Avery Nelson --- 0 late days used

010920903

amn018@uark.edu

Artificial Intelligence Assignment 5 Report

Description of Implementation

The goal of this assignment was to study deep convolutional neural networks for image classification and train a classification model. The assignment consisted of two main parts: using specific models to classify images and training a custom model on the CIFAR-10 dataset.

For the first part, I worked with specific model types to classify images. I used three different backbones: ResNet-50, ResNet-101, and DenseNet-121. The implementation required downloading ten random images of different objects from the internet and using these models to predict the class names and their corresponding probabilities. I then visualized the top five predictions (the five classes with the highest probabilities) for each image across all three models.

The implementation started with defining appropriate image transformations. These transformations resized the images, converted them to tensors, and normalized them using the mean and standard deviation values that the models were trained with. I loaded the ImageNet class names from a pickle file to interpret the numeric predictions from the models.

For visualizing the top five predictions, I implemented a function that would:

1) Load and preprocess the image.
2) Pass it through the model.
3) Extract the top five predictions and their probabilities.
4) Display the image alongside a bar chart showing these predictions.

For the second part of the assignment, I implemented a Convolutional Neural Network (CNN) for the CIFAR-10 dataset. I started by defining the training data loader. I then designed a CNN architecture with the following components:

1) Three convolutional layers with batch normalization and max pooling
2) Fully connected layers with dropout for regularization
3) ReLU activation functions

The training framework included:

1) Forward pass to compute outputs
2) Loss computation using CrossEntropyLoss.
3) Accuracy calculation
4) Backpropagation and optimization using Adam optimizer.
5) Tracking training metrics (loss and accuracy)

For evaluation, I implemented a similar code for the test set, adding top five accuracy calculations. I also added code to generate a confusion matrix using sklearn and visualize correctly predicted samples, which helped in analyzing the model's strengths and weaknesses across different classes.

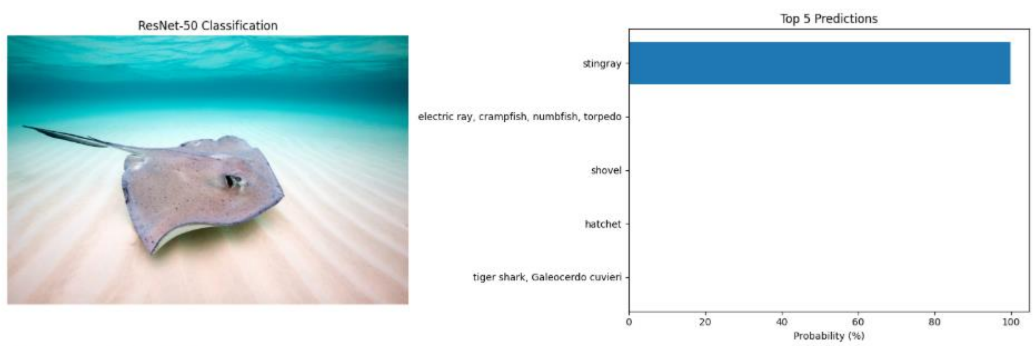
Results

I downloaded ten images of different objects and ran them through ResNet-50, ResNet-101, and DenseNet-121 models. For each image, I visualized the top five predictions from each model.

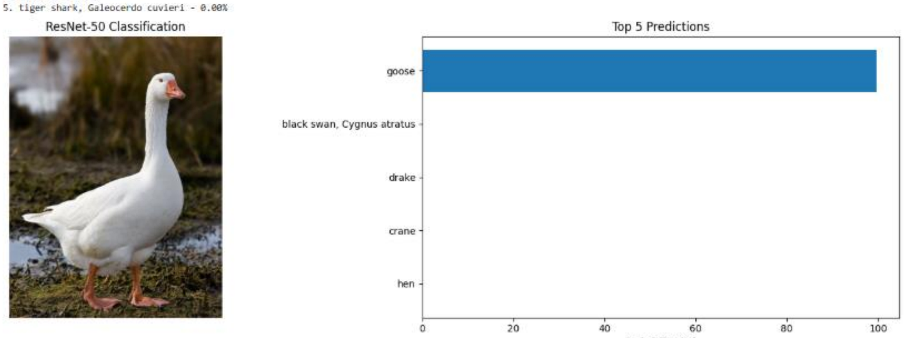
All three models showed strong performance, often correctly identifying the primary object in each image. There were some interesting variations between models, with DenseNet-121 sometimes providing more accurate predictions for certain objects than the ResNet models, while the ResNet models performed better on others.

For example, on images containing animals, all three models consistently identified the correct species with high confidence. On more complex images with multiple objects, the models sometimes focus on various aspects of the image, resulting in varying predictions.

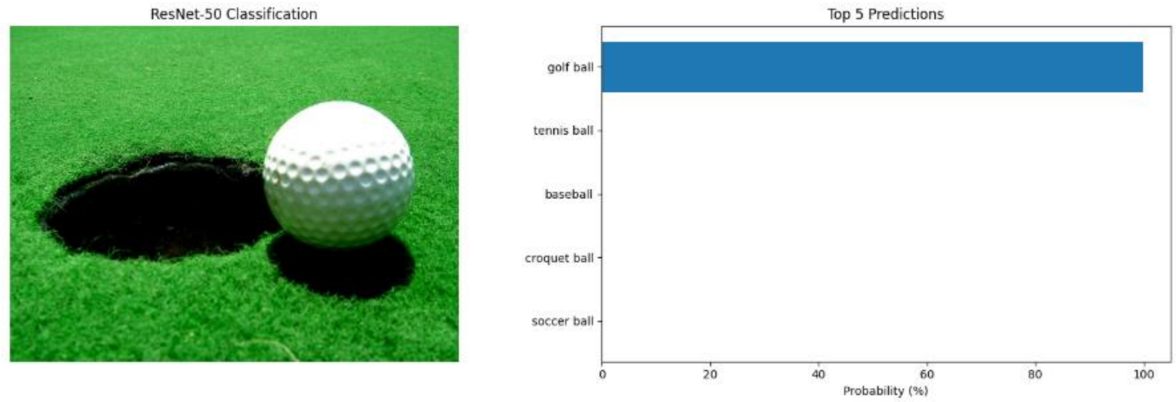
Resnet 50 Predictions:



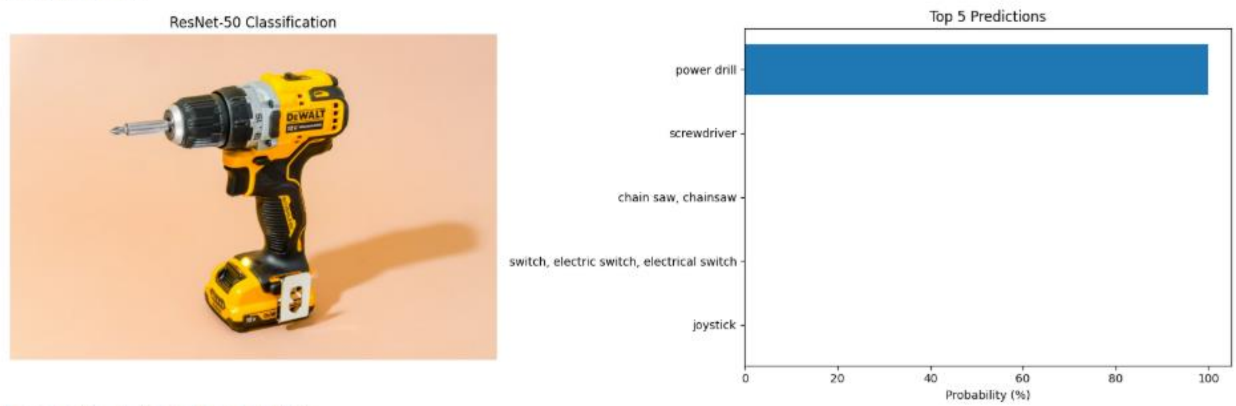

```
Top 5 predictions for image1.jpg using ResNet-50:
1. stingray - 99.80%
2. electric ray, crampfish, numbfish, torpedo - 0.20%
3. shovel - 0.00%
4. hatchet - 0.00%
5. tiger shark, Galeocerdo cuvieri - 0.00%
```



