Introducing AB-Image Classification

The AB-Image Classification project aims to develop a robust machine-learning model to classify images from the CIFAR-10 dataset. The CIFAR-10 dataset consists of 60,000 32x32 color images divided into 10 classes, each containing 6,000 images. The primary goal of this project is to accurately classify images into one of the ten categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

This report outlines the methods used in developing the model, the results obtained during evaluation, and a discussion of the findings and implications of the model's performance.

Image set CIFAR 10

By Ankita & Bruno

Methods

Data Preprocessing

- **Dataset Loading:** The CIFAR-10 dataset was loaded and split into training, validation, and test sets.
- **Image Augmentation:** Techniques such as rotation, zoom, and flipping were applied to increase dataset variability and help prevent overfitting.
- **Normalization:** The pixel values of the images were normalized to a range of [0, 1] to improve the convergence of the neural network during training.

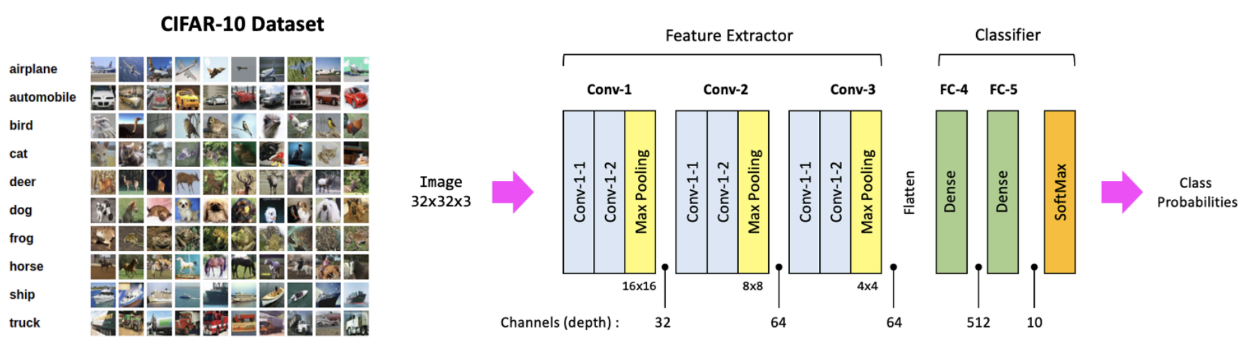
Feature Engineering

- **One-Hot Encoding:** Class labels were converted into a binary matrix representation, allowing the model to better understand and process categorical data.

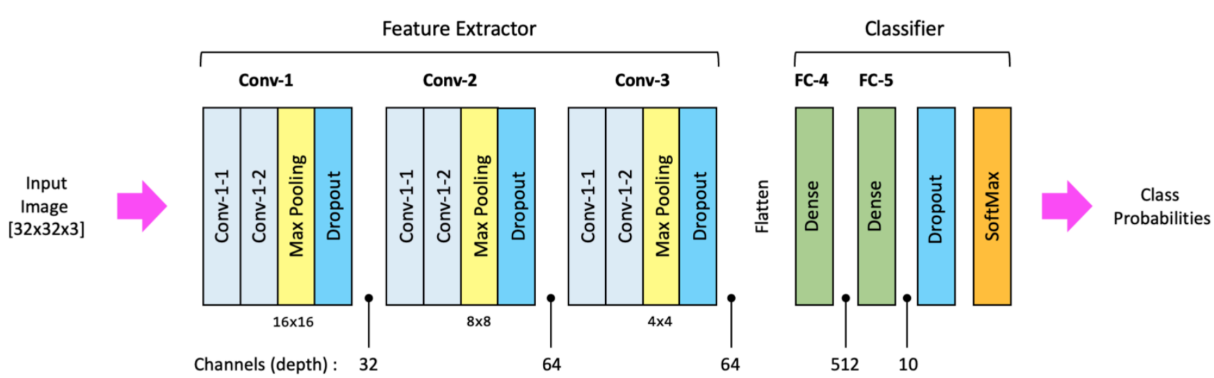
Model Architecture

- **Convolutional Neural Network (CNN):** A CNN architecture was designed, consisting of multiple convolutional layers followed by pooling layers to extract features from the images. The architecture included:
    - Convolutional layers with ReLU activation functions (Feature extractor)
        - The first layer consists of 32 filters, (3, 3) size of the filter, padding = 'same' - this ensures the output feature map has the same spatial dimensions (32X32) as the input.
        - Similarly, the second layer and the third layer consist of 64 filters, (3, 3) size of the filter, padding = 'same' - this ensures the output feature map has the same spatial dimensions (64X64) as the input.
    - Max pooling layers to reduce dimensionality
    - Dropout layers to prevent overfitting
        - We have mentioned the dropout percentage of 0.25, 0.25, 0.25, 0.5 for each layer.

- Classification layer (Classifier)
    - It takes the 1D vector from the previous layer as an input.
    - It's a final dense layer with softmax activation for multi-class classification

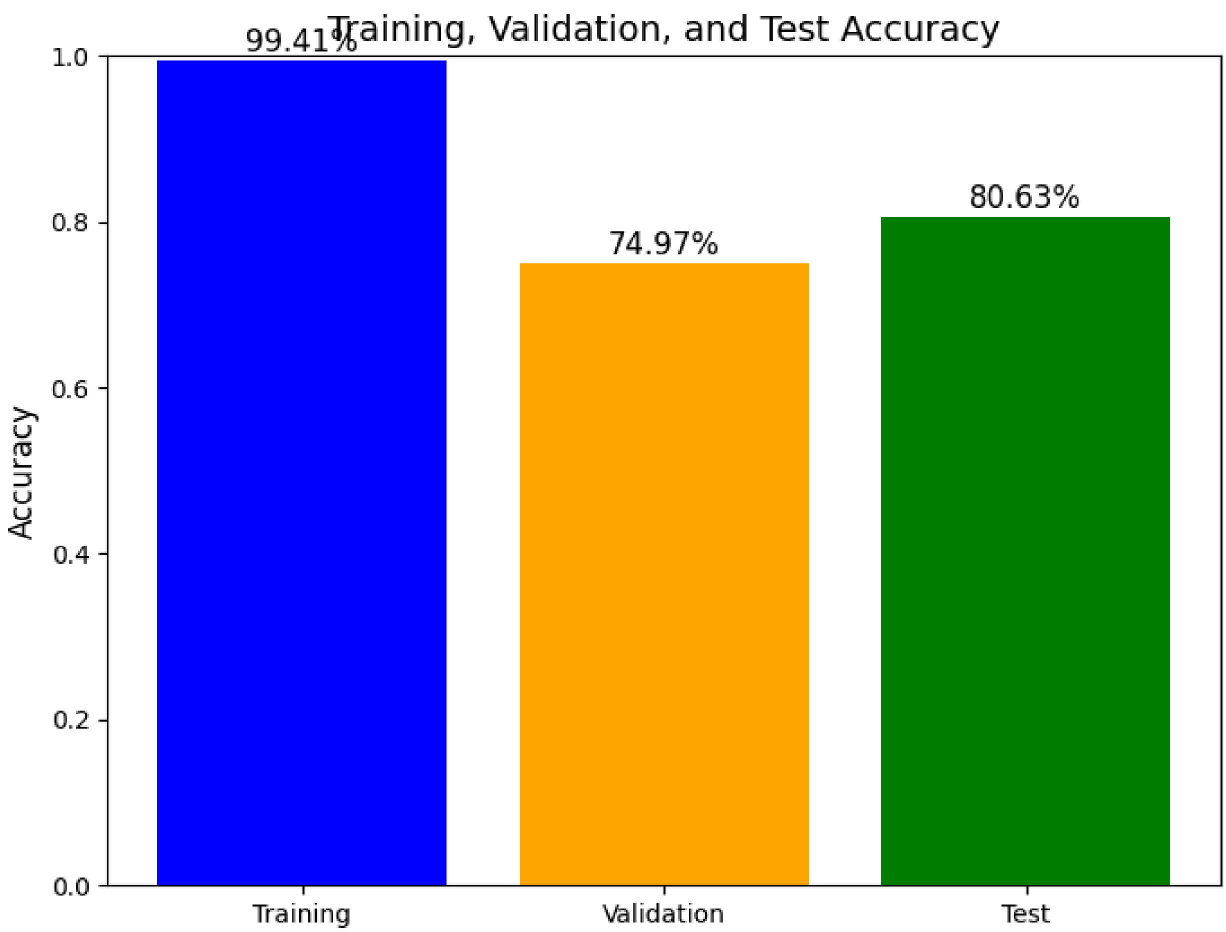


CNN architecture



CNN architecture with Dropout

- **Model Performance:** The model achieved an accuracy of over 80% on the test set after training for 50 number of epochs. Measured using accuracy check and the confusion matrix.



Accuray after optimization (before fine tunning)

