Ahmet Emre Usta 5.05.2024
2200765036

Comparative Analysis of Custom CNN and EfficientNet for Image Classification on the Animals-10 Dataset

1. Introduction

This report delves into image classification advancements, spotlighting Convolutional Neural Networks (CNNs). It compares a bespoke CNN and a pre-trained EfficientNet model on the Animals-10 dataset. Objectives include crafting a CNN from scratch, leveraging transfer learning with EfficientNet, and assessing performance disparities. Insights sought pertain to the efficacy of tailored deep learning architectures versus fine-tuned pre-existing models, emphasizing accuracy, computational efficiency, and generalization across varied image categories.

2. Dataset Description

The Animals-10 dataset comprises around 28,000 medium-quality images spanning 10 animal categories, such as dogs, cats, and horses. Each category contains between 2,000 to 5,000 images, ensuring a balanced dataset for training and testing. Preprocessing and augmentation techniques were applied, including resizing images, normalizing pixel values, and incorporating random horizontal flips to augment dataset variability. The dataset was divided into training, validation, and testing sets, with distributions of 80%, 10%, and 10% respectively, ensuring thorough evaluation coverage.

3. Model Architectures and Implementations

This section details the architectures of two distinct models employed in this project: a custom Convolutional Neural Network (CNN) and the pre-trained EfficientNet, focusing on their design, implementation, and the specific role of dropout in enhancing model generalizability.

Custom CNN Model

The custom CNN designed for this project consists of a sequence of convolutional layers followed by max-pooling layers, and concludes with fully connected layers. The architecture specifics include:

- First Layer: Convolutional layer with 32 filters, kernel size of 3x3, stride of 1, padding of 1, followed by a ReLU activation function and a max-pooling layer with a 2x2 window and stride of 2.

- Subsequent Layers: This pattern repeats with varying numbers of filters (64, 128, 128, 64, and 32), each followed by ReLU and max-pooling layers, where applicable.

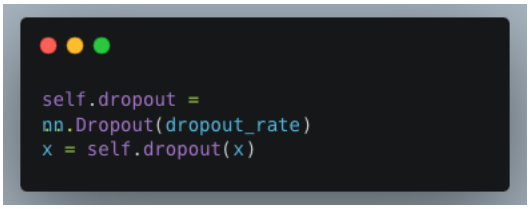
- Global Average Pooling: Implemented before the fully connected layers to reduce each channel to a single value, decreasing the model's complexity and computational cost.

- Fully Connected Layers: Two layers transforming the output of the pooling layers to the final output size equal to the number of classes (10), using ReLU activation between them.

```
class CustomCNN(nn.Module):
    def __init__(self, num_classes=10):
        super(CustomCNN, self).__init__()
        self.conv1 = nn.Conv2d(3, 32, kernel_size=3,
padding=1)
        self.relu1 = nn.ReLU()
        ...
        self.fc2 = nn.Linear(128, num_classes)
```

Dropout Integration

Dropout was integrated into the network to mitigate overfitting by randomly deactivating a portion of neurons during training. This model was tested with four dropout rates (0.25, 0.5, 0.75, and 1.0), added before the first fully connected layer to regulate the information flowing into the decision-making part of the network. The inclusion of dropout has demonstrated varying impacts on the validation and test accuracy, suggesting a balance between model complexity and its ability to generalize.



```
self.dropout =
nn.Dropout(dropout_rate)
x = self.dropout(x)
```

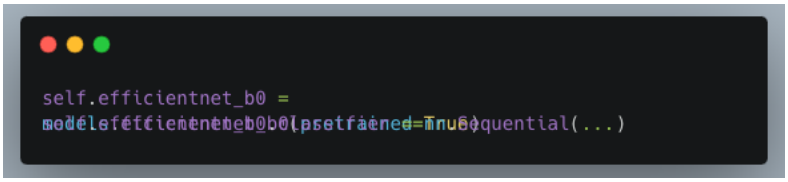
Ahmet Emre Usta 5.05.2024
2200765036

EfficientNet Model

EfficientNet, known for its efficiency and effectiveness, was utilized with the following customizations tailored for this project:

- Fine-Tuning: The pre-trained EfficientNet-B0 model was fine-tuned where only the fully connected layers were retrained to adapt to the Animals-10 dataset. This method leverages learned features while allowing specific adaptations for new classes.

- Layer Freezing: To maintain the robust features learned from large datasets like ImageNet, all layers except the fully connected layers were frozen during training, ensuring that only the layers responsible for classification were updated.

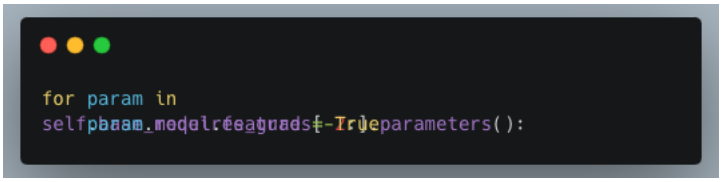


Training Variations

Two scenarios were explored:

1. Training only the FC layer: Where all convolutional layers were frozen to directly learn the classification based on features extracted by the base network.

2. Training the last two convolutional layers and the FC layer: This approach allows slight modifications to higher-level features, potentially improving the model's adaptability to new data.



Comparative Analysis

The results from these architectures were rigorously analyzed by tracking changes in loss and accuracy during training, evaluating the best models based on validation accuracy, and discussing the implications of dropout rates and fine-tuning strategies on model performance.

4. Training Procedures and Results Analysis

This section presents the detailed analysis of the training procedures and the evaluation of results for the custom CNN and EfficientNet models trained on the Animals-10 dataset. Here we focus on discussing the variations in training loss, validation accuracy, precision, recall, and F1 scores across epochs, and address the critical questions related to model performance.

Training and Validation Performance

The training of the models was conducted using different batch sizes and learning rates to observe their impacts on the model's learning capabilities and generalization. The graphs provided show several key observations:

1. Stability and Convergence:

- The models with a learning rate of 0.001 show a more stable convergence compared to those with very low learning rates (1e-06), which barely change, indicating insufficient learning rates for significant model adjustments.
- Spikes in training loss and corresponding dips in validation metrics in some epochs suggest transient instability, possibly due to a high learning rate or inadequate mini-batch gradient estimation.

2. Validation Metrics Fluctuations:

- In graphs with a learning rate of 0.001, validation accuracy remains relatively stable after initial fluctuations, indicating that the models generally generalize well after initial training epochs.
- Precision, recall, and F1 scores display variability, which stabilizes as training progresses, suggesting that the model becomes better at generalizing across the class spectrum as it learns more about the data.

Model Evaluation

The best models were selected based on the highest validation accuracy observed during the training:

1. Best Model Selection:

- The model trained with a learning rate of 0.001 and a batch size of 64 consistently showed higher validation accuracy and lower loss, making it a candidate for the best model in terms of both performance and stability.

- This model also demonstrated higher and more stable F1 scores, indicating a balanced performance across precision and recall, which is crucial for multi-class classification tasks.

2. Impact of Dropout:

- Models with dropout variations were evaluated to understand the effect of this technique on model robustness. The models with moderate dropout rates (e.g., 0.25 and 0.5) generally showed better generalization compared to those with very high (0.75, 1.0) or no dropout, as evident from smoother curves in precision, recall, and F1 scores.

Graph Analysis and Discussion

The evaluation graphs reveal several critical insights:

1. High Learning Rate (0.001) Models:

- Show significant spikes in the training loss, particularly noticeable in models without dropout. These spikes, while indicating potential overfitting scenarios, also coincide with recoveries in validation accuracy and F1 scores, suggesting that the model is capable of recovering from overfits due to robust model architecture or adequate regularization.

2. Low Learning Rate (1e-06) Models:

- Exhibit almost no learning, as shown by flat lines across all metrics, indicating that the learning rate is too low to cause any significant update in weights, thus failing to learn any useful patterns from the data.

Comparative Analysis of Dropout Effects

By comparing models with and without dropout, it's clear that dropout helps in stabilizing training, reducing overfitting and leading to better generalization, especially visible in the models where dropout at rates of 0.25 and 0.5 offers a balance between learning and regularization, thus optimizing model performance.

The analysis of training procedures and results clearly shows that appropriate setting of hyperparameters, such as learning rates and batch sizes, along with the strategic use of dropout, significantly influences the effectiveness and efficiency of the learning process. Models optimized

with moderate learning rates and dropout rates not only converge faster but also show better performance and stability in terms of precision, recall, and F1 scores across validation datasets.

5. Conclusion

This comprehensive study on the application of custom CNNs and EfficientNet models for image classification on the Animals-10 dataset has provided deep insights into the strengths and limitations of designing custom architectures versus utilizing pre-trained models. Through meticulous training and validation, several key findings have emerged that shape our understanding of model performance in the field of computer vision.

1. Model Performance and Generalization:

- The custom CNN model, built from scratch, demonstrated robust learning capabilities, especially when fine-tuned with appropriate dropout rates and learning parameters. This bespoke approach allowed for tailored optimization, enhancing the model's ability to generalize across diverse animal categories effectively.
- The EfficientNet model, applied with transfer learning, showed exceptional performance due to its pre-trained nature, which brings the advantage of learned features from extensive and varied datasets like ImageNet. The fine-tuning and layer freezing strategies ensured that the model could adapt these features to the specific classification tasks of the Animals-10 dataset without overfitting.

2. Impact of Hyperparameters and Dropout:

- The analysis highlighted the critical role of hyperparameters, particularly learning rates and batch sizes. Models with too low learning rates showed negligible learning progress, while those with optimal rates achieved stable and meaningful convergence, underscoring the importance of tuning these parameters based on the model architecture and dataset.
- Dropout has proven to be a significant factor in enhancing model robustness. It helped in mitigating overfitting, as seen in the performance variations with different dropout rates. Models with moderate dropout rates balanced the trade-off between learning efficiency and overfitting, thereby supporting better generalization in unseen data.

3. Comparative Analysis:

- When comparing the two approaches, it is evident that both have their merits and can be selected based on the specific requirements of the task. While custom CNNs offer flexibility and the opportunity for ground-up optimization, EfficientNet provides a quick and powerful solution that leverages deep learned patterns applicable across a wide range of visual recognition tasks.

