara, Turkiye
irem.topsakal@metu.edu.tr

***Abstract*—This project explores implementing of 1D ResNet model to predict crystal violet concentrations from Raman spectra. After preprocessing with baseline correction and normalization, the model is trained with data augmentation to improve generalization. It achieved 94.8% accuracy and an R^2 score of 0.96 and this outperforms linear regression baseline. Results show that deep learning can effectively capture subtle spectral differences.**

***Index Terms*—Raman spectroscopy, crystal violet, concentration, deep learning**

I. Introduction

Raman spectroscopy is a rapid and non-destructive analytical technique used to analyze molecular structures by detecting the inelastic scattering of monochromatic light as it interacts with a sample. The laser with a known wavelength is used for spectral analysis. Scattering provides a molecular fingerprint that can reveal the presence of specific functional groups, bond types and chemical compositions. Because Raman spectroscopy requires minimal or no sample preparation and can be performed directly on solids, liquids or gases, it is widely applied across various fields, such as chemistry, biology and materials science

In a Raman spectrum, each peak corresponds to a specific molecular bond. The position of the peak on the x-axis, waveshift, is unique to each type of bond in the molecule and helps identify the whole molecules in a sample. The height of the peak on the y-axis is intensity. The intensity of the Raman signal depends on the properties of the molecule, the strength of the bond and the concentration of the substance. However, when the concentration is low, the Raman signal can become very weak and more easily affected by background noise. A common issue is fluorescence which also produces signal that the detector picks up, creating a background that can interfere with or hide the true Raman peaks.

Some molecules have well-known and strong Raman spectrum pattern. One of them is crystal violet (CV), known for its intense and well-characterized Raman signature. CV is some kind of dye used in detection systems for biochemical systems. In this project, dataset of Raman spectra of CV molecule is used which is provided by Prof. Dr. Alpan Bek.

The main goal of this project is to develop a deep learning model which predicts accurately the concentration of a sample based on its Raman spectrum. Analyzing Raman spectra at low concentrations comes with several challenges. First, the Raman signal becomes weaker, making it harder to detect as background noise often dominates. Traditional methods struggle to accurately get peaks. Additionally, it can be difficult to clearly distinguish between different concentrations because the differences in signal intensity are subtle at low concentrations. Another complication is that Raman intensity is influenced by both concentration dependent and concentration independent factors and this makes it an inverse problem that is hard to solve with standard approaches. Deep learning offers solution because it can learn and handle complex, non-linear relationships in the data which makes it well-suited for this type of analysis.

II. Literature review

For this project, the studies covering both classification and regression based problems are reviewed. One key study focused on identifying bacteria types using approximately 60000 Raman spectra from 30 different bacteria classes [1]. The data had a low signal-to-noise ratio (SNR of 4.1). Before training the model, several preprocessing steps were applied: background signals were removed using a 5th-order polynomial fit, intensity values were normalized to a range between 0 and 1 and outliers with unusually high intensities were excluded. The researchers used a one-dimensional convolutional neural network (1D CNN) based on the ResNet architecture with 25 layers. The final output layer used a softmax function to perform multi-class classification. The model was trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 10. They used 5-fold cross-validation during training, and leave-one-out cross-validation for fine tuning. The performance was evaluated using classification accuracy (82.2%), confusion matrices and ROC curves. Compared to simpler models such as logistic regression (75%) and support vector machines (74.9%), CNN showed better performance especially under noisy conditions. The source code for this work is publicly available and will serve as a base for this project.

In another classification type of study [2], researchers created a deep learning model to automatically identify different forms of TiO_2 (titanium dioxide) using Raman data. They used a combination of 1D convolutional layers and memory layer (LSTM) to learn patterns in the data. The model was trained on the public RRUFF dataset and it contains many Raman spectra of different minerals. The model includes four convolutional layers with ReLU activations, and then followed by pooling and LSTM layer. It ends with fully connected layers and a softmax output. It was trained using the RMSprop optimizer with a learning rate of 0.001. The model achieved high accuracy: %99.12 for its top prediction (Top-1 accuracy). It also worked well on raw data. The model is tested with real samples of TiO_2 which is prepared in the lab. It successfully identified standard and defect-rich forms of TiO_2. Removing LSTM layer or using different activation functions made the model less accurate. Overall, it shows good results for fast analysis of materials without needing expert input.

In terms of regression type of study, the researchers used a convolutional neural network to estimate very small chemical concentrations from Raman spectra [3]. They focused on two chemical compounds: Rhodamine 800 (R800) and Methylene Blue (MB) with wide concentration range from 1 femtomolar (fM) to 1 micromolar (µM). The dataset included around 8960 spectra for R800 and 3200 spectra for MB. The preprocessing steps included Savitzky–Golay smoothing, background subtraction and non negative matrix factorization (NMF). The model is consisted of four convolutional layers followed by a fully connected layer. Unlike classification tasks, model was trained to predict concentration values directly on a logarithmic scale that means no softmax layer was used at the end. Each input was structured as a map of 2D Raman spectra matrix where each element represented a 1D spectrum. The model's performance was evaluated using the coefficient of determination (R^2), achieving an average R^2 score of 0.958 through 5-fold cross-validation. For R800, mean squared error (MSE) was reported as 0.111. The authors reused model trained on R800 for MB using transfer learning. Although the authors did not provide their code, approach they used serves as reference for CNN based regression tasks for Raman spectra.

III. Dataset Overview

The raw Raman spectroscopy dataset used in this project includes measurements at five different concentrations: 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8} and 10^{-9} M. For each concentration, 900 measurements were taken. The spectral range is from 169.84 cm^{-1} to 2339.42 cm^{-1} and each measurement contain average of 2300 data points. This uniform structure across concentrations provides a robust dataset for analysis and model training.

The raw data for each measurement is provided in two separate files. Therefore, the first step is to merge these files to obtain the full spectral range which is 169.84 - 2339.42 cm^{-1}. Due to the nature of Raman spectroscopy, background signals and fluorescence radiation must be removed and baseline correction is necessary. To address this, asymmetric least squares method (ALS) is applied. The result of this correction is shown in Figure 1.

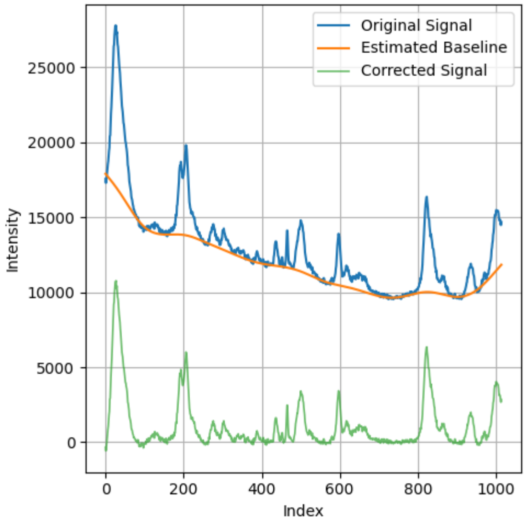

Fig. 1. Example of baseline correction with ALS

One of the baseline corrected spectrum as input to the model is presented in Figure 2. All input data are preprocessed in a such a way that contains merging and baseline correction prior to being fed into the model.

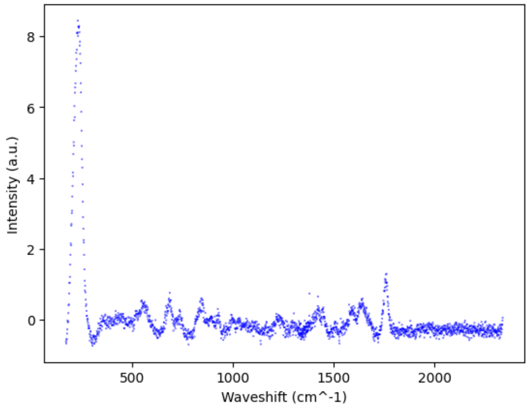

Fig. 2. Example of the Raman spectra of CV molecule at 10^{-6} M concentration after baseline correction and normalization

Z-score normalization is applied during training to improve model performance. It standardizes each Raman spectrum by subtracting its mean and dividing by its standard deviation so that data is with zero mean and unit variance. This helps stabilize the training process and leads to better accuracy. Although it was initially stated that no scaling would be used, results showed that applying normalization improved performance so it is included.

