Convolutional Neural Network (CNN) for Blood Sample Classification

Coursework, ARTIFICIAL INTELLIGENCE (COM2028)

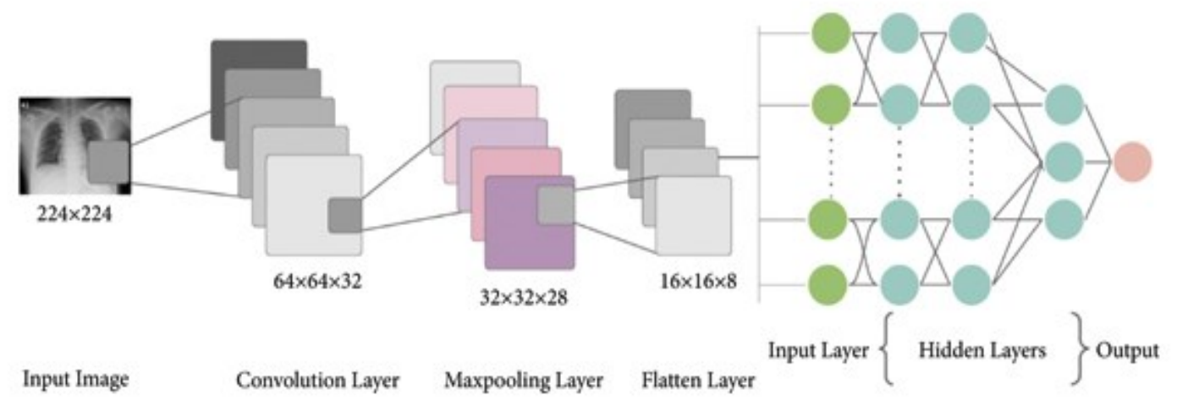
Islam Nayeemul

UNIVERSITY OF SURREY

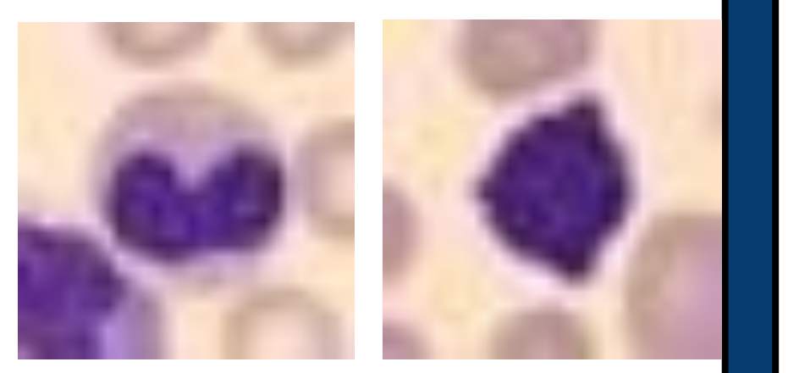
Introduction: This project aims to develop a blood sample classification model using a convolutional neural network (CNN). The dataset consists of 10,000 RGB images of 28*28 resolution, which are labelled with a value of 0-7 based on their health condition. The model will be trained on this dataset to identify unique features of each health condition and use this information to predict the correct label for new, unseen blood samples.

In recent years, machine learning has been increasingly used in disease detection, medical diagnosis, drug discovery, and medical robotics. The use of machine learning assists healthcare professionals in making informed decisions regarding patient care and treatment.

Recently, there have been some significant advancements in medical research by utilizing Convolutional Neural Networks (CNNs). For instance, one study titled "Study on Convolutional Neural Network to Detect COVID-19 from Chest X-Rays" (1) used X-ray images of lungs to detect the COVID-19 disease. Another research work called "Breast cancer histopathological image classification using a hybrid deep neural network" (2) employed a hybrid deep neural network to classify histopathological images of breast cancer, and "Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification" (3) used transfer learning with CNNs for the classification of fundus images for diabetic retinopathy. These studies have achieved high accuracy, with nearly more than 96% accuracy rate in some cases, in developing ML models to improve healthcare facilities.



Problem overview: The aim of this project was to develop an effective blood sample classification model using machine learning techniques. In order to achieve this, I explored various approaches including K-nearest neighbour (KNN), Logistics regression and convolutional neural network (CNN). Among these techniques, CNN showed the highest efficiency in classification accuracy. However, there were some challenges encountered during the development of these models.



One major issue was the lack of sufficient data for training the model. The dataset contained only 10,000 RGB images of 28*28 resolution, which may not be enough for the model to learn all the features required for accurate classification. Another challenge was the low dimensions of the images, which could potentially affect the accuracy of the classification.

