**Report**

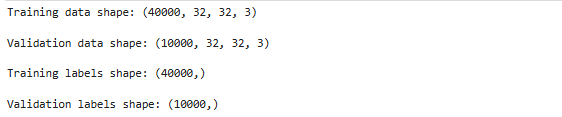
**Jawad Ahmad**

**Image Classification Using Convolutional Neural Networks (CNN) with CIFAR-10 Dataset**

This report outlines the implementation of a Convolutional Neural Network (CNN) for image classification, following the assignment objectives. It discusses the dataset selection, preprocessing, CNN architecture, training and evaluation processes, and results.

**Data Preprocessing**

* **Dataset**: CIFAR-10, consisting of 32x32 color images across 10 categories.
* **Loading** **Data**: The CIFAR-10 dataset was loaded using the **keras.datasets** module.
* **Normalization**: Pixel values were normalized to the range [0, 1] by dividing by 255.0.
* **Reshaping**: The dataset was reshaped into 4D tensors compatible with TensorFlow.
* **Data Split**: The training data was divided into training and validation sets (80:20 split).



**CNN Architecture Design**

**Architecture Components:**

* **Convolutional Layers: Used to extract features from the images.**
* **Pooling Layers: Used to reduce the spatial dimensions of the feature maps.**
* **Fully Connected Layers: Used for classification based on the extracted features.**
* **Activation Functions: ReLU was used for hidden layers, and softmax was used for the output layer.**

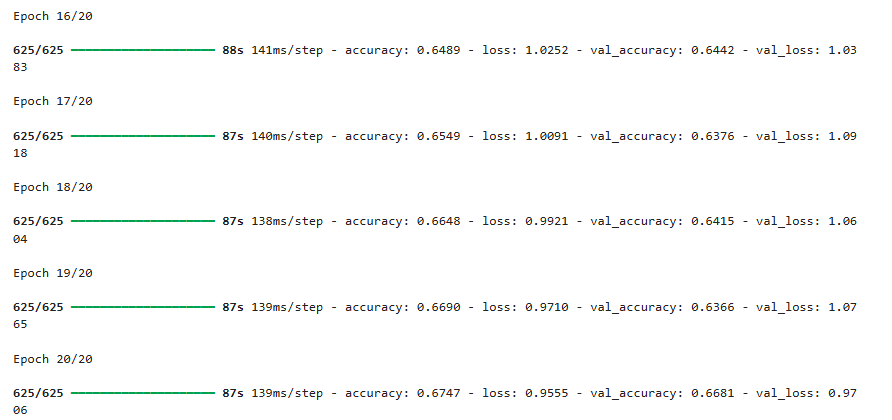
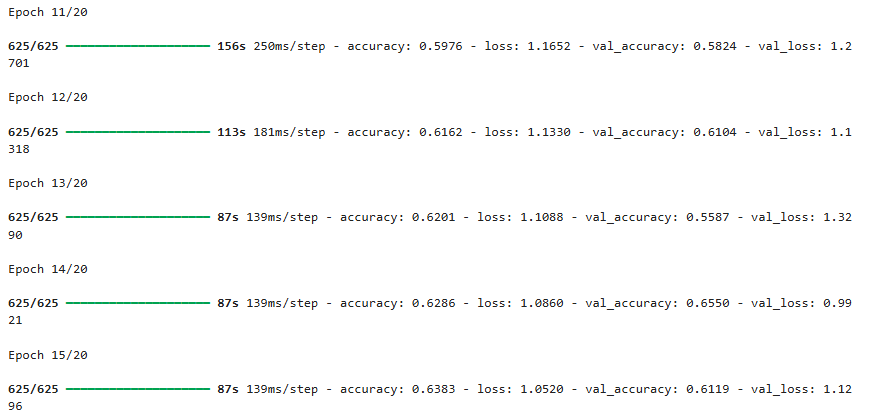
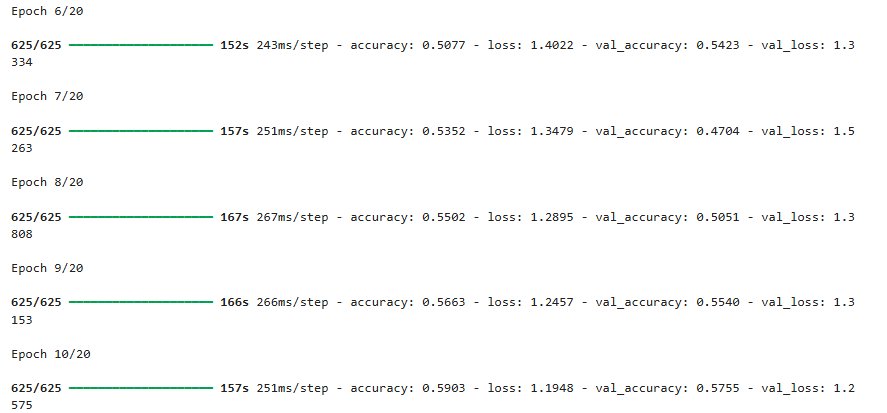
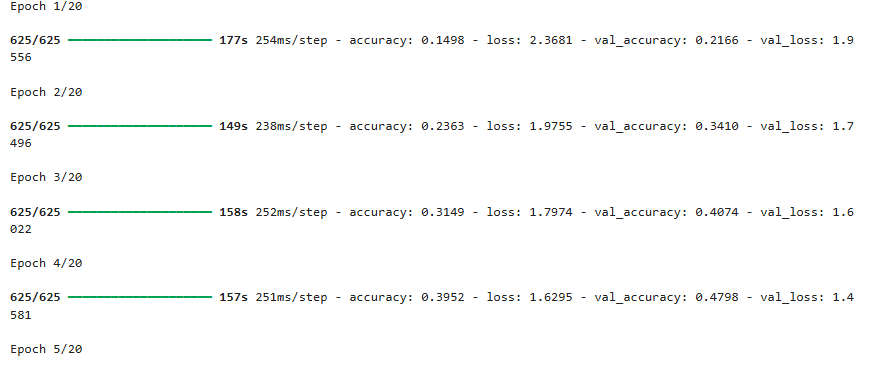
**Model Compilation**

* **Loss Function:**Categorical cross-entropy was used as the loss function.
* **Optimizer:**Adam optimizer was selected for training the model.
* **Metrics:**Accuracy was chosen as the evaluation metric.