IV. Methodology

A. Preprocessing

As it is explained in more detail in previous section, each Raman measurement in the dataset initially comes in two separate files, which are merged to obtain the full spectral range between 169.84 and 2339.42 cm^{-1}. Baseline correction with ALS is applied to each data for removing background signal. Following this, z-score normalization is performed individually on each spectrum. In addition, data augmentation is used to improve model generalization. This includes random spectral shifting, intensity scaling, and Gaussian noise addition. By this way, data amount in training part is increased by the factor of 3.

B. Model Architecture

The model architecture is primarily adapted from the open source implementation by [1] with modifications inspired by [3]. Key changes include adapting the input dimensions to fit this project's dataset and modifying the final fully connected layer to output a single continuous value for regression. The detailed architecture of the ResNet model and its residual block structure is presented in Table I and Table II.

TABLE I
Architecture of Custom 1D ResNet Model where k = kernel size, s = stride, p = padding, BN = Batch Normalization, FC = Fully Connected layer. C_1, C_2, C_3 are tunable channel sizes selected via hyperparameter optimization

Layer	Output Shape	Details
Input	(1, 2300)	—
Conv1D + BN + ReLU	(C_1, 2300)	k=5, s=1, p=2
ResBlock ×2	(C_1, 2300)	$C_1 \rightarrow C_1$, k=5, s=1
ResBlock ×2	(C_2, 1150)	$C_1 \rightarrow C_2$, k=5, s=2
ResBlock ×2	(C_3, 575)	$C_2 \rightarrow C_3$, k=5, s=2
Flatten	(C_3 × 575)	—
FC Layer	(1,)	C_3 × 575 → 1

TABLE II
Structure of a 1D Residual Block

Sub-layer	Details
Conv1D_1 + BN + ReLU	k=5, s=stride, p=2
Conv1D_2 + BN	k=5, s=1, p=2
Shortcut (if needed)	Conv1D k=1, s=stride
Add + ReLU	Element-wise addition

C. Training Strategy

First, the dataset is randomly split into training (70%), validation (15%), and test (15%) sets. This gives 3150 training samples and 675 for both validation and test. Augmentation is applied only to training set. Each original training sample is used to create three new samples, increasing the training set size to 9450. The model is trained as a regression task using Mean Squared Error loss between the predicted and actual concentration values. The model is optimized using Adam optimizer as it is used in related studies and has demonstrated effective performance in similar tasks. Hyperparameters such as learning rate and channel depth are tuned using Optuna by trying 10 different combinations. After determining best hyperparameters with Optuna, model is trained with these parameters. Model training is continued for 20 epochs.

D. Performance Evaluation

The model's performance is evaluated using a combination of regression and classification metrics to reflect both the continuous nature of the predictions and the discrete levels of the target concentrations. During training, Mean Squared Error is used as the loss function. Evaluation metrics (MAE, RMSE, R^2, and weighted kappa score) are calculated after rounding predictions to the nearest concentration level. For evaluation, additional regression metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R^2 are computed. Because target concentration values should be discrete, predictions are rounded to the nearest class. This step works as a post-regression classification. Based on these rounded values, classification performance is measured using test accuracy, confusion matrix and weighted kappa score.

V. Results and Discussions

This section presents the performance of the proposed 1D ResNet model on the Raman spectroscopy dataset along with comparisons to baseline methods and an ablation study.

Hyperparameter optimization phase using Optuna where involved 10 trials with 5 epoch training each takes approximately 20 minutes to complete. Following this, the final model is trained for 20 epochs which requires around 10 minutes.

Best hyperparameters obtained using Optuna are:

- Hidden Layer Sizes: 128- 256 - 512
- Learning Rate: 1.07 x 10^{-4}

Rest of the evaluations are done by using these hyperparameters in the models.

