Technical Summary: Gender Classification using DenseNet169 Transfer Learning- Task A

1. Approach

The core strategy revolves around **transfer learning**, a technique that reuses a pre-trained deep learning model as a starting point for a new, related task. This approach significantly reduces training time and data requirements, especially when dealing with limited datasets, by leveraging features learned from vast datasets like ImageNet.

Data Pipeline: A highly optimized data pipeline is made. Images are loaded from specified training and validation directories, automatically inferred with binary labels, and resized to a uniform 224x224 resolution.

Data Augmentation:
To enhance model generalization and mitigate overfitting, data augmentation is applied exclusively to the training dataset. This includes:

- **Random Flip**: Randomly flips images horizontally.
- **Random Rotation**: Applies slight random rotations.
- **Random Zoom**: Introduces random zooming.
- **Random Contrast**: Adjusts image contrast randomly. These transformations ensure the model is exposed to diverse variations of the input images during training. This technique is used in the female class as there are less data in this class.

Normalization and Prefetching: All image pixel values are normalized to the range between 0 and 1 using a rescaling layer. This standardization aids in faster and more stable model convergence. Furthermore, an automatic tuning mechanism is utilized for prefetching data, which overlaps data loading and preprocessing with model execution, significantly accelerating the training process by minimizing GPU/CPU idle time.

2. Architecture

The model architecture is built upon a pre-trained **DenseNet169** convolutional neural network, adapted for the specific binary classification task.

Base Model (Feature Extractor):

- **Model:** A DenseNet169 neural network.
- **Input Shape:** (224, 224, 3) (standard for ImageNet-trained models).

- **Pre-trained Weights:** Loaded with weights from the imagenet dataset. This allows the model to leverage rich, hierarchical features learned from millions of diverse images.
- **Exclusion of Top Layer:** The crucial decision to exclude DenseNet169's original classification head. This allows for the integration of a custom classification layer tailored to the specific problem.
- **Freezing:** The weights of the DenseNet169 are frozen. This prevents updates to its pre-trained layers during initial training, preserving the learned features and preventing catastrophic forgetting, especially important with smaller datasets.

Custom Classification Head:
A new classification head is appended to the frozen DenseNet169 base:

- **Global Average Pooling**: Reduces the spatial dimensions of the feature maps from the base model into a single vector per feature map by averaging, effectively summarizing the features. This layer significantly reduces the number of parameters compared to a flatten layer, making the model more efficient and less prone to overfitting.
- **Dropout Layer**: A dropout layer with a 30% dropout rate. This regularization technique randomly sets a fraction of input units to zero at each update during training, which helps prevent co-adaptation of neurons and improves the model's generalization capabilities.
- **Dense Layer with Sigmoid Activation**: The final output layer for binary classification. A single neuron outputs a probability score, transformed by the sigmoid activation function to be between 0 and 1 or (male and female).

3. Innovations and Key Features

- **Efficient Transfer Learning:** By freezing the powerful DenseNet169 feature extractor, the model quickly adapts to the new task with minimal training on the custom head, saving computational resources and time.
- **Robust Data Augmentation:** The on-the-fly data augmentation strategy ensures that the model sees a wide variety of input data, making it more robust to real-world variations and reducing overfitting.
- **Class Imbalance Handling:** The use of a class weight computation utility with a 'balanced' setting dynamically calculates and applies weights to the loss function during training. This ensures that the model pays adequate attention to under-represented classes, preventing bias towards the majority class.
- **Automated Best Model Saving:** A Model Checkpoint callback automatically monitors the validation accuracy and saves the model weights only when an improvement is observed, ensuring that the best performing model throughout

the training process is preserved.

4. Training Details

The model is compiled with an Adam optimizer, known for its adaptive learning rate capabilities, and binary cross-entropy as the loss function, which is standard for binary classification tasks. Training is conducted for 50 epochs, with validation performed on a separate dataset to monitor performance on unseen data. The calculated class weights are incorporated into the training process to address any dataset imbalance.

Conclusion

This solution employs a well-structured transfer learning pipeline, combining a powerful pre-trained DenseNet169 with a custom classification head. Coupled with robust data augmentation, class imbalance handling, and performance optimizations like prefetching and model checkpointing, this approach provides an effective and efficient method for developing high-performing image binary classification models. We have used the T4 GPU for faster training.