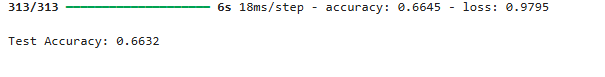
**Model Training**

* The model was trained for 20 epochs with a batch size of 64.
* Training and validation accuracy and loss were monitored.

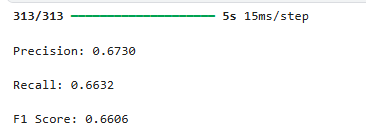
****

**Model Evaluation**

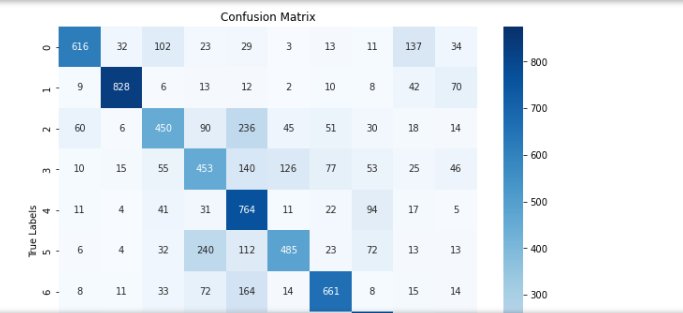
* The trained model was evaluated on the test dataset.
* Key performance metrics such as accuracy, precision, recall, and F1 score were calculated**.**

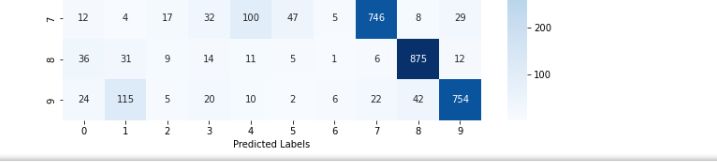
****

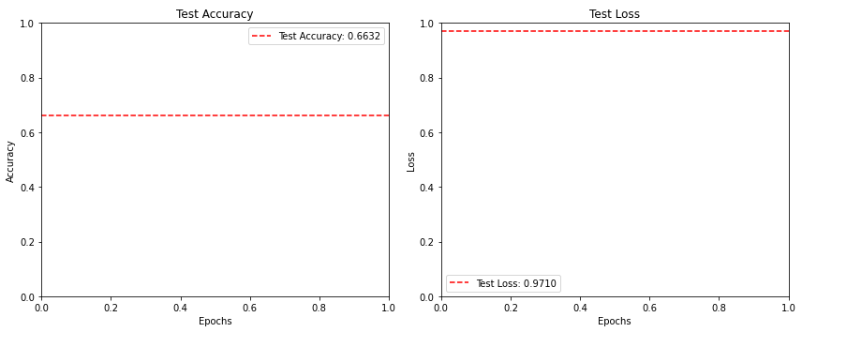
****

**  
Confusion Matrix:**

**This was used to visualize the performance of the model across different classes, helping to identify which classes were misclassified and to what extent.**

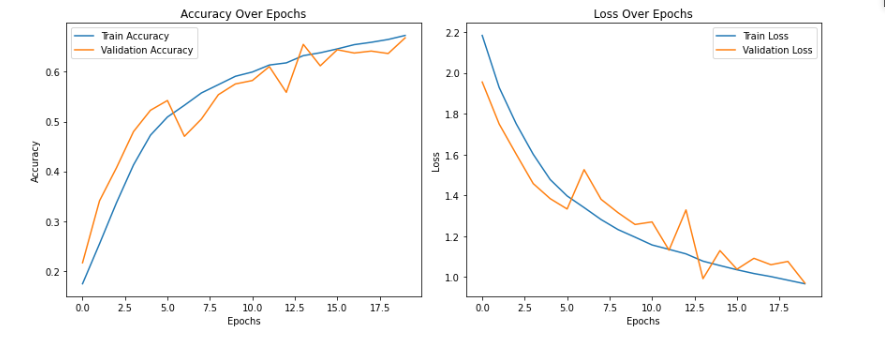
****

**  
Test\_data visualization:**

****

**Train\_data & Validation data visualization**

These plots were generated to visualize the model's performance over epochs, allowing us to assess whether the model is learning effectively and whether overfitting is occurring**.**

****

**Model Training after augmentation**

Data augmentation is a powerful technique that I have utilized to enhance the performance of our CNN model on the CIFAR-10 dataset. By increasing the diversity of the training data, I have improved the model's ability to generalize, which is reflected in the accuracy achieved on the test data. This technique is essential in modern deep learning practices, especially for image classification tasks**.**

**Results Obtained**

**Performance Metrics**

Despite implementing several effective techniques to enhance the performance of the Convolutional Neural Network (CNN) on the CIFAR-10 dataset, the accuracy achieved on the test data did not show a substantial increase**.**

* **Test Accuracy: 62.57%**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**