Classification of XRD Data Using Neural Networks

Surya Y[a],Harish Adhithiya K[b],Hemachandra M[c],Kalyan Srinivas K[d],Arun Kumar P[e](Group 14)

Abstract

The identification of space groups from XRD data is a critical yet time-consuming process in materials science, typically requiring human intervention. To address this challenge, we explore the application of machine learning techniques to automate the classification process. This project uses a large dataset of 50,000 XRD data points from the Materials Project database to improve the speed and accuracy of identifying space groups. The results demonstrate the potential of neural networks to effectively classify XRD patterns.

1.Introduction

X-ray diffraction is an analysis technique used to analyze physical properties such as crystal structure and space groups of a sample. We irradiate the sample with X-rays and record the intensities and the scattering angles of the diffracted X-Rays. Diffraction occurs as the wavelength of the X-rays used is similar to the interatomic spacing.

Crystalline materials have periodic arrangements of atoms. Crystallography categorizes unit cell parameters into 7 crystal systems to describe symmetries, and identifies 230 space groups to represent the symmetries of atoms' spatial positions. Powder XRD has been the principal method for determining the arrangement of atoms. To decode XRD patterns, traditional methods based on the Rietveld refinement method require correct peak indexing. So it involves human intervention and is time consuming. It is even harder to identify low-symmetry atom arrangements with powder XRD patterns, since peaks can be overlapped and hard to separate and index.
The approach of applying deep learning to XRD patterns and to classify their symmetries automatically without human interference is explored here.

2.Data Acquisition and Visualization

XRD data is available from various sources such as the **Materials Project (MP)** and **Crystallography Open Database (COD)**.
The Materials Project API was used to obtain the structure and symmetry of nearly 50,000 materials (hereby referred to as 50k dataset). Using the API the symmetry information is automatically converted to XRD data with the help of the XRD Calculator tool in the PyMatGen library. From COD, around 8,000 data points were obtained in the form of CIF (Crystallographic Information File) files. These were converted to XRD Data using the XRD Calculator tool in the PyMatGen library(hereby referred to as 8k dataset).

PyMatGen

PyMatGen is a Python library primarily designed for materials analysis and modeling, offering a wide range of tools for working with crystal structures, electronic structures, and properties of materials.

XRD Calculator

Once a crystal structure is defined, the XRD Calculator can compute the corresponding XRD pattern. It simulates the diffraction pattern by calculating the intensities of X-ray reflections from the crystal lattice based on its atomic arrangement and the properties of the X-rays.

There are certain points to note here:

- The output generated by these library functions represents an idealized scenario, wherein the resulting XRD diffractogram is devoid of noise. In practical applications, experimental XRD data inevitably contains noise due to factors such as instrumental limitations, sample imperfections, and environmental influences. Therefore, subsequent analysis and interpretation of experimental XRD data typically necessitate noise reduction and data processing techniques to accurately identify and quantify the crystalline phases present in the sample.
- The sampling of data points from the diffractogram is done in an irregular manner. There are many data points around a peak but very few data points everywhere else.
- This would mean that there isn't a fixed abscissa step size for the obtained XRD TwoThetha vs Intensity 2D arrays.
- Since the amount of peaks in XRD of a space group would be different from that of a different space group, this essentially means that the size of our 2D arrays are going to be highly varying.
- In order to use these arrays in NN models as the input, they must all be of the same shape/dimension, this is achieved by padding the smaller arrays with zeros.