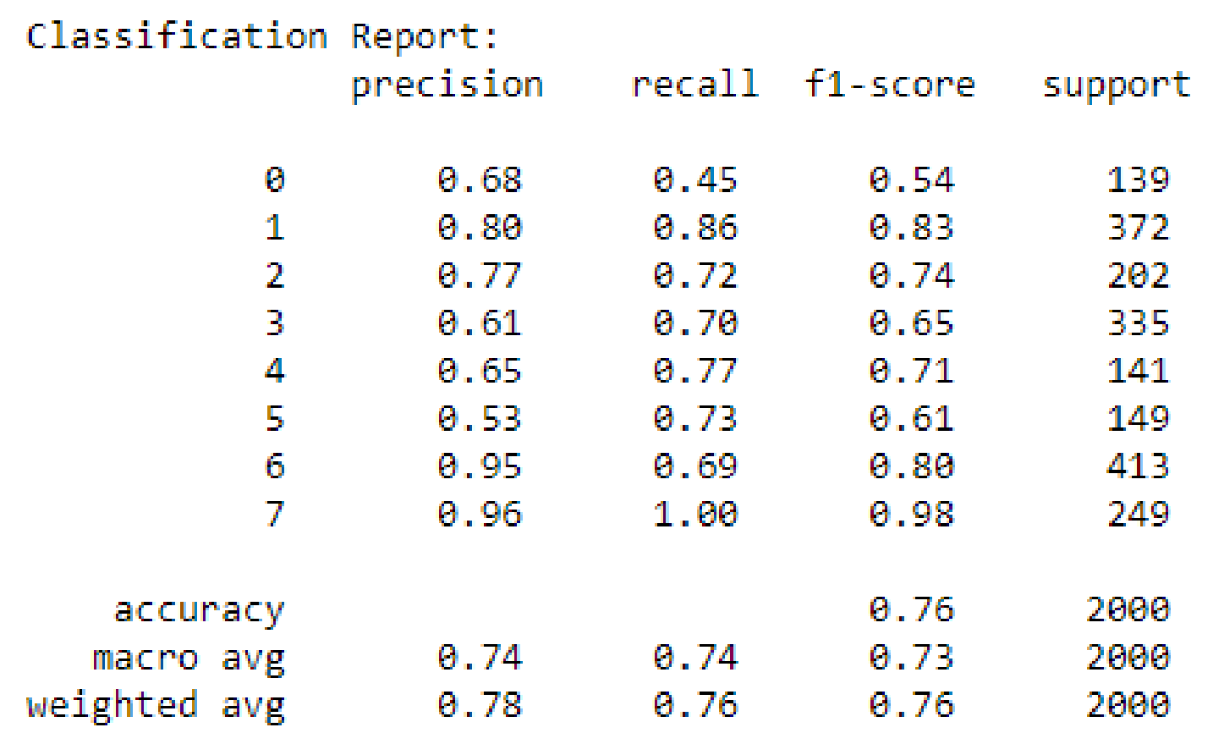
Furthermore, the dataset lacked important information that could have been useful in developing a more sophisticated classification model. For instance, data such as cell type counts, hemoglobin, white blood cell count, blood type, lipid panel, pH level, medical history and records of patients could provide additional insights for more accurate classification.

Various implementation and evaluation:

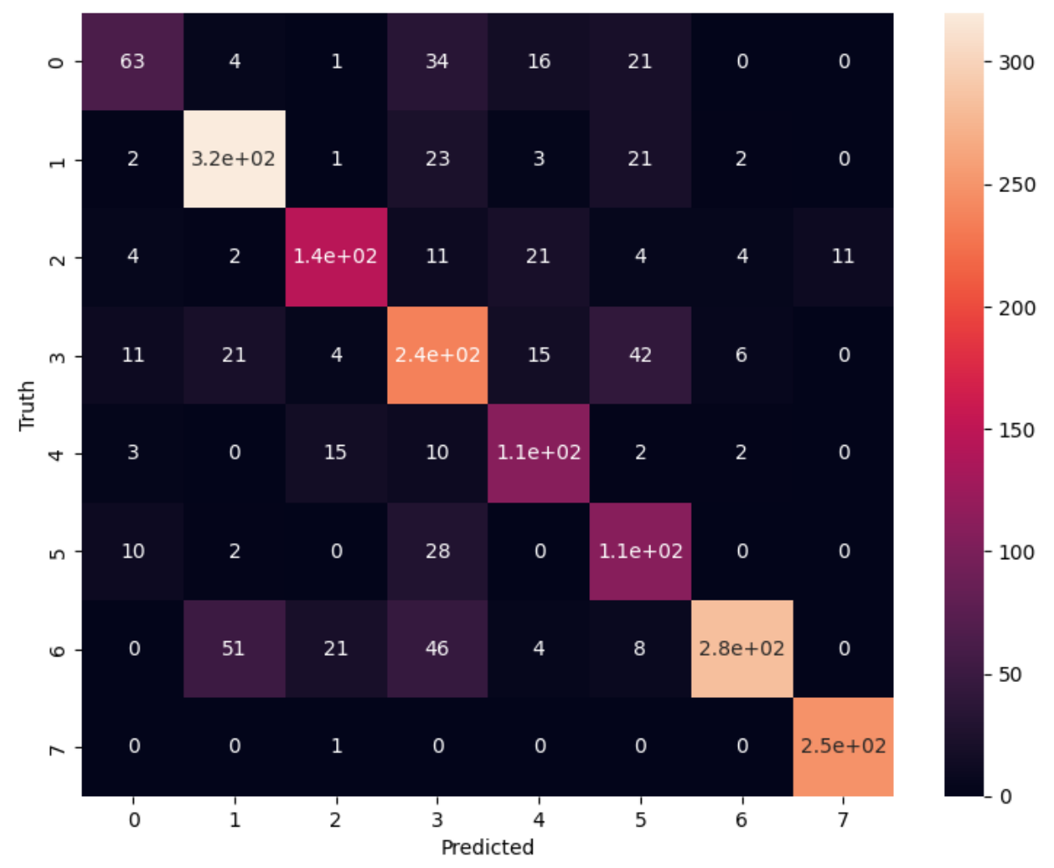
The datasets used for all models were split into 80% training data and 20% validation data. This approach ensured that the models were trained on a sufficiently large dataset while also allowing for an independent evaluation of their performance on unseen data. The validation data was used to fine-tune the hyperparameters of the models and prevent overfitting. Finally, the performance of the models was evaluated on the remaining 20% of the data to measure their generalization ability.

KNN: It was selected due to its simplicity and ability to perform well with data of low dimensionality. However, KNN can be sensitive to irrelevant features and outliers, and may not perform well on high-dimensional data. To determine the optimal value of k, different values of k ranging from 5-40 were tested, and it was found that the best accuracy was achieved with k=15.

Cross-validation was also used to evaluate the performance of the KNN model, and it achieved an accuracy of [0.7535 0.7555 0.757 0.7525 0.762].



