2024

FISH CLASSIFICATION USING PSEUDO LABELLING

Name: Sameer Afzal

Matricola: 1000053143

Course: Deep Learning II (Data Science)

Professors: Concetto Spampinato & Giovanni Bellitto

Contents

1. Introduction 2
2. Model Description 2
   Pre-trained Networks 3
      ResNet18 and ResNet50 3
      VGG16 3
      DenseNet121 3
3. Dataset 3
   Data Source 3
   Data Size 4
   Preprocessing 4
4. Training Details and Procedure 4
   Evaluation Metrics 4
   Training Phase 4
   Pseudo Labeling 5
   Re-Training Phase 5
5. Experimental Results 5
   Training Curves 5
   Performance and Comparison 7
6. Conclusion 8
References 8

Sameer Afzal afzal1998sameer@gmail.com

1. Introduction

The task of classifying different species of fish using deep learning models presents several significant challenges. The primary objective of this challenge is to design and validate a deep learning model capable of accurately classifying fish species. The dataset provided consists of a small labeled training dataset of annotated images, a test dataset, and an auxiliary dataset of unannotated images. The small size of the labeled training dataset and the large volume of unlabeled data introduce unique difficulties and opportunities in model training and validation.

One of the main challenges is the limited size of the labeled training dataset. With only 150 annotated images, training a deep learning model from scratch is impractical due to the high risk of overfitting and poor generalization to unseen data. Additionally, the dataset contains 15 different species, each exhibiting significant inter-species variation and intra-species similarity, making the classification task even more complex.

To address these challenges, our proposed solution leverages a combination of transfer learning and semi-supervised learning techniques, specifically pseudo-labeling, to effectively utilize both the small labeled dataset and the large auxiliary dataset of unannotated images. By combining the strengths of pre-trained models with the augmentation benefits of semi-supervised learning, our approach enhances the model's performance and ensures it generalizes well to new data. The key innovations lie in the strategic integration of pre-trained networks, confidence-based pseudo-labeling, and iterative retraining, providing a robust solution for the fish species classification task.

2. Model Description

The proposed solution leverages multiple pre-trained convolutional neural networks (CNNs) to classify fish species, taking advantage of transfer learning to overcome the challenge of limited labeled data. The networks used include ResNet18, ResNet50, VGG16, and DenseNet121. Among these, the best results were obtained using DenseNet121. This section provides a comprehensive description of the architectures, activation functions, regularization techniques, optimization algorithms, and loss functions used.

Pre-trained Networks

ResNet18 and ResNet50

- **Architecture**: ResNet (Residual Networks) use skip connections (or shortcuts) to jump over some layers. ResNet18 has 18 layers, and ResNet50 has 50 layers.
- **Activation Functions**: ReLU for all convolutional layers.
- **Regularization**: Batch normalization is applied after each convolution layer to stabilize the learning process and improve convergence.

VGG16

- **Architecture**: VGG16 consists of 13 convolutional layers followed by 3 fully connected layers. The convolutional layers use small receptive fields (3x3) and max-pooling layers.
- **Activation Functions**: ReLU for all layers.
- **Regularization**: Dropout layers after the fully connected layers.

DenseNet121

- **Architecture**: DenseNet (Densely Connected Convolutional Networks) consists of dense blocks where each layer is connected to every other layer in a feed-forward fashion. DenseNet121 has 121 layers.
- **Activation Functions**: ReLU for all layers.
- **Regularization**: Batch normalization is applied after each convolution layer.

Among these architectures, DenseNet121 yielded the best performance. The DenseNet121 model is initialized with pre-trained weights from training on a large dataset (ImageNet). The final fully connected layer (classifier) is replaced to adapt the model to our specific classification task with 15 fish species. The number of input features to the classifier is set to match the number of output classes (15). Furthermore, Stochastic Gradient Descent (SGD) is used for training the model as optimizer and Cross-Entropy Loss is used, which is suitable for multi-class classification problems.