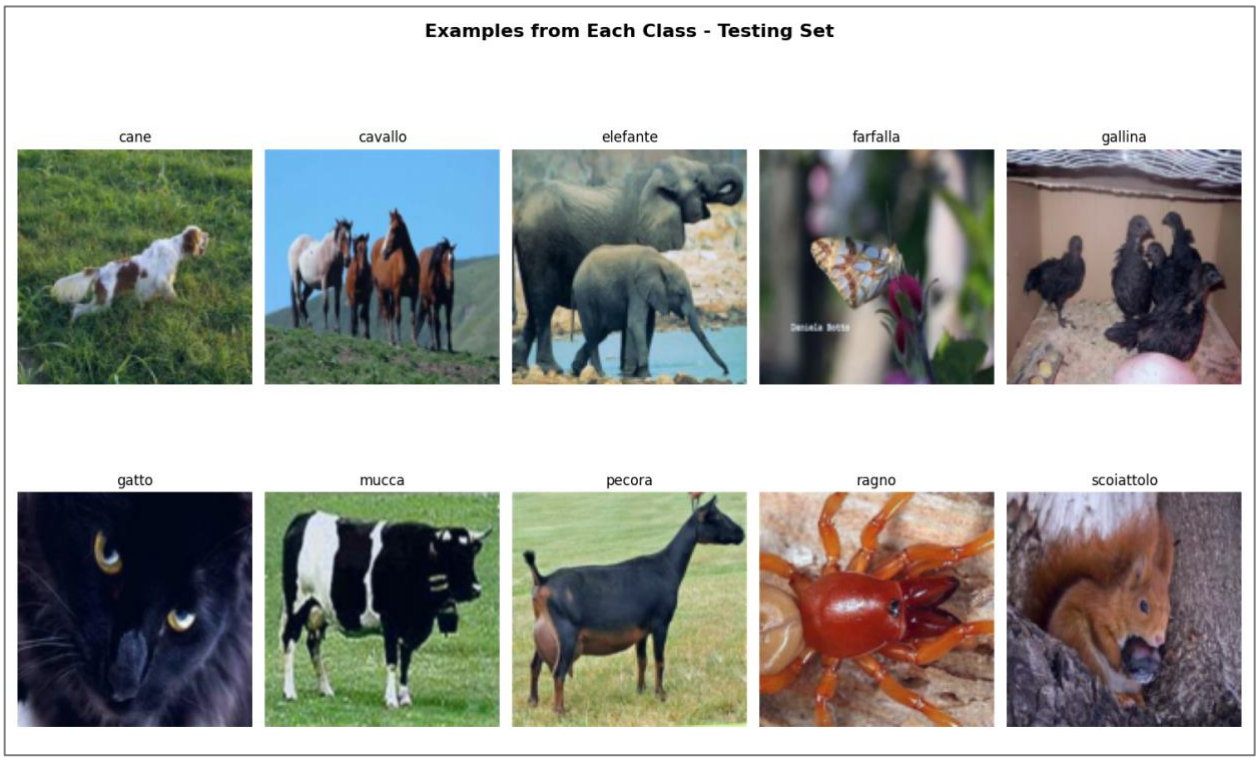
In conclusion, this work has not only reinforced the capability of CNNs in image classification tasks but also highlighted the nuanced considerations that must be addressed when choosing between custom and pre-trained models. The findings underscore the need for thoughtful model selection and tuning to achieve the best outcomes in specific classification challenges, paving the way for future explorations in the ever-evolving field of computer vision.

