CLASSIFYING FISH BY SPECIES USING CONVOLUTIONAL NEURAL NETWORKS

Abdullah Albattal*, Anjali Narayanan†

*Electrical and Computer Engineering, University of California San Diego, La Jolla, CA,
†Scripps Institution of Oceanography, University of California San Diego, La Jolla, CA

ABSTRACT

There is an increasing need for automated fish classification to help properly identify fish species and characteristics in a standardized, non-invasive, and cost-effective manner. Machine learning is a promising method to do this. In this paper, we present the results of a convolutional neural network (CNN) used to identify fish species across datasets. Our proposed model improves on a previously-built model by Rathi et al. (2018). The performance of our improved model is demonstrated with real-world data from a research organization called The Nature Conservancy.

Index Terms—convolutional neural network, fish identification, fish species classification, the Nature Conservancy, Fish4Knowledge

1. INTRODUCTION

Identifying fish by their characteristics is crucial to many industries and personnel. Fish populations are being increasingly impacted by environmental pressures such as global warming, marine responses to climate change and pollution, and societal pressures such as unregulated and overfishing and sustainable exploitation of marine natural resources [1–3]. The consequences of these provide further motivation to develop a standardized, cost-effective, and reliable method to monitor fish across habitats [4].

Manual methods to identify fish can be problematic: they can be time-consuming; they may require large sampling efforts; sampling can be destructive to the marine environment; they may be expensive but result in limited data; and lack of fish specialists may cause incorrect and subjective identification [3–8]. Automated systems can help accurately classify these fish consistently. There has been a growing interest in exploiting electronic monitoring, electronic reporting, and artificial intelligence for fish identification purposes and to improve current methods [3]. Using video and images of fish (either underwater or aboard ships) are becoming common. These methods are effective, portable, non-invasive, and non-destructive and can provide high quality, high-resolution images at an affordable cost [1, 4, 9]. Machine learning techniques provide a means to automate image processing and can be tailored to conduct efficient fish species identification and segmentation [2, 6, 8].

Current automation approaches include many learning protocols and features such as color, shape and contour, specific landmarks, and texture [1, 8]. Specific example of learning protocols that have been used include principal component analysis, (multiclass) support vector machine, artificial neural networks, and convolutional neural networks (CNNs) [4, 6]. It has been observed that deep learning methods tend to achieve the highest performance, and CNNs in particular can be quite successful [1, 5, 7]. CNNs require the availability of large datasets and the accuracy of such models are contingent on the extent and quality of training data. It is possible to mitigate these issues by employing transfer learning and/or augmentation [2, 5, 7] In this study, we employ the methodology of Rathi et al. (2018) [9] to build a CNN to classify fish images, initially collected from underwater video, and then from a dataset provided by the Nature Conservancy. The goal of this project is to be able to build a CNN that can be applied to datasets collected by research organizations such as the Nature Conservancy and help identify fishes for research and fisheries purposes.

2. NETWORK ARCHITECTURE

Figure 1 displays the basic architecture of a CNN, which is the basis on which our models are built. Input images are normalized and are of the same size. Convolutional layers in the models presented result in either 32 or 64 feature maps that represent relevant features in the image detected by the convolutional layer.

Fig. 1: Example architecture of the CNN, adapted from [9]
Following the convolutional layer is a maxpooling layer, which can reduce spatial size of the representation and the number of parameters and computation, which can help control overfitting [10]. The convolutional layers and maxpooling layers have a kernel size of 5 x 5 pixels. The kernel matrices help extract local features. There are three activation functions that are tested in this study, ReLU (rectified linear unit), tanh, and sigmoid, applied on the convolutional layers. The ReLU defined in (1) performance was superior to the others and our paper will focus on the performance of models using it as the activation function:

\[ h = \max(0, a) ; a = W \cdot x + b \]  

ReLU

Then, the matrix is flattened into a vector and is input for a fully connected network with dropout layers. These dropouts randomly disable a chosen number of neurons in each layer, which can improve performance and reduce overfitting of the model. The models in this study were built using Keras API on Tensor Flow, which conducts backpropagation automatically [10–12]. Models were compiled with sparse categorical cross entropy loss and the adam optimizer, as recommended in [9]. For the implementation we tested with several combinations of EPOCHs, learning rates (adaptive and fixed), and batch sizes. For the results provided in this paper, the number of EPOCHs we used was 100 EPOCHs, learning rate of 0.001, and a batch size of 20 or 50.