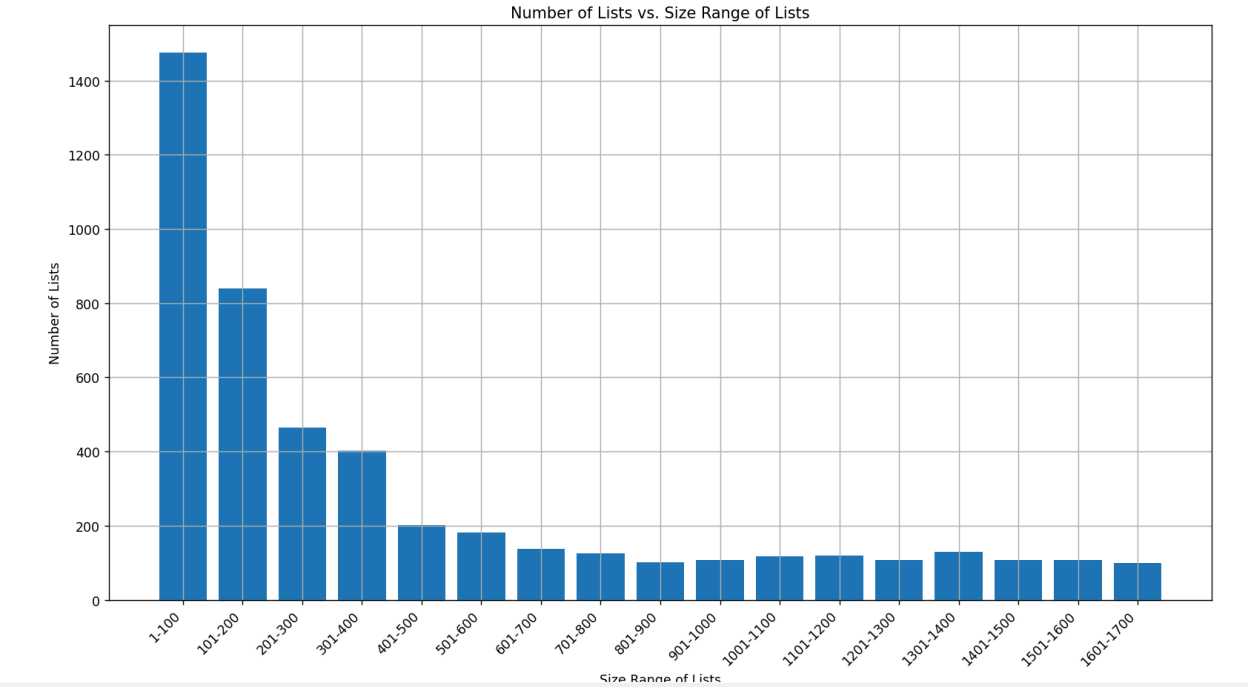

Fig 2.1 - A histogram of the size of numpy array (X-axis) vs Frequence (Y-Axis) of data points obtained from the MP Dataset.

The padding technique results in uniformly shaped arrays, but the downside is that the memory requirement for storing all these arrays during model training is very high, hence the arrays with more than 5000 points(about 10% of the dataset) are discarded to save time and make computation easier.

Visualization of Dataset

Visualization allows us to identify patterns and potential biases in the dataset. Understanding the distribution helps in selecting appropriate ML algorithms and preprocessing techniques.

The occurrence of different space groups is plotted. The data is a little biased. The occurrence of some space groups is a lot higher than the others. A similar plot is made for crystal structures in the form of a pie chart. The MP dataset with 50k data points was used for all models.

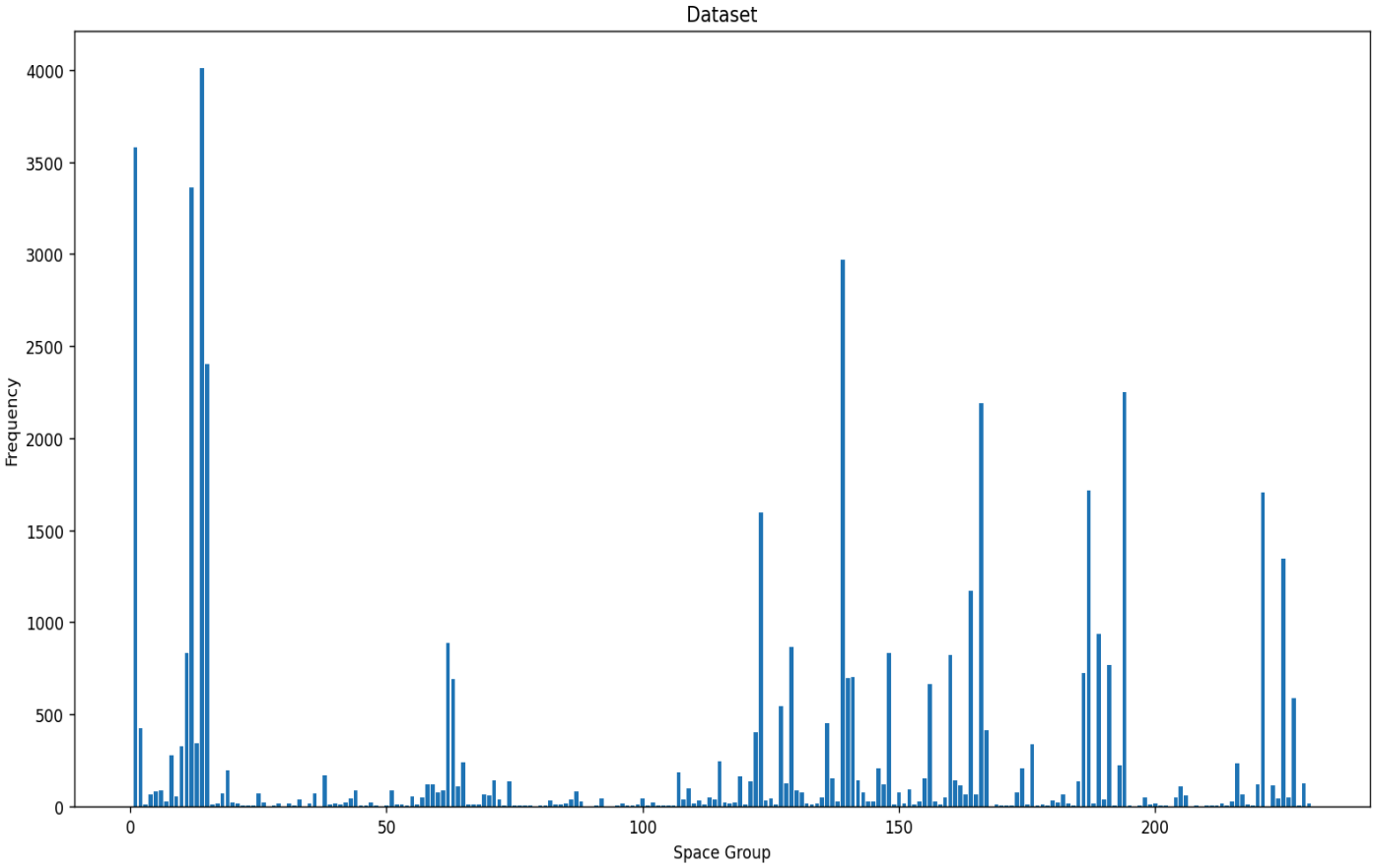

Fig 2.2 - Space Group vs Frequency in MP Dataset

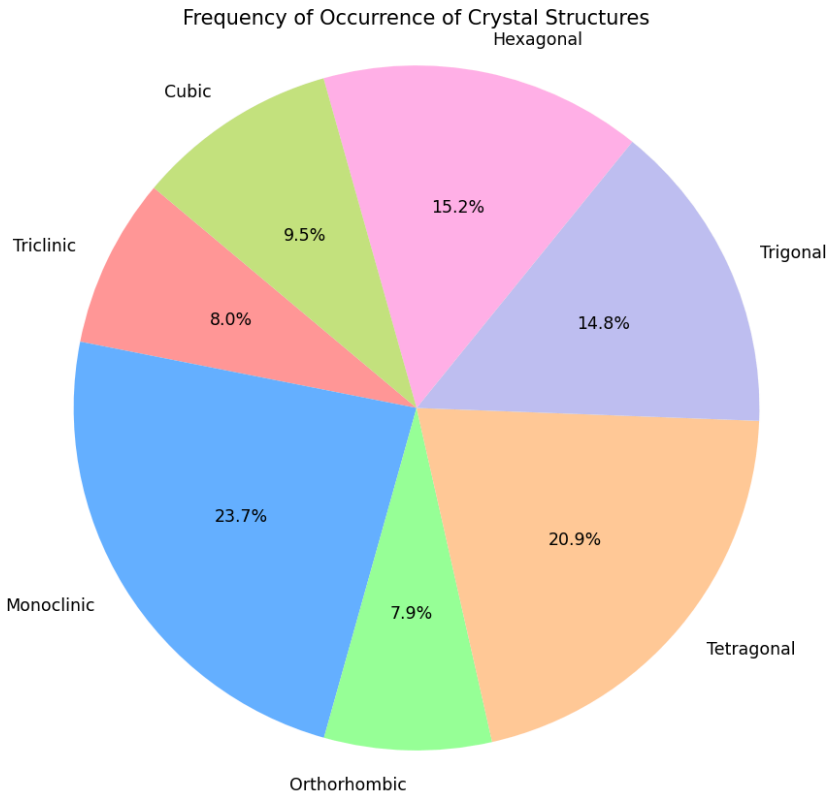

Fig 2.3 - Pie chart of crystal structure split in the MP Dataset

3.Methods

Analyzing XRD data (identifying the peaks), fitting the curve to the best extent possible and matching it with the existing database of materials is a very complicated task. So, neural networks which work well on such tasks are the best suited model to obtain highest accuracy.

Three main models were trained on the dataset.