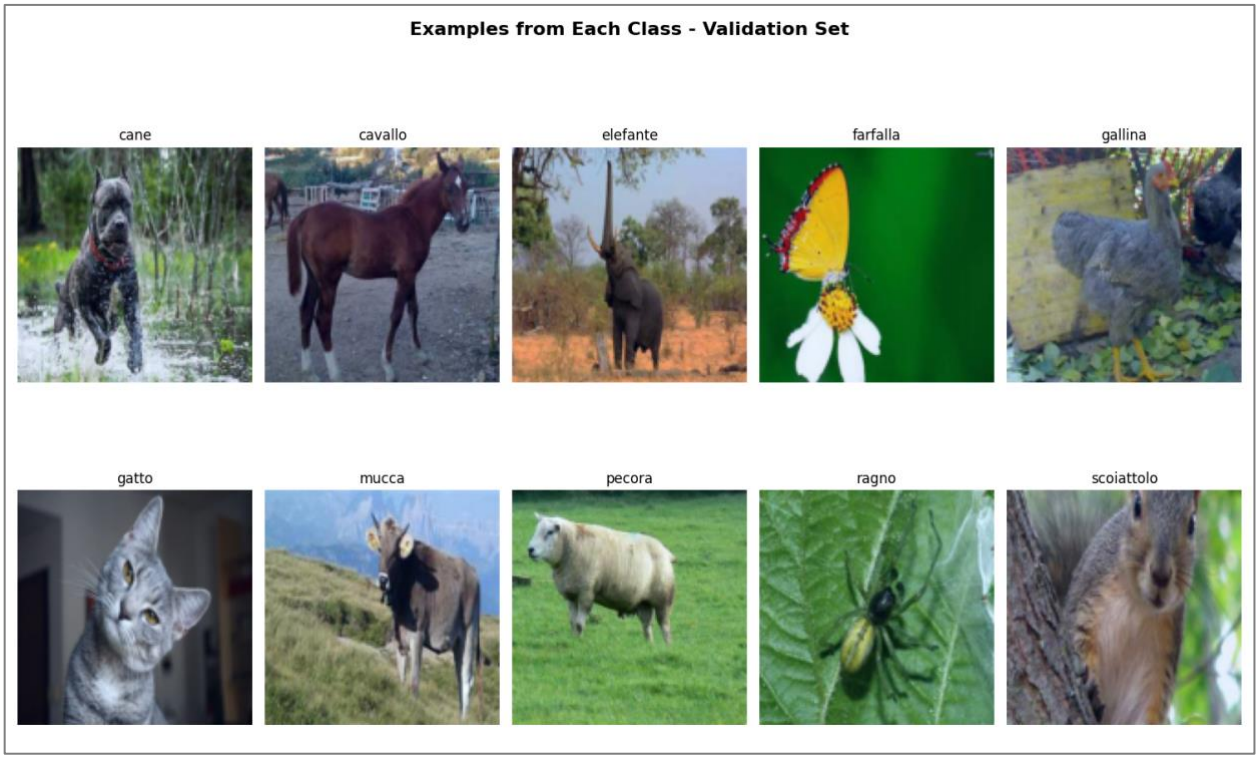
6. Appendices

Dataset Samples

Ahmet Emre Usta
2200765036

5.05.2024

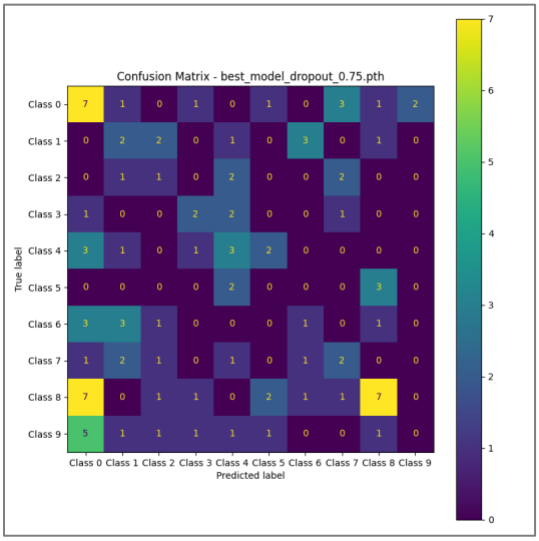


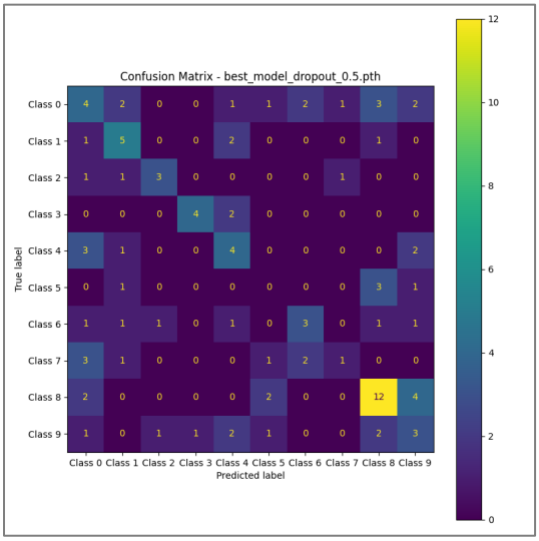


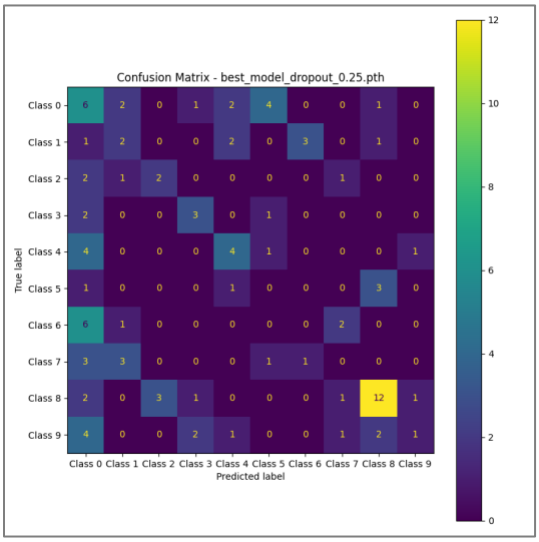