3. DATASET

The initial dataset was provided by Fish4Knowledge project [13], which includes snapshots of the fish in their underwater habitat and binary masks to isolate the fish from these backgrounds. Rathie et al. (2018) utilize this dataset to train a CNN to classify the fish in the dataset by species [9]. The distribution of images across classes is seen in Figure 2. To deal with the highly unbalanced dataset, image augmentation is implemented in the classes with less than 200 images (blue in Figure 2) as part of the pre-processing step. The data were then divided, where 80% of the complete dataset was randomly chosen as the training and validation datasets, whereas the remaining 20% was used as the testing dataset. Of the 80% allotted for training and validation, 80% was randomly selected and used as the training dataset, and the remaining 20% was used as the validation dataset. The same approach was taken to append images to the dataset provided to us by the Nature Conservancy, which exhibited a similar distribution as in Figure 2.

3.1. Pre-processing

Before feeding the images within the dataset to the CNNs, the images were pre-processed by scaling the intensity values to be between zero and one. The images have also been resized to a uniform size of 100 by 100 pixels. Both of the datasets that we worked with are highly unbalanced as can be seen from the distribution in Figure 2 of the number of images per class for the Fish4Knowledge dataset. Hence, we performed data augmentation on the datasets where copies of images in classes with few examples were added to the dataset after translation, rotation, flipping (across the x-axis and/or y-axis), and/or brightness adjustments [14]. Once this was done, the masks were appended to the RGB images (forming the final 100 x 100 x 4 input) and the images and labels were stored in separate binary (.npy) files.

3.2. Training on original vs. complete datasets

The CNNs were tested multiple times with both the original (without the addition of augmented images) and complete (with the addition of augmented images) datasets. For brevity, we discuss in detail the results produced by the model trained, validated, and tested on the complete dataset; overall, the original dataset resulted in less accurate or skew among classes. For example, using the original dataset would result in images belonging in the first five species (which had a higher number of representative images in their class) being very well classified and a high test accuracy. However, this did not take into account the lack of representation of the other species classes and therefore did not provide model metrics that could be translated to more generalized models. To alleviate this, we appended images to the initial dataset. This ensured that images from all classes were represented in all training, validation, and testing sets. After analyzing the model performance on both the original and complete dataset, it was determined that the complete dataset was more dependable in justifying the model metrics, and thus will be discussed in the rest of the paper.

4. RESULTS

4.1. Initial Fish4Knowledge dataset

There are a total of 23 species represented in this dataset and a total of 27,370 images across all classes. After augmenta-
tion, the total number of images across all classes was 54,001. The first model is built similar to Figure 1. This is referred to as the "Rathi" model as it is a reproduction of the model presented in Rathi et al. (2018) [9]. The first layer is an input convolutional layer of 32 feature maps, then a maxpooling layer both with kernel sizes of 5 x 5, followed by two more sets of convolutional and maxpooling layers with 64, then 32 feature maps. The rest of the architecture is a fully connected network with a dense layer of 200 perceptrons, a dropout of 20%, and another dense layer of 100 perceptrons, a dropout of 20%, and finally a fully connected layer with a softmax function providing the probability of the image belonging to each of the 23 classes.

While the model seems to have a high accuracy when evaluated on the test dataset, the loss plot (Figure 3) shows evidence of overfitting as seen with the increasing values of the validation loss. This suggests the model may not generalize well, however, our results show a relatively high accuracy and well-classified test images.

The model achieves a test accuracy of 94.31%. The confusion matrix (Figure 4) shows that all classes are well classified with at least 81% probability. The average precision for all classes is 0.965 and the average recall is 0.927.

To improve the model, we implemented several different changes in an attempt to increase the number of trainable parameters, make the loss less volatile, and reduce overfitting. This included changing the kernel size from 5 x 5 to 3 x 3, increasing the number of dropout layers and the level of dropout, and changing the number of convolutional layers to improve the number of trainable parameters.

The final architecture that provided the best model metrics involves an input convolutional layer with 64 feature maps, a dropout layer of 40%, a maxpooling layer, followed by another convolutional layer with 32 feature maps, then the same dropout and maxpooling layers. These layers implement the same 5 x 5 kernel size. After this, the model continues into a fully-connected network, same as before, except with 40% dropout.

Our model exhibits greatly improved results. There was no more overfitting and the training and validation losses were consistently decreasing (Figure 5). When evaluated on the test data, the model achieves an accuracy of 98.6% - an improvement from the Rathi model. The confusion matrix (Figure 6) displays that all classes are much better classified with only two classes being classified with a probability less than 95%. The average precision for all classes is 0.989 and the average recall is 0.983. The increase of both the average precision and recall indicates better retrieval of true values.

4.2. Dataset provided by the Nature Conservancy

The data provided to us by the Nature Conservancy consists of 25 classes (species) of fish with 1306 images across all classes. This dataset does not have a mask as a 4th channel as is the case with the Fish4Knowledge dataset. To create the mask, which would isolate the fish from the background and other objects in the images, we tried several methods. Non-machine neural network methods such as active contour methods [15], graph-based methods [16], and watershed [17] (together with morphological methods). However, none of these methods were successful in segmenting/isolating the fish. That is why we resolved to using a convolutional neural network based method, which is the Mask-RCNN with a Res-
5. DISCUSSION

Overall, our model improved on the Rathi model. When applied to testing data from the Fish4Knowledge dataset, our model achieved an accuracy that was more than 3% higher than that of the original model described in [9]. Driven by the improved performance of our model we tested it on the data provided by the Nature Conservancy. Our model performed well on the testing data. With an accuracy of 94.25%, the results are promising that this model could be used in the future for fish species identification and classification. Figure 7 shows a visual example of this.

Our efforts show that by adding augmented images to the dataset, the complete dataset becomes relatively well-distributed and results in improved accuracy even when tested on unseen data, and more reliable metrics. Our work shows that our model can be applied to real-life datasets that are not necessarily noiseless, well-distributed, or robust such as the dataset provided by the Nature Conservancy. As well, we have shown that a convolutional neural network is a useful method to classify fish images by species accurately. These networks can be modified and scaled with relative ease. This makes it an ideal method for the entirety of the fishing and fisheries community as it proves to be accessible, cost-effective, and reliable means to fish classification.

6. CONCLUSION

Our model achieved the goal of improving on the Rathi model and the secondary goal of classifying fish images by species with a real-world dataset provided to us by researchers. We hope to further implement deep learning techniques that can help elucidate more information about sampled fish using image data. In addition to species classification, fish morphology and external characteristics, such as fin type, length, and color, may be resolved from the images in the datasets we used in this project. This is what we envision the next step for this project would be through the use of instance segmentation and object detection neural networks. Another path for future work would be to use a larger, and more diverse dataset to create a standardized pre-processing and augmentation approaches that will further improve trained models and use K-fold cross validation to glean more information on model metrics. We are optimistic about the transferability of this model to other real-world datasets and the potential of using it for high level classification of fish.

### Table 1: Accuracy per class of our model on the Nature Conservancy dataset

<table>
<thead>
<tr>
<th>Class Number</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.99</td>
</tr>
<tr>
<td>2</td>
<td>0.85</td>
</tr>
<tr>
<td>3</td>
<td>0.95</td>
</tr>
<tr>
<td>4</td>
<td>0.93</td>
</tr>
<tr>
<td>5</td>
<td>0.93</td>
</tr>
<tr>
<td>6</td>
<td>0.85</td>
</tr>
<tr>
<td>7</td>
<td>0.94</td>
</tr>
<tr>
<td>8</td>
<td>0.91</td>
</tr>
<tr>
<td>9</td>
<td>0.95</td>
</tr>
<tr>
<td>10</td>
<td>0.7</td>
</tr>
<tr>
<td>11</td>
<td>0.94</td>
</tr>
<tr>
<td>12</td>
<td>0.99</td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.95</td>
</tr>
<tr>
<td>15</td>
<td>0.95</td>
</tr>
<tr>
<td>16</td>
<td>0.98</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>0.96</td>
</tr>
<tr>
<td>19</td>
<td>0.93</td>
</tr>
<tr>
<td>20</td>
<td>0.98</td>
</tr>
<tr>
<td>21</td>
<td>0.82</td>
</tr>
<tr>
<td>22</td>
<td>0.95</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>0.98</td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>
References


A. TEAM CONTRIBUTION

A.1. Abdullah Albattal

- Helped identify literature for background on motivation and model building
- Created masks for dataset provided by the Nature Conservancy and conducted image augmentation
- Helped create and test CNNs (reproducing and improving on the Rathi model)
- Helped create documentation (including support on LaTeX) and PowerPoint presentation
- Put all the code together to upload to GitHub and wrote README file
- Facilitated group meetings and provided insight into new findings, updates, and information

A.2. Anjali Narayanan

- Helped identify literature for background on motivation and model building
- Helped create and test CNNs (reproducing and improving on the Rathi model)
- Added to code to include model metrics and prepared visuals for paper and presentation (e.g. CNN architecture)
- Helped create documentation and PowerPoint presentation
- Started report on LaTeX and troubleshooted formatting issues
- Facilitated group meetings and provided insight into new findings, updates, and information
B. RESPONSE TO GROUP CRITIQUES

B.1. Response to Group 7’s Critique

Comments: Group 11 used CNN and deep learning for fish species classification using RGB images and binary masks. They attempted to balance out their heavily skewed dataset and then compared the performance of the model on the original dataset and their augmented dataset. (Thank you for a good summary of our work.)

Suggestions offered by group 7 to us with responses:

• The augmented dataset still looks heavily skewed for the “Dascyllus reticulatus” class (class 1). Doesn’t this affect your results? and How did you choose which classes to augment? Why did you choose those classes? Why not augment all of the classes?

Our Response: Our aim with the addition of augmented images was mainly to account for the fact that there were a number of classes that had significantly less images (i.e. an order of magnitude lower) than the first five classes. This caused our testing set to be largely made up of only images from the first five classes, however, by adding the augmented data, we were able to get more classes in our testing which allowed us to evaluate our model on a more generalized testing dataset (see Figure 2). We did not want to augment all classes, as we already had more than enough images for the first five classes and wanted to bring the others to the same level. Yes, there was a disproportionate amount still in class 1, however, our model improved with the addition of the augmented data and all of the classes were represented properly in the training, validation, and testing sets we had (see section 3.2).

• Could overfitting of the original model also have been resolved with early stopping during training?

Our Response: We decided not to do early stopping because as seen in our loss plots, the validation loss was typically higher than the training loss, and due to this we decided to implement other methods to try and reduce overfitting that would have a better response in this situation and allow us to achieve higher accuracies (see section 4.1).

B.2. Response to Group 28’s Critique

Comments: This presentation did a good job on explaining the background, previous work, dataset etc. with enough figures so that I can easily understand what they are talking about. From my understanding, this project mainly focuses on fish classification based a previous paper. The dataset comes from University of Edinburgh, school of informatics. The improvement of their work are mainly three parts: (1) Use data augmentation to reduce the effect of highly unbalanced dataset. (2) By cutting one convolutional layer and adding several dropout layers, they prevented the overfitting problem. (3) By adding more parameters in each layer, they increased the final validation accuracy. (Glad it was understandable and thank you!)

Suggestions offered by group 28 to us and our responses:

• Maybe regular deep learning model is not the best model to fit this practical problem. Did they try other models like AE or transfer learning from other pre-trained models like X-ception / InceptionV3?

Our Response: The purpose of this paper was to recreate a specific CNN from [9], and so we focused our efforts on using that model architecture. Moreover, we were motivated by the fact that deep learning and CNNs have been successful in the context of fish classification (see section 1 for more information on this)

• How many combinations of hyperparameters did they try? It will be much better if we can see the comparisons on different hyperparameters.

Our Response: We had tried a number of different hyper-parameter options, and combinations, as well as a number of changes to the CNN model itself. We only report on the most accurate/best model for our purposes as going through all of the models that we tested would take too much time (see section 2)

• This dataset seems have too many similar images (same object captured in different times). Did they consider use other datasets instead?

Our Response: At the point of critique, we had not tried our model on the dataset provided by the Nature Conservancy. We have since added that to our report and thus have used another dataset (see section 4.2).

B.3. Response to Group 39’s Critique

Comments: This presentation shows a lot of details about the data set, data augmentation, building and improving model. By using data augmentation and convolutional neural network, the group solve the fish classification problem very well. (Thank you!)

Suggestions offered by group 39 to us and our responses:

• Why do you choose Adam as optimizer, did you try other optimizers like SGD?

Our Response: We used Adam because that was the optimizer used in the model we were trying to recreate (see section 2). We also observed that Adam performs better than SGD in our model.

• Can you explain why do you remove the first convolutional layer when improving the model?
Our Response: Our intuition in improving the model was to increase the number of features that are fed to the fully connected layers. The convolutional layers are to an extent feature extraction/engineering layers and the fully connected layers are more of a classification engine within the model. In the original model there are not enough features from the CNNs fed to the fully connected layers (only 32 features) and we thought it would improve the model to allow for more feature from the CNNs to be fed to the fully connected layers as a way to improve the model. After testing our intuition, it appears to have worked. In numerical details, it is as follows: We remove the first convolutional layer to increase the number of trainable parameters (responsible for feature extraction) for the model. The group is referring to a convolutional layer with 32 feature maps that was removed from the model to improve it. The removal of this layer resulted in only two convolutional layers, the first with 64 feature maps instead of 32, which ultimately increased the number of features fed to the fully connected layer to 512 instead of 32 (see paragraph 3 in section 4.1, after Figure 3).

- It might be helpful to mention how many numbers and classes you used on training, validation and testing on pptx.

Our Response: We took this advice and did indicate the total number of images in each dataset used. We also mention in the paper how this was split into training and validation (80% of total dataset) and testing (20% of total dataset) (see section 3, and details in section 4).