STUDENT NAME: MANPREET KAUR

STUDENT ID: C0917927

Report: Image Classification with Convolutional Neural Networks (CNN)

Objective

The objective of this project was to build and evaluate a Convolutional Neural Network (CNN) model for the classification of images from the Segmentation dataset. This dataset contains labeled images across multiple classes, and the goal is to predict the correct class of each image based on its features.

Process Overview

1. **Dataset Preparation**:
    - The dataset consists of two parts: the training set and the test set. The training set is used to train the model, while the test set is used for evaluating the model's performance.
    - The data was organized into directories with each class of images stored in a separate folder. We used the `ImageDataGenerator` from Keras for data augmentation and rescaling.
2. **Data Augmentation and Normalization**:
    - Data augmentation was applied during the training phase to prevent overfitting and improve generalization. Specifically, we used rescaling (normalization) of pixel values to be between 0 and 1.
    - The dataset was split into a training set (80%) and a validation set (20%) for model training and hyperparameter tuning.
3. **Model Architecture**:
    - A Sequential CNN model was designed with three convolutional layers followed by max-pooling layers. Each convolutional layer helps the model learn hierarchical features in the images.
    - The model has:
        - Three convolutional layers with increasing filter sizes (32, 64, 128).
        - Max-pooling layers to downsample the feature maps.
        - A dense fully connected layer with 512 neurons and ReLU activation.
        - A final softmax output layer with 6 neurons, representing the 6 classes in the dataset.
4. **Model Compilation and Training**:
    - The model was compiled using the Adam optimizer and categorical cross-entropy as the loss function since this is a multi-class classification problem.
    - The model was trained for 10 epochs with a batch size of 32 images. The training was evaluated using the validation set.
5. **Model Evaluation**:
    - After training, the model was evaluated on a separate test set.
    - The classification report was generated to assess precision, recall, and F1-score for each class.
    - The confusion matrix was visualized to identify misclassified instances.
6. **Misclassification Analysis**:

- A visualization of misclassified images was created to identify any patterns or common mistakes made by the model.
- This helps in understanding which classes the model struggles with and can guide future improvements.

Model Architecture

- **Input Layer**: The input shape was set to (150, 150, 3), corresponding to the resized images.
- **Convolutional Layers**: Three convolutional layers with 32, 64, and 128 filters, respectively, each followed by a max-pooling layer.
- **Fully Connected Layer**: A dense layer with 512 neurons and a dropout layer to reduce overfitting.
- **Output Layer**: A softmax output layer with 6 neurons, one for each class.

Classification Report

After evaluating the model on the test set, the following classification report was obtained:

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

           0       0.84      0.70      0.76       437
           1       0.97      0.94      0.95       474
           2       0.77      0.79      0.78       553
           3       0.80      0.75      0.78       525
           4       0.77      0.87      0.82       510
           5       0.81      0.86      0.84       501

    accuracy                           0.82      3000
   macro avg       0.83      0.82      0.82      3000
weighted avg       0.82      0.82      0.82      3000
```

- **Accuracy**: 82%
- The model performs well with high recall and precision in most classes, particularly class 1, which shows a high recall of 0.94.
- **Macro average** shows an overall balanced performance across all classes.

Findings

1. **Training Performance**:
   - The model showed steady improvement in both training and validation accuracy across epochs. The validation accuracy was close to the training accuracy, indicating that the model generalized well to unseen data.

2. **Test Set Evaluation**:
    - On the test set, the model performed reasonably well, achieving an overall accuracy of 82%. Class 1 achieved the highest performance, while class 0 exhibited lower recall, indicating that it was harder for the model to classify images from this class.
3. **Confusion Matrix**:
    - The confusion matrix helped identify which classes were frequently confused with each other. This insight could be used to further tune the model or apply additional preprocessing techniques to enhance performance on these particular classes.
4. **Misclassified Images**:
    - Misclassified images were reviewed, and it was observed that the model struggled with certain classes, which could be due to insufficient data or inherent similarities between those classes.

Conclusion and Future Work

- The CNN model was successful in performing multi-class image classification with an overall accuracy of 82%.
- The misclassified images suggest that further data augmentation or model architecture adjustments (e.g., adding more convolutional layers or using transfer learning) could improve performance.
- Future work could also involve fine-tuning hyperparameters or applying more advanced techniques such as pre-trained models (e.g., ResNet, VGG) for better results.