Applications of AI

Group Assignment

Submitted by:

AMPBA - Batch 19

Student Name	PG ID
Muhammad Ashraf Hussain	12220048
Aditya Sharma	12210048
Adarsh Balan	12220062
Thaneshwar Prasad Sahu	12220083
Prathyusha Thatipelli	12220044

PART A:

1. Image Captioning

Business Domain: Social Media

Input Data: User-uploaded images

Expected Output: Automatically generated descriptive captions for images.

Company: Facebook

2. Visual Question Answering

Business Domain: E-learning

Input Data: Images from educational content, accompanied by text-based questions.

Expected Output: Accurate answers to the questions based on the visual content.

Company: Coursera

3. Object Detection

Business Domain: Retail

Input Data: Surveillance camera footage, weight sensor

Expected Output: Real-time identification and tracking of products on shelves.

Company: Big Basket

4. Image Segmentation

Business Domain: Healthcare

Input Data: Medical images (e.g., X-rays, MRIs).

Expected Output: Precise segmentation of organs or abnormalities within medical images.

Company: NVIDIA, Siemens Healthcare

5. Image Similarity Computation

Business Domain: E-commerce

Input Data: Product images from an online marketplace.

Expected Output: Recommendations of visually similar products to enhance user shopping experience.

Company: Amazon, Myntra, Ajio etc.

6. Action Recognition

Business Domain: Interactive Gaming

Input Data: Player's movements and gestures captured through cameras, motion sensors, or wearable devices (e.g., VR headsets, Kinect-like sensors).

Expected Output: Real-time detection and classification of players' physical actions and gestures within the game environment.

Company: Microsoft (for Kinect-based gaming) or Oculus (for VR-based gaming)

Reference: G3D: A gaming action dataset and real time action recognition evaluation framework | IEEE Conference Publication | IEEE Xplore

7. Multi-object Tracking

Business Domain: Autonomous Vehicles (Self Driving Cars)

Input Data: Sensor data (e.g., LiDAR, cameras) from self-driving cars.

Expected Output: Real-time tracking of multiple objects (e.g., other vehicles, pedestrians) to ensure safe navigation.

Company: Waymo (Alphabet Inc.), Uber, ABB

Reference: Autonomous Vehicle Research - Our Latest Publications - Waymo

PART B:

Dataset Overview: The dataset consists of 805 images across 15 categories, derived from the larger Caltech 101 dataset. The task involves creating a 15-class image classification model. For each category, images from 0001 to 0040 are used for training, and the remaining images are used for testing.

Google Collab has been used for code implementation along with standard python libraries.

Question 1

- Zip file is unzipped and relevant class folders were extracted and stored in google drive with folder name: dataset and file path: /content/drive/MyDrive/AAI Group Assignment/dataset
- Dataset divided into training and test by keeping first 40 images for training and remaining for test

```
Extracted to /content/drive/MyDrive/AAI Group Assignment/dataset
Contents of extracted directory: ['dataset']
Training and testing datasets created at /content/drive/MyDrive/AAI Group Assignment/train and /content/drive/MyDrive/AAI Group Assignment/test
Contents of train directory: ['accordion', 'bass', 'camera', 'crocodile', 'crocodile_head', 'cup', 'dollar_bill', 'emu', 'gramophone', 'hedgehog', 'nautilus', 'pizza', 'pyramid', 'sea_
Contents of test directory: ['accordion', 'bass', 'camera', 'crocodile', 'crocodile_head', 'cup', 'dollar_bill', 'emu', 'gramophone', 'hedgehog', 'nautilus', 'pizza', 'pyramid', 'sea_
```

Question 2

Three different pretrained models – ResNet18, DenseNet121, and VGG19 – were employed for the classification task. The models were trained on the dataset, and the per-class classification accuracy was evaluated in terms of precision and recall. This involved modifying the models to suit the 15-class classification and adapting their final layers accordingly.

Training and evaluation functions were defined and then used for each model to calculate precision and recall.

```
Training and evaluating resnet18...
Epoch 1/25, Loss: 0.14063434302806854
Epoch 2/25, Loss: 0.057409320026636124
Epoch 3/25, Loss: 0.06868451833724976
Epoch 4/25, Loss: 0.037219174206256866
Epoch 5/25, Loss: 0.028178133070468903
Epoch 6/25, Loss: 0.06909924000501633
Epoch 7/25, Loss: 0.026505939662456512
Epoch 8/25, Loss: 0.02454829216003418
Epoch 9/25, Loss: 0.031990908086299896
Epoch 10/25, Loss: 0.022165261209011078
Epoch 11/25, Loss: 0.02897891402244568
Epoch 12/25, Loss: 0.01501509826630354
Epoch 13/25, Loss: 0.02373841218650341
Epoch 14/25, Loss: 0.018202783539891243
Epoch 15/25, Loss: 0.020262477919459343
Epoch 16/25, Loss: 0.040584344416856766
Epoch 17/25, Loss: 0.015272463671863079
Epoch 18/25, Loss: 0.017581652849912643
Epoch 19/25, Loss: 0.011857635341584682
Epoch 20/25, Loss: 0.011391010135412216
Epoch 21/25, Loss: 0.011890620924532413
Epoch 22/25, Loss: 0.01605450175702572
Epoch 23/25, Loss: 0.008651270531117916
Epoch 24/25, Loss: 0.022066481411457062
Epoch 25/25, Loss: 0.012296359986066818
```

```
Training and evaluating densenet121...
Epoch 1/25, Loss: 0.08740967512130737
Epoch 2/25, Loss: 0.028316093608736992
Epoch 3/25, Loss: 0.06206175684928894
Epoch 4/25, Loss: 0.04529522359371185
Epoch 5/25, Loss: 0.021236605942249298
Epoch 6/25, Loss: 0.023629480972886086
Epoch 7/25, Loss: 0.011831761337816715
Epoch 8/25, Loss: 0.024832041934132576
Epoch 9/25, Loss: 0.044277623295784
Epoch 10/25, Loss: 0.016931651160120964
Epoch 11/25, Loss: 0.015888331457972527
Epoch 12/25, Loss: 0.008100301958620548
Epoch 13/25, Loss: 0.013920933939516544
Epoch 14/25, Loss: 0.029740264639258385
Epoch 15/25, Loss: 0.013623036444187164
Epoch 16/25, Loss: 0.010365628637373447
Epoch 17/25, Loss: 0.007874789647758007
Epoch 18/25, Loss: 0.015897197648882866
Epoch 19/25, Loss: 0.0119498111307621
Epoch 20/25, Loss: 0.009250419214367867
Epoch 21/25, Loss: 0.01770429126918316
Epoch 22/25, Loss: 0.016909828409552574
Epoch 23/25, Loss: 0.020906632766127586
Epoch 24/25, Loss: 0.030112646520137787
Epoch 25/25, Loss: 0.030381230637431145
```

```
Training and evaluating vgg19...
Epoch 1/25, Loss: 0.002315773628652096
Epoch 2/25, Loss: 0.0007475065067410469
Epoch 3/25, Loss: 0.00016324575699400157
Epoch 4/25, Loss: 0.0002515343076083809
Epoch 5/25, Loss: 6.392181239789352e-05
Epoch 6/25, Loss: 0.00021703507809434086
Epoch 7/25, Loss: 0.0007854479481466115
Epoch 8/25, Loss: 6.370686605805531e-05
Epoch 9/25, Loss: 0.00020656462584156543
Epoch 10/25, Loss: 2.7972355383099057e-05
Epoch 11/25, Loss: 0.0003426851471886039
Epoch 12/25, Loss: 9.86254817689769e-05
Epoch 13/25, Loss: 0.0008384441025555134
Epoch 14/25, Loss: 0.003664429998025298
Epoch 15/25, Loss: 0.0004360201710369438
Epoch 16/25, Loss: 0.0013736592372879386
Epoch 17/25, Loss: 0.000646266620606184
Epoch 18/25, Loss: 1.427945517207263e-05
Epoch 19/25, Loss: 0.0010791533859446645
Epoch 20/25, Loss: 0.00022229035675991327
Epoch 21/25, Loss: 0.00018855281814467162
Epoch 22/25, Loss: 3.391576683497988e-05
Epoch 23/25, Loss: 0.004325329791754484
Epoch 24/25, Loss: 6.295423372648656e-05
Epoch 25/25, Loss: 0.011011590249836445
```

<u>Precision and Recall Outputs for each model:</u>

```
Model: resnet18
Per-class Precision and Recall:
accordion: Precision - 1.0000, Recall - 1.0000
bass: Precision - 1.0000, Recall - 1.0000
camera: Precision - 1.0000, Recall - 1.0000
crocodile: Precision - 0.8000, Recall - 0.8000
crocodile_head: Precision - 0.8182, Recall - 0.8182
cup: Precision - 1.0000, Recall - 1.0000
dollar_bill: Precision - 1.0000, Recall - 1.0000
emu: Precision - 1.0000, Recall - 1.0000
gramophone: Precision - 1.0000, Recall - 1.0000
hedgehog: Precision - 0.9333, Recall - 1.0000
nautilus: Precision - 1.0000, Recall - 1.0000
pizza: Precision - 1.0000, Recall - 1.0000
pyramid: Precision - 1.0000, Recall - 1.0000
sea_horse: Precision - 1.0000, Recall - 0.9412
windsor_chair: Precision - 1.0000, Recall - 1.0000
```

```
Model: densenet121
Per-class Precision and Recall:
accordion: Precision - 1.0000, Recall - 1.0000
bass: Precision - 1.0000, Recall - 1.0000
camera: Precision - 1.0000, Recall - 1.0000
crocodile: Precision - 0.9000, Recall - 0.9000
crocodile_head: Precision - 0.9091, Recall - 0.9091
cup: Precision - 1.0000, Recall - 1.0000
dollar_bill: Precision - 1.0000, Recall - 1.0000
emu: Precision - 1.0000, Recall - 1.0000
gramophone: Precision - 1.0000, Recall - 1.0000
hedgehog: Precision - 0.9286, Recall - 0.9286
nautilus: Precision - 0.9375, Recall - 1.0000
pizza: Precision - 1.0000, Recall - 1.0000
pyramid: Precision - 1.0000, Recall - 1.0000
sea_horse: Precision - 1.0000, Recall - 0.9412
windsor_chair: Precision - 1.0000, Recall - 1.0000
```

```
Model: vgg19
Per-class Precision and Recall:
accordion: Precision - 1.0000, Recall - 1.0000
bass: Precision - 1.0000, Recall - 1.0000
camera: Precision - 1.0000, Recall - 1.0000
crocodile: Precision - 0.7500, Recall - 0.9000
crocodile_head: Precision - 0.9000, Recall - 0.8182
cup: Precision - 1.0000, Recall - 0.8824
dollar_bill: Precision - 0.9231, Recall - 1.0000
emu: Precision - 1.0000, Recall - 1.0000
gramophone: Precision - 1.0000, Recall - 1.0000
hedgehog: Precision - 1.0000, Recall - 0.9286
nautilus: Precision - 0.9375, Recall - 1.0000
pizza: Precision - 1.0000, Recall - 1.0000
pyramid: Precision - 1.0000, Recall - 0.9412
sea_horse: Precision - 0.9412, Recall - 0.9412
windsor_chair: Precision - 0.9412, Recall - 1.0000
```

Question 3

Here, the focus is on fine-tuning the VGG19 model. The pretrained VGG19 model was adapted to output 15 classes, aligning with the dataset's requirements. The model underwent a training process on the dataset, followed by an evaluation phase where its performance was measured in terms of precision and recall for each class.

Model Training

```
Epoch 1/25, Loss: 0.00029038841603323817
Epoch 2/25, Loss: 0.003390142461284995
Epoch 3/25, Loss: 0.001954454928636551
Epoch 4/25, Loss: 0.0009592715068720281
Epoch 5/25, Loss: 0.0007679020636714995
Epoch 6/25, Loss: 2.6104991775355302e-05
Epoch 7/25, Loss: 0.0002920116239693016
Epoch 8/25, Loss: 7.186943548731506e-06
Epoch 9/25, Loss: 0.00355253997258842
Epoch 10/25, Loss: 0.00022705852461513132
Epoch 11/25, Loss: 4.281125438865274e-05
Epoch 12/25, Loss: 6.314789061434567e-05
Epoch 13/25, Loss: 0.0011150952195748687
Epoch 14/25, Loss: 2.4520792067050934e-05
Epoch 15/25, Loss: 1.689718737907242e-05
Epoch 16/25, Loss: 0.0006182850920595229
Epoch 17/25, Loss: 0.0030808441806584597
Epoch 18/25, Loss: 0.00032424964592792094
Epoch 19/25, Loss: 0.007072964683175087
Epoch 20/25, Loss: 0.00010252150968881324
Epoch 21/25, Loss: 6.611769640585408e-05
Epoch 22/25, Loss: 5.265396976028569e-05
Epoch 23/25, Loss: 3.058983929804526e-05
Epoch 24/25, Loss: 6.584721995750442e-05
Epoch 25/25, Loss: 7.923463272163644e-05
```

Precision and Recall for each class.

```
Per-class Precision and Recall:
accordion: Precision - 1.0000, Recall - 1.0000
bass: Precision - 1.0000, Recall - 1.0000
camera: Precision - 1.0000, Recall - 1.0000
crocodile: Precision - 0.7778, Recall - 0.7000
crocodile_head: Precision - 0.7500, Recall - 0.8182
cup: Precision - 1.0000, Recall - 0.9412
dollar_bill: Precision - 0.9231, Recall - 1.0000
emu: Precision - 1.0000, Recall - 1.0000
gramophone: Precision - 1.0000, Recall - 1.0000
hedgehog: Precision - 1.0000, Recall - 0.9286
nautilus: Precision - 0.9375, Recall - 1.0000
pizza: Precision - 1.0000, Recall - 1.0000
pyramid: Precision - 1.0000, Recall - 1.0000
sea_horse: Precision - 1.0000, Recall - 1.0000
windsor_chair: Precision - 1.0000, Recall - 1.0000
```

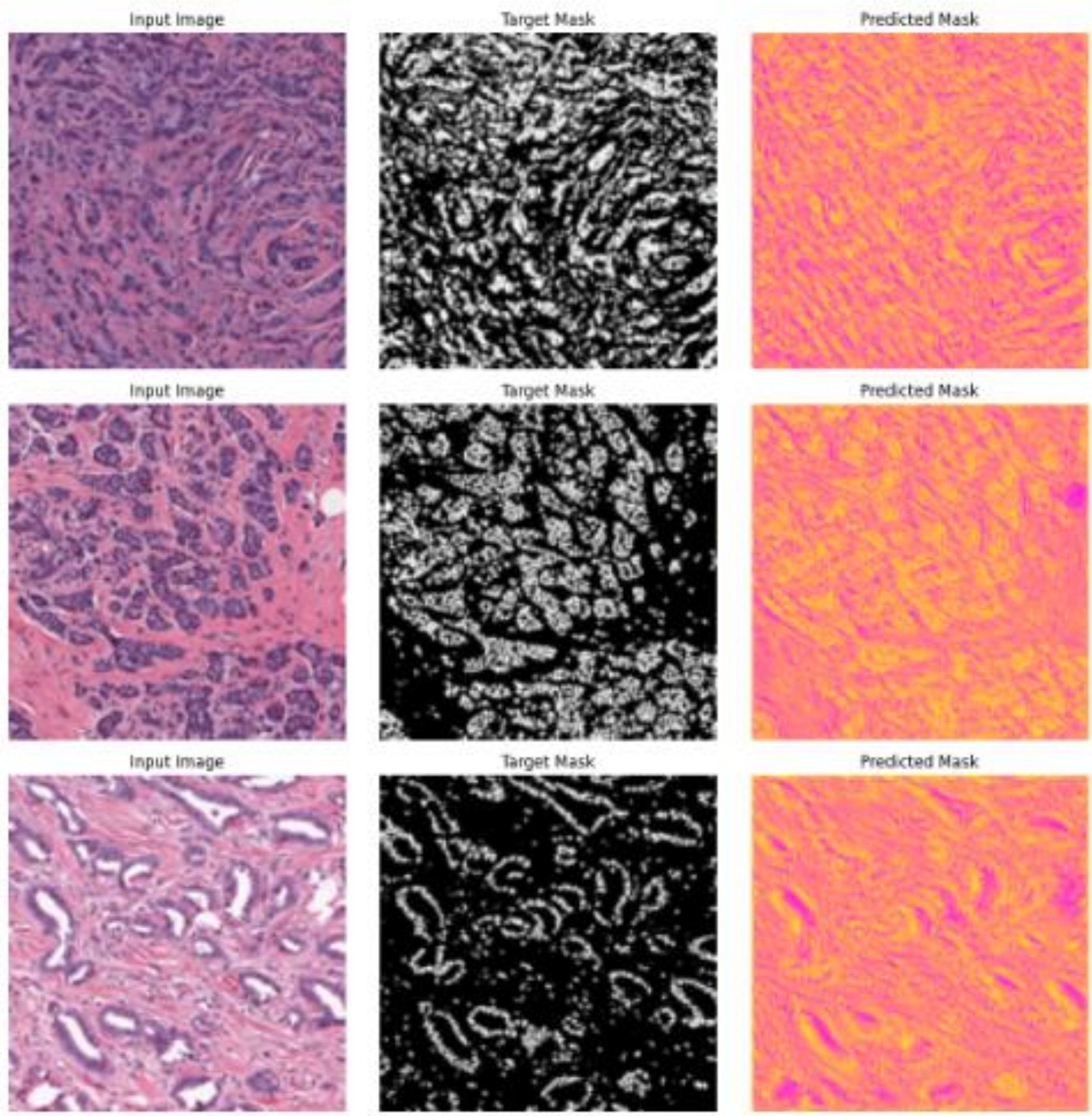
PART C:

The focus was to fine tune a UNet model with ResNet backbone for image segmentation on a set of pathology images. The model underwent a training process on the dataset, where multiple iterations were tried to fine tune the model with different combinations of gamma, batch size, learning rate, epochs, step_size etc.. Following are the parameters, results, and outputs of the chosen model:

```
num_class = 3
model = ResNetUNet(num_class).to(device)

# freeze backbone layers
for l in model.base_layers:
  for param in l.parameters():
    param.requires_grad = False

optimizer_ft = optim.Adam(filter(lambda p: p.requires_grad, model.parameters()), lr=1e-6, weight_decay=0.001)

#Decays the learning rate of each parameter group by gamma every step_size epochs
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_ft, step_size=7, gamma=0.85)

model = train_model(model, optimizer_ft, exp_lr_scheduler, num_epochs=5)
```

```
Epoch 0/4
----------
train: bce: 0.415276, dice: 0.010455, loss: 0.212865
LR 1e-06
val: bce: 0.418885, dice: 0.010474, loss: 0.214679
saving best model to checkpoint.pth
0m 22s
Epoch 1/4
----------
train: bce: 0.360146, dice: 0.009912, loss: 0.185029
LR 1e-06
val: bce: 0.374856, dice: 0.010040, loss: 0.192448
saving best model to checkpoint.pth
0m 24s
Epoch 2/4
----------
train: bce: 0.305694, dice: 0.009395, loss: 0.157544
LR 1e-06
val: bce: 0.321751, dice: 0.009521, loss: 0.165636
saving best model to checkpoint.pth
0m 24s
Epoch 3/4
----------
train: bce: 0.252940, dice: 0.008870, loss: 0.130905
LR 1e-06
val: bce: 0.266092, dice: 0.008974, loss: 0.137533
saving best model to checkpoint.pth
0m 23s
Epoch 4/4
----------
train: bce: 0.199874, dice: 0.008351, loss: 0.104112
LR 1e-06
val: bce: 0.203664, dice: 0.008363, loss: 0.106013
saving best model to checkpoint.pth
```

```
0m 23s
Best val loss: 0.106013
```




```
Dice Coefficient: 0.0083891469825516
IoU: 0.004212242050170816
```

The Dice Coefficient and IoU values you obtained are quite low, indicating a limited overlap between the predicted masks and the ground truth masks.
For image segmentation tasks, especially in medical imaging, metrics like the Dice Coefficient and Intersection over Union (IoU) are indeed often more informative than accuracy. These metrics are particularly useful for evaluating the performance of models in tasks where the classes are imbalanced, such as in our case with pathology images.