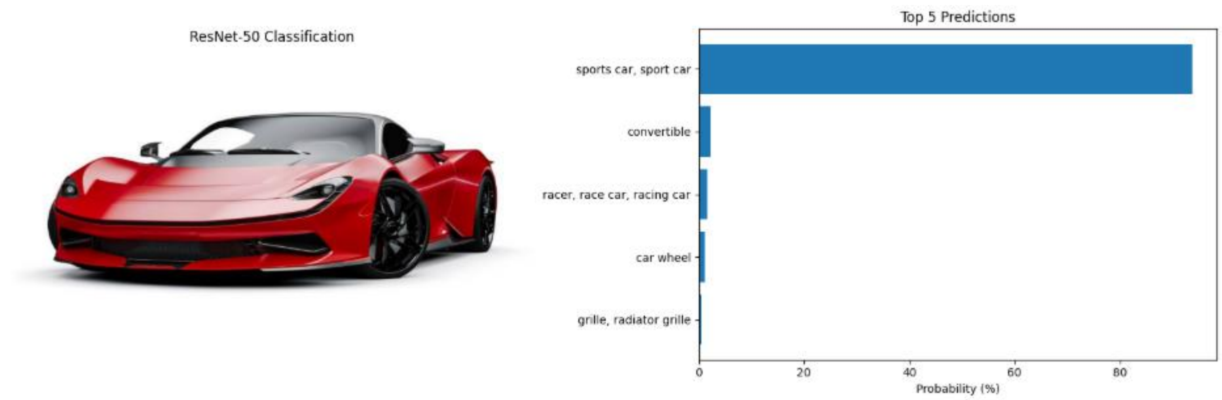

```
Top 5 predictions for image2.jpg using ResNet-50:
1. goose - 99.68%
2. black swan, Cygnus atratus - 0.15%
3. drake - 0.05%
4. crane - 0.03%
5. hen - 0.02%
```



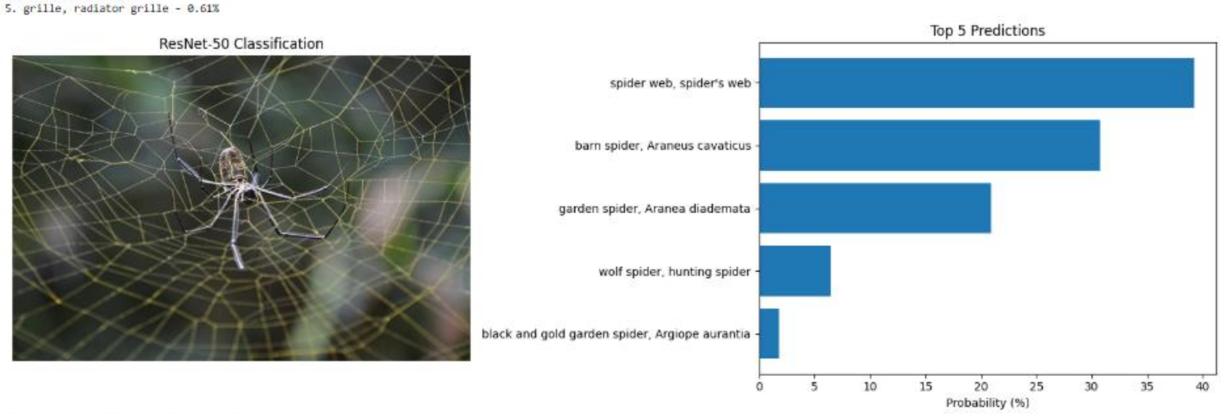

```
Top 5 predictions for image3.jpg using ResNet-50:
1. golf ball - 99.98%
2. tennis ball - 0.01%
3. baseball - 0.01%
4. croquet ball - 0.00%
5. soccer ball - 0.00%
```



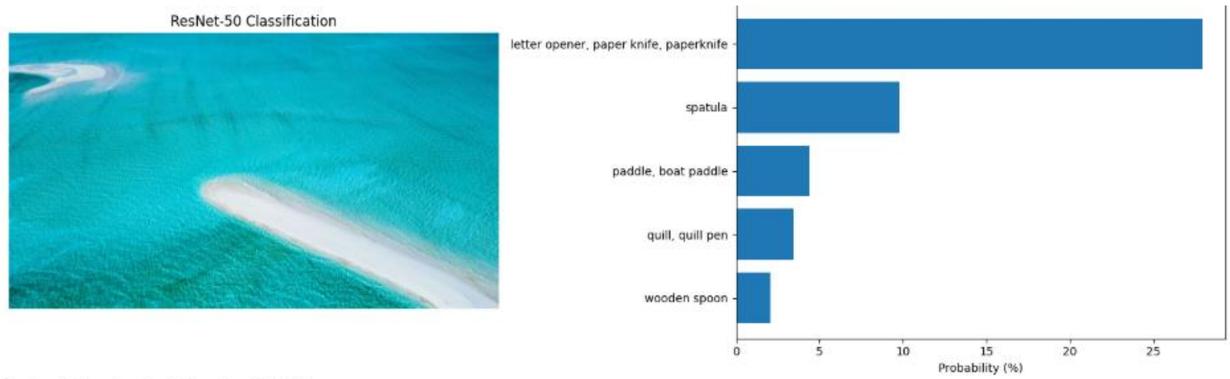

```
Top 5 predictions for image4.jpg using ResNet-50:
1. power drill - 99.94%
2. screwdriver - 0.03%
3. chain saw, chainsaw - 0.02%
4. switch, electric switch, electrical switch - 0.00%
5. joystick - 0.00%
```



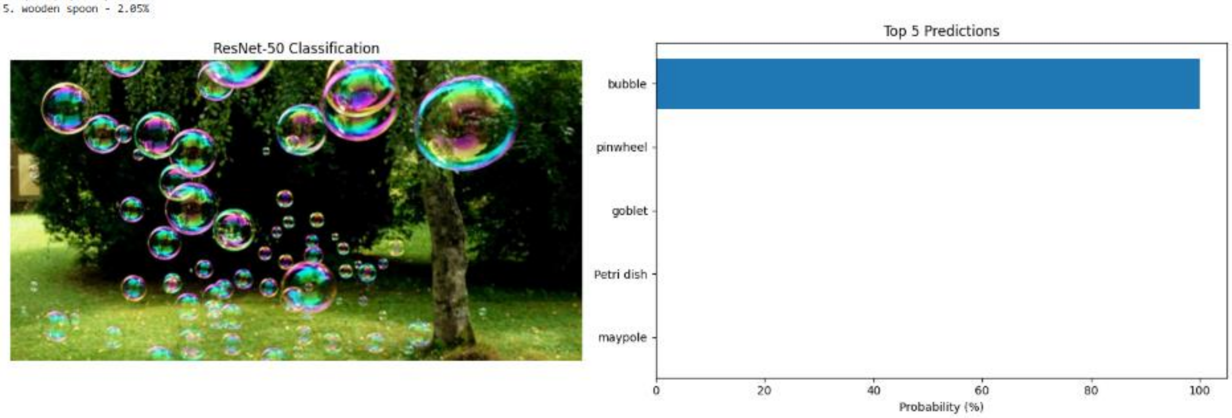

```
Top 5 predictions for image5.jpg using ResNet-50:
1. sports car, sport car - 93.76%
2. convertible - 2.18%
3. racer, race car, racing car - 1.62%
4. car wheel - 1.18%
5. grille, radiator grille - 0.61%
```