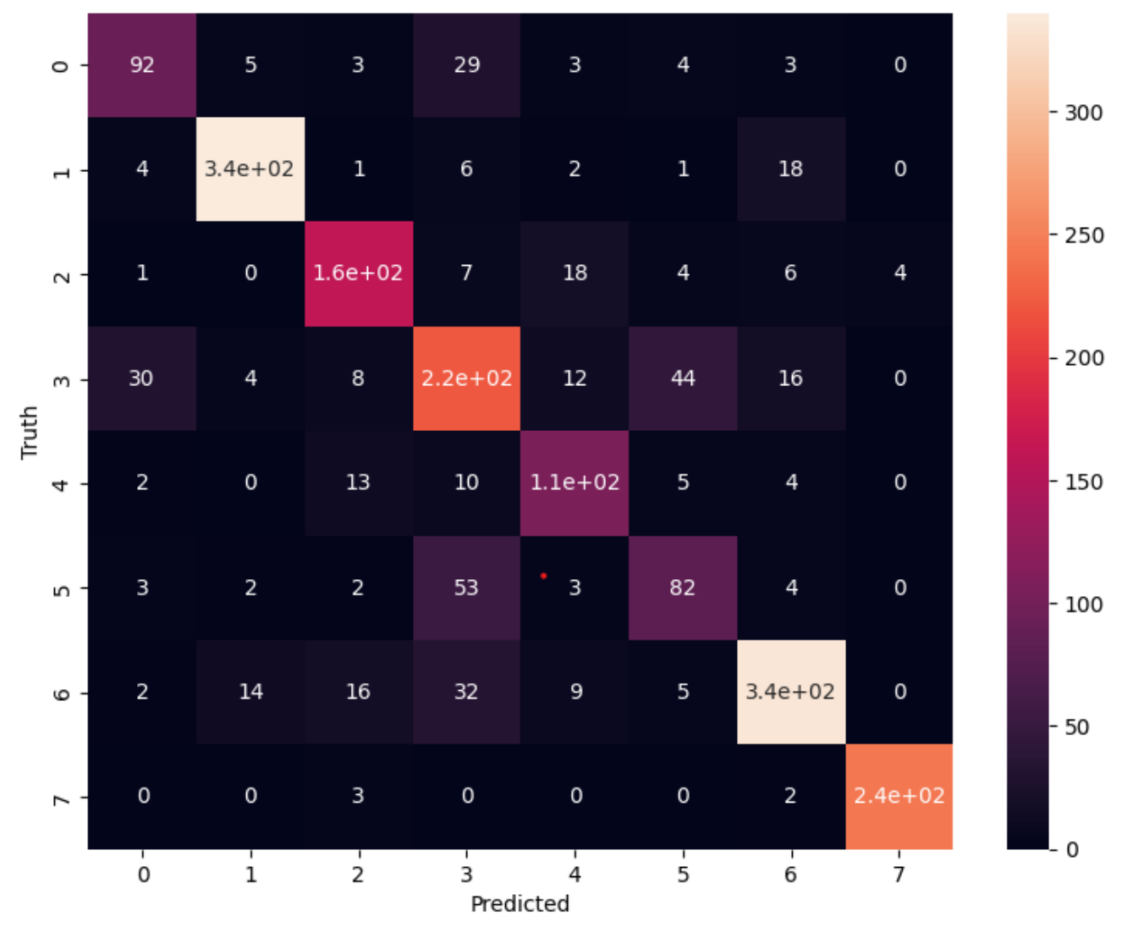
```
Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.45      0.54       139
           1       0.80      0.86      0.83       372
           2       0.77      0.72      0.74       202
           3       0.61      0.70      0.65       335
           4       0.65      0.77      0.71       141
           5       0.53      0.73      0.61       149
           6       0.95      0.69      0.80       413
           7       0.96      1.00      0.98       249

    accuracy                           0.76      2000
   macro avg       0.74      0.74      0.73      2000
weighted avg       0.78      0.76      0.76      2000
```



