Session:3 Hours

Architecture Comparison via Guided Exercises

"Compare and Analyze Multiple CNN Architectures for Image Classification and Segmentation"

Exercise Flow: Build → Train → Compare → Reflect

Dataset:

- For **classification**: CIFAR-10 or Fashion-MNIST
- For **segmentation**: Oxford Pet Dataset (or any toy dataset with masks)

Exercise A: Custom CNN (30 mins)

1. **Build** a CNN with 3–4 Conv layers, MaxPooling, Dropout, BatchNorm, and EarlyStopping.
2. **Train** the model for 15–20 epochs and **record**:
    - Accuracy
    - Loss
    - Epoch at which EarlyStopping triggered

Questions:

- What are the total trainable parameters?
- What happens if Dropout is removed?
- Is the model overfitting or underfitting?

Exercise B: ResNet50 (30 mins)

1. **Load** pretrained ResNet50 (include_top=False) with weights='imagenet'
2. **Freeze** base layers and add custom head
3. **Train**, then **unfreeze last few layers** and fine-tune

Questions:

- Compare training time with Custom CNN
- What changes after fine-tuning?
- How does ResNet prevent vanishing gradients?

Exercise C: DenseNet121 (30 mins)

1. Use DenseNet121 as base model
2. Add GlobalAveragePooling + Dense classifier
3. Train and evaluate

Questions:

- How does DenseNet's parameter count compare to ResNet?
- Why does DenseNet perform better on smaller datasets?
- What effect does feature reuse have?

Exercise D: U-Net for Segmentation (30 mins)

1. Build or load a U-Net model
2. Train on a segmentation task (use small image-mask dataset)
3. Evaluate using IoU / Dice Score

Questions:

- How is U-Net architecture different from classification CNNs?
- What role do skip connections play in segmentation?
- What loss function did you use — and why?

Final Comparison Table (15 mins)

Fill this table based on their results:

Model	Task	Params	Train Time	Accuracy / Metric	Pros	Cons
Custom CNN	Classification					
ResNet50	Classification					
DenseNet121	Classification					
U-Net	Segmentation					

Wrap-Up Questions for Discussion (15 mins)

1. Which model gave the best generalization performance?
2. Which was most computationally expensive?
3. If you had limited data, which architecture would you prefer?
4. When would you avoid using transfer learning?
5. Which architecture would you use for:
    - A large image classification task?
    - A medical image segmentation task?
    - A mobile application?

Deliverables from Each Student

- A notebook with:
    - Code for all 4 models
    - Metric plots (loss/accuracy)
    - Completed comparison table
    - Answers to reflection questions