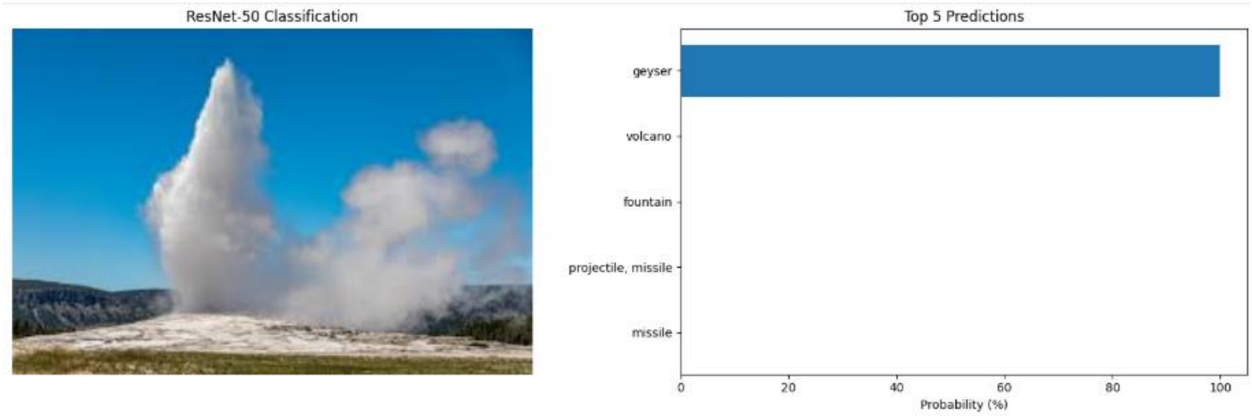

```
Top 5 predictions for image6.jpg using ResNet-50:
1. spider web, spider's web - 39.26%
2. barn spider, Araneus cavaticus - 30.73%
3. garden spider, Aranea diademata - 20.93%
4. wolf spider, hunting spider - 6.45%
5. black and gold garden spider, Argiope aurantia - 1.81%
```



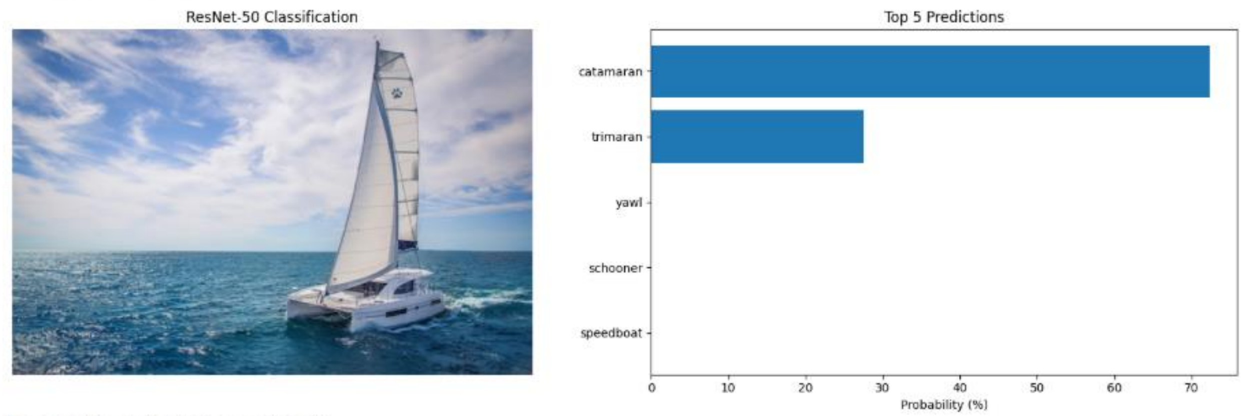

```
Top 5 predictions for image7.jpg using ResNet-50:
1. letter opener, paper knife, paperknife - 27.92%
2. spatula - 9.81%
3. paddle, boat paddle - 4.40%
4. quill, quill pen - 3.46%
5. wooden spoon - 2.05%
```




```
Top 5 predictions for image8.jpg using ResNet-50:
1. bubble - 100.00%
2. pinwheel - 0.00%
3. goblet - 0.00%
4. Petri dish - 0.00%
5. maypole - 0.00%
```

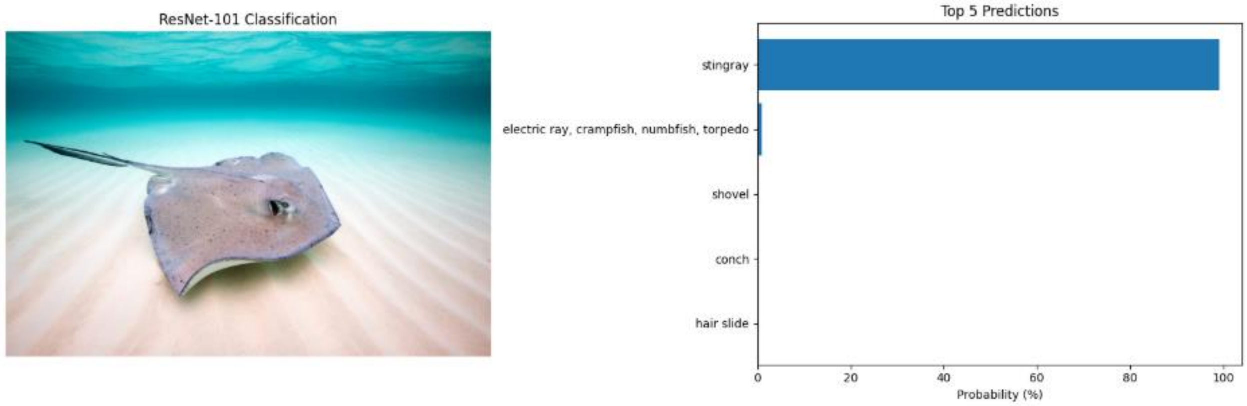



```
Top 5 predictions for image9.jpg using ResNet-50:
1. geyser - 99.98%
2. volcano - 0.01%
3. fountain - 0.00%
4. projectile, missile - 0.00%
5. missile - 0.00%
```

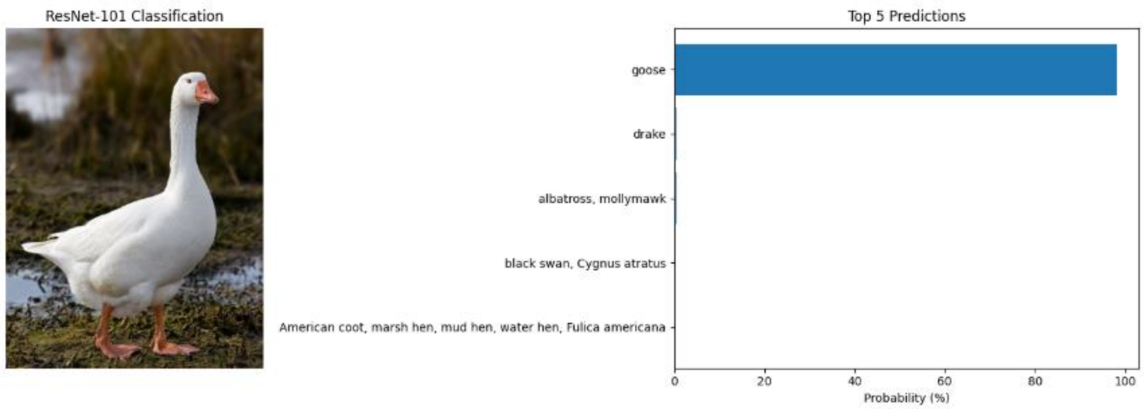



```
Top 5 predictions for image10.jpg using ResNet-50:
1. catamaran - 72.30%
2. trimaran - 27.57%
3. yawl - 0.08%
4. schooner - 0.04%
5. speedboat - 0.00%
```