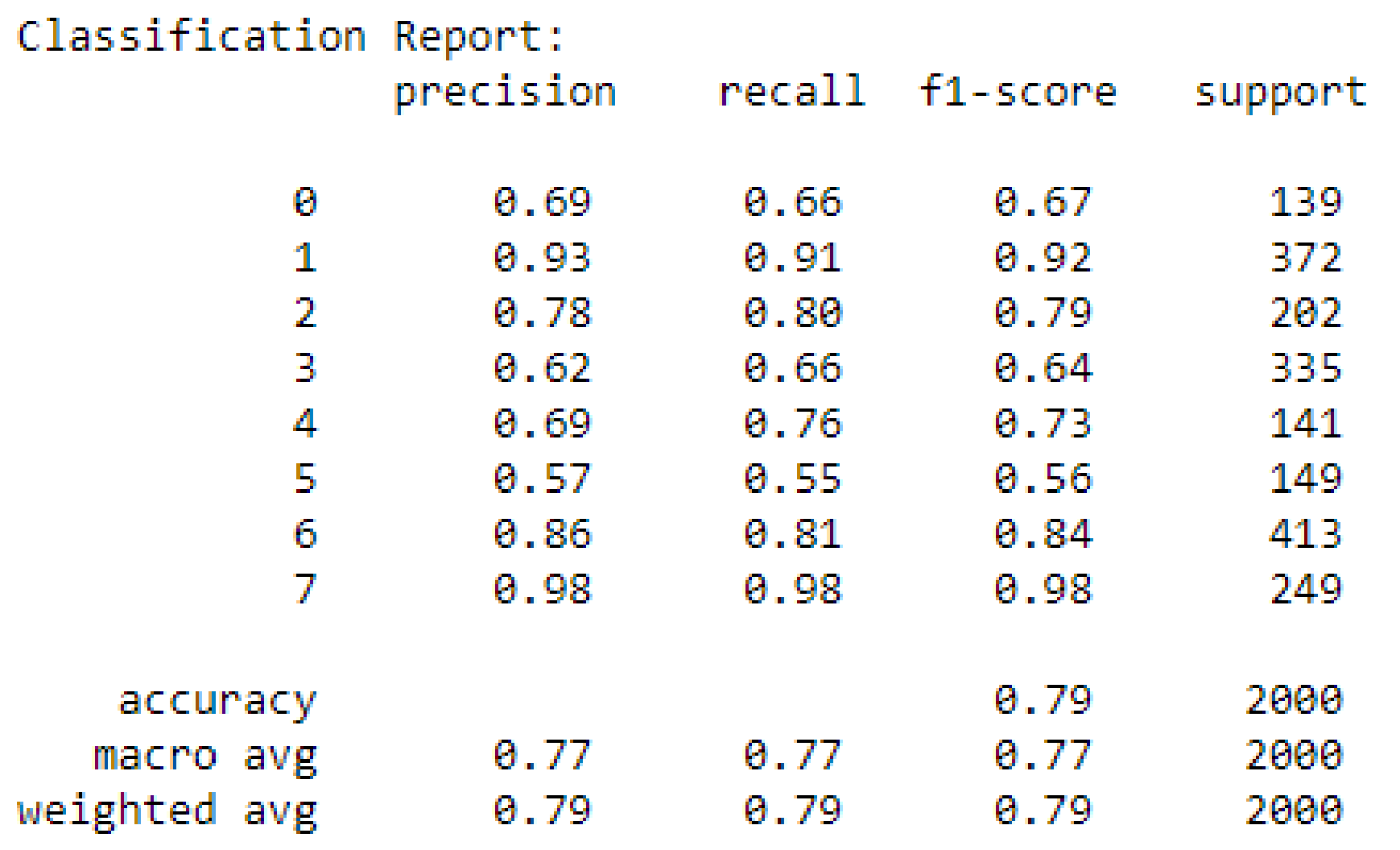
Truth \ Predicted	0	1	2	3	4	5	6	7
0	63	4	1	34	16	21	0	0
1	2	3.2e+02	1	23	3	21	2	0
2	4	2	1.4e+02	11	21	4	4	11
3	11	21	4	2.4e+02	15	42	6	0
4	3	0	15	10	1.1e+02	2	2	0
5	10	2	0	28	0	1.1e+02	0	0
6	0	51	21	46	4	8	2.8e+02	0
7	0	0	1	0	0	0	0	2.5e+02

Logistic regression:

Logistic Regression was selected as the second model to compare with the KNN model. Similar to KNN, logistic regression may also be affected by class imbalance, leading to biased predictions towards the majority class. The assumption of linearity can also limit the model's performance, as it may not be able to capture complex, nonlinear relationships between the features and labels. Despite these limitations, logistic regression can be useful for understanding the data and exploring the relationship between the features and labels. The performance of the logistic regression model was found to be less than 80% accuracy, indicating that it may not be the best model for this particular dataset.



Truth \ Predicted	0	1	2	3	4	5	6	7
0	92	5	3	29	3	4	3	0
1	4	3.4e+02	1	6	2	1	18	0
2	1	0	1.6e+02	7	18	4	6	4
3	30	4	8	2.2e+02	12	44	16	0
4	2	0	13	10	1.1e+02	5	4	0
5	3	2	2	53	3	82	4	0
6	2	14	16	32	9	5	3.4e+02	0
7	0	0	3	0	0	0	2	2.4e+02



```
Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.66      0.67       139
           1       0.93      0.91      0.92       372
           2       0.78      0.80      0.79       202
           3       0.62      0.66      0.64       335
           4       0.69      0.76      0.73       141
           5       0.57      0.55      0.56       149
           6       0.86      0.81      0.84       413
           7       0.98      0.98      0.98       249

    accuracy                           0.79      2000
   macro avg       0.77      0.77      0.77      2000
weighted avg       0.79      0.79      0.79      2000
```

CNN Models:

A few CNN models with different epochs and parameters are shown. Also, plotted graphs are included for better understanding. Many combinations of epochs and parameters were used but only few are mentioned

CNN_1: Epochs = 15

Conv2D(Filters=16, size = (3,3), relu)

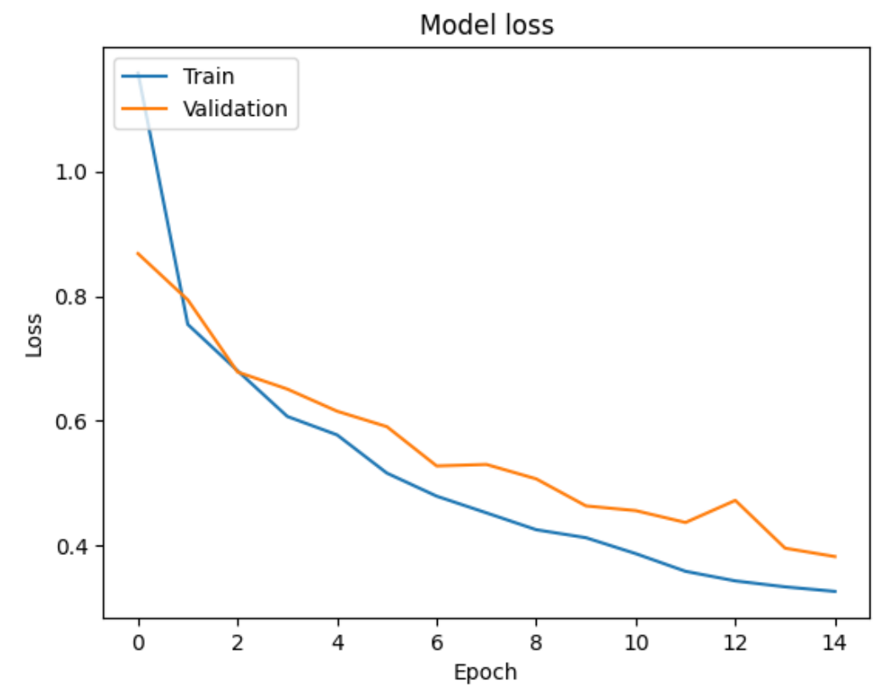
MaxPool(2,2)