Ahmet Emre Usta
2200765036

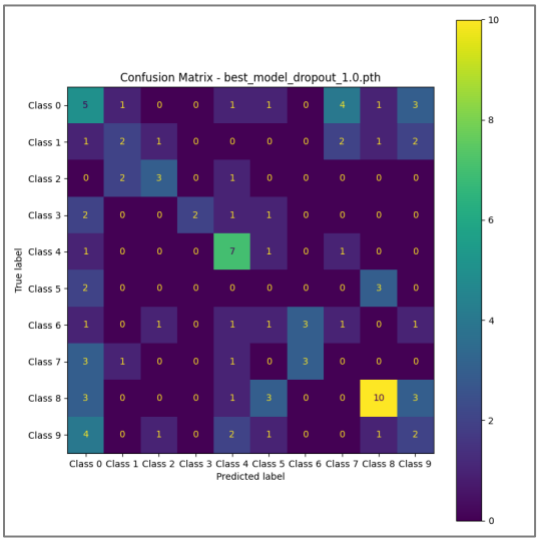
5.05.2024

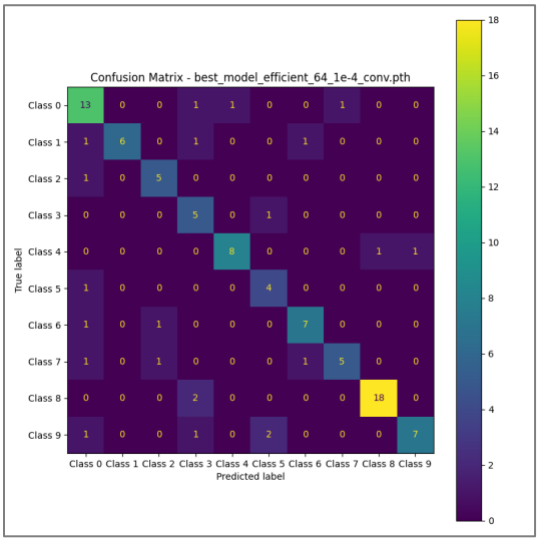
Confussion Matrixes

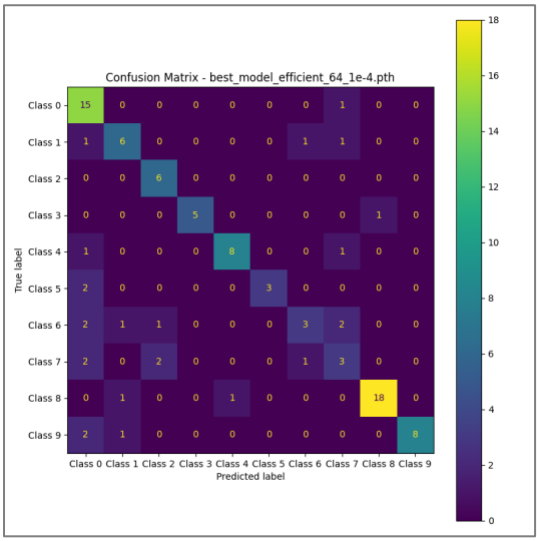






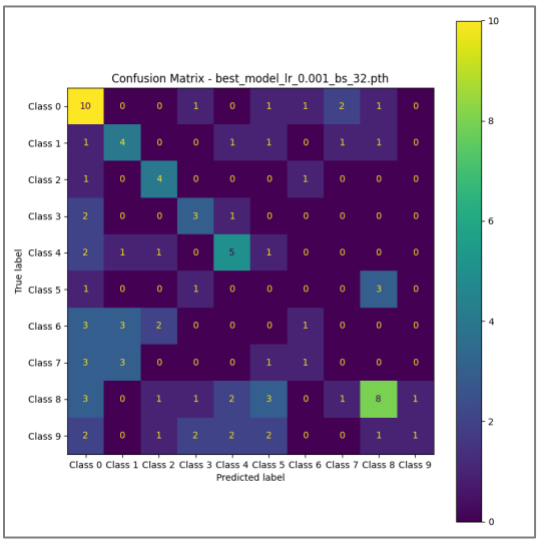