3. Dataset

Data Source

The data for this project is provided by the professor, consisting of annotated training images, unannotated auxiliary images, and test images.

Data Size

- **Training Data**: 150 annotated samples (15 samples per class).
- **Unannotated Data**: Approximately 38,000 unannotated images.
- **Test Data**: 148 test images.

Preprocessing

Preprocessing is a crucial step in preparing the data for training and testing deep learning models. It ensures that the input data is in a suitable format and enhances the model's ability to generalize from the training data. The preprocessing steps undertaken in this project involve data normalization and augmentation techniques, implemented through the torchvision.transforms module. The specific transformations applied to the training and test datasets are detailed below:

- **Training Transformations**: RandomResizedCrop, RandomHorizontalFlip, ToTensor, Normalize.
- **Test Transformations**: Resize, CenterCrop, ToTensor, Normalize.

4. Training Details and Procedure

Evaluation Metrics

The primary metrics used for evaluating model performance are accuracy and loss. These metrics provide insight into the model's ability to correctly classify the images and the effectiveness of the training process.

Training Phase

- **Epochs**: 100 for initial training.
- **Batch Size**: 32 for initial training.
- **Validation Strategy**: Validation is performed during training by calculating the accuracy at the end of each epoch. This allows for tracking the model's learning progress and identifying issues such as overfitting or underfitting.
- **Optimizer**: Stochastic Gradient Descent (SGD) with a learning rate of 0.001 and momentum of 0.9 is used.
- **Initialization**: The weights for the convolutional and other intermediate layers are initialized using pre-trained weights from models trained on the ImageNet dataset. The final fully connected layer, adapted to the target task with 15 classes, is newly initialized.

Pseudo Labeling

After initial training, the models were used to predict labels for the unlabeled dataset. This process involves generating pseudo-labels for the unannotated data based on the model's predictions.

Re-Training Phase

- **Epochs**: 500 for re-training with combined labeled and pseudo-labeled data.
- **Batch Size**: 64 for re-training.
- **Optimizer**: SGD with the same parameters as the initial training phase.

5. Experimental Results

Training Curves

The following plots illustrate the training phase accuracy and loss, as well as the re-training phase accuracy and loss.