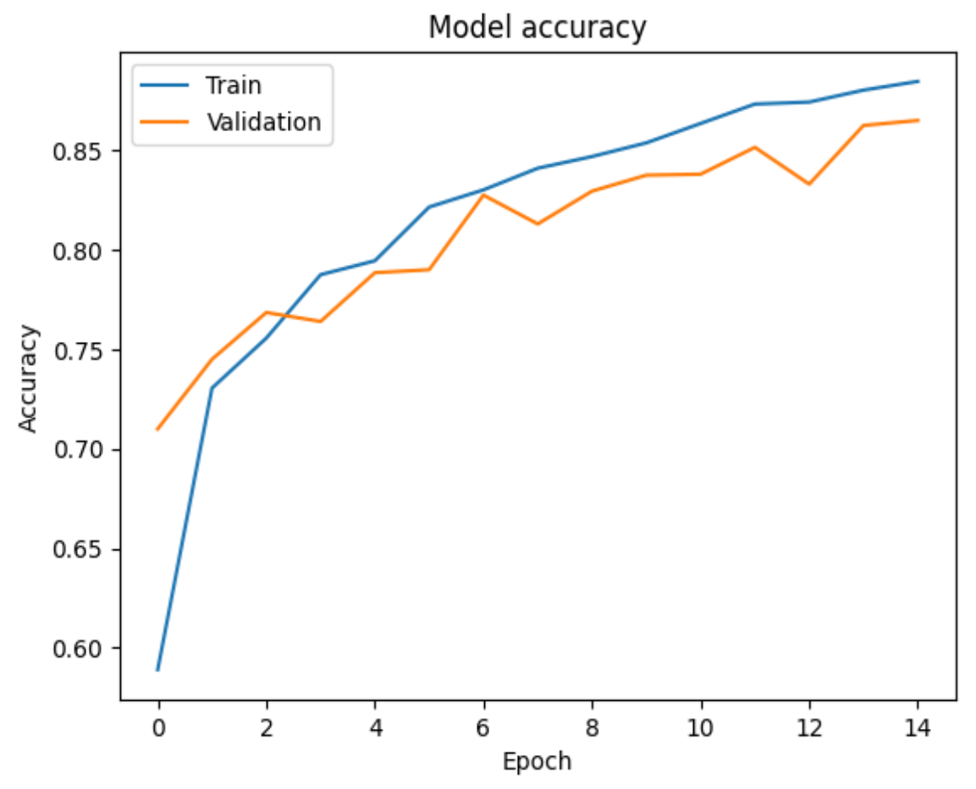
Conv2D(Filters=32, size = (3,3), relu)

MaxPool(2,2)

FlattenLayer(64, relu)

DenseLayer(8, softmax)





CNN_2: Epochs = 35

Conv2D(Filters=32, size = (3,3), relu)

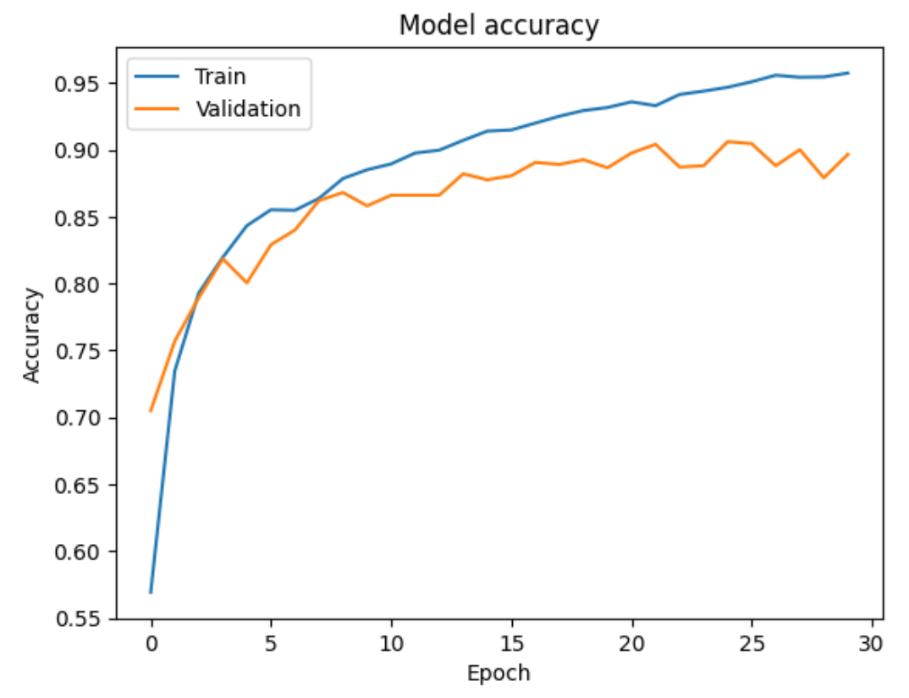
MaxPool(2,2)