1. Baseline ANN Model
2. ANN Model with some Optimizations
3. CNN Model

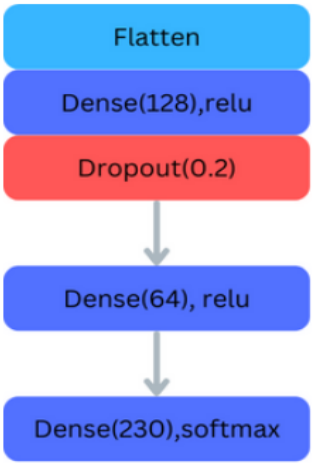

Fig 3.1 - Baseline ANN Model

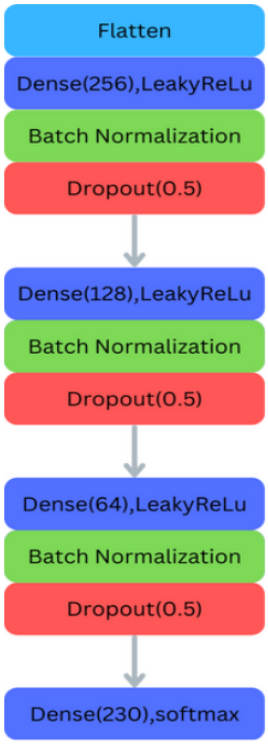

Fig 3.2 - ANN + Optimizations

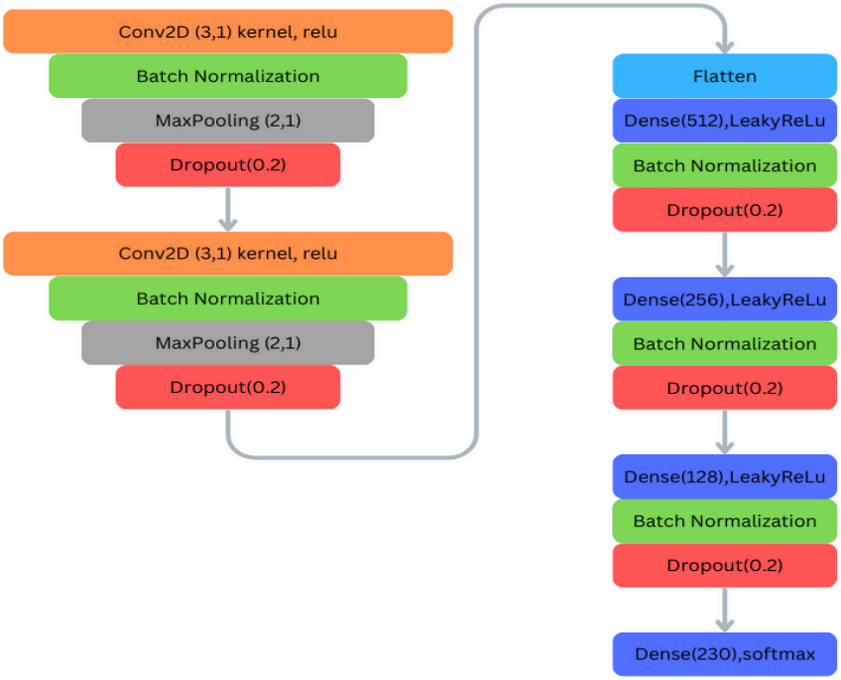

Fig 3.3 - CNN Model

The architecture of the three models are shown in the above flow charts

4.Results and Discussion

4.1 Baseline ANN Model

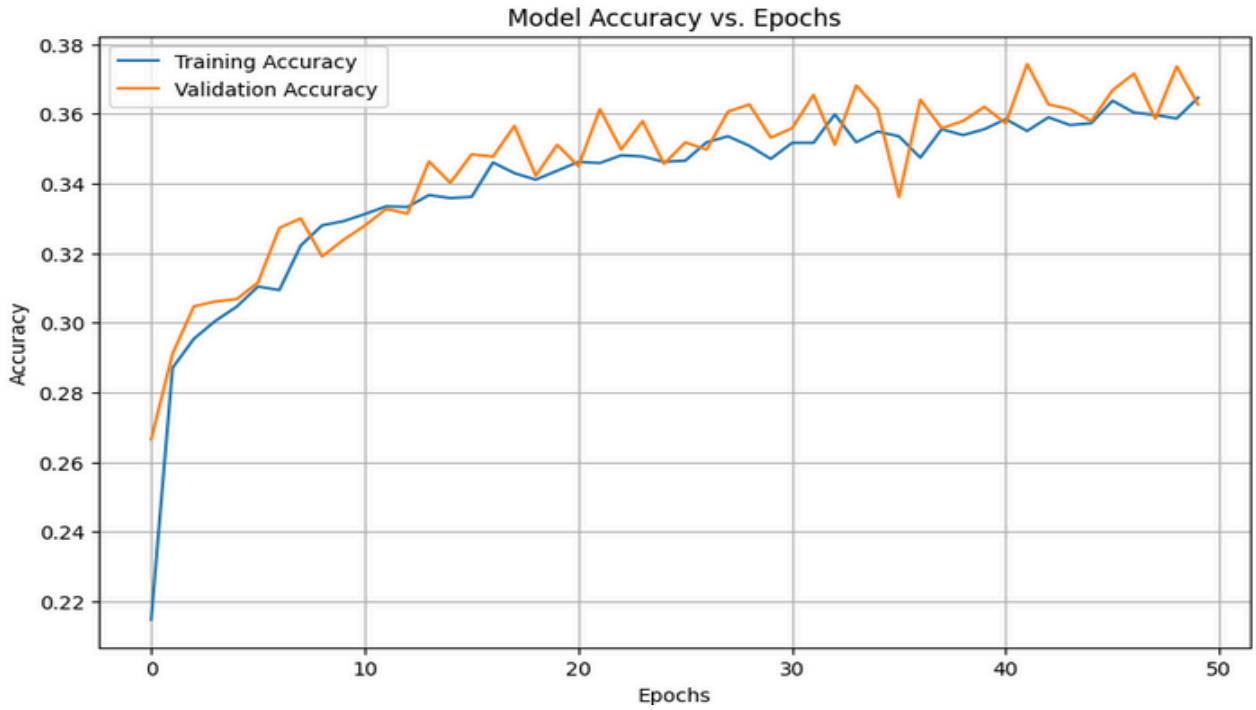

Fig 4.1.1 - Baseline ANN Model Accuracy

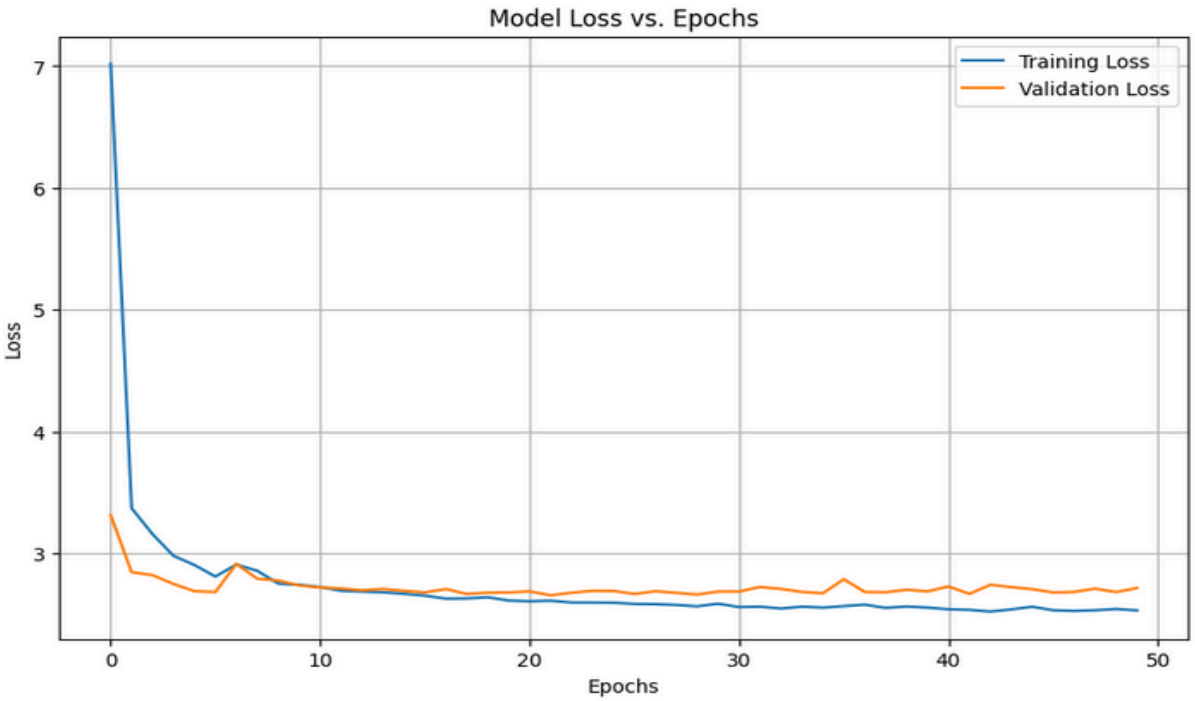

Fig 4.1.2 - Baseline ANN Model loss

As we observe the **baseline ANN model gives an accuracy of about 33%**.

4.2 Optimised ANN Model

Optimisations Used :

1. **Batch Normalisation**: This technique normalizes the activations of each layer, facilitating faster convergence and reducing sensitivity to weight initialization.
2. **OneHotEncoding:** This technique converts the target variable (space group) which is an integer into a vector of zeros. The vector contains a one in the index of the space group being referred. It allows the model to effectively capture the relationships between different categories without assuming ordinality.
3. **ReduceLROnPlateau:** We take a specified metric, such as val_accuracy, and if no improvement is seen for a 'patience' number of epochs, it reduces the learning rate. By doing so, it helps the model to get out of local minima and converge to a better solution.
4. **LeakyReLU**: The LeakyReLU activation function addresses the problem of "dying neurons" by allowing a small, non-zero gradient when the unit is not active.
5. **L1_L2 regularization:** This is required to prevent the model having excessive weights to particular features. This is important to prevent overfitting.

These optimization techniques were able to bring up the accuracy of classification to around 43%, there is still room for improvement.

Drawbacks of this way of implementing ANN :

- In this implementation, the 2D array(input) is being flattened to a 1D array and then fed into the neural network. During the flattening, the spatial information carried by the points in the 2D array is now lost(although not completely).
- Feature extraction: for any ML model to train effectively, there has to be appropriate feature extraction from the dataset. This has to be done using domain-knowledge of the problem in question, for our case of XRD, the features that most dominantly affect the space groups are peak position, peak intensity, peak spacing(distance between the peak positions), and peak shape. In a traditional XRD analysis with Rietveld refinement, these are done by experts and are the most crucial and time-consuming steps in the whole process. Simple algorithms will not be capable of accurately extracting these features, and their viability is questionable.

4.3 Convolutional Neural Networks (CNN)

The necessity for using CNN arises from the fact that XRD data consists of many spatial elements like peak height and peak width.
So, the accuracy of the ML model can be further improved by using CNN that works well in such cases. This is proven experimentally by the increase in accuracy of nearly 10% when the model was switched to CNN from ANN.

