**Project Report: Image Classification with CelebA Dataset**

1. **Introduction** This project aims to develop and evaluate a deep learning model for classifying images from the CelebA dataset into ten selected attributes. The objective is to build a robust model capable of accurate classification by preprocessing data, designing a Convolutional Neural Network (CNN), tuning hyperparameters, and evaluating performance.
2. **Dataset Preparation**
   * **Dataset:** CelebA
   * **Original Dataset Size:** 202,599 images
   * **Selected Classes:** 10 attributes randomly chosen from the dataset
   * **Filtered Dataset:**
     + **Images per Class:** 1,000
     + **Total Images:** 9,987 (after ensuring no duplicates and sampling as needed)
   * **Steps Taken:**
     + **Data Filtering:** Selected 10 attributes and sampled 1,000 images per class, ensuring the total dataset consisted of 9,987 images.
     + **Data Sampling:** In case of insufficient images per class, additional samples were drawn from the full dataset to meet the required size.
     + **Data Storage:** Saved the filtered dataset as 'filtered\_dataset.csv'.
3. **Model Architecture**
   * **Model:** Improved Convolutional Neural Network (CNN)
   * **Layers:**
     + **Convolutional Layers:**
       - Conv2d (3 input channels, 32 output channels, 3x3 kernel)
       - Conv2d (32 input channels, 64 output channels, 3x3 kernel)
       - Conv2d (64 input channels, 128 output channels, 3x3 kernel)
     + **Fully Connected Layers:**
       - Linear (128 \* 16 \* 16 input features, 256 output features)
       - Linear (256 input features, 10 output features for classification)
   * **Activation Functions:** ReLU
   * **Pooling:** MaxPool2d with 2x2 kernel
4. **Training Process**
   * **Training:**
     + **Number of Epochs:** 10
     + **Optimizer:** Adam
     + **Learning Rate:** 0.0005
   * **Hyperparameter Tuning:**
     + **Learning Rates:** [0.0001, 0.0005, 0.001]
     + **Batch Sizes:** [16, 32, 64]
   * **Best Hyperparameters:**
     + **Learning Rate:** 0.0005
     + **Batch Size:** 32
5. **Evaluation Results**
   * **Metrics:**
     + **Accuracy:** 43.0%
     + **Precision:** 46.0%
     + **Recall:** 43.0%
     + **F1-Score:** 40.0%
   * **Confusion Matrix:** A confusion matrix was plotted to visualize the performance of the model across the different classes.
6. **Insights and Conclusion**
   * **Performance:** The model achieved an accuracy of 43.0% with balanced precision, recall, and F1-score, indicating effective classification but room for improvement.
   * **Challenges:** Handling class imbalance and ensuring sufficient representation of all classes required careful sampling and augmentation.
   * **Future Work:** Further improvement can be achieved through advanced architectures like ResNet or EfficientNet, and by incorporating additional data augmentation techniques or transfer learning.
   * **Model Saving:** The final model was saved as 'final\_model.pth' for future use and deployment.