A. Performance of Proposed Model

The proposed ResNet-based model was trained as a regression model and evaluated on a test set. Performance metrics at the end of 20 epochs are as follows:

- Mean Squared Error: 0.0893
- Mean Absolute Error: 0.0785
- Root Mean Squared Error: 0.2802
- R^2 Score: 0.9598
- Weighted Kappa score: 0.9793
- Accuracy for test split: 0.948

Even though the model performs with good accuracy, it is possible to say that the amount of data is not sufficient even with data augmentation because the validation loss curves fluctuate around the training loss by looking at Figure 3 and 4. The issue might be the limited size of the validation set. Apart from that, validation loss is mostly higher than the training loss, which is a positive sign.

The model struggles to predict the 10^{-9} M concentration which is expected since the spectra for 10^{-8} M and 10^{-9} M are visually very similar. In addition, misclassifications at 10^{-9} M are likely due to extremely weak Raman signals where

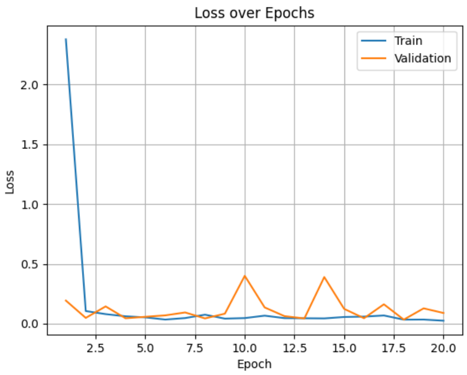

Fig. 3. Loss calculation for the proposed model per epoch

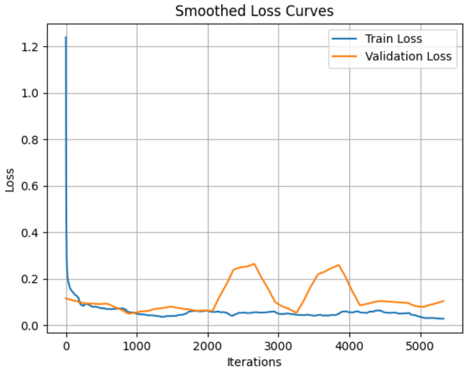

Fig. 4. Loss calculation for the proposed model per iteration

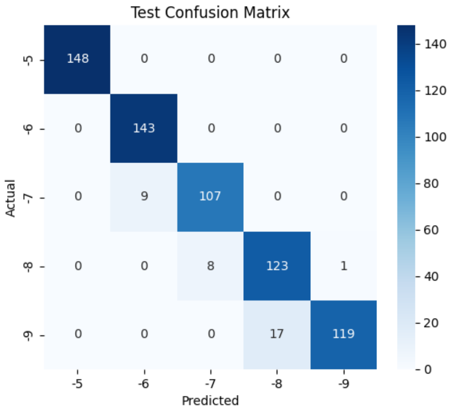

Fig. 5. Confusion matrix of test split for the proposed model

spectrum becomes dominated by noise. On the other hand, the model performs quite well for higher concentrations, such as 10^{-5} M and 10^{-6} M.

Even though they are not formally recorded but observed, across multiple training runs, the model consistently achieved over 90% accuracy. It demonstrate stable and robust generalization when data augmentation is applied.

B. Baseline Model Comparison

To evaluate the effectiveness of the proposed architecture, a simple baseline model (linear regression) is trained on same dataset. Performance metrics at the end of 20 epochs are as follows:

- Mean Squared Error: 0.1207
- Mean Absolute Error: 0.0963
- Root Mean Squared Error: 0.3103
- R^2 Score: 0.9539
- Weighted Kappa score: 0.9776
- Accuracy: 0.904

The linear regression model performed reasonably well considering its simplicity. However, it is worse than ResNet model especially in capturing complex, non-linear relationships in the spectral data. As it can see from Figure 7, it performs worse for 10^{-6} M concentration and labeled them as 10^{-5}. This result in drop in accuracy score. Also, loss curves can be seen in Figure 6.

