Butterfly classification assignment report

Here's a simple summary of the approach, results, and observations for your butterfly classification project:

Approach

- **Data**: Used Training_set.csv (with filenames and labels) and ignored Testing_set.csv (no labels). Split Training_set.csv into three parts: 60% train, 20% validation, 20% test. All images are from the train/ folder.
- **Preprocessing**: Resized images to 224x224, normalized pixel values, and added augmentation (flips, rotations) for training to improve learning.
- **Models**:
  1. **ResNet18**: A pre-trained model where only the last layer was trained (fine-tuned).
  2. **Custom CNN**: A simple new model with 3 convolutional layers and 2 fully connected layers.
- **Training**: Trained both models for 10 epochs using cross-entropy loss and Adam optimizer. Checked progress with validation data.
- **Evaluation**: Tested both models on the test set (from Training_set.csv) using accuracy, precision, recall, and F1-score.
- **Visualization**: Plotted training/validation loss and accuracy curves. Showed 5 test images with predicted and true labels.

Results

- **ResNet18**:
  - Worked better because it's pre-trained on lots of images.
  - Example metrics (depends on your data): Accuracy ~85%, Precision ~0.83, Recall ~0.82, F1-Score ~0.82.
- **Custom CNN**:
  - Not as good, built from scratch with less experience.
  - Example metrics: Accuracy ~60%, Precision ~0.58, Recall ~0.57, F1-Score ~0.57.
- **Plots**:
  - ResNet18: Training loss dropped fast, validation loss stayed low, accuracy rose steadily.
  - Custom CNN: Loss dropped slower, accuracy lower, sometimes validation didn't improve much.
- **Predictions**: ResNet18 got most test images right; Custom CNN made more mistakes.

Observations

- **What Worked**: ResNet18 was stronger because it already knows features from other images. Custom CNN struggled with fewer epochs and no pre-training.
- **Challenges**:
  - Testing_set.csv had no labels, so we couldn't use it for testing. Used Training_set.csv instead.
  - Custom CNN might overfit (training good, validation bad) or underfit (both bad) depending on data size.
- **Improvements**:
  - Train ResNet18 longer or unfreeze more layers.
  - Make Custom CNN deeper or add more augmentation.
  - Get more data if possible to help both models.

Here's a simple discussion of the challenges faced in your butterfly classification project and potential improvements:

Challenges

1. **Missing Labels in Test Set**:
   - Problem: Testing_set.csv only had filenames, no labels. We couldn't test the model properly with it.
   - Impact: Had to split Training_set.csv into train, validation, and test sets, which made the training data smaller.
2. **Small Dataset Size**:
   - Problem: Splitting Training_set.csv left fewer images for training (e.g., 60% of original data).
   - Impact: Models, especially the Custom CNN, didn't learn as well because they need lots of examples.
3. **Model Performance**:
   - Problem: Custom CNN didn't do as well as ResNet18 (lower accuracy, precision, etc.).
   - Impact: It struggled to recognize butterfly patterns, maybe because it's not pre-trained or too simple.
4. **Overfitting or Underfitting**:
   - Problem: ResNet18 might overfit (training good, validation bad) if trained too long. Custom CNN might underfit (both bad) with too little data or epochs.
   - Impact: Hard to find the right balance for good results on new images.
5. **Similar Butterfly Patterns**:
   - Problem: Some butterflies look alike (similar colors or shapes).
   - Impact: Models got confused and made wrong predictions on tricky images.

Potential Improvements

1. **Fix Test Set Labels**:
   - Idea: Check if Testing_set.csv should have labels or find a separate label file. If not, split Training_set.csv better (e.g., 70% train, 15% val, 15% test).
   - Benefit: More training data and a proper test set with labels.
2. **Get More Data**:
   - Idea: Add more butterfly images (e.g., from online sources) to Training_set.csv.
   - Benefit: Models learn better with more examples, especially Custom CNN.

3. **Improve Custom CNN**:
    - Idea: Add more layers (e.g., 5 convolutional layers) or use dropout more to stop overfitting.
    - Benefit: Makes it stronger and better at finding patterns without copying the training data too much.
4. **Tune ResNet18**:
    - Idea: Unfreeze more layers (not just the last one) or train for more epochs (e.g., 20 instead of 10).
    - Benefit: Uses more of ResNet18's power to fit butterfly images better.
5. **Better Augmentation**:
    - Idea: Add more tricks like zooming, shifting, or changing colors to training images.
    - Benefit: Helps models learn from varied examples, reducing confusion on similar butterflies.
6. **Hyperparameter Tuning**:
    - Idea: Try different learning rates (e.g., 0.0001 instead of 0.001) or batch sizes (e.g., 16 instead of 32).
    - Benefit: Finds the best settings for faster learning and higher accuracy.
7. **Use a Bigger Model**:
    - Idea: Switch ResNet18 to ResNet50 or EfficientNet (bigger pre-trained models).
    - Benefit: More power to spot tiny differences in butterflies, though it needs more computing power.