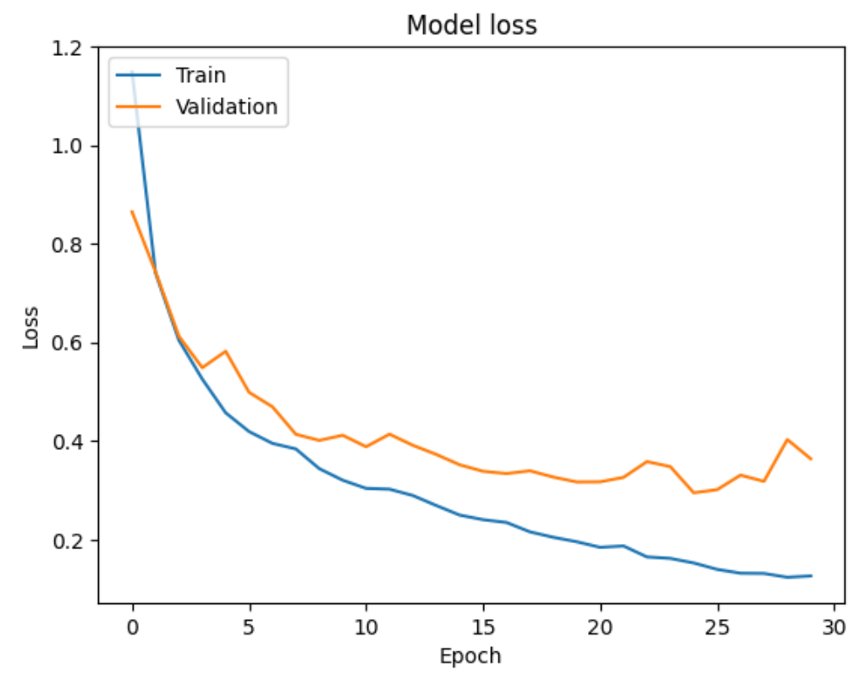
Conv2D(Filters=64, size = (3,3), , relu)

MaxPool(2,2)

FlattenLayer(64, relu)

DenseLayer(8, softmax)





CNN_3/ Final_CNN: Epochs = 50

Conv2D(Filters=32, size = (3,3), Padding = Same,relu)

MaxPool(2,2)

Conv2D(Filters=64, size = (3,3),Padding = Same, relu)

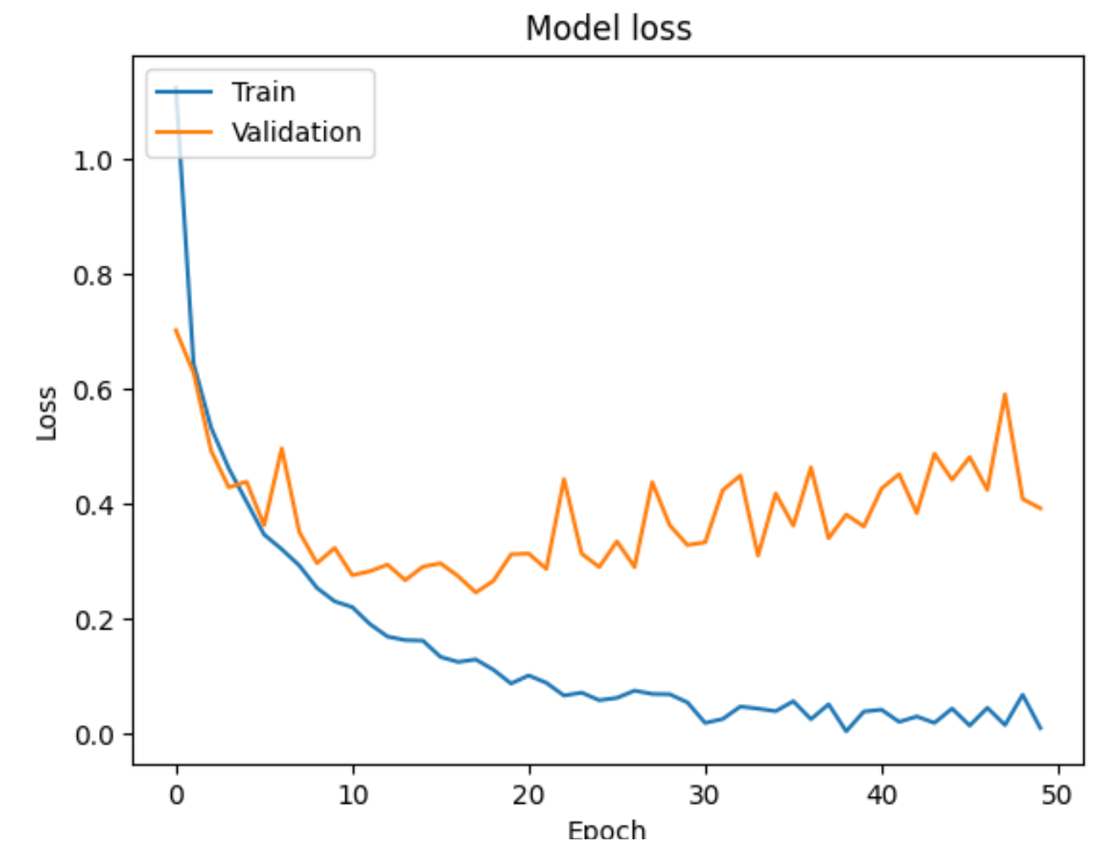
MaxPool(2,2)

