Assignment 10: Dataset Preparation and Image Classification with PyTorch

Objective:

To develop practical skills in handling image data, designing convolutional neural networks, and deploying a trained model for classification using PyTorch. This assignment will guide you through the steps of preparing a dataset, implementing and training a model, and evaluating its performance.

Instructions:

1. **Prepare the Dataset**:
    - Collect images of muffins and chihuahua. Organize them into a structured folder format:

```
dataset/
├── train/
│   ├── muffins/
│   └── chihuahua /
└── val/
    ├── muffins/
    └── chihuahua /
```

    - Apply image transformation techniques to standardize the input size and format for your model. Recommended transformations include resizing to 256x256 pixels and normalizing pixel values.
    -

2. **Model Training:**
    - Implement a simple convolutional neural network (Design by your own) for classification.
    - Train your model using the PyTorch framework with the prepared dataset. Experiment with different learning rates and other hyperparameters to optimize performance.

3. **Model Evaluation:**
    - Evaluate your model on a validation set to check its performance.

4. **Save and Load Model:**

   - Save your trained model to a file using the following code:

     ```
     # Save the model
     torch.save(model.state_dict(), 'model.pth')
     ```

   - Ensure your model can be reloaded successfully for later use with this code:

     ```
     # Load the model
     model_loaded = SimpleCNN(num_classes=2)
     model_loaded.load_state_dict(torch.load('model.pth'))
     model_loaded.eval()
     ```

5. **Prepare for Testing:**

   - On January 24, 2025, your model will be tested with 20 new images. If your model achieves more than 50% accuracy, you will receive full marks.

Submission Requirements:

Ensure your submission includes all source code, a report detailing your model architecture, training process, and a discussion on the model's performance (Evaluation phase). Include screenshots or logs of your model's training progress and final accuracy.