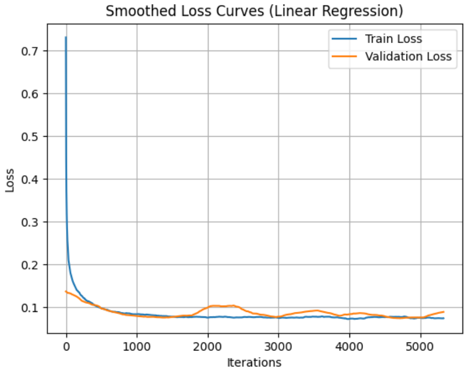

Fig. 6. Loss calculation for the baseline model per iteration

C. Ablation Study: No Data Augmentation

To measure the impact of data augmentation ResNet model is also trained using only to dataset without augmentation. Results show how augmentation influences model performance. Performance metrics at the end of 20 epochs are as follows:

- Mean Squared Error: 0.1956
- Mean Absolute Error: 0.2415
- Root Mean Squared Error: 0.4914
- R^2 Score: 0.8696
- Weighted Kappa score: 0.9289
- Accuracy: 0.759

Without data augmentation, data performed considerable worse as it can be seen from Figure 9. It makes much more

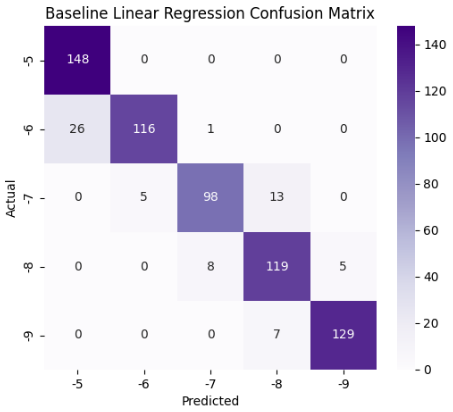

Fig. 7. Confusion matrix of test split for the baseline model which is linear regression

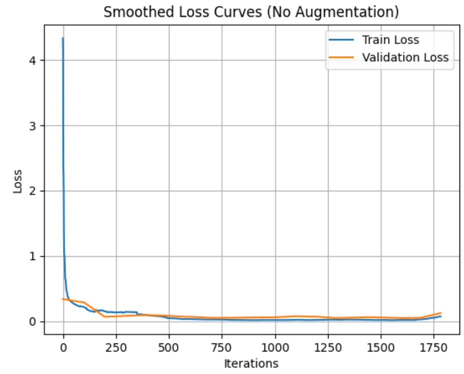

Fig. 8. Loss calculation for ablation study per iteration

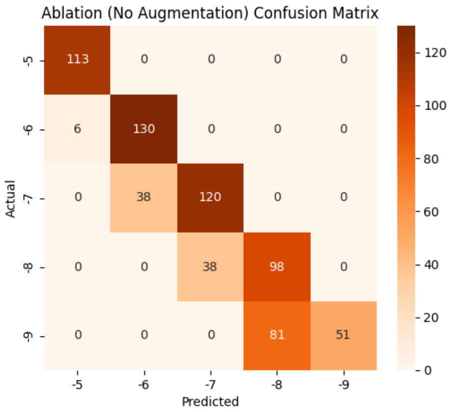

Fig. 9. Confusion matrix of test split for ablation study

mistake while predicting the class. Moreover, loss function for training starts with higher values than the other models as shown in Figure 8. It can be concluded that data augmentation increases the model performance.

References

[1] C. S. Ho, N. Jean, C. A. Hogan, L. Blackmon, S. S. Jeffrey, M. Holodniy, N. Banaei, A. A. E. Saleh, S. Ermon, and J. A. Dionne, "Rapid identification of pathogenic bacteria using raman spectroscopy and deep learning," *Nature Communications*, vol. 10, no. 1, p. 4927, 2019. [Online]. Available: https://doi.org/10.1038/s41467-019-12898-9

[2] A. Bhattacharya, J. A. Benavides, L. F. Gerlein, and S. G. Cloutier, "Deep-learning framework for fully-automated recognition of tio2 polymorphs based on raman spectroscopy," *Scientific Reports*, vol. 12, no. 1, p. 21874, 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-26343-3

[3] W. J. Thrift and R. Ragan, "Quantification of analyte concentration in the single molecule regime using convolutional neural networks," *Analytical Chemistry*, vol. 91, no. 21, pp. 13 337–13 342, 2019, pMID: 31589030. [Online]. Available: https://doi.org/10.1021/acs.analchem.9b03599