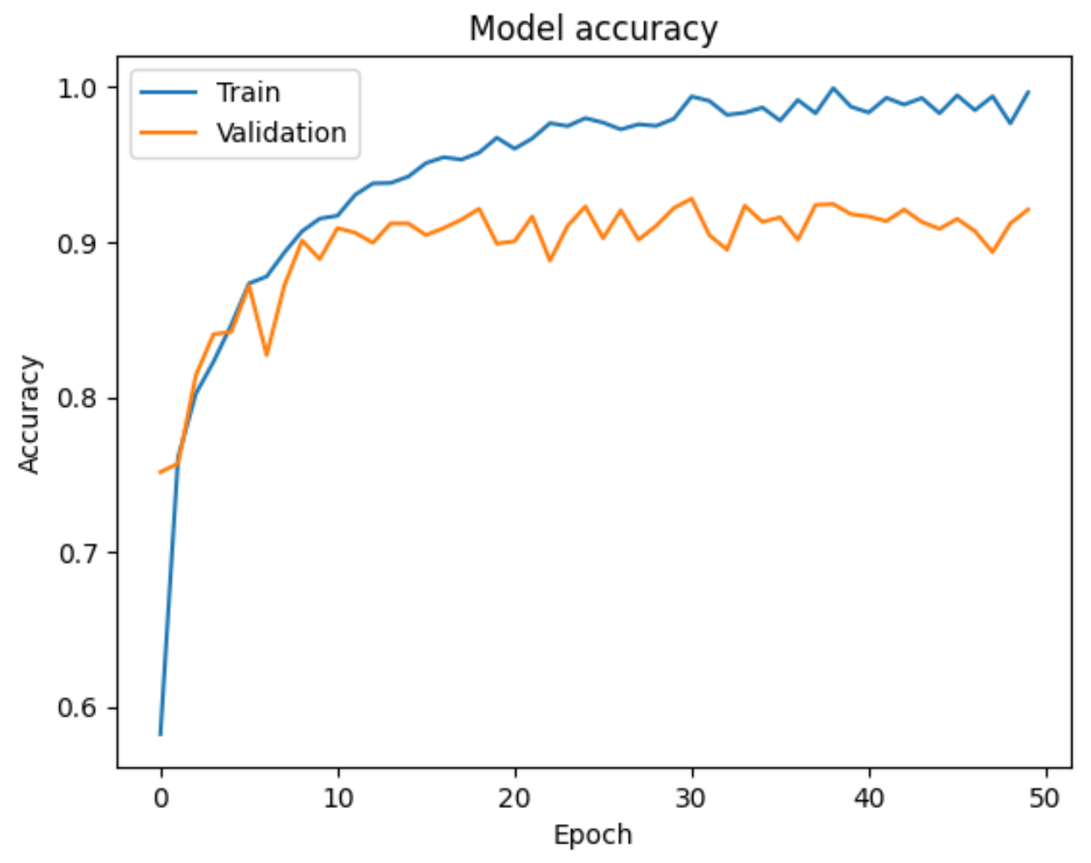
Conv2D(Filters=128, size = (3,3),Padding = Same, relu)

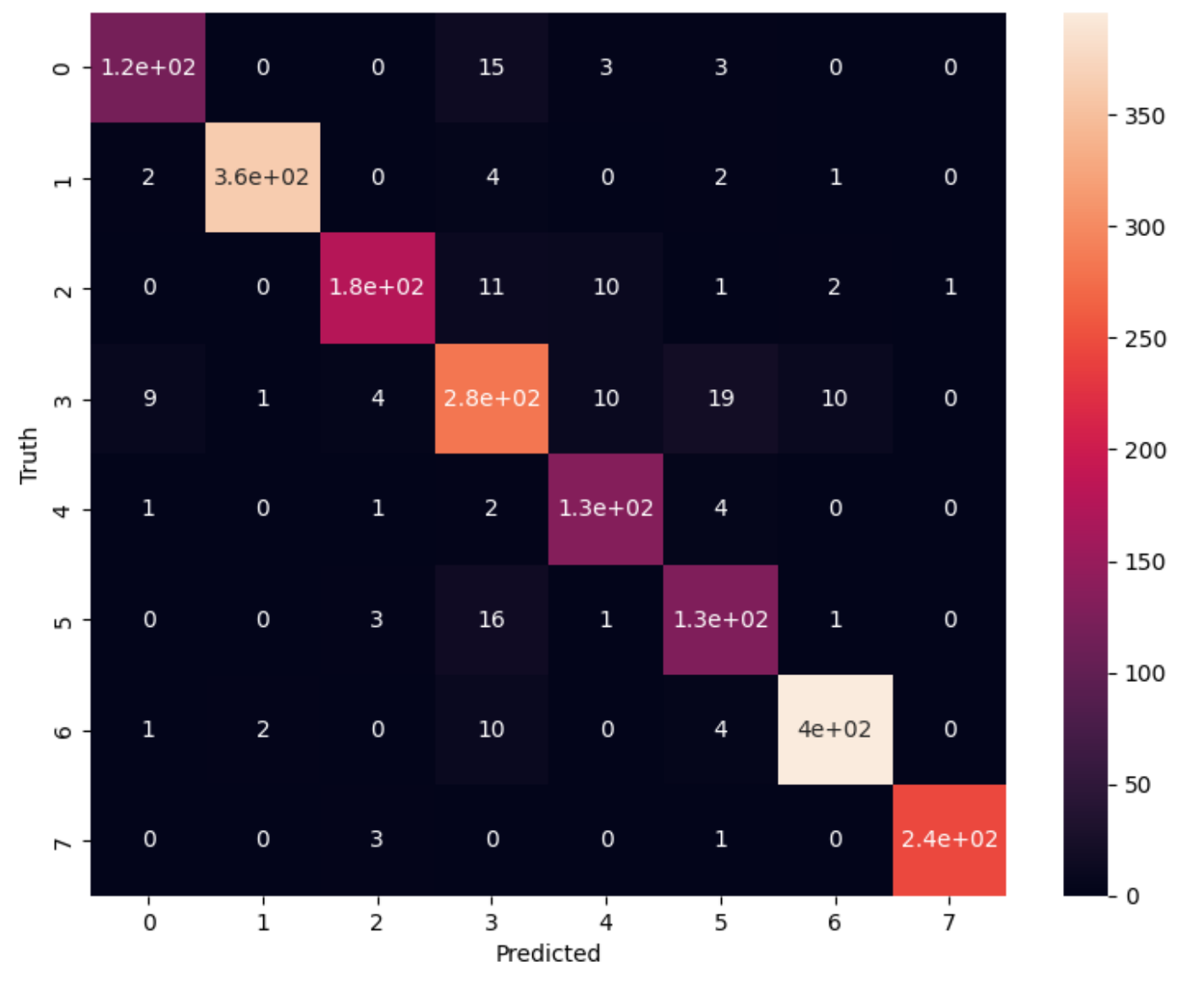
MaxPool(2,2)