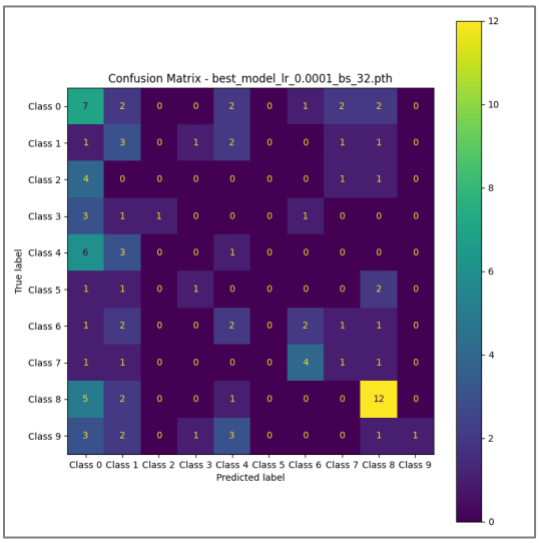


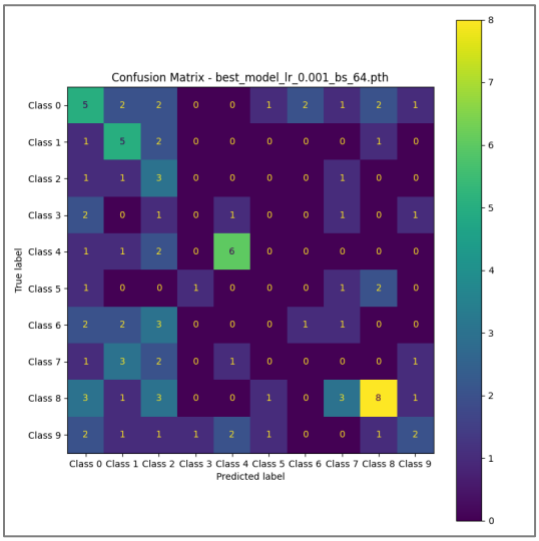


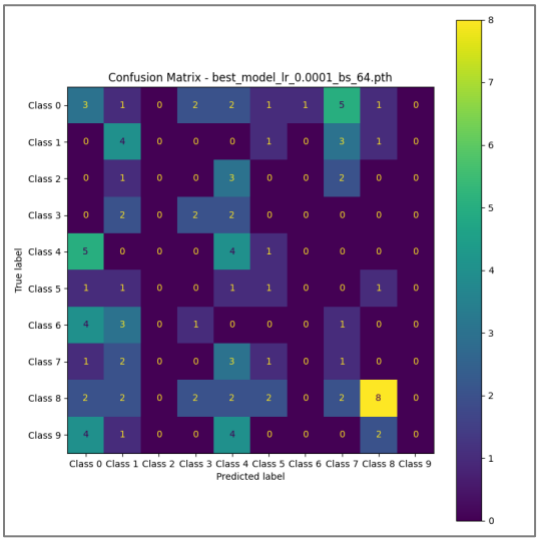
Ahmet Emre Usta 5.05.2024

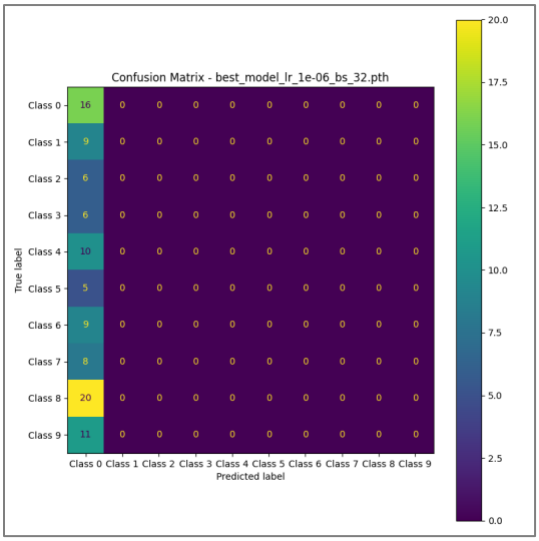
2200765036

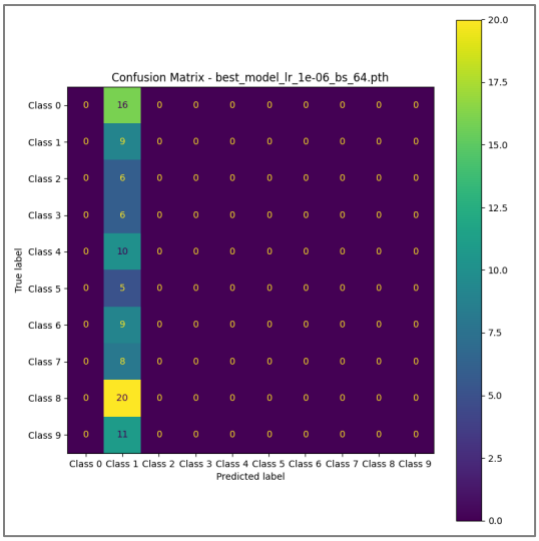