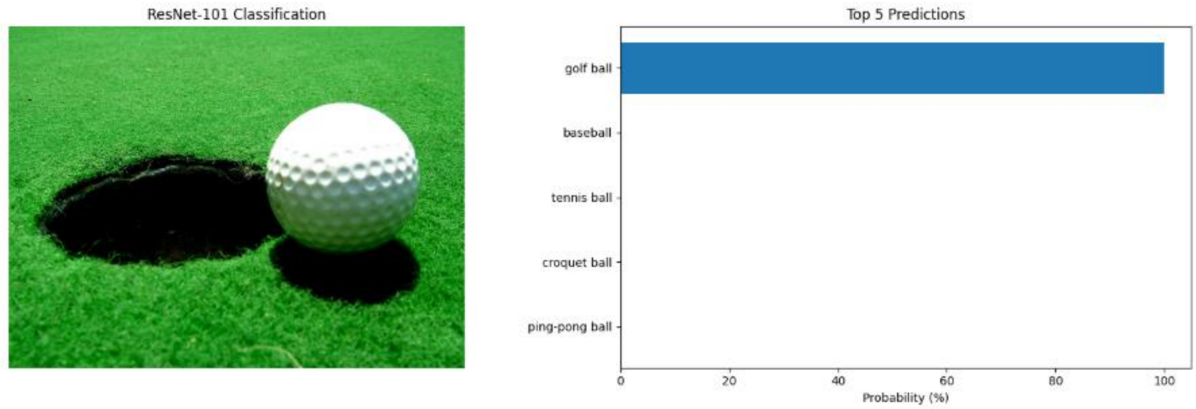
Resnet 101 Predictions:



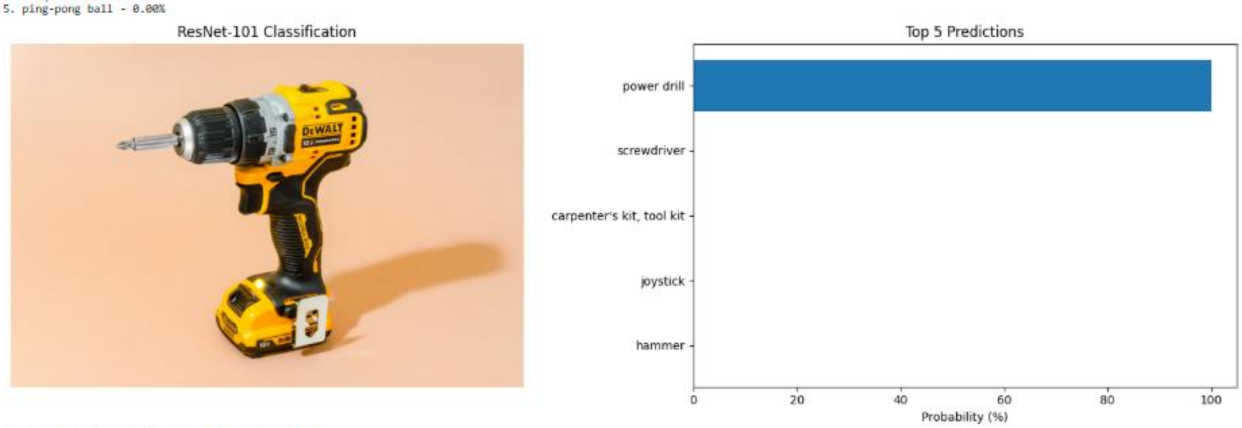

```
Top 5 predictions for image1.jpg using ResNet-101:
1. stingray - 99.09%
2. electric ray, crampfish, numbfish, torpedo - 0.90%
3. shovel - 0.00%
4. conch - 0.00%
5. hair slide - 0.00%
```



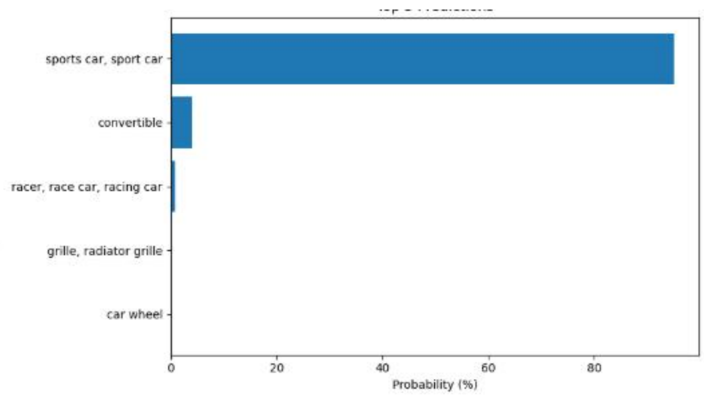

```
Top 5 predictions for image2.jpg using ResNet-101:
1. goose - 98.10%
2. drake - 0.49%
3. albatross, mollymawk - 0.37%
4. black swan, Cygnus atratus - 0.28%
5. American coot, marsh hen, mud hen, water hen, Fulica americana - 0.15%
```



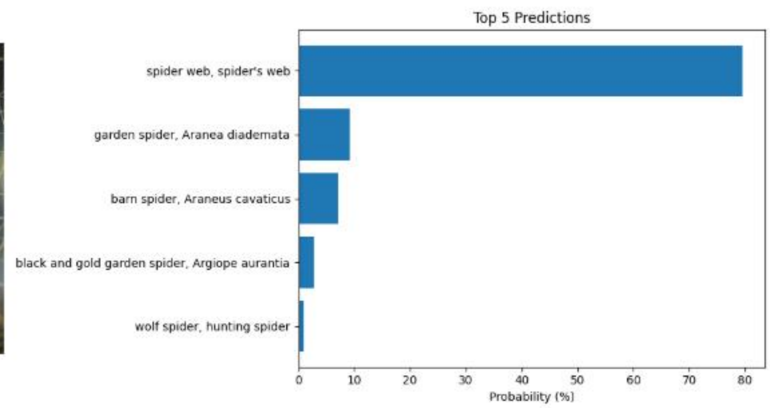

```
Top 5 predictions for image3.jpg using ResNet-101:
1. golf ball - 99.99%
2. baseball - 0.00%
3. tennis ball - 0.00%
4. croquet ball - 0.00%
5. ping-pong ball - 0.00%
```



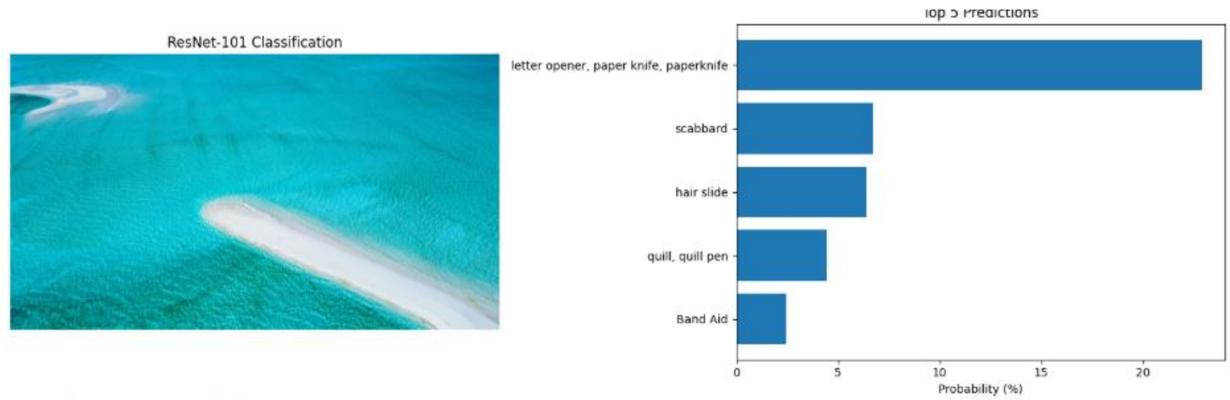

```
Top 5 predictions for image4.jpg using ResNet-101:
1. power drill - 99.99%
2. screwdriver - 0.01%
3. carpenter's kit, tool kit - 0.00%
4. joystick - 0.00%
5. hammer - 0.00%
```



