University of Osnabrueck

Implementing ANNs with TensorFlow

Turtle Recognition Using Deep Convolutional Neural Networks

Dennis Hesenkamp, Lennart Zastrow, Madhuri Ramesh

supervised by
Charlotte Lange

April 8, 2022

Contents

Abstract

Here is room for an abstract.

1 Background

Three-fifths of all turtle species worldwide are on the verge of extinction or are already severely threatened. According to a U.S. study published in the journal BioScience, this makes turtles the most endangered vertebrates in the world, ahead of mammals, birds, fish, and amphibians.

Sea turtles are known as indicator species which means that their presence and abundance reflect the health of the wider ecosystem. Therefore, increasing our ability to identify and understand them can enhance our ecological understanding.

Not only for their own sake, but they also contribute significantly to a healthy ecosystem. The herbivores, carnivores, or omnivores are at the same time hunters, pest controllers, and food sources for other animals. The scavenging species, for example, ensure a clean environment, and the herbivorous turtles make an important contribution to spreading plant seeds.

For many sea turtles, sea grass is the main food source. Sea grass grows in thick beds on shallow seabeds. Constant feeding by sea turtles on this grass keeps the beds in order and prevents them from becoming long and unhealthy. Because these sea grass beds are prime locations for small fish to breed and spawn, healthy sea grass beds are critical to populations of small fish living in the oceans. Without this contribution from sea turtles, the ocean ecosystem would be out of balance.

While sea turtles spend most of their lives in the ocean, they come to the beach to lay their eggs. This important part of a turtle's life also has an important impact on a beach's ecosystem. Without plants like beach grasses, the beach would succumb to erosion. These plants are fertilized by eggs that do not hatch and by turtle droppings on the beach. This nutrition is essential for the survival of the beach ecosystem.

Humans are making it difficult for the turtles to survive. The animals are suffering from climate change, habitat destruction, excessive trade in live animals, the sale of their meat and shells, and environmental pollution. For example, Australian scientists estimated that one in five sea turtles now dies from eating too much plastic. With 14 swallowed plastic particles, the risk of death increases to 50 percent. At 200 pieces swallowed, the sea turtle is no longer viable. The plastic pieces get stuck or cause internal injuries.

The greatest threat to turtles, however, is and remains humans. In some regions of the world, turtles are considered a delicacy; worldwide, they are kept as exotic pets. The meat of larger species ends up on markets in Asia, Africa, and Latin America, for example, while smaller species are traded internationally primarily as pets. The turtle shell ends up in the powdered form in pills and pastes used in traditional Asian medicine (including TCM). Some species, which are said to have a special medicinal effect, such as the three-striped hinge back turtle (Cuora trifasciata) achieve prices of several thousand US dollars per animal. Even in Europe, turtles can still be ingredients in TCM formulations.

2 Grant Species Preservation Through Population Census