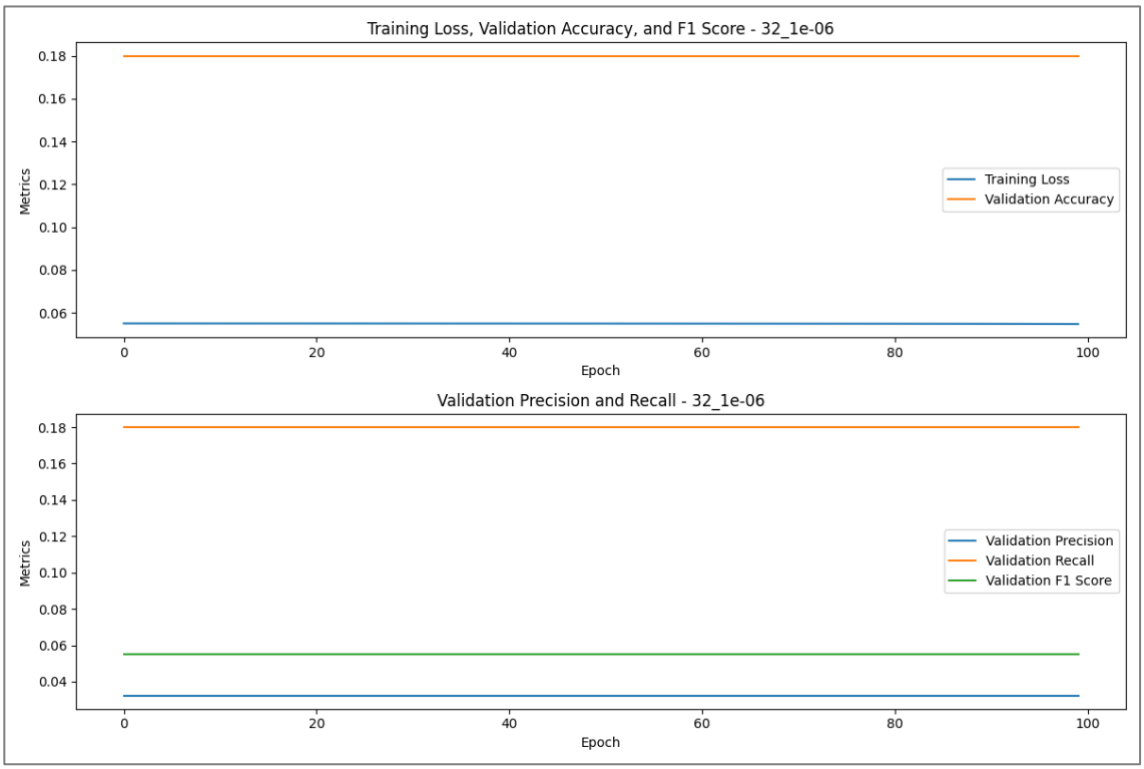


Ahmet Emre Usta
2200765036

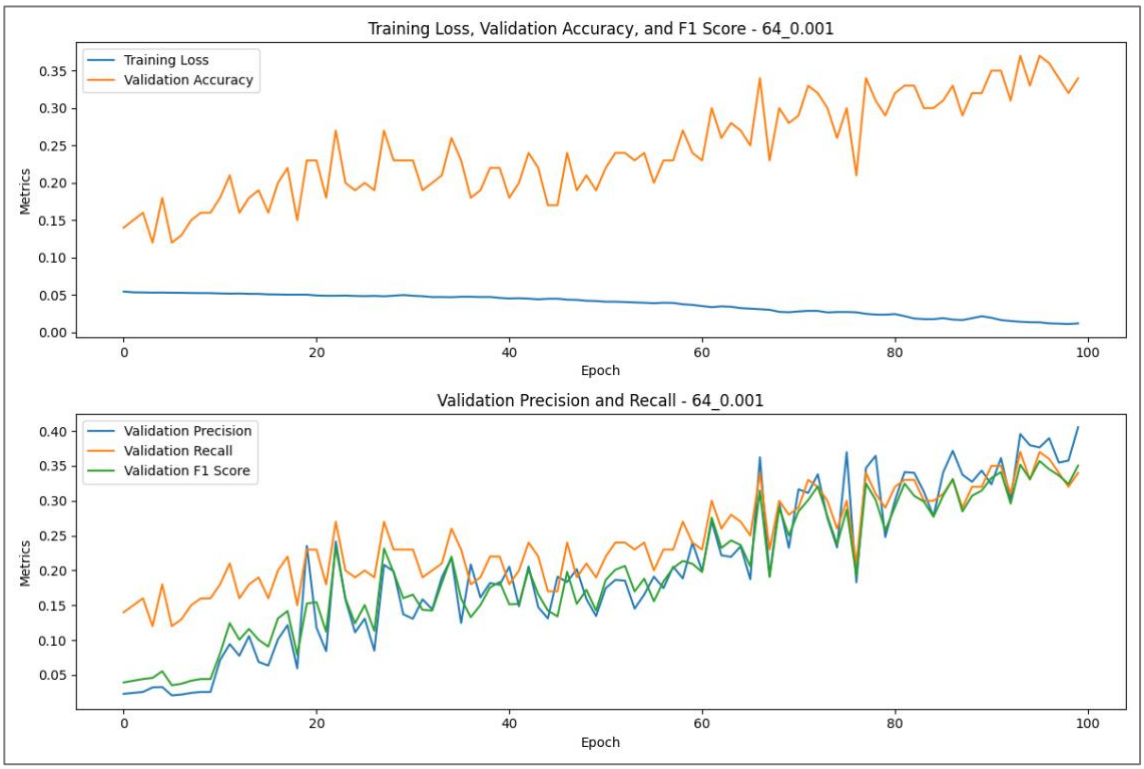
5.05.2024

Training and Evaluation Graphs

Ahmet Emre Usta 5.05.2024
2200765036

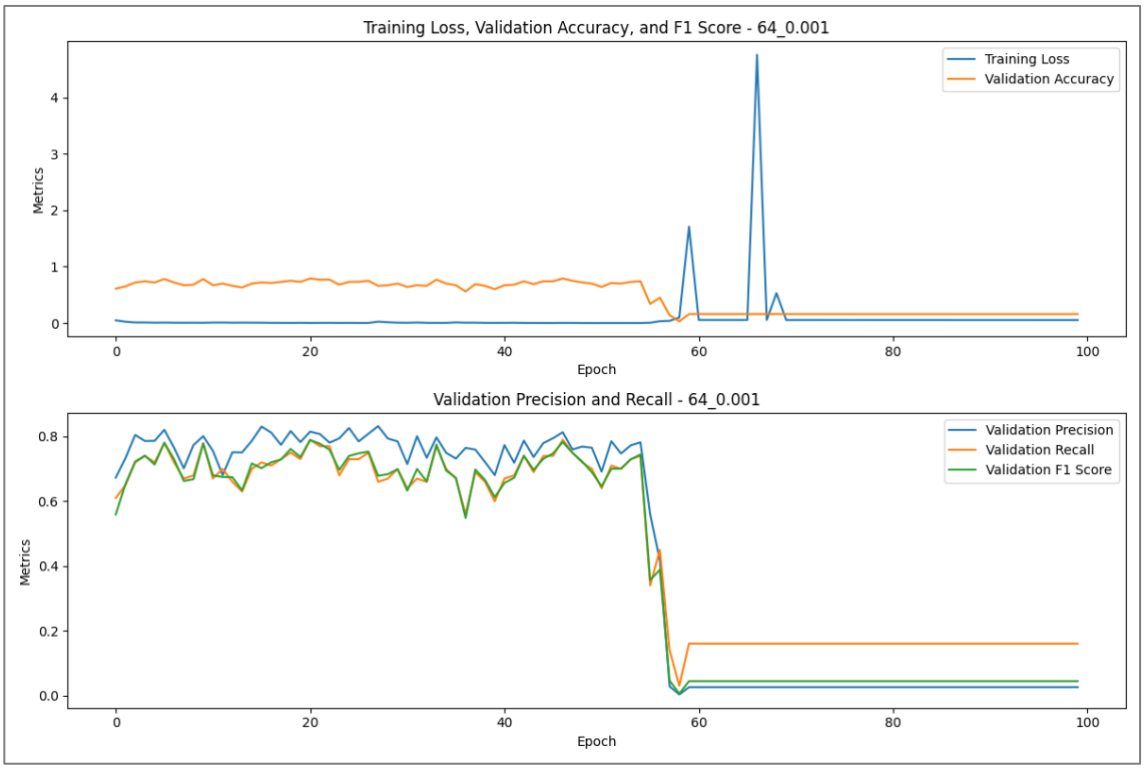


