PROJECT REPORT

Title: Image Classification using Transfer Learning Model

Course Title: Machine Learning Lab

Course Code: CSE 432

Github Link: CashMahmood/Image-Classification-using-Transfer-Learning-Model-in-Machine-Learning

Submitted by

Name: Kayes Mahmood

ID: 2215151030

Batch: 51

Section: 7A2

Submitted to

Mrinmoy Biswas Akash

Lecturer & Course Coordinator, Department of CSE

University of Information Technology and Sciences

Md. Yousuf Ali

Lecturer, Department of CSE

University of Information Technology and Sciences

Submission Date: 07 July 2025

Introduction

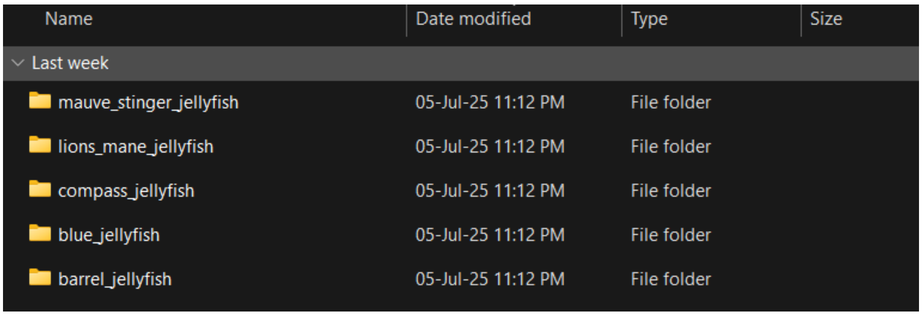
This project demonstrates a deep learning-based image classification approach to classify jellyfish species using convolutional neural networks (CNNs). The main objective is to develop a model that can accurately classify images of jellyfish into one of five predefined classes. To improve performance and training speed on a limited dataset, the project utilizes transfer learning with the VGG16 architecture pre-trained on the ImageNet dataset.

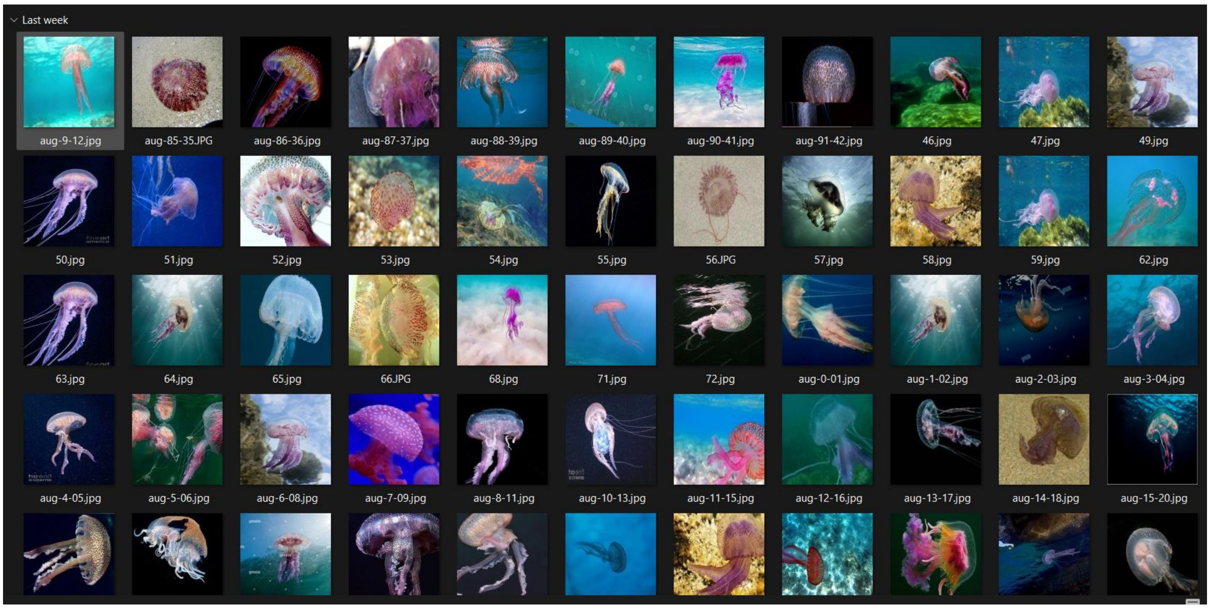
Dataset Overview

The dataset consists of 750 images divided into 5 classes:

- barrel_jellyfish
- blue_jellyfish
- compass_jellyfish
- lions_mane_jellyfish
- mauve_stinger_jellyfish

Each class contains 150 images. The dataset is stored in Google Drive and loaded using TensorFlow's image_dataset_from_directory() function. The images are resized to 224x224 pixels, and a batch size of 32 is used.





Preprocessing and Augmentation

To enhance generalization and avoid overfitting, data augmentation techniques were applied:

- Random Horizontal Flip
- Random Rotation (0.2 radians)
- Random Zoom
- Random Brightness
- Random Contrast

Additionally, the dataset is split into:

- 80% training (19 batches)
- 20% testing (5 batches)

Data is prefetched using TensorFlow's AUTOTUNE to optimize GPU utilization.

Transfer Learning with VGG16

A pre-trained VGG16 model is loaded without the top classification layers (include_top=False). Its layers are frozen to preserve learned features from the ImageNet dataset.

