Project 3

Name: Ram sai kiran Chintala
ID: 12594885

Links:
Panopto Link :
https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=5b295a7e-ab9a-462b-9fe9-b230005f989d

Part A –
Experiment 1: Object Detection and Segmentation
https://www.kaggle.com/code/ramsaikiran/object-detection

Experiment 2 : UNet Segmentation
https://www.kaggle.com/code/kiranchowdary29/unet-kiran

Experiment 3 : Fine_tune_SegFormer
https://colab.research.google.com/drive/10eINr25eyuC8jK0vsyPINyXcSw9CPTmP?usp=sharing

Part B-
Experiment 1 VGG
https://colab.research.google.com/drive/1mrTzglCaXJ-zmR2HVQ4eBo0ZDENuRur9?usp=sharing

Experiment 2 Resnet
https://colab.research.google.com/drive/10eINr25eyuC8jK0vsyPINyXcSw9CPTmP?usp=sharing

Discussion:
1. Compare the results of your experiments for Object Detection.
a. Display and discuss the results of object detection on images and video using YOLOv8.

Images: High localization and classification accuracy with mAP50-95(B) =0.806. Static nature allowed for precise bounding boxes and classifications.
Videos: Consistent tracking and detection across frames; occasional motion blur or rapid object movement reduced accuracy.

b. Which setting (images or videos) provides the best object localization and classification performance?

Images: Provided better object localization and classification due to static frames and no motion blur.
Videos: Faced challenges with motion blur and frame transitions, slightly reducing performance.

c. What challenges did you encounter in detecting partially visible objects, and how did the model perform in these cases?

The model struggled with detecting occluded or partially visible objects, often missing or misclassifying them. This issue arises from the lack of sufficient contextual features for incomplete objects..

d. In which scenarios did you observe false positives or false negatives, and what might be the reason behind these misclassifications?

False Positives: Occurred in cluttered backgrounds due to similar patterns confusing the model.
False Negatives: Often observed with small or occluded objects, as the model missed them due to insufficient feature resolution or contrast.
Configuration 3, with a lower learning rate and a larger batch size, performed the best overall. It worked particularly well for images, while videos handled dynamic scenarios better but struggled with motion-related issues.

2. Compare the results of your experiments for Segmentation.
a. Display and discuss the results of the image segmentation using the UNet model.
Best Configuration: Configuration 1 delivered the highest performance with a Test Dice score of 0.8364 and Test IoU of 0.7223. This was achieved with 20 epochs, a learning rate of 0.001, and a batch size of 60.
Observation: Increasing the learning rate and batch size in Configurations 2 and 3 reduced performance, indicating that UNet performs better with smaller learning rates and moderate batch sizes.
b. Display and discuss the results of the image segmentation using SegFormer.
Best Configuration: Configuration 2 achieved the highest Mean IoU (0.3268) and Mean Accuracy (0.5037) with 7 images. Configuration 1, however, showed the best overall accuracy (0.9143), suggesting better generalizability.
Observation: SegFormer's performance varied with the number of images fine-tuned, where balancing the dataset and configurations was key to better results.
c. Which model do you feel provides the best structural understanding of geometric structures?

UNet provided a better understanding of geometric structures due to its superior Dice and IoU scores, making it more reliable for precise segmentation structures.

d. Did you observe any issues with overlapping objects?

Overlapping objects were a challenge for both models. UNet sometimes struggled with boundaries, causing minor inaccuracies. SegFormer, due to its transformer-based approach, was more prone to misclassifications in crowded scenes or regions with object overlaps.

3. Compare the results of your experiments for Transfer Learning.

a. Display and discuss the selected classification layer training performance.

During training, ResNet displayed better performance than VGG, with its accuracy improving steadily across epochs, reaching 60.2% in the final epoch. The validation accuracy for ResNet peaked at 67.5%, significantly higher than VGG's 58.6%. Additionally, ResNet had lower training and validation loss throughout, indicating more stable learning. VGG struggled with slower convergence and higher losses, suggesting its architecture was less effective for this task.

b. Display and discuss the results of the test performance for each experiment.

In testing, ResNet demonstrated superior metrics compared to VGG. It achieved a higher AUC-ROC score (0.9911 vs. 0.9833), mean average precision (0.7574 vs. 0.6517), and overall accuracy (67.54% vs. 59.45%). These results highlight ResNet’s ability to generalize better to unseen data, further supporting its effectiveness in feature extraction.

c. Which model do you think provided the better set of features for training between the VGG and RESNET? How did you come to this conclusion?

ResNet provided the better set of features for training due to its residual connections, which help avoid vanishing gradients and improve learning. Its consistently higher accuracy, precision, and AUC-ROC score, along with lower losses, make it the clear choice over VGG for this task.

4.

In these experiments, I observed that ResNet's residual connections significantly enhanced feature extraction, resulting in better generalization and performance compared to VGG. This aligns with the concepts of deeper architectures and gradient flow we covered in the course, as ResNet effectively mitigated vanishing gradients while training. My efforts in tuning hyperparameters highlighted the importance of learning rates and batch sizes for model convergence, which ties directly to our discussions on optimization strategies. The experiments also demonstrated the trade-offs between model complexity and performance, reinforcing the balance required between computational efficiency and accuracy in real-world applications. Overall, these hands-on efforts provided practical insights into how theoretical concepts like transfer learning and architecture design impact outcomes in practice.

Results

In these experiments, I observed how the choice of architectures and training configurations significantly impacts model performance. For example, ResNet's use of skip connections clearly demonstrated its ability to avoid vanishing gradients, something we covered in class, which allowed it to converge faster and generalize better than VGG. Working on these experiments reinforced my understanding of how deeper networks like ResNet can outperform traditional architectures by preserving information across layers.

The results also highlighted the importance of tuning hyperparameters like learning rates and batch sizes—something we discussed in depth during optimization lectures. Seeing the direct impact of these choices on training and validation performance made those theoretical concepts feel much more practical and relevant. Overall, the hands-on work tied closely to the foundational ideas we studied in class, like overfitting, loss functions, and model generalization. It was rewarding to see how these principles applied to solving real problems.

The graphs, plots are all in links.