Convolutional Neural Networks are designed specifically for the purpose of processing grid-like data, such as images. They introduce 2 key operations before the Neural layers:

- **Convolution:** Applies filters to the input data to extract features like edges, corners, and patterns.
- **Pooling:** Reduces the spatial dimensions of the feature maps, helping to control overfitting and improve computational efficiency.

Using CNN has improved the accuracy of the model to around 65%.

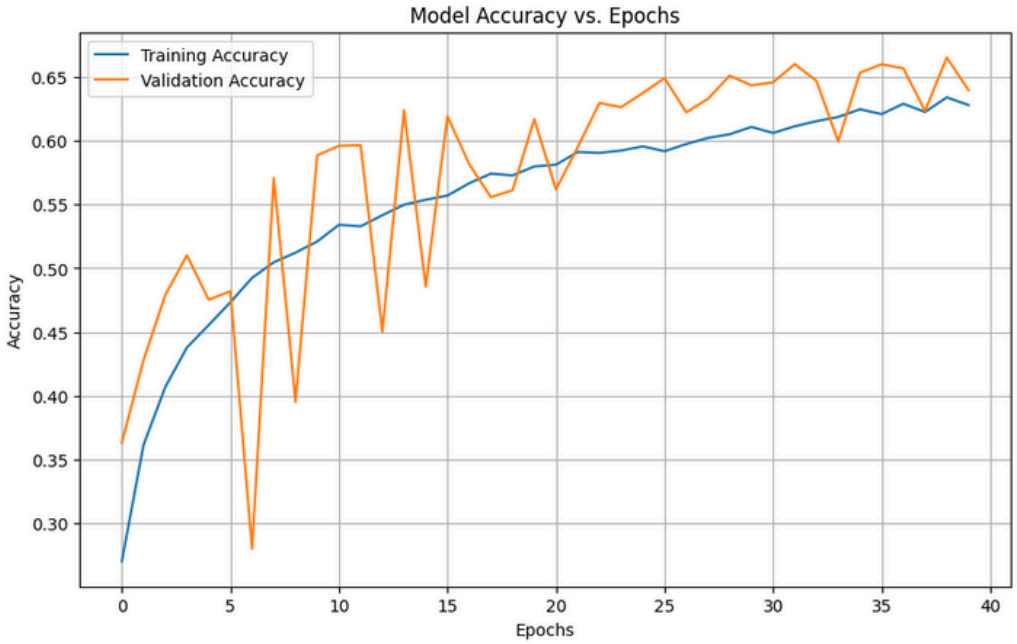

Fig 4.3.1 - CNN Model Accuracy vs Epochs

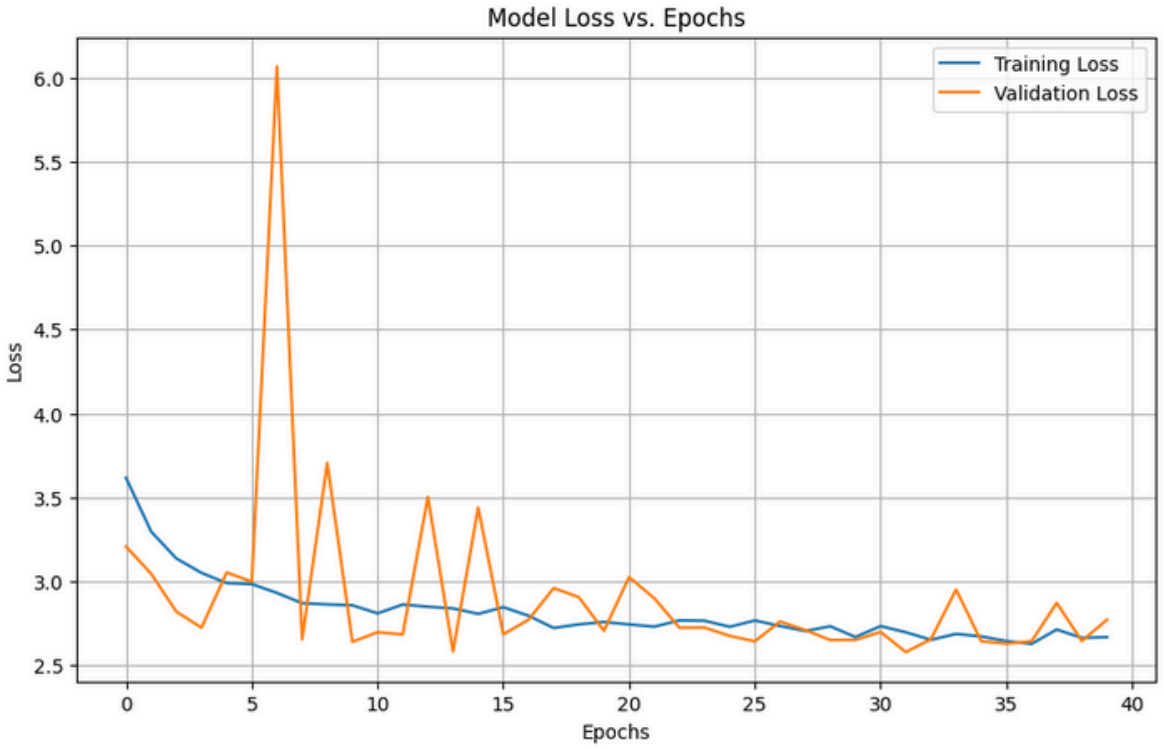

Fig 4.3.2 - CNN loss

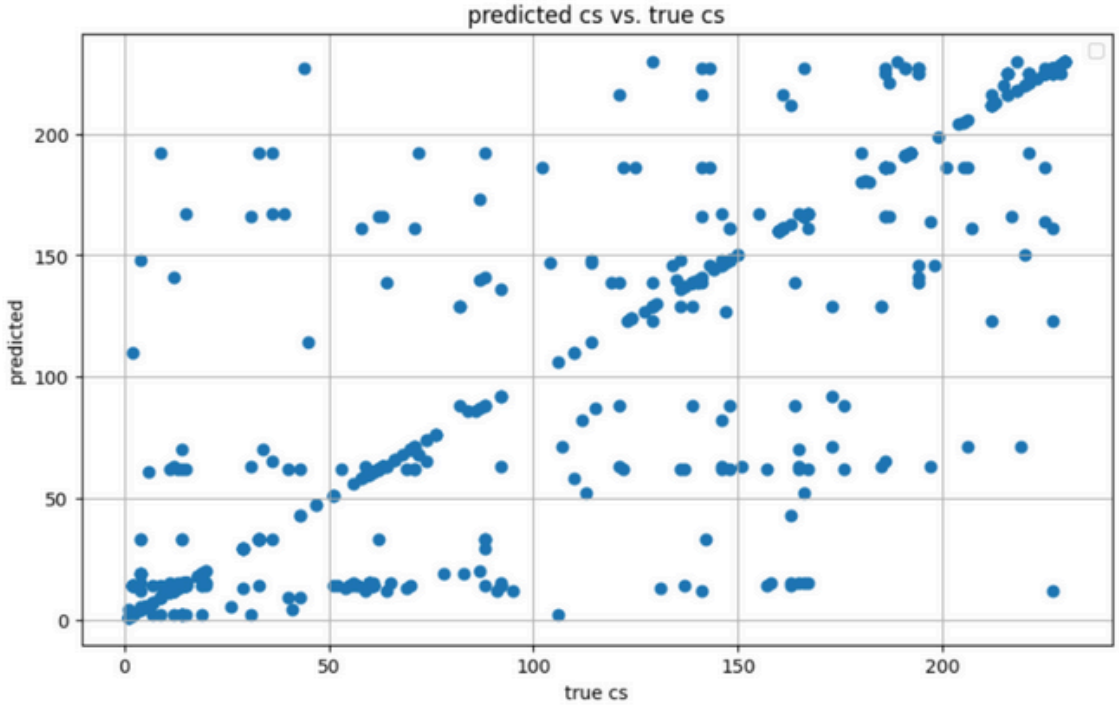

Fig 4.3.3 - Predicted vs True values of Space Groups in CNN Model

There is a major flaw in this approach so far, given that the dataset that we are using is a skewed/asymetric dataset(as in there aren't the same amount of XRD data in each of space groups), using accuracy as a metric to evaluate the performance of the models is not a good choice, due to the following reasons:

- **Focus on Majority Class:** Accuracy measures the proportion of correct predictions out of the total number of predictions. In skewed datasets, the accuracy from the majority class dominates, and a model can achieve high accuracy simply by predicting the majority class for all instances. This doesn't reflect the model's ability to identify the minority class.
- **Lack of Sensitivity to Class Imbalance:** Accuracy doesn't account for the relative sizes of the classes. A model might have high accuracy but perform poorly on the minority class, which can be problematic as all classes are equally important to us.

5.Testing the Model

The RRUFF database has a huge catalog of experimental XRD data. We have evaluated the accuracy of this model on data from RRUFF dataset, only 200 points were considered for evaluation, this was due to the fact that RRUFF website doesn't have an API to automatically download data from, so we had to manually extract data.

