CS229 – Image Segmentation, and Expected Calibration Error

In this report, I evaluate deep learning models for the tasks of image segmentation and pixel-wise classification. I have done the following tasks for this given homework-

1. Preprocessing of images.
2. Training and validation of Resnet50 model, including visualization of predictions for a sample image and a confidence calibration curve.
3. Comparison of Resnet50 model performance with pretrained Resnet50 and Resnet101 models with fine-tuning.
4. Comparison of train loss, validation loss, accuracies, and ECE values for Resnet50 and Resnet101 models with and without fine-tuning.
5. Hyperparameter tuning of Resnet50 model training, specifically focusing on the effects of learning rate and weight decay regularization on performance.

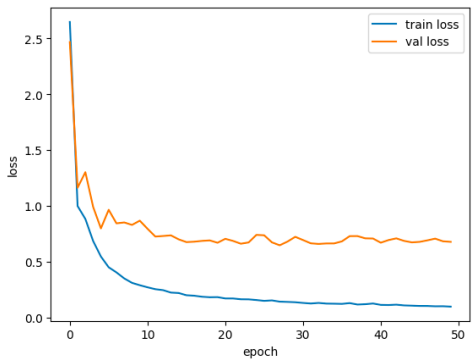
(a) Hyper-parameter choices:

I have used the following hyper parameters. I experimented different values for learning rate, and weight decay parameters. Before that, I preprocessed the images according to the instructions.

- Batch Size = 64
- Epochs = 50
- Learning Rate = 0.01, 0.05, 0.001, 0.0001
- Optimizer = SGD
  - Momentum = 0.9
  - Weight Decay = 0.0 , 0.01, 0.05, 0.001
- Preprocessing:
  - Resize image to 128x128
  - Center Cropping
  - Normalize the image in [-1,1] range

(b) Plot train and validation loss curves

This graph shows the training and validation losses of a neural network model, Resnet 50, during 50 epochs of training. Both losses decrease over time, but the training loss keeps dropping while the validation loss seems like to be plateaus. This suggests the model may have overfit to the training data and needs further fine-tuning to improve its ability to generalize.

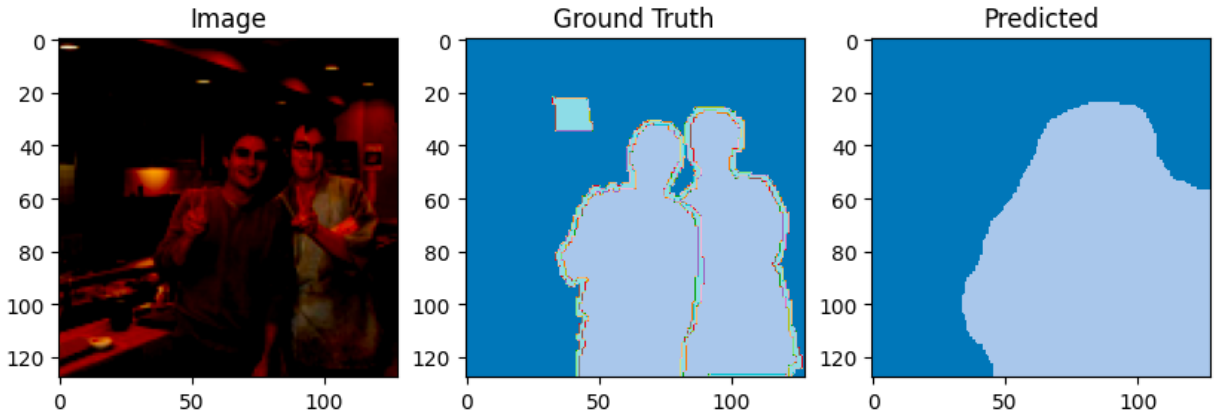


(c) Accuracy metric:

Accuracy on the validation set = `0.817 (at epoch 50)`

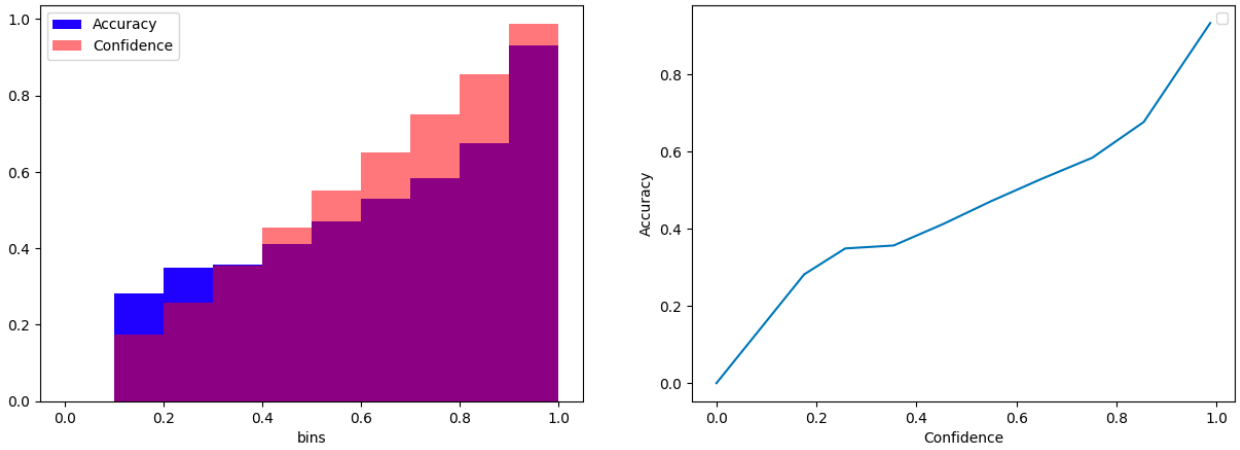
(d) Visualize an image from the validation set, its true segmentation, and the predicted segmentation:

I evaluated the performance of the trained model on an image from the validation set. The model achieved an accuracy of 81.7%, and although it failed to identify the detailed outline of the two people in the image, it was able to correctly recognize their general shape. The ground truth class labels and predicted labels for the image are shown in the figure.



(e) Plot confidence calibration curve on test data:

The figure depicts the accuracy and confidence of a model in 10 bins, along with the corresponding confidence calibration curve. The results indicate that there is not a significant difference between the accuracy and confidence of the model.



(f) Expected calibration error:

ECE: `0.0733`

The ECE value of 0.0733 for the Resnet50 model indicates that the model is reasonably well-calibrated, meaning that its predicted probabilities are generally close to the true probabilities of the predicted classes. On average, the model's predicted probabilities are off by approximately 7.3% from the true probabilities. While a lower ECE value is usually preferred, an ECE of 0.0733 suggests that the Resnet50 model is reliable in predicting the probabilities of the output classes. We can also see this in the confidence calibration curve.

(g) AI collaboration statement:

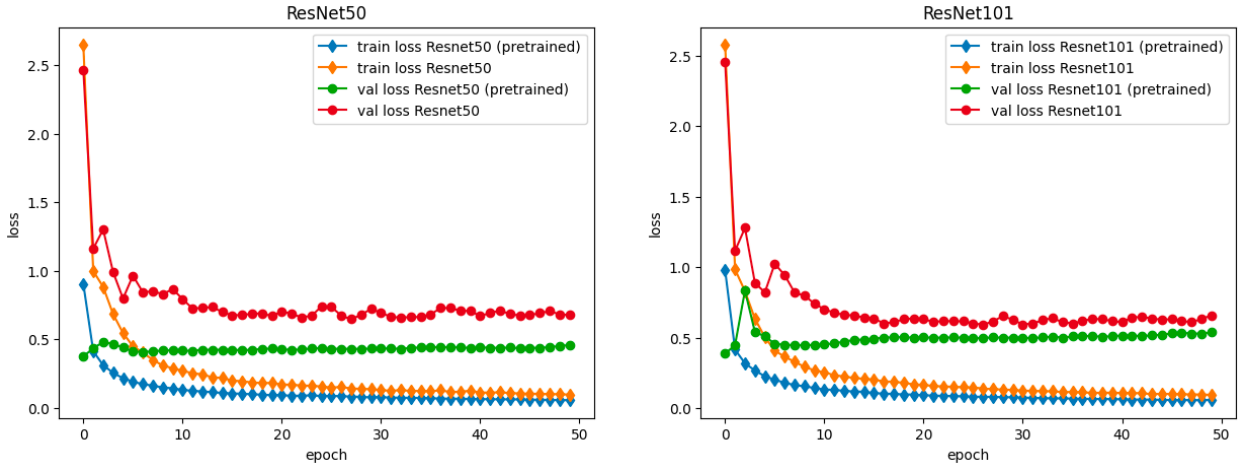
I used ChatGPT to rewrite the explanation of the figures.

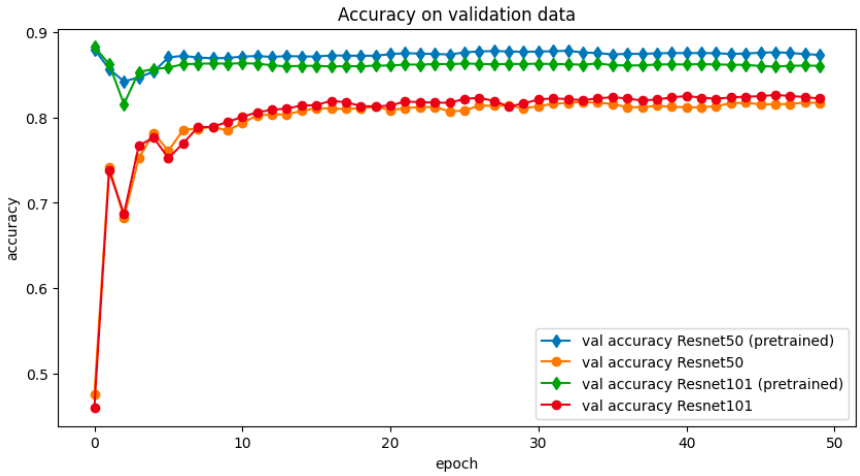
(h) Extra Credit:

⇒ ***Explore Transfer Learning:***

Trained Vs. Pretrained-Fine tuned model:

The graph below displays the training and validation losses of Resnet50 and Resnet101 models for both trained and pretrained-finetuned experiments. The training loss for all experiments decreases in a monotonic fashion, while the validation loss plateaus for the pretrained model. This suggests that fine-tuning the parameters of the pretrained model on the dataset improves its performance, but only to a certain extent. Interestingly, the pretrained model outperforms the raw model that is trained solely on this dataset. The accuracy graph on the validation set supports this observation. Among all experiments, the pretrained and fine-tuned Resnet50 model shows the best performance.



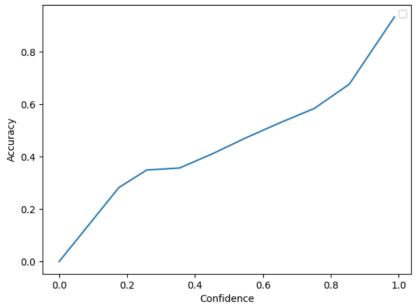


Confidence calibration curve:

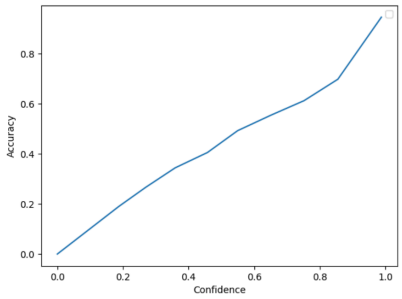
I also calculated the expected calibration error (ECE) and plotted the confidence calibration curve for each setting. The Resnet50 pretrained-finetuned model exhibits the lowest ECE value, indicating better calibration. Additionally, the confidence calibration curve for this model suggests acceptable performance.

Model	**ECE**
Resnet50	0.0733
Resnet50 pretrained+finetuned	**0.0543**
Resnet101	0.0741
Resnet101 pretrained+finetuned	0.0685

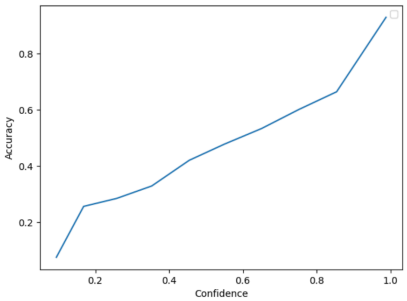
Resnet50:



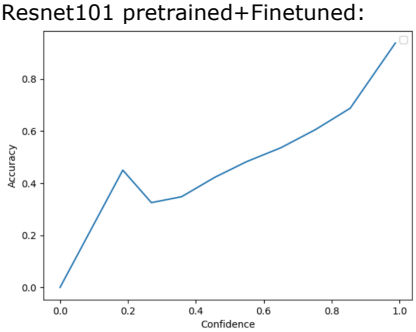
Resnet50 pretrained+Finetuned:



Resnet101:

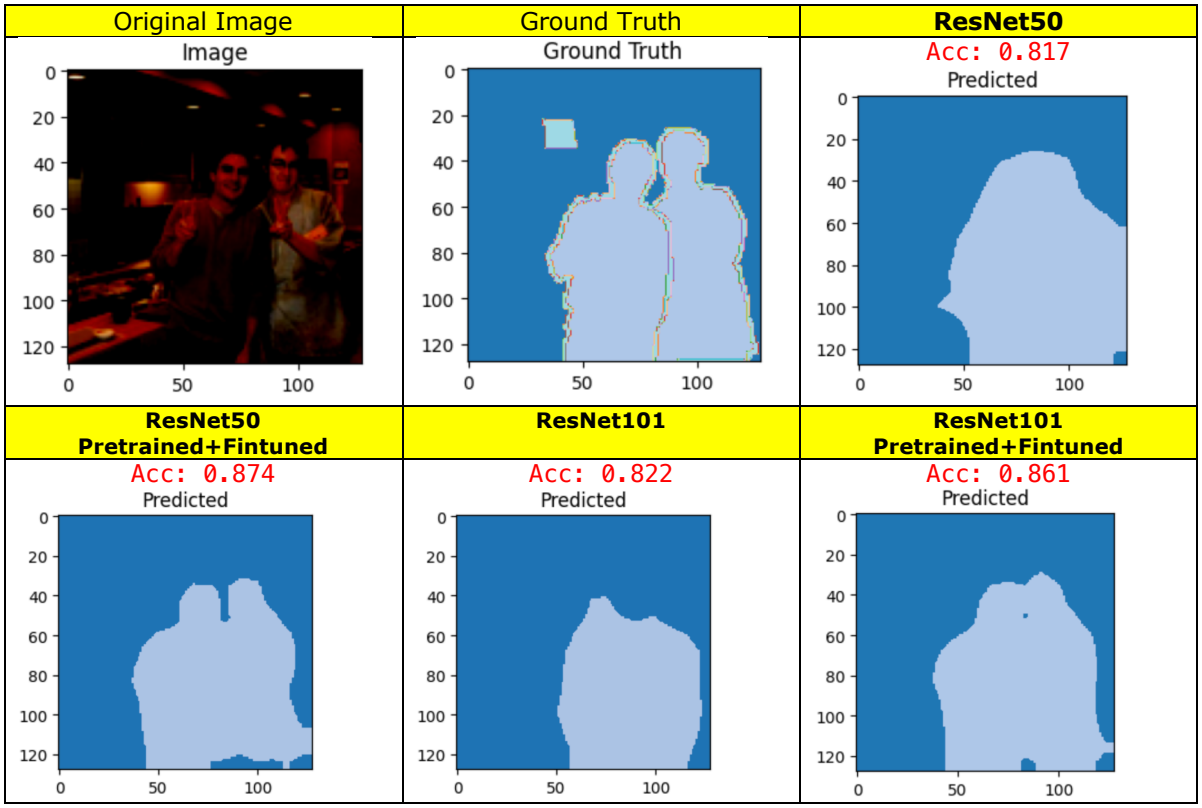


Resnet101 pretrained+Finetuned:



Visualization of segmentation:

The visualization of the segmentation on an image also shows that Resnet50 with finetuning can identify the persons better than other models.



⇒ ***Hyper-parameter Tuning:***

I conducted experiments to tune hyperparameters for the Resnet50 model, specifically the learning rate and weight decay. I tried weight decay values of 0.0, 0.01, 0.05, and 0.001, and learning rate values of 0.01, 0.05, 0.001, and 0.0001. The table below shows the resulting train loss, validation loss, and validation accuracy for each setting. Interestingly, regularization did not seem to improve performance, and a learning rate of 0.01 consistently yielded optimized loss across various cases.

Weight Decay	**Learning Rate**	**Train Loss**	**Val Loss**	**Val Accuracy**
0.0	0.01	0.098	0.679	0.817
0.0	0.05	0.040	0.727	0.819
0.0	0.001	0.034	0.766	0.819
0.0	0.0001	0.869	1.410	0.695
0.01	0.01	0.037	0.733	0.811
0.01	0.05	0.503	1.417	0.678

0.01	0.001	0.191	1.101	0.740
0.01	0.0001	0.860	1.406	0.693
0.05	0.01	0.215	1.481	0.708
0.05	0.05	1.306	1.666	0.504
0.05	0.001	1.212	1.338	0.698
0.05	0.0001	1.201	1.331	0.704
0.001	0.01	1.031	1.446	0.693
0.001	0.05	1.003	1.503	0.619
0.001	0.001	0.869	1.409	0.697
0.001	0.0001	0.865	1.409	0.695