Ahmet Emre Usta 5.05.2024
2200765036

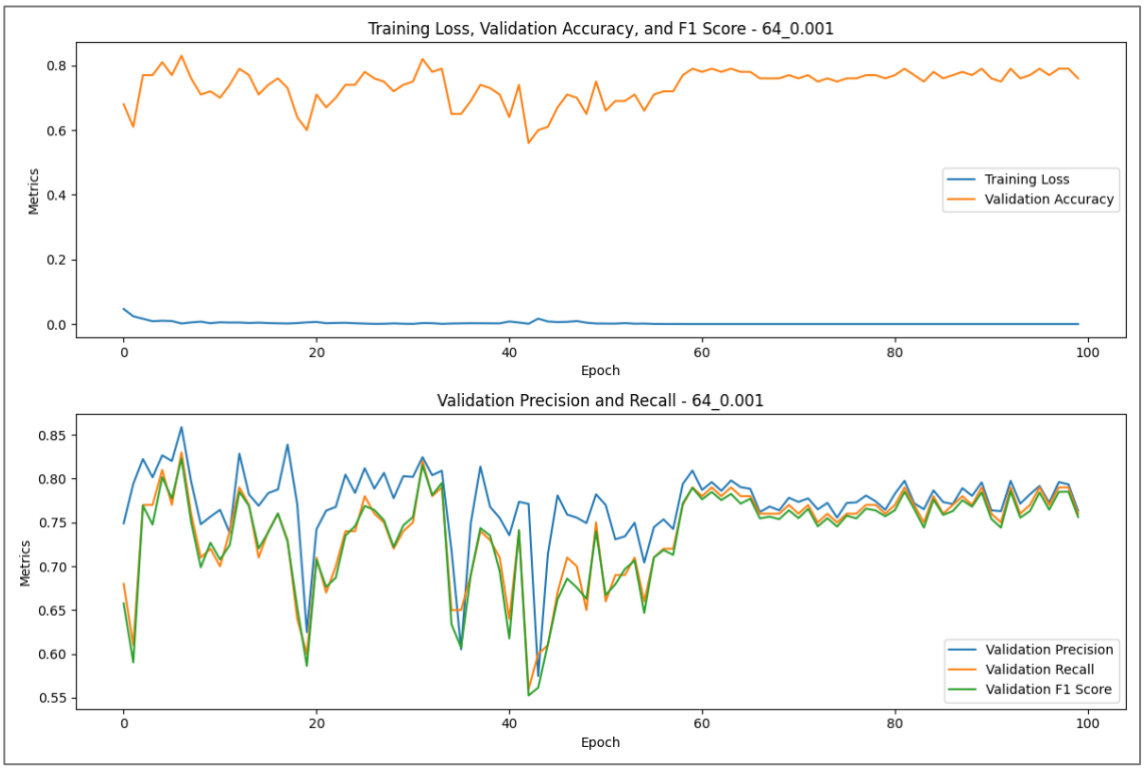


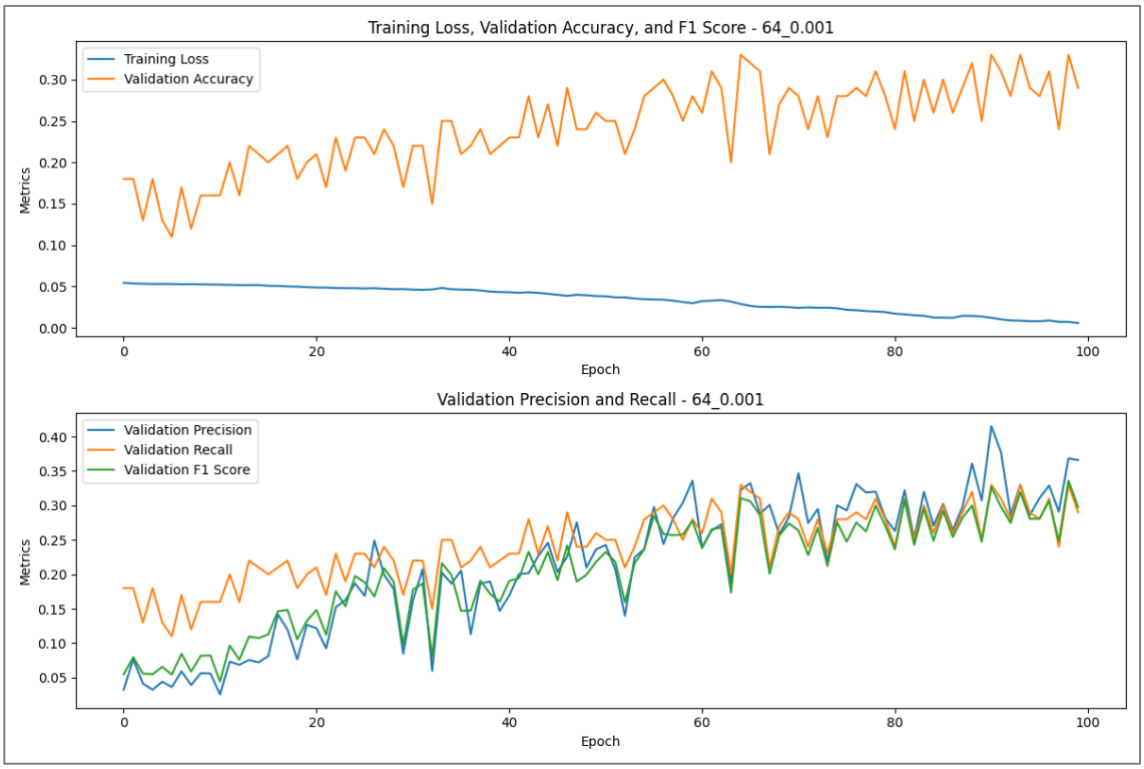
Ahmet Emre Usta 5.05.2024

2200765036



Ahmet Emre Usta 5.05.2024
2200765036





Ahmet Emre Usta 5.05.2024

2200765036