The model architecture used:

```
Sequential([
    base_model,
    Flatten(),
    Dense(256, activation='relu'),
    Dense(5, activation='softmax')
])
```

- Optimizer: Adam
- Loss Function: Sparse Categorical Crossentropy
- Metrics: Accuracy

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14,714,688
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 256)	6,422,784
dense_1 (Dense)	(None, 5)	1,285

```
Total params: 21,138,757 (80.64 MB)
Trainable params: 6,424,069 (24.51 MB)
Non-trainable params: 14,714,688 (56.13 MB)
```

Training Process

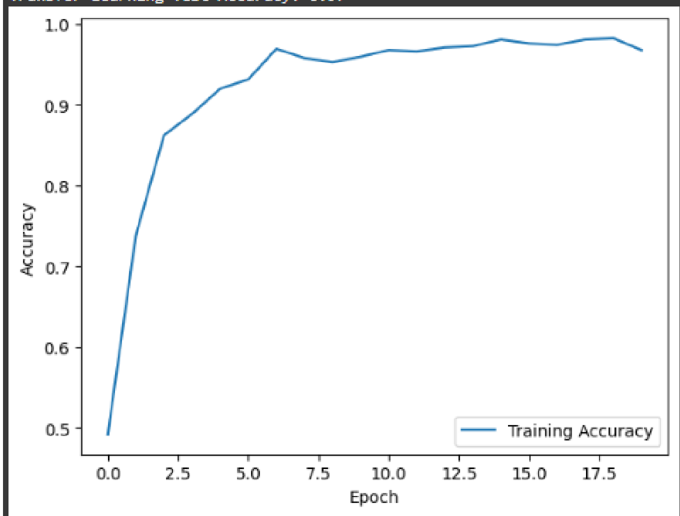
The model is trained for 20 epochs. The training accuracy improves steadily, starting from 37% and reaching 97%.

The performance on the test dataset was:

- Test Accuracy: 96.46%
- Test Loss: 0.0781

These results indicate that the transfer learning approach with VGG16 successfully learned to classify jellyfish images with high accuracy.

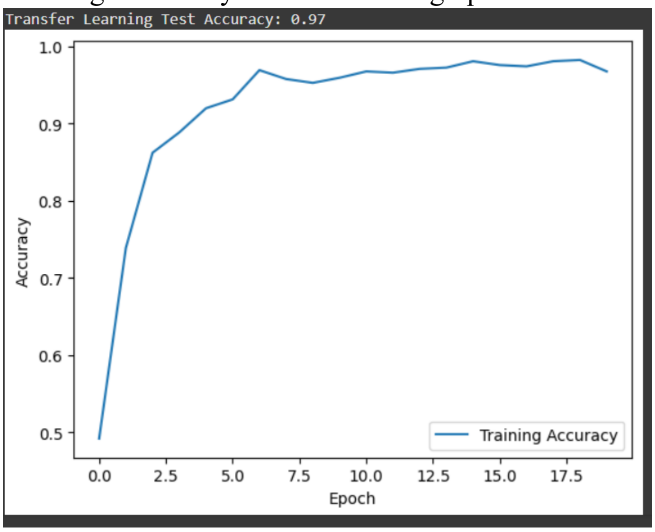
```
Epoch 1/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 111s 5s/step - accuracy: 0.3705 - loss: 58.3735
Epoch 2/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 56s 1s/step - accuracy: 0.7027 - loss: 1.3910
Epoch 3/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 28s 517ms/step - accuracy: 0.8473 - loss: 0.5112
Epoch 4/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 10s 484ms/step - accuracy: 0.8792 - loss: 0.3185
Epoch 5/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 8s 387ms/step - accuracy: 0.9323 - loss: 0.2257
Epoch 6/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 10s 349ms/step - accuracy: 0.9348 - loss: 0.1878
Epoch 7/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 8s 424ms/step - accuracy: 0.9746 - loss: 0.1013
Epoch 8/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 7s 340ms/step - accuracy: 0.9573 - loss: 0.1282
Epoch 9/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 10s 352ms/step - accuracy: 0.9508 - loss: 0.1426
Epoch 10/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 10s 336ms/step - accuracy: 0.9596 - loss: 0.1203
Epoch 11/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 8s 427ms/step - accuracy: 0.9629 - loss: 0.1471
Epoch 12/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 9s 330ms/step - accuracy: 0.9546 - loss: 0.1233
Epoch 13/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 10s 339ms/step - accuracy: 0.9770 - loss: 0.0601
Epoch 14/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 9s 431ms/step - accuracy: 0.9801 - loss: 0.0926
Epoch 15/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 9s 340ms/step - accuracy: 0.9791 - loss: 0.0866
Epoch 16/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 9s 398ms/step - accuracy: 0.9771 - loss: 0.0947
Epoch 17/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 8s 407ms/step - accuracy: 0.9798 - loss: 0.0625
Epoch 18/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 7s 336ms/step - accuracy: 0.9847 - loss: 0.0378
Epoch 19/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 8s 415ms/step - accuracy: 0.9839 - loss: 0.1028
Epoch 20/20
19/19 ━━━━━━━━━━━━━━━━━━━━ 9s 346ms/step - accuracy: 0.9698 - loss: 0.1190
5/5 ━━━━━━━━━━━━━━━━━━━━ 10s 2s/step - accuracy: 0.9646 - loss: 0.0781
Transfer Learning Test Accuracy: 0.97
```



Visualization

- Sample Augmented Images: 25 images from one batch were visualized with class labels.

- Training Accuracy Plot: A line graph shows accuracy improvement over 20 epochs.



Conclusion

The use of transfer learning with VGG16 proved to be highly effective for this classification task. Despite the relatively small dataset size, the model achieved excellent accuracy due to the robust pre-trained features of VGG16 and proper augmentation strategies.