FlattenLayer(128, relu)

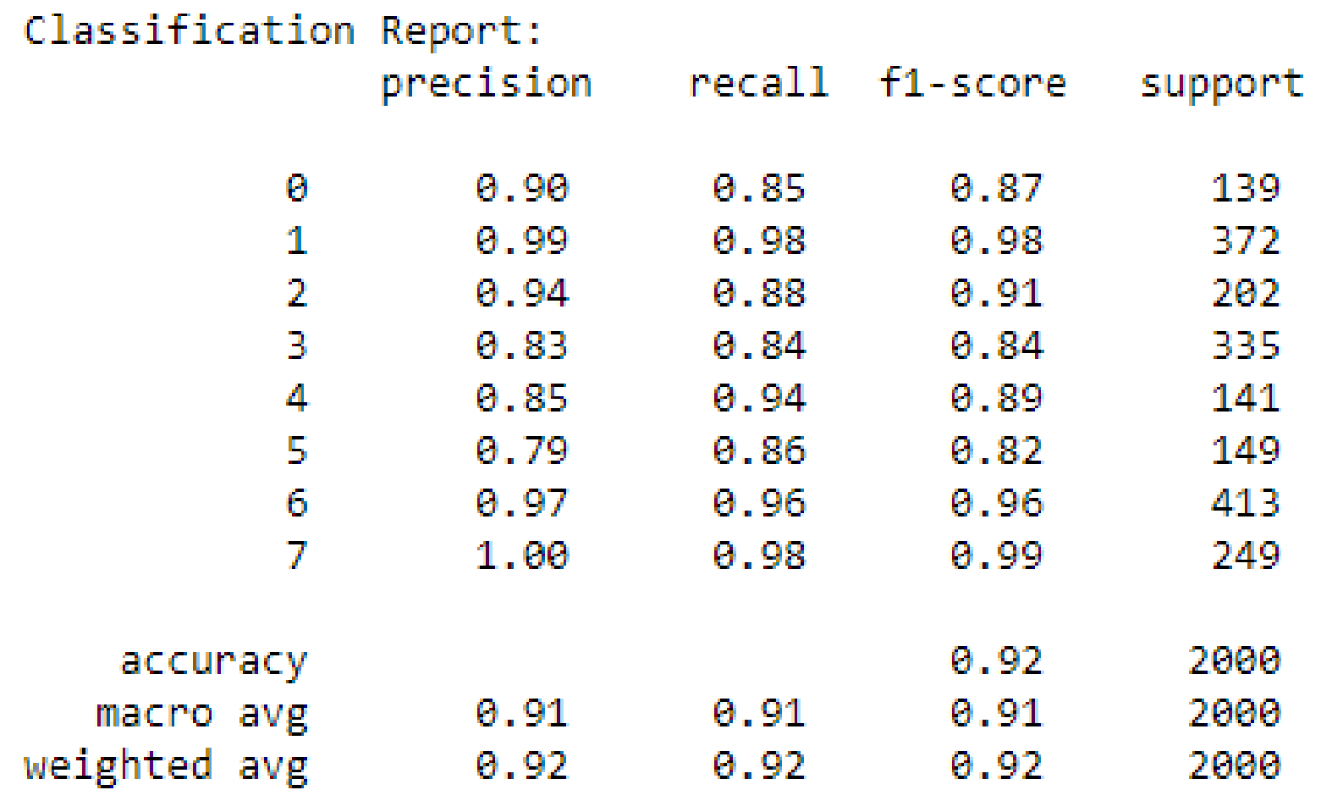
DenseLayer(8, softmax)







Truth \ Predicted	0	1	2	3	4	5	6	7
0	1.2e+02	0	0	15	3	3	0	0
1	2	3.6e+02	0	4	0	2	1	0
2	0	0	1.8e+02	11	10	1	2	1
3	9	1	4	2.8e+02	10	19	10	0
4	1	0	1	2	1.3e+02	4	0	0
5	0	0	3	16	1	1.3e+02	1	0
6	1	2	0	10	0	4	4e+02	0
7	0	0	3	0	0	1	0	2.4e+02



```
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.85      0.87       139
           1       0.99      0.98      0.98       372
           2       0.94      0.88      0.91       202
           3       0.83      0.84      0.84       335
           4       0.85      0.94      0.89       141
           5       0.79      0.86      0.82       149
           6       0.97      0.96      0.96       413
           7       1.00      0.98      0.99       249

    accuracy                           0.92      2000
   macro avg       0.91      0.91      0.91      2000
weighted avg       0.92      0.92      0.92      2000
```

Conclusion:

The implementation of different models and parameters was limited due to the availability of computational resources and a small number of samples. However, techniques such as histogram equalization and data augmentation were used to improve the performance of the CNN model in different trail and error check. Despite these limitations, the models were trained and evaluated on many different metrics including accuracy, precision, recall, f1-score and overall 90% accuracy is achieved.

Reference: 1.https://www.hindawi.com/journals/mpe/2021/3366057/   2. . https://www.sciencedirect.com/science/article/pii/S1046202319300349 3.https://ieeexplore.ieee.org/abstract/document/8301998

Image of Neural network: Reference-1, Blood Samples: Project's own dataset