Figure 1: Training Phase Accuracy

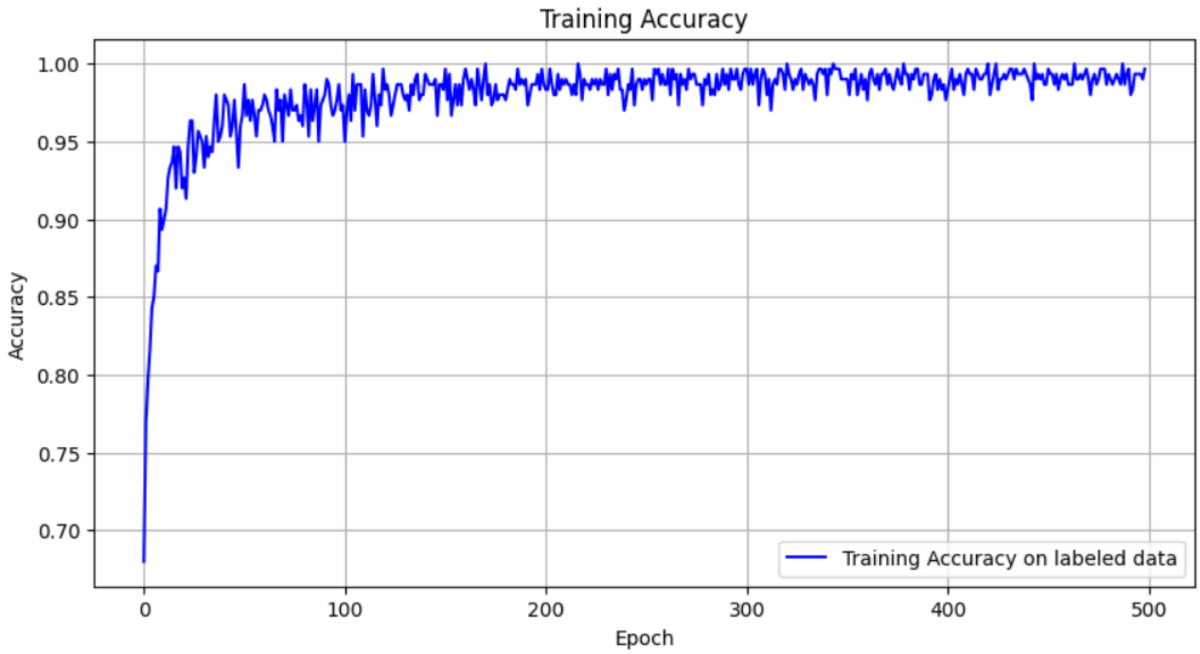

Figure 2: Re-Training Phase Accuracy

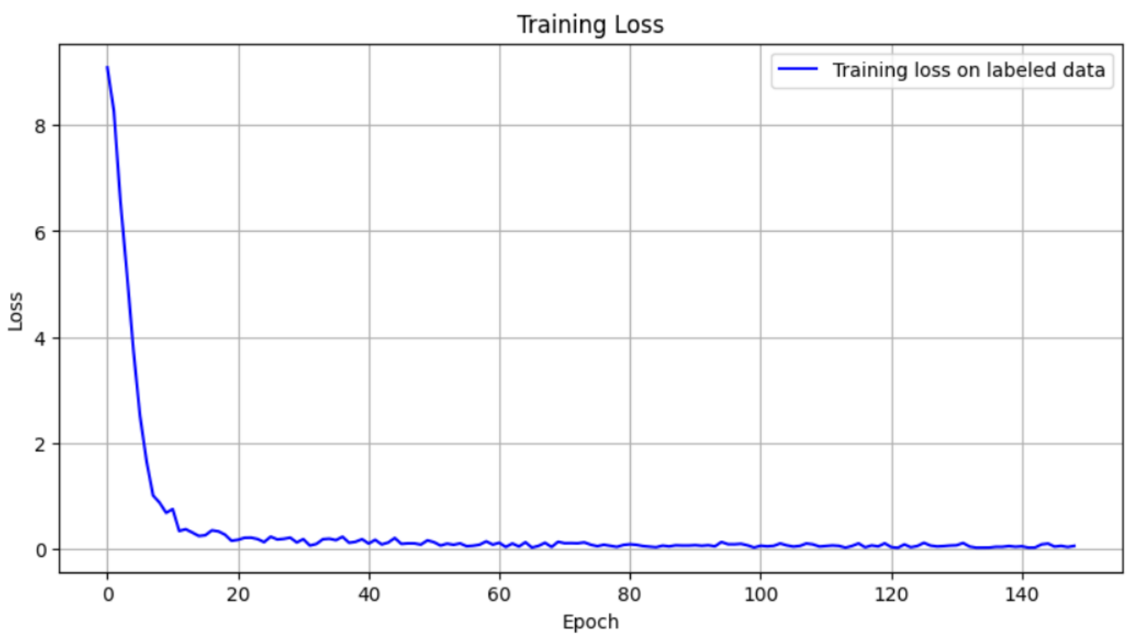

Figure 3: Training Loss

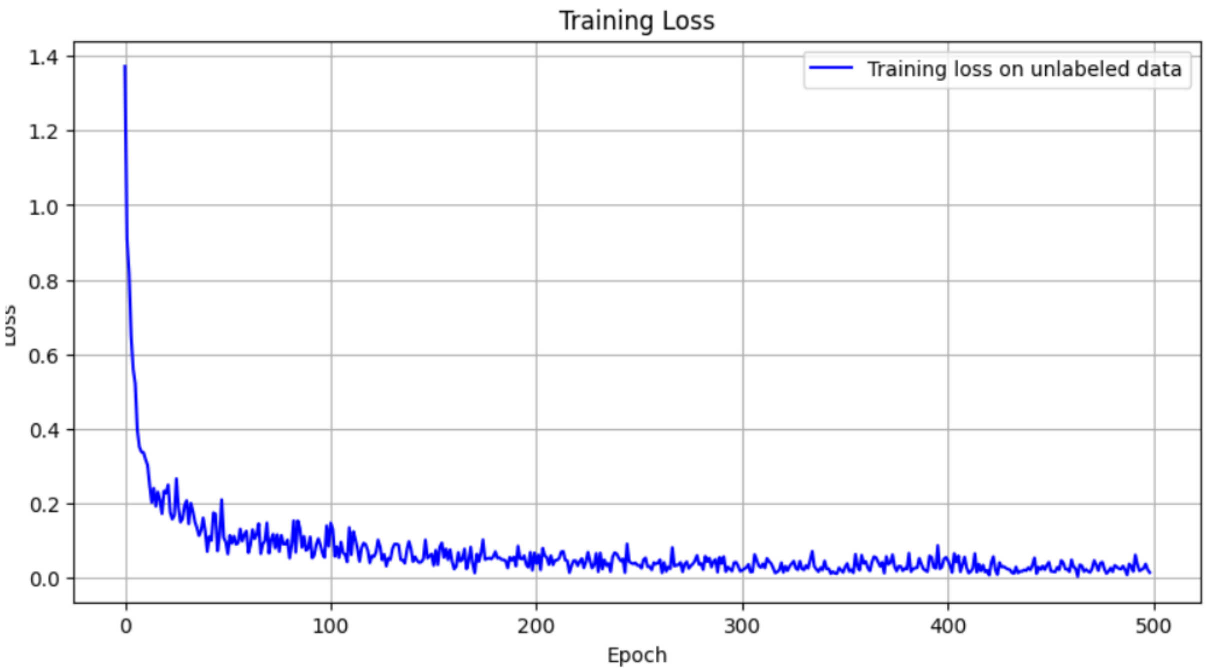

Figure 4: Re-Training Loss

Performance and Comparison

The final performance metrics on the test set are reported, demonstrating the effectiveness of the proposed solution. DenseNet121 achieved the highest accuracy, followed by VGG16 and ResNet18.

Submission No	Model	batch	lr	taining epoc	epochs for unlabel da	optimizer	unannotated sample si	ac	momentum
1	resnet50	32	0.001	100	0	SGD	0	0.9	0.9
7	resnet18	64	0.001	100	300	SGD	100	0.945	0.9
8	VGG16	64	0.0001	150	500	ADAM	100	0.945	
11	densenet121	64	0.001	150	500	SGD	150	0.97	0.9

Figure 5: Detailed comparison with multiple hyper-parameters

6. Conclusion

In this project, we aimed to design and validate a deep learning model for the classification of 15 fish species, leveraging both a small annotated training dataset and a large unannotated dataset. By employing multiple pre-trained models and experimenting with various combinations of hyperparameters, we were able to identify the most effective approach for this task. Among the tested models, DenseNet121 emerged as the most successful, achieving an impressive accuracy of 0.972 on the test data. The key hyperparameters contributing to this performance included a learning rate of 0.01, a batch size of 64, and training over 500 epochs. In comparison, VGG16 and ResNet18 also performed well, with accuracies of 0.945 and 0.932, respectively. These results highlight the robustness and effectiveness of transfer learning techniques, especially when dealing with small annotated datasets. My approach demonstrates that careful selection and fine-tuning of pre-trained models can lead to significant improvements in classification accuracy, even with limited labeled data. This study not only contributes a reliable solution for fish species classification but also underscores the potential of leveraging large unannotated datasets to enhance model performance through methods like pseudo-labeling.

References

1. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.
3. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
4. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition.