Part A:

1. Image Captioning

Scenario 1

- Domain: E-commerce
- Input data: Product images
- Expected output: Descriptive captions for each product
- Company which has deployed: Amazon
- Paper/blog reference: https://aclanthology.org/2021.acl-short.36/

Scenario 2

- Domain: Social Media
- Input data: User-uploaded photos
- Expected output: Automated photo descriptions to improve accessibility
- Company which has deployed: Facebook
- Paper/blog reference: https://dl.acm.org/doi/fullHtml/10.1145/3529836.3529856

Scenario 3

- Domain: Digital Asset Management
- Input data: Photos and videos in large archives
- Expected output: Captions and tags for easy search and retrieval
- Company which has deployed: Getty Images
- Paper/blog reference: https://web.library.uq.edu.au/research-tools-techniques/search-techniques/find-everything-your-topic/cited-reference-searching

Scenario 4

- Domain: Healthcare
- Input data: Medical images
- Expected output: Detailed descriptions of findings in medical images
- Company which has deployed: Zebra Medical Vision
- Paper/blog reference: https://ieeexplore.ieee.org/document/9521159

2. Visual Question Answering

Scenario 1

- Domain: Retail
- Input data: Images of products with questions about them
- Expected output: Answers related to product details, availability, etc.
- Company which has deployed: Walmart
- Paper/blog reference: https://www.researchgate.net/publication/368572075_Product_Question_Answering_in_E-Commerce_A_Survey

Scenario 2

- Domain: Education
- Input data: Educational diagrams and images
- Expected output: Answers to questions related to the content of the diagrams
- Company which has deployed: Quizlet
- Paper/blog reference: https://www.researchgate.net/figure/The-statistical-analysis-of-the-answers-to-the-questions-in-the-form-of-diagrams_fig4_340833744

Scenario 3

- Domain: Customer Support
- Input data: Product images with user inquiries
- Expected output: Automated responses to user questions
- Company which has deployed: IKEA
- Paper/blog reference: https://www.smartinsights.com/marketplace-analysis/customer-analysis/how-ikea-are-innovating-in-customer-research/

Scenario 4

- Domain: Automotive
- Input data: Car interior images with questions about features
- Expected output: Answers regarding car features and specifications
- Company which has deployed: Tesla
- Paper/blog reference: https://arxiv.org/html/2406.09203v1

3. Object Detection

Scenario 1

- Domain: Security
- Input data: Surveillance footage
- Expected output: Detection of suspicious objects and activities
- Company which has deployed: Hikvision
- Paper/blog reference: https://www.researchgate.net/publication/380182564_A_Critical_Study_on_Suspicious_Object_Detection_with_Images_and_Videos_Using_Machine_Learning_Techniques

Scenario 2

- Domain: Retail
- Input data: Store camera feeds
- Expected output: Monitoring product placement and stock levels
- Company which has deployed: Amazon Go
- Paper/blog reference: https://fastercapital.com/topics/monitoring-and-evaluating-the-success-of-product-placement-efforts.html

Scenario 3

- Domain: Agriculture
- Input data: Drone images of crops
- Expected output: Detection of weeds and pests
- Company which has deployed: John Deere
- Paper/blog reference: https://www.researchgate.net/publication/377150443_Precision_Weed_Management_using_Artificial_Intelligence_Tools_and_Techniques_for_Sustainable_Agriculture

Scenario 4

- Domain: Automotive
- Input data: Vehicle camera feeds
- Expected output: Detection of pedestrians, other vehicles, and road signs
- Company which has deployed: Tesla
- Paper/blog reference: https://www.researchgate.net/publication/347016104_Recognition_of_Vehicles_Pedestrians_and_Traffic_Signs_Using_Convolutional_Neural_Networks

4. Image Segmentation

Scenario 1

- Domain: Healthcare
- Input data: Medical scans (e.g., MRI, CT)
- Expected output: Segmentation of tumors and other anomalies
- Company which has deployed: Siemens
- Paper/blog reference: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10527911/

Scenario 2

- Domain: Autonomous Vehicles
- Input data: Road scene images
- Expected output: Segmentation of lanes, vehicles, pedestrians
- Company which has deployed: Waymo
- Paper/blog reference: https://www.sciencedirect.com/science/article/abs/pii/S0921889020304632

Scenario 3

- Domain: Agriculture
- Input data: Satellite images of farmland
- Expected output: Segmentation of different crop types
- Company which has deployed: Climate Corporation
- Paper/blog reference: segmentation_using_satelite_images