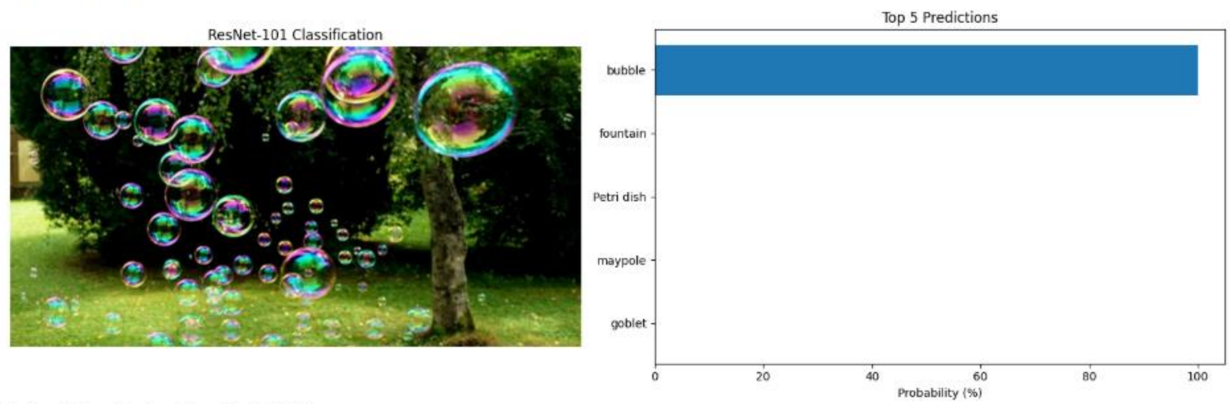

```
Top 5 predictions for image5.jpg using ResNet-101:
1. sports car, sport car - 95.09%
2. convertible - 3.97%
3. racer, race car, racing car - 0.83%
4. grille, radiator grille - 0.05%
5. car wheel - 0.04%
```



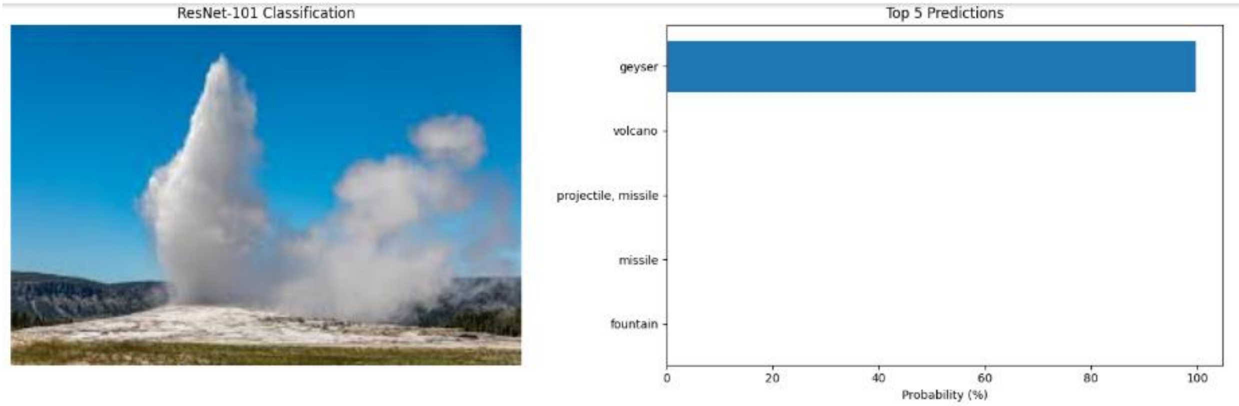

```
Top 5 predictions for image6.jpg using ResNet-101:
1. spider web, spider's web - 79.62%
2. garden spider, Aranea diademata - 9.26%
3. barn spider, Araneus cavaticus - 7.04%
4. black and gold garden spider, Argiope aurantia - 2.69%
5. wolf spider, hunting spider - 0.95%
```



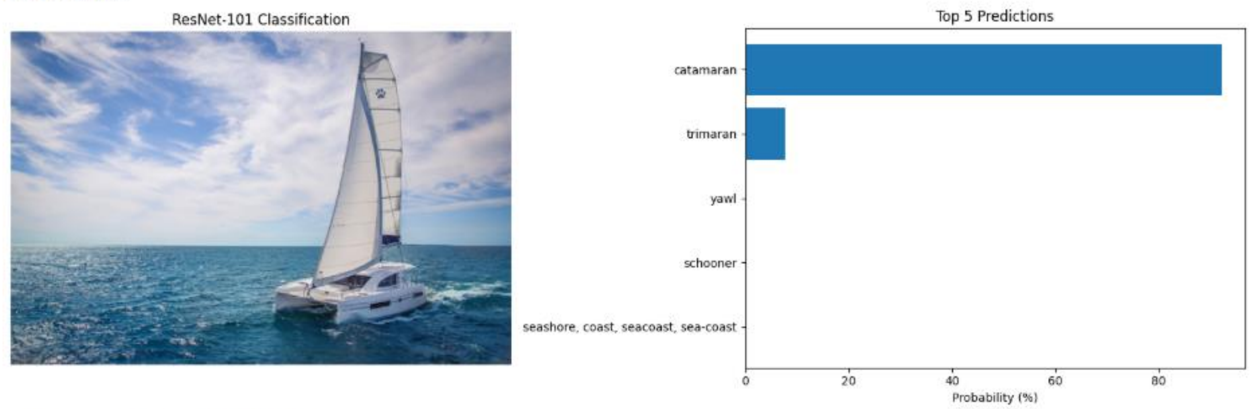

```
Top 5 predictions for image7.jpg using ResNet-101:
1. letter opener, paper knife, paperknife - 22.90%
2. scabbard - 6.74%
3. hair slide - 6.39%
4. quill, quill pen - 4.43%
5. Band Aid - 2.44%
```




```
Top 5 predictions for image8.jpg using ResNet-101:
1. bubble - 100.00%
2. fountain - 0.00%
3. Petri dish - 0.00%
4. maypole - 0.00%
5. goblet - 0.00%
```

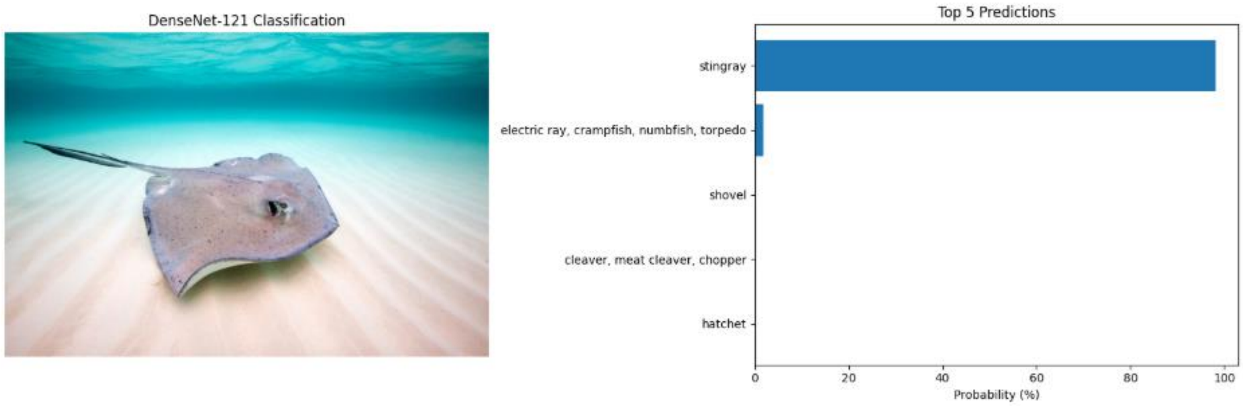


Top 5 predictions for image9.jpg using ResNet-101:
1. geyser - 99.86%
2. volcano - 0.11%
3. projectile, missile - 0.01%
4. missile - 0.00%
5. fountain - 0.00%

