Classification of Fish Species

Calvin Spencer and **Rida Shahid**
{caspencer,rishahid}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

Employing image recognition software to classify fish species from images has the potential to aid in valuable data collection in real world environments. Convolutional neural networks (CNNs) are the best models to utilize with color image data. We trained a CNN on 427 images, taken at a seafood market, of 9 different fish species. Image data augmentation decreased the performance of our best model from 87.2% accuracy to 66.18%, likely due to the small capacity of our model designed for unaugmented data. Our results support accepted understandings of CNNs as effective image classifiers, and demonstrate that the differences between fish species can be captured through color imagery.

1 Introduction

Wild fish stocks are becoming increasingly fragile as commercial fisheries, climate change, and environmental degradation ruin their aquatic ecosystems. To support and restore these populations, researchers rely on data to inform their conservation strategies. However, collecting data such as population size and migration patterns is logistically challenging when studying fish; Even more so when each fish must be classified by species. While analog methods have been sufficient in some historical cases, interdisciplinary applications of machine learning algorithms could aid biologists in their data collection, facilitating more efficient and accurate research (Atlas et al. 2021).

Imagery is likely the most practical and abundant data type to train species-classifying machine learning algorithms. Convolutional neural networks (CNNs) are renowned for their compatibility with image processing, leading us to employ a TensorFlow CNN in our model. Previous research has shown that CNNs are capable of classifying tree species with high accuracy using color image data (Egli and Höpke 2020). We suspect that CNNs could provide similar accuracy when trained on images of fish.

Ulucan et al. (2020) published a dataset which can aid in the creation and development of fish species classification models. They generated the data by taking 427 color pictures of 9 different species of fish found in a seafood market. In their literature, the researchers propose that models could be trained to identify species of fish that may spoil faster, which could help prevent seafood-related illnesses (Ulucan, Karakaya, and Turkan 2020). However, we imagine this type of image classification could be applied to field settings. For instance, bycatch could be automatically photographed and run through this type of species classifier to count the number of each species.

This paper will describe our preprocessing techniques, CNN models performance using TensorFlow, and the use of data augmentation. Furthermore, we discuss the broader impacts of developing CNN classifiers and conclude with recommendations for future improvement.

2 Background

Convolutional Neural Networks

CNNs are a subset of neural networks that have at least one convolutional layer. They are known for their ability to outperform regular neural networks when given images as an input. Neural networks generally do not improve much when designed with more than 3 layers, while CNNs function better with a greater number of layers due to the 3-dimensional neural structure created by layering convolutional layers, pooling layers, and fully-connected layers. Intuitively, this fits image processing because structures within images have hierarchies which are better represented by the additional layers of the CNN (sta 2021).

Data Preparation

The Large-Scale Dataset for Fish Segmentation Classification originally consisted of a mix of JPG and PNG images and since there were more PNG images, we created a PNG copy of each JPG image. The fish images were saved in different folders corresponding to the different species; we created a dataframe consisting of the path to access each image (image name) and the fish specie the image belongs to (label).

We rescaled the images by dividing each pixel value by 255 and thus, limiting the range of our features to between 0 and 1. Next, we extracted each image using its path and augmented the data by flipping, shifting, rotating, and applying zoom and shear effects to each image. We set the size of the images to 250 by 250 and use a batch size of 32 for augmentation.

3 Experiments

The purpose of this work is to construct a classifier that can correctly identify fish species in images from a Large Scale Fish Dataset (Ulucan, Karakaya, and Turkan 2020). We explore several different CNN models and vary the number of layers and hyperparameters for each. Another intent behind this work is to compare the effects of data augmentation on image classification. In addition to this, we observe the differences between the predictions generated for each specie in the dataset. The model with the highest accuracy is determined to be the best one.

Dataset and Algorithms

A Large Scale Fish dataset contains 9 different seafood types collected from a supermarket in Turkey (Ulucan, Karakaya, and Turkan 2020). It consists of a total of 427 different images which are either in JPG or PNG format. The different classes (fish species) are slightly imbalanced as seen in Table 1. The data set is split into train, validation and test sets; the train set contains 272 images, validation set contains 69 images and the test set contains 86 images with varying numbers of images from each class.

Fish Species	# of Images
Black Sea Sprat	50
Gilt Head Bream	50
Horse Mackerel	49
Red Mullet	49
Red Sea Bream	49
Sea Bass	50
Shrimp	50
Striped Red Mullet	50
Trout	30

Table 1: Summary of Data Set.

We develop CNN models using unaugmented data with differing numbers of layers and hyperparameters. We then determine the best model and train it on the original augmented data to observe the effects of data augmentation. We choose accuracy as the metric for determining the best model and its corresponding set of hyperparameters. All of our work is conducted using Scikit-Learn, Keras and TensorFlow (Pedregosa et al. 2011), (Abadi et al. 2015).

Models

Baseline The baseline model always predicts the most frequent label in the data set. Since several classes have the majority image number of 50, the baseline predicts any one of them.

CNNs We use a sequential model to design CNN models because our model should have a single input and output (Abadi et al. 2015). We vary the number of convolutional layers between 2 and 3 and their corresponding number of filters between 16, 32, 64, 128, and 256. We use ReLU as the activation function because of its proven success and wide usage (Ramachandran, Zoph, and Le 2017). After every 2-D convolutional layer, we add a max pooling layer in order to control overfitting by reducing the amount of parameters and computation in the network (sta 2021). Next, we flatten the model so it can be passed to fully connected layers and add dense layers to add the fully connected layers to the neural network. The activation function for the last dense layer is set to softmax because it is most suitable for the last layer of a classification network (Abadi et al. 2015).

We compile the model with Adam optimization to implement gradient descent, and categorical crossentropy loss as we have more than two label classes in our dataset (Abadi et al. 2015). We vary the learning rate between 0.001, 0.0005, and 0.00001 and fit the model with a batch size of 32 or 64 and run it for 50 epochs. The model is validated after each epoch and the best model is evaluated on the test set.

4 Results

Table 2 summarizes the results of our baseline model and the best CNN models generated using the methods described previously.

Model	Accuracy (%)
Baseline	13.97
CNN (original data)	87.20
CNN (augmented data)	66.18

Table 2: Summary of Experimental Results.