Scenario 4

- Domain: Manufacturing
- Input data: Images of assembly lines
- Expected output: Segmentation of different components for quality control
- Company which has deployed: Siemens
- Paper/blog reference: https://www.researchgate.net/publication/375913224_Improving_Quality_Inspections_with_Image_Analysis_and_Artificial_Intelligence

5. Image Similarity Computation

Scenario 1

- Domain: E-commerce
- Input data: Product images
- Expected output: Finding visually similar products
- Company which has deployed: ASOS
- Paper/blog reference: https://www.researchgate.net/publication/377339515_Artificial_Intelligence_Algorithms_For_Object_Detection_and_Recognition_In_video_and_Images

Scenario 2

- Domain: Social Media
- Input data: User-uploaded photos
- Expected output: Finding similar images for recommendation
- Company which has deployed: Pinterest
- Paper/blog reference: https://www.csail.mit.edu/news/researchers-use-ai-identify-similar-materials-images

Scenario 3

- Domain: Digital Asset Management
- Input data: Large image datasets
- Expected output: Finding duplicate or near-duplicate images
- Company which has deployed: Shutterstock
- Paper/blog reference: https://www.researchgate.net/publication/361912810_Detecting_Near_Duplicate_Dataset_with_Machine_Learning

Scenario 4

- Domain: Healthcare
- Input data: Medical images
- Expected output: Finding similar cases for diagnostic assistance
- Company which has deployed: IBM Watson Health
- Paper/blog reference: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8754556/

6. Action Recognition

Scenario 1

- Domain: Sports Analytics
- Input data: Videos of sports matches
- Expected output: Recognizing and analyzing player actions
- Company which has deployed: Hudl
- Paper/blog reference: https://www.mdpi.com/1424-8220/20/11/3040

Scenario 2

- Domain: Security
- Input data: Surveillance footage
- Expected output: Detecting suspicious or abnormal activities
- Company which has deployed: Hikvision
- Paper/blog reference: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9252660/

Scenario 3

- Domain: Healthcare
- Input data: Videos of patients
- Expected output: Monitoring and analyzing patient movements
- Company which has deployed: Philips Healthcare
- Paper/blog reference: https://www.researchgate.net/publication/366904680_Remote_patient_monitoring_using_artificial_intelligence_Current_state_applications_and_challenges

Scenario 4

- Domain: Retail
- Input data: Store surveillance videos
- Expected output: Recognizing customer behaviors and interactions
- Company which has deployed: Walmart
- Paper/blog reference: https://www.researchgate.net/publication/355069187_AI_in_Consumer_Behavior

7. Multi-object Tracking

Scenario 1

- Domain: Security
- Input data: Surveillance video feeds
- Expected output: Tracking multiple individuals in crowded areas
- Company which has deployed: Hikvision
- Paper/blog reference: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8157856/

Scenario 2

- Domain: Autonomous Vehicles
- Input data: Vehicle camera feeds
- Expected output: Tracking other vehicles, pedestrians, and obstacles
- Company which has deployed: Tesla
- Paper/blog reference: https://www.mdpi.com/2078-2489/15/2/104

Scenario 3

- Domain: Sports Analytics
- Input data: Videos of sports matches
- Expected output: Tracking players and ball movements
- Company which has deployed: Hawk-Eye Innovations
- Paper/blog reference: https://www.researchgate.net/publication/3832521_Tracking_players_and_a_ball_in_soccer_games

Scenario 4

- Domain: Wildlife Conservation
- Input Data: Footage from camera traps in a wildlife reserve
- Expected Output: Tracking multiple animals to study their behavior and monitor their populations
- Company: National Geographic
- Paper/blog reference: https://www.nature.com/articles/s41592-022-01426-1

8. Image Classification

Scenario 1

- Domain: Healthcare
- Input Data: Medical images (e.g., X-rays, MRIs)
- Expected Output: Classification of images into different categories, such as healthy or various types of anomalies
- Company: Zebra Medical Vision
- Paper/Blog Reference: https://www.researchgate.net/publication/375952278_Deep_Learning_for_Image_Authentication_A_Comparative_Study_on_Real_and_AI-Generated_Image_Classification

Scenario 2

- Domain: Agriculture
- Input Data: Images of crops
- Expected Output: Classification of different crop types and detection of diseases
- Company: John Deere
- Paper/Blog Reference: Plant Disease Detection Using Deep Learning

Scenario 3

- Domain: Finance
- Input Data: Document images (e.g., invoices, receipts)
- Expected Output: Classification of documents into predefined categories for automated processing
- Company: JP Morgan Chase
- Paper/Blog Reference: https://www.docsumo.com/blog/document-classification

Scenario 4

- Domain: Environmental Monitoring
- Input Data: Satellite images
- Expected Output: Classification of land cover types (e.g., urban, forest, water bodies)
- Company: Planet Labs
- Paper/Blog Reference: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9853425/

Part B:

1. As the initial step, we loaded the data and split it into 2 folders: train and test. The train folder contains 1:40 images of each class; the rest are stored in the test folder. Loaded data was collected using different loaders for training and testing.
2. The pre-trained **ResNet18, DenseNet, and VGG19** models were loaded as the next step to perform **image classification**. For training, these model weights were frozen and replaced in the final layer with a number of classes in this data set, and 3 models were trained. The metrics for these models are listed below.

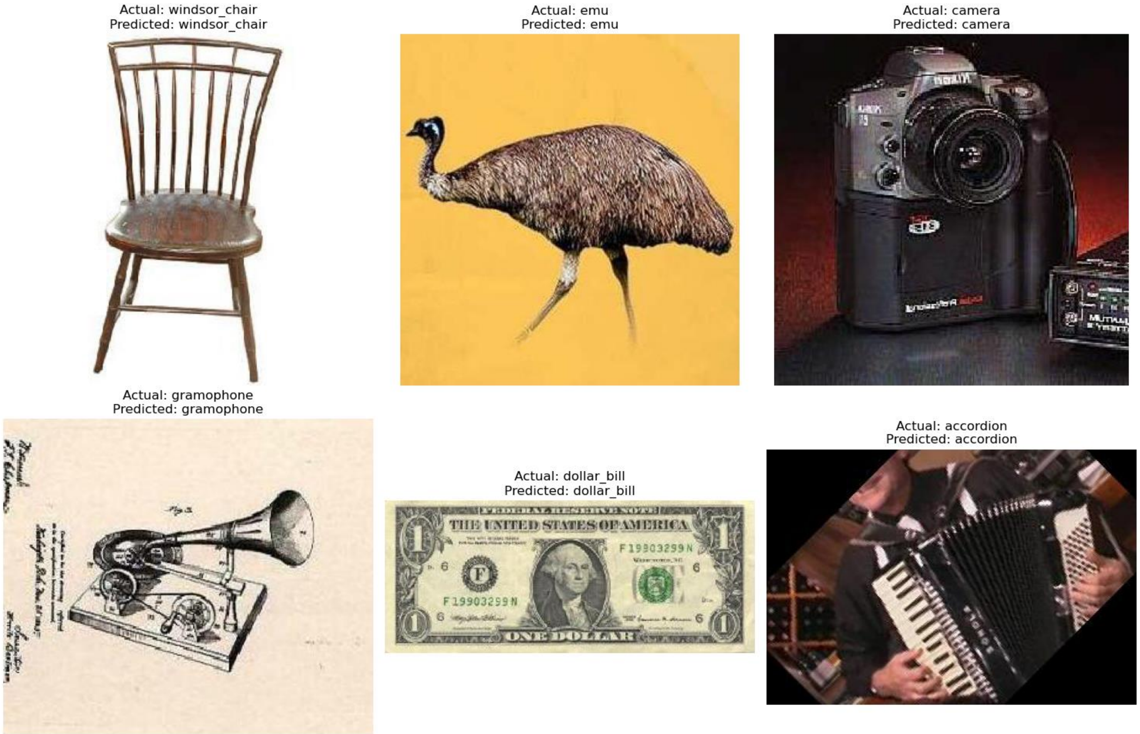
   ResNet18:

   - On training over 10 epochs, the **ResNet18** model shows an increase in training accuracy, from **72.67% to 98.17%.**
   - Test accuracy also shows an increase, from **71.22% to 89.27%.** Train losses gradually decrease, while test losses also decrease with certain variables.
   - Each class's precision and recall parameters reflect model performance variations across classes.
   - By the final epoch, both metrics improve significantly for most classes, indicating an increase in model robustness and generalizability.
   - Taken randomly 6 images from the test folder. Evaluated using the model trained. The results are as follows.

From the randomly chosen 6 images, the ResNet18 model predicts all the labels correctly.