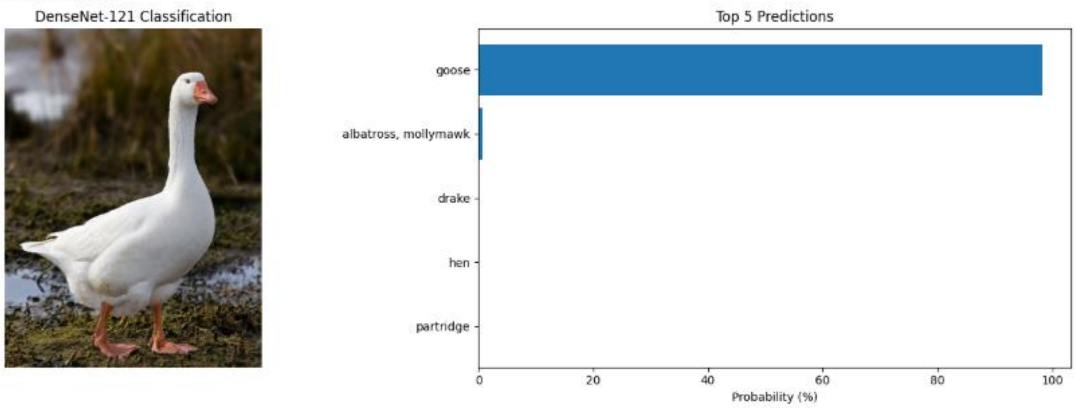


Top 5 predictions for image10.jpg using ResNet-101:
1. catamaran - 92.15%
2. trimaran - 7.76%
3. yawl - 0.04%
4. schooner - 0.02%
5. seashore, coast, seacoast, sea-coast - 0.00%

DenseNet 121 Predictions:

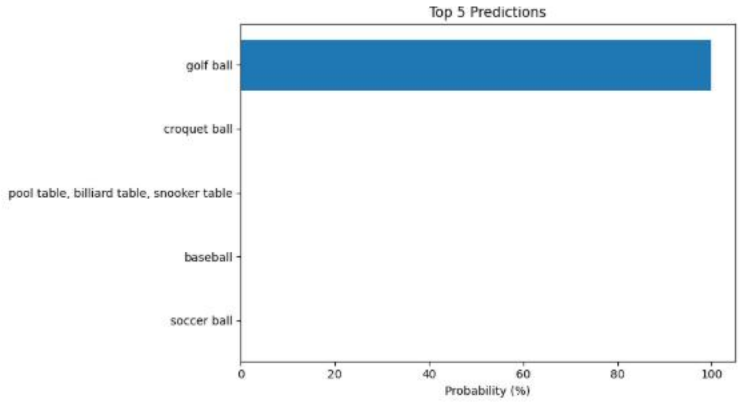



```
Top 5 predictions for image1.jpg using DenseNet-121:
1. stingray - 98.08%
2. electric ray, crampfish, numbfish, torpedo - 1.90%
3. shovel - 0.01%
4. cleaver, meat cleaver, chopper - 0.00%
5. hatchet - 0.00%
```



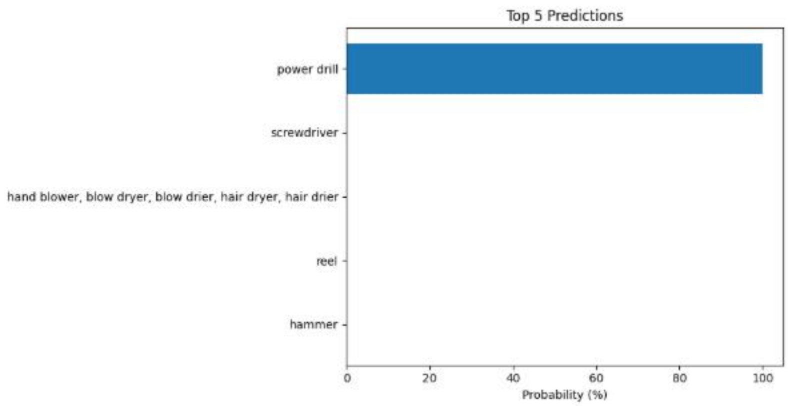

```
Top 5 predictions for image2.jpg using DenseNet-121:
1. goose - 98.32%
2. albatross, mollymawk - 0.77%
3. drake - 0.15%
4. hen - 0.11%
5. partridge - 0.10%
```

DenseNet-121 Classification

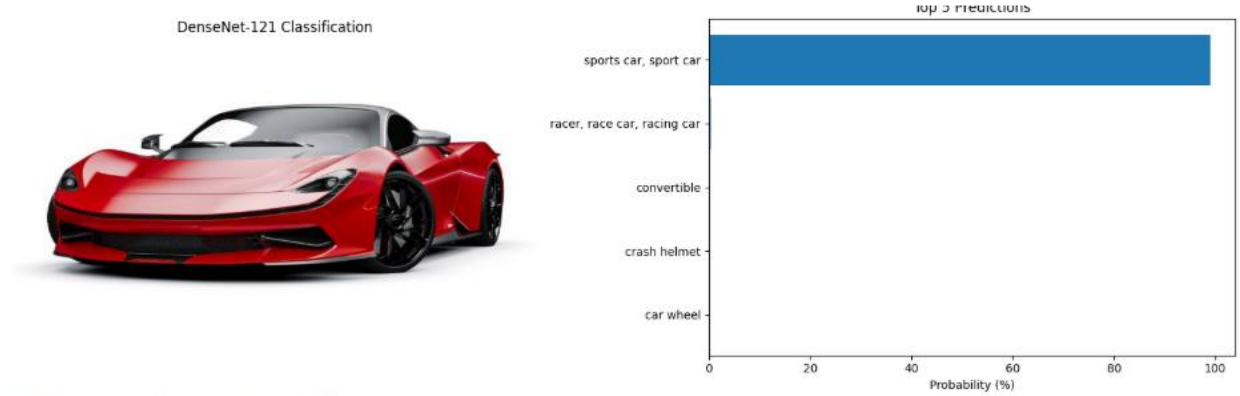



```
Top 5 predictions for image3.jpg using DenseNet-121:
1. golf ball - 100.00%
2. croquet ball - 0.00%
3. pool table, billiard table, snooker table - 0.00%
4. baseball - 0.00%
5. soccer ball - 0.00%
```

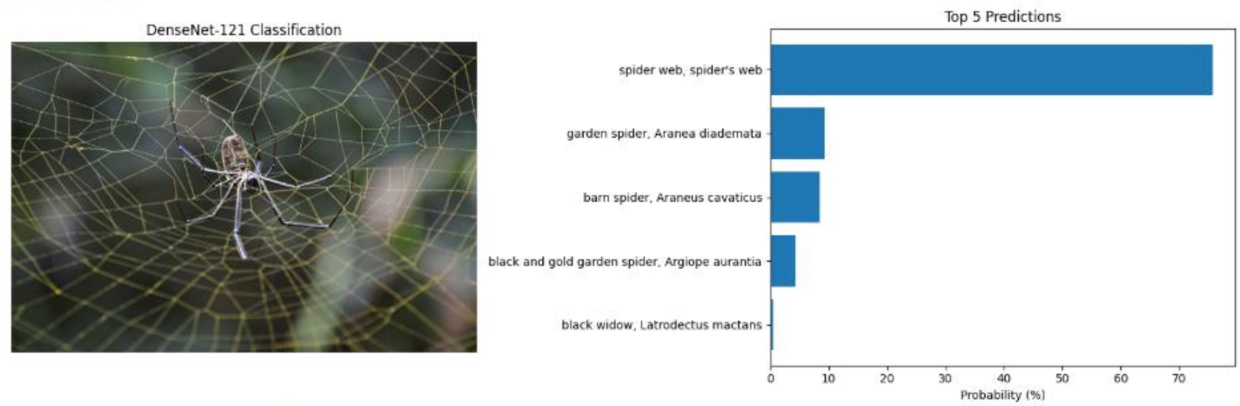
DenseNet-121 Classification




```
Top 5 predictions for image4.jpg using DenseNet-121:
1. power drill - 99.97%
2. screwdriver - 0.02%
3. hand blower, blow dryer, blow drier, hair dryer, hair drier - 0.00%
4. reel - 0.00%
5. hammer - 0.00%
```

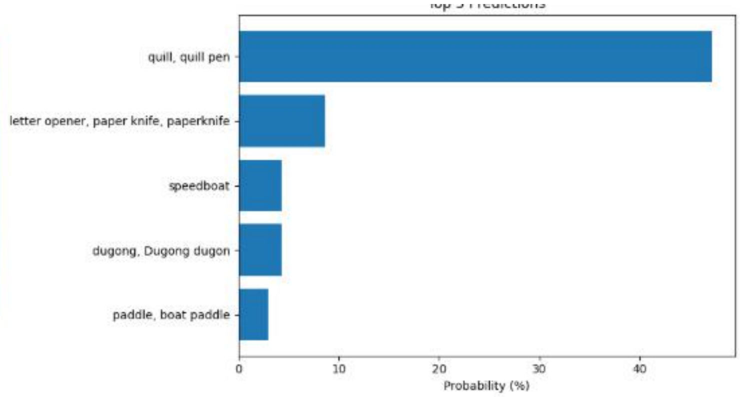



```
Top 5 predictions for image5.jpg using DenseNet-121:
1. sports car, sport car - 98.99%
2. racer, race car, racing car - 0.34%
3. convertible - 0.27%
4. crash helmet - 0.20%
5. car wheel - 0.11%
```



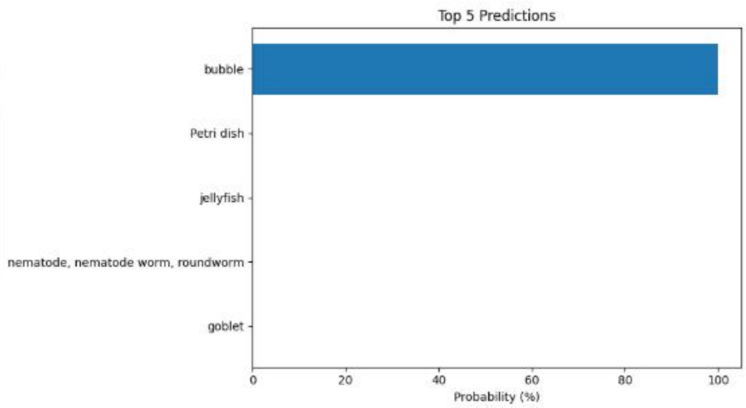

```
Top 5 predictions for image6.jpg using DenseNet-121:
1. spider web, spider's web - 75.83%
2. garden spider, Aranea diademata - 9.27%
3. barn spider, Araneus cavaticus - 8.49%
4. black and gold garden spider, Argiope aurantia - 4.29%
5. black widow. Latrodectus mactans - 0.55%
```

DenseNet-121 Classification

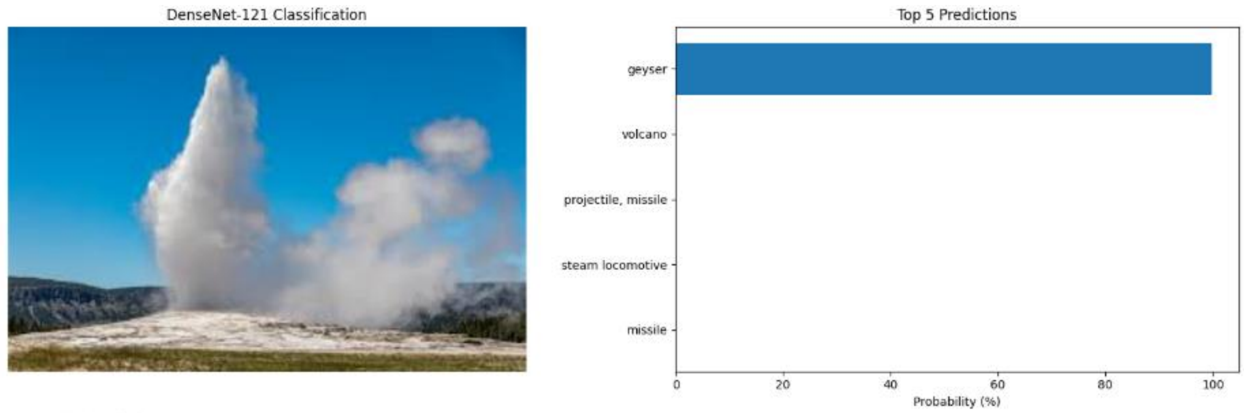



```
Top 5 predictions for image7.jpg using DenseNet-121:
1. quill, quill pen - 47.19%
2. letter opener, paper knife, paperknife - 8.70%
3. speedboat - 4.35%
4. dugong, Dugong dugon - 4.31%
5. paddle, boat paddle - 2.98%
```

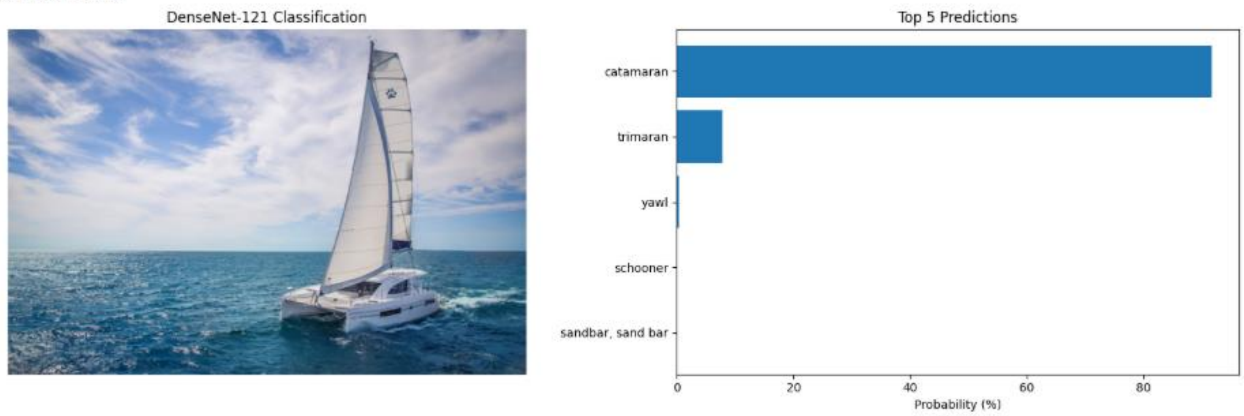
DenseNet-121 Classification




```
Top 5 predictions for image8.jpg using DenseNet-121:
1. bubble - 100.00%
2. Petri dish - 0.00%
3. jellyfish - 0.00%
4. nematode, nematode worm, roundworm - 0.00%
5. goblet - 0.00%
```