These results indicate that both of our models are significantly more accurate than the baseline model. There is a difference of > 50% between the accuracies of the trained models and baseline model. This is significant enough to conclude that both of these models outperform the baseline model.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 248, 248, 16)      448
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 124, 124, 16)      0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 122, 122, 32)      4640
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 61, 61, 32)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 59, 59, 64)        18496
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 29, 29, 64)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 53824)             0
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              55116800
_________________________________________________________________
dense_5 (Dense)              (None, 9)                 9225
=================================================================
Total params: 55,149,609
Trainable params: 55,149,609
Non-trainable params: 0
_________________________________________________________________
```

Figure 1: A summary of the architecture of our final CNN model.

The best CNN model is implemented with 3 2-D convolutional layers with 16, 32, and 64 filters respectively where each layer is followed by a max pooling layer. The overall model architecture can be seen in Figure 1. This model was trained with a learning rate of 0.0001 and a batch size of 32. While comparing the accuracies with augmented data and unaugmented data, we observe that the unaugmented data actually performs better. This disagrees with previous research as data augmentation is known to improve the performance of models and expand limited datasets to take advantage of the capabilities of big data (Shorten and Khoshgoftaar 2019). We speculate that this may be because we trained our model on unaugmented data first and the model did not have the capacity to learn complex augmented images well. It also might be because the unaugmented data matches the test data more closely and this particular data set does not require a larger and more complex training set.

Label Name	Precision	Recall	F1-Score	Support
Black Sea Sprat	1.00	0.90	0.95	10
Gilt Head Bream	0.82	0.90	0.86	10
Horse Mackerel	0.88	0.70	0.78	10
Red Mullet	0.90	0.90	0.90	10
Red Sea Bream	1.00	1.00	1.00	10
Sea Bass	0.71	1.00	0.83	10
Shrimp	0.91	1.00	0.95	10
Striped Red Mullet	0.86	0.60	0.71	10
Trout	0.83	0.83	0.83	6

Table 3: Summary of Final Results for Each Label.

Table 3 summarizes the results for each label generated using our best model. Support is the number of actual occurrences of the class in the specified dataset (Pedregosa et al. 2011). The key findings of these results are:

1. Red Sea Bream has the highest precision, recall and F-1 score which means that it has no false positives, false negatives and best overall performance.
2. Striped Red Mullet has the lowest recall, one of the lowest precisions and the lowest F-1 score which means that it has the worst overall performance.

We describe the performance of our classifier using the confusion matrix shown in Figure 2. Based on the colorbar, darker colors correspond to larger numbers and in the matrix, the darkest colors lie on the diagonal between true and predicted labels. This shows that almost all of the predicted labels match their true labels which indicates that most of the fish species are correctly predicted by the classifier.

A comparison between the labels shows that "Trout" and "Striped Red Mullet" have the lowest number of true positives. We predict that this may be because Trout had the lowest images in the dataset so in reality, the lowest number of true positives is for Striped Red Mullet; the number of true positives for other labels is similar to one another's. We observe the following trends in mispredictions:

1. Horse Mackerel is most frequently mispredicted as Shrimp.
2. Red Sea Bream is never mispredicted.
3. Striped Red Mullet has the most mispredictions.

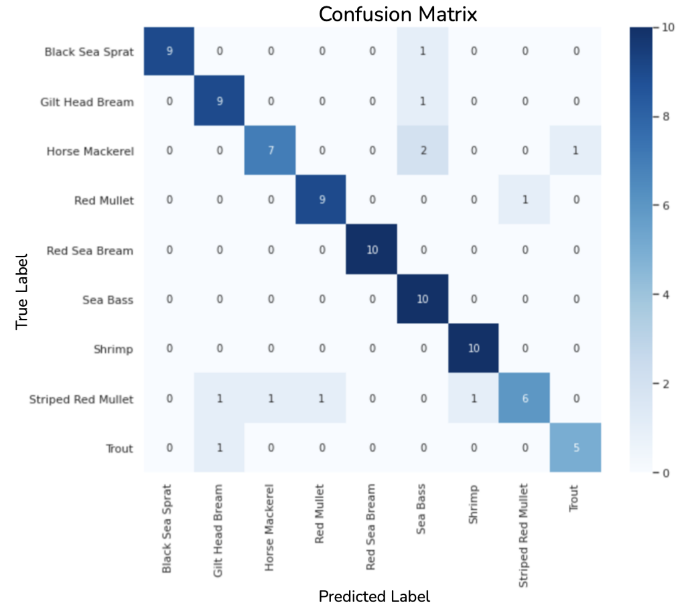

Figure 2: A confusion matrix of the true labels versus the predicted labels.

Based on these results, we conclude that species that look similar to one another are frequently mispredicted and those with unique characteristics are never mispredicted. Moreover, the model performs poorly for species with less samples in the dataset which shows that more images does improve the performance of the model. Overall, our model was able to obtain a reasonably high accuracy despite the small dataset. We speculate that in future, working with a larger dataset may yield even better results. We also intend on training the model on the augmented data first and then training it on the unaugmented data, not the other way around.

5 Broader Impacts

The development of a highly accurate fish species image classifier would greatly benefit researchers who work with laborious data collection methods in the field. By automating the species classification process, datasets could be generated faster, and potentially from multiple locations at once. This application of machine learning models for image species classification is just one example of how computational power can supplement our ability to research natural sciences (Fernandes et al. 2015). Adoption of these models as reliable techniques may help transition what is considered reliable methodology in the field to include more machine learning.

The application mentioned by the authors of our dataset is potentially concerning if inaccurate classifications were made. If we were to use our species classifier to prevent consumption of spoiled seafood as Ulucan et al. mentioned, the consequences of a misclassification could directly harm an individual. In the research settings we hypothesized this model could be useful for, the influence of model misclassifications / bias is limited to the scope of the research itself.

However, many concerns regarding the concept of image-trained machine learning models are valid. Literature regarding the application of CNNs in medicine suggest that governance structures oversee how the technology is used in order to mitigate issues such as biases, inaccuracies, liability, and privacy (Reddy et al. 2019). Yet, we posit that most of these concerns do not apply to our model because the training data and intended use is only images of fish, not human medical data.

6 Conclusions

Our model trained on Ulucan et al.'s dataset was able to accurately predict the species of a fish image with an accuracy of 87.20%. Data augmentation did not provide the benefits we needed to reduce overfitting in a model trained on a minimal dataset. While our model is likely limited in its ability to classify fish images taken in different settings, it serves as a valid proof of concept that fish can be classified by species using CNNs, and of the value that data augmentation provides. We recommend surveying those researchers who would employ a model such as this to learn more specifically how automating image classification could be used in their methodology. Images of fish collected in a living, underwater environment would present additional challenges that we did not address, but may be more practical to researchers studying aquatic ecology.

7 Contributions

R.S. and C.S. researched data sources and found an appropriate set of images. R.S. preprocessed the data and converted all JPGs to PNG files. C.S. developed CNN models without augmented data. R.S. developed augmented CNN models. R.S. and C.S. experimented with trained models and analyzed performance. C.S. wrote the abstract, introduction, CNN background, broader impacts, and conclusions. R.S. wrote the preprocessing background, experiments, and results. Both authors documented code and proofread the report.

8 Acknowledgements

We would like to thank Dr. Raghu Ramanujan for guiding us to relevant CNN resources and providing us with relevant feedback throughout this work.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; and Zheng, X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Atlas, W. I.; Holt, C. A.; Selbie, D. T.; Connors, B. M.; Cox-Rogers, S.; Carr-Harris, C.; Hertz, E.; and Moore, J. W. 2021. Linking habitat and population dynamics to inform conservation benchmarks for data-limited salmon stocks.

Egli, S., and Höpke, M. 2020. CNN-based tree species classification using high resolution RGB image data from automated UAV observations. *Remote Sensing* 12(23):3892.

Fernandes, J. A.; Irigoien, X.; Lozano, J. A.; Inza, I.; Goikoetxea, N.; and Pérez, A. 2015. Evaluating machine-learning techniques for recruitment forecasting of seven north east atlantic fish species. *Ecological Informatics* 25:35–42.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Ramachandran, P.; Zoph, B.; and Le, Q. V. 2017. Searching for activation functions. *CoRR* abs/1710.05941.

Reddy, S.; Allan, S.; Coghlan, S.; and Cooper, P. 2019. A governance model for the application of AI in health care. *Journal of the American Medical Informatics Association* 27(3):491–497.

Shorten, C., and Khoshgoftaar, T. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6.

2021. Convolutional neural networks for visual recognition.

Ulucan, O.; Karakaya, D.; and Turkan, M. 2020. A large-scale dataset for fish segmentation and classification. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 1–5.