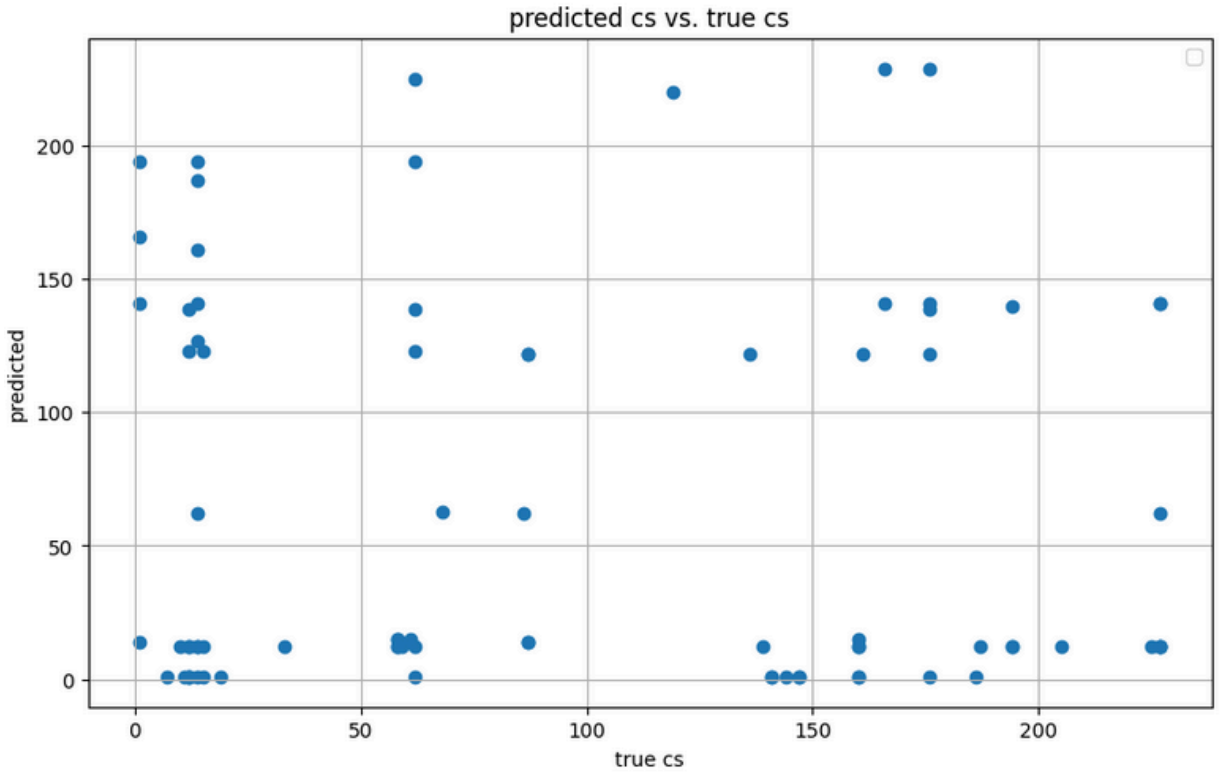

Fig 5.1 - Performance of Model on Test Data

It can be seen that the performance is very poor. It has an accuracy of 2.5%.
This result can be a cause of various factors:

- Difference in training and test data: the real world data contains a lot of noise, this noise can disrupt the convolution part where patterns are identified.

- Difference in the organization of data. In the MP data, the angle array on average had 2000 points and did not have any standard step value. But the RRUFF data has a standard step value of 0.01 and every angle array has 8500 points on average.
- Using a biased dataset. The distribution of various space groups was not equal and no special weights were declared to take this into account.

6.Conclusion

In this project, we investigated the feasibility of using machine learning to classify XRD patterns into space groups. We employed a deep neural network architecture with convolutional layers to automatically extract relevant features from the XRD patterns. Our findings demonstrate the potential of machine learning for XRD pattern classification tasks.

7. Future Work

- Some real data with noise was also used to test the model. This was obtained from RRUFF. The unavailability of such real data was a hindrance in training the model. With sufficient data from the RRUFF dataset a better model could be made.
- With all data points having a standard step in the angles array, just the intensities array can be fed to the model for training.
- A proper model with the above improvements made can be integrated with softwares like EVA which are meant for XRD analysis to save time and reduce human intervention.

8. Contributions from the authors

[a] - Data acquisition(COD), Cleaning of Data, Training ANN- (MM21B065)
[b] - Training ANN, Training and Optimizing CNN, Testing with RRUFF data- (MM21B028)
[c] - Data Acquisition (MP), Cleaning of Data- (MM21B029)
[d] - Optimization of ANN Model- (MM21B033)
[e] - Literature review, Data Acquisition (RRUFF)- (MM21B011)

9. References

[1] Salgado, J.E., Lerman, S., Du, Z. et al. Automated classification of big X-ray diffraction data using deep learning models. npj Comput Mater 9, 214 (2023). https://doi.org/10.1038/s41524-023-01164-8
[2] Rongzhi Dong, Yong Zhao, et al. DeepXRD, a Deep Learning Model for Predicting of XRD spectrum from Materials Composition. https://doi.org/10.48550/arXiv.2203.14326
[3] A Boulle and A Debelle 2023 Mach. Learn.: Sci. Technol. 4 015002. https://doi.org/10.1088/2632-2153/acab4c
[4] PyMatGen Documentation
[5] Tensorflow Documentation
[6] Materials Project Dataset
[7] COD Dataset
[8] RRUFF Dataset