DenseNet121

- On training over 10 epochs, the model exhibits a steady improvement as training accuracy increases from **94.17% to 99%.**
- Test accuracy follows a similar trend, improving from **70.73% to a maximum of 92.20%.** However, the experimental loss is variable, increasing mainly at time 8 before consolidation.
- The precision and recall metrics in each class exhibit variability. In general, the model shows high accuracy and recall for many classes, and there is a remarkable improvement in both metrics until the final period.
- Overall performance shows effective curriculum and flexibility, although at times, it shows signs of overfitting. Early stopping can be employed to as the test loss increases in the final epochs.
- Randomly chosen 6 images from the test folder were evaluated based on the model trained. The predictions are as follows.



The DenseNet121 model has predicted all 6 images correctly.

VGG19

- On the model training over epochs, it resulted in an accuracy of **38% to 45%.**
- The precision across classes. Few classes resulted in higher precision, while few had very low. We can infer that the model predicted all positives correctly.
- Similarly with recall. A few classes resulted in higher recall, and a few had low recall. High recall for a few classes indicates that the model is thorough in capturing the most true instances of those classes, but it may also misclassify some negatives as positives.
- Randomly chosen 6 images from the test folder and evaluated on the model. The results are as follows.

2/6 images are classified correctly.

3. Fine Tuning VGG-19 Model:

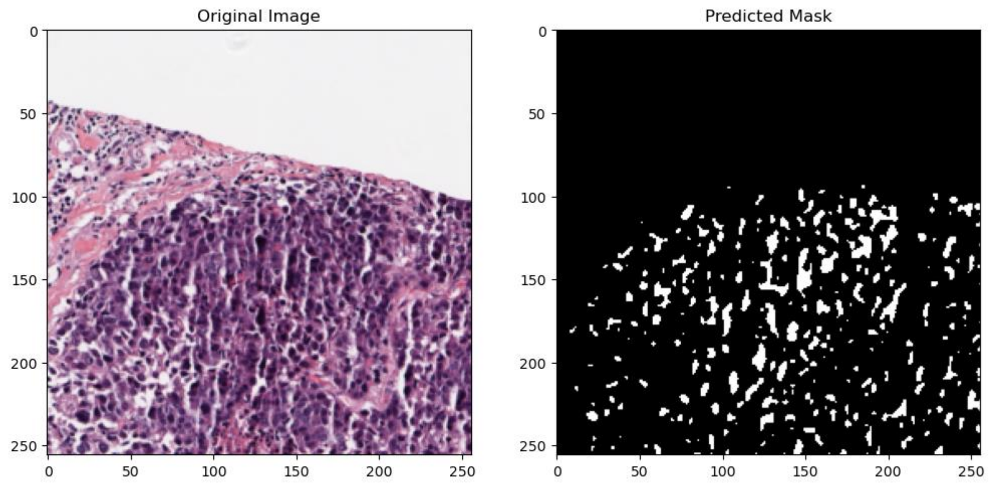
- Performed data loading and splitting as mentioned. Using Tensor-flow, performed data augmentation.
- As a next step defined the VGG19 model. Unfreezed a few weights and trained the model over 50 epochs for fine-tuning.
- Observed accuracy, precision, and recall improved over a few epochs. Initially, they were low, but the accuracy, precision, and recall increased over the few epochs and iterations.
- Early stopping in the code was used to stop the model when it started to overfit.
- Finally got a **test accuracy of 92%, a precision of 91%, and a recall of 84%.** This shows the model is able to identify true positives correctly.
- The loss of the model was high initially, but over the epochs, it was reduced significantly in the final epochs, which is a 33% loss in the final epochs. This shows the efficiency of the model.
- Tested model on a few test images by randomly choosing them. And on few unseen data (not from the dataset).

The model correctly classified 6/6 images.

From the above, we see how the VGG19 model's accuracy improved by fine-tuning our dataset. Initially using a pre-trained model it showed very less accuracy. However, fine-tuning it on our dataset has improved the results significantly.

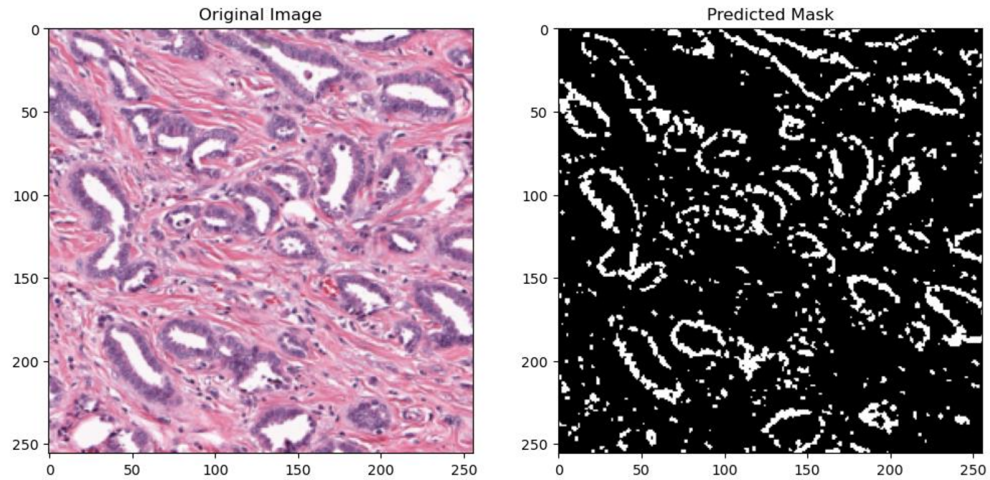
Part C:

- As an initial step, the data loading was performed. Split the images into train and test folders.
- Using Pytorch to transform the data. Later, we developed a U-Net model with a ResNet backbone to perform **Image Segmentation**.
- Defined the model metrics and trained the model over 24 epochs. The training loss was loss, i.e., 33%, but the test loss was higher. Which shows model overfitting.
- Tested the model on test images by randomly choosing images from the test folder.



Model -1

- The bce loss and Dice were on the higher side. Later, we used the existing U-Net model with a ResNet backbone from PyTorch.
- Defined the model and metrics for evaluation of the model. Trained it over 25 epochs. Got an improved test accuracy of 78%, and loss is also minimized.
- Tested both models on randomly choosing images from test folders. The results are as follows.



Model-2

- Second model segmentation is more compared to model 1. Model-1 can be remodeled by changing layers and re-arranging convolution layers to improve the accuracy of the model.

Part D:

- Data loading is performed as an initial step. Images and annotations were split into test and train folders. The train folder contains 100 images and annotations in XML format. Test folders contains 28 images and annotations in XML format.
- Defined functions to extract text from XML annotated files. The data set contains 4 classes (water bottle, milk bottle, tetra pack, can) + background.
- Defined a Faster RCNN model with metrics per class to perform **Object Detection.**
- Trained the model over 10 epochs. The loss was reduced significantly over the epochs. Finally, it achieved a validation l**oss of 0.03.** This shows that the model is able to make accurate predictions.
- The precision values are higher across all classes. Class water bottles and tetra packs can have higher values than milk bottles.
- The recall values fluctuated over the epochs across classes.
- Since we give higher weightage to false positives in this model, we can consider precision over recall. Precision gives importance to false positives rather than false negatives.
- Tested the model by taking random images from the test folder. The results as follows.

Image 2 - Predictions

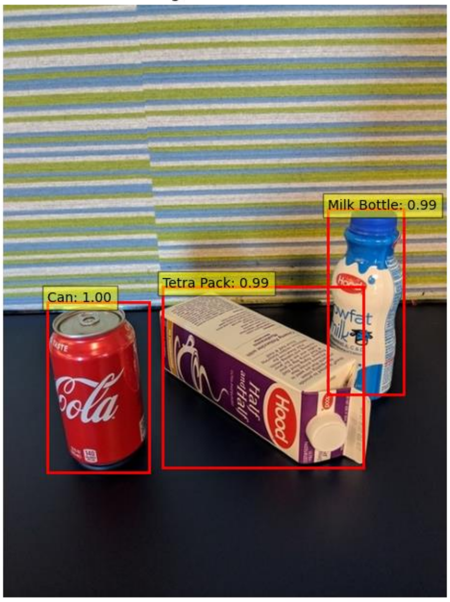
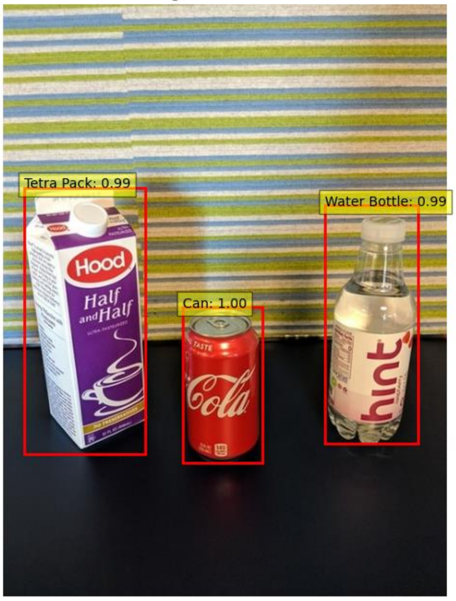


Image 3 - Predictions



- We can see the model correctly identifies the objects with higher precision.