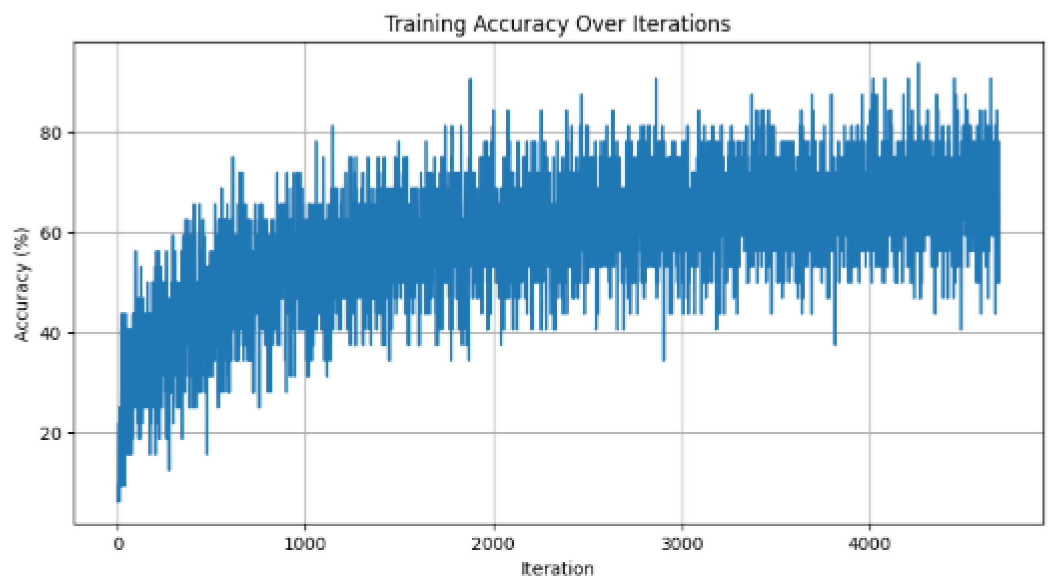

```
Top 5 predictions for image9.jpg using DenseNet-121:
1. geyser - 99.81%
2. volcano - 0.08%
3. projectile, missile - 0.02%
4. steam locomotive - 0.01%
5. missile - 0.01%
```




```
Top 5 predictions for image10.jpg using DenseNet-121:
1. catamaran - 91.77%
2. trimaran - 7.90%
3. yawl - 0.32%
4. schooner - 0.01%
5. sandbar, sand bar - 0.00%
```

The visualizations show the input image alongside a bar chart of the top five predictions, clearly showing how each model processed the images. The implementation successfully handles image pre-processing through appropriate resizing, normalization, and tensor conversion, which are essential steps for working with pre-trained models.

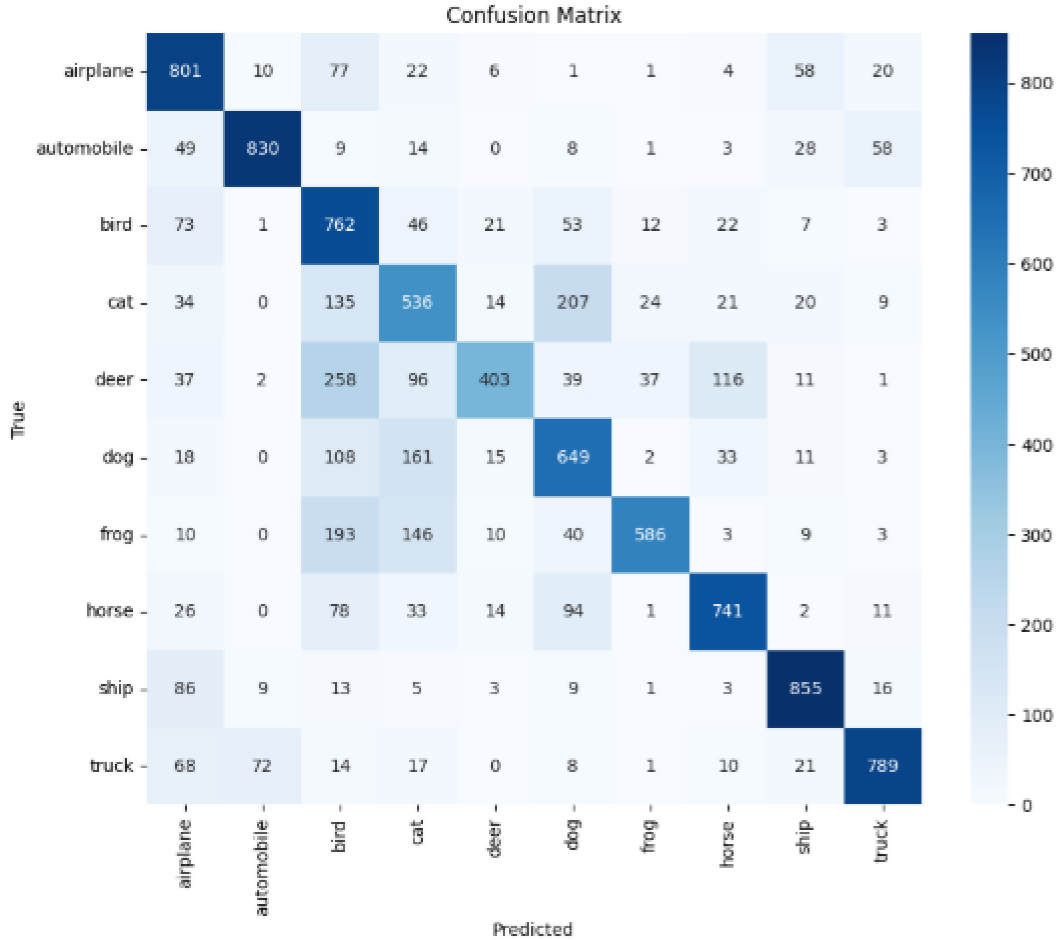
For the CIFAR-10 classification task, I implemented a CNN architecture with three convolutional layers followed by max pooling and batch normalization. The training process showed consistent improvement over the three epochs. The model quickly learned to distinguish between the classes, with training accuracy improving steadily.



The training loss decreased steadily throughout the training process, indicating that the model was learning effectively. The accuracy curve showed consistent improvement, with occasional fluctuations due to the stochastic nature of mini-batch training.

When evaluated on the test set, the model achieved a final accuracy of 69.52% and a top five accuracy of around 97%. This indicates that while the model sometimes confuses the primary class, it usually includes the correct class within its top five predictions.

```
Iter [1/100]. Accuracy: 69.00
Final Accuracy: 69.52
Top 5 Accuracy: 0.970
```



The confusion matrix revealed interesting patterns about the model's performance across different classes. The model performed best on distinct classes like automobiles, ships, and trucks, but had more difficulty distinguishing between visually similar classes like cats and deer, or birds and airplanes.

The ten correctly predicted samples from the test set showed that the model was capable of correctly identifying objects across all classes. The model demonstrated robustness to variations in orientation, background, and style within each class.

Overall, this assignment provided valuable insights into the capabilities of different CNN architectures and the challenges of training a custom model for image classification. The pretrained models (ResNet-50, ResNet-101, and DenseNet-121) demonstrated impressive performance on a variety of images, while the custom model for CIFAR-10 showed reasonable performance given its relatively simple architecture and limited training time.

In the first part, I observed that the more complex models (ResNet-101 and DenseNet-121) sometimes produced more accurate classifications than ResNet-50, particularly for challenging images with multiple objects or unusual perspectives. This demonstrates the ability of more robust architectures to capture more nuanced features. However, these improvements were not consistent across all images, suggesting that the simpler ResNet-50 architecture is often sufficient for many classification tasks.

For the CIFAR-10 classification task, my CNN achieved satisfactory performance with just 3 epochs of training. The confusion matrix revealed that the model particularly struggled with distinguishing between animals (classes that are visually similar), which is a common challenge in image classification. The model performed best on classes with distinctive shapes and features, such as airplanes and ships.

The results highlight the effectiveness of CNNs for image classification tasks and demonstrate how pretrained models can be leveraged for new applications without extensive retraining. They also show the importance of proper data preprocessing, model architecture design, and evaluation metrics in developing effective image classification systems.