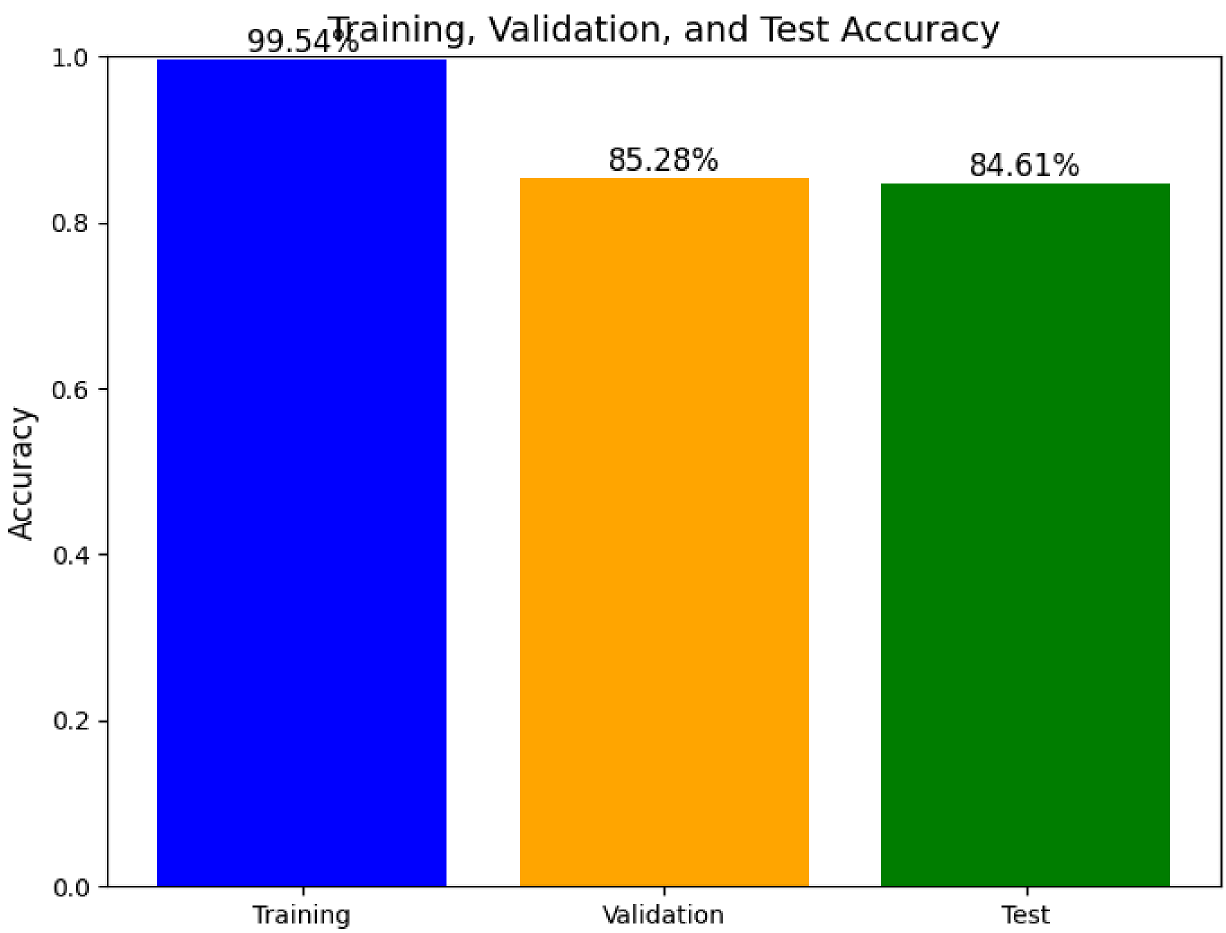
Transfer Learning

- **Pre-trained Model Utilization:** A pre-trained model (e.g., VGG16) was employed to leverage learned features, followed by fine-tuning to adapt the model to the CIFAR-10 dataset.

- **Model Performance:** The model achieved an accuracy of over 84% on the test set after training for 101 epochs using the EarlyStopping optimization technique.

Optimization Techniques

- **EarlyStopping**: This technique monitored validation loss and halted training if no improvement was observed for a specified number of epochs (patience), restoring the best weights.
- **ModelCheckpoint**: This technique saved the model's weights at the end of each epoch, ensuring the best-performing model was available for further training or inference.



Accuracy comparison after the fine tunning

Evaluation Metrics

- The model's performance was assessed using metrics such as accuracy, and a confusion matrix was generated to visualize the classification results.

Results

The model was evaluated on the test dataset, and the following results were observed:

- **Test Accuracy**: The model achieved an accuracy of **84%** on the test dataset.
- **Confusion Matrix**: The confusion matrix revealed the classification performance across all classes, highlighting areas where the model performed well and struggled.

Discussion

The results indicate that the AB-Image Classification model effectively classifies images from the CIFAR-10 dataset, demonstrating the benefits of leveraging transfer learning with a pre-trained model. The application of data augmentation and dropout layers played a crucial role in enhancing the model's robustness and generalization capabilities.

Strengths

- **High Accuracy**: The model achieved a high accuracy rate, suggesting that the transfer learning approach effectively utilized the knowledge from the pre-trained model.
- **Generalization**: Data augmentation helped the model generalize better to unseen data, improving overall performance.

Limitations

- **Class Imbalance**: Some classes may have been underrepresented, potentially leading to lower performance in those categories.
- **Model Complexity**: The complexity of the VGG16 architecture may lead to longer training times and resource consumption, which could be a limitation for deployment in resource-constrained environments.

Future Work

- Further research could focus on exploring other architectures and techniques, such as ensemble methods, to improve accuracy further.
- Investigating more advanced data augmentation strategies and fine-tuning hyperparameters can enhance performance.
- Deployment of the model to the public domain using Tensorflow serving.
- Error handling and conditions to check the image file using Flask for deployment.
- Include Precision, Recall, and F1-Score for checking performance.
- Change the image size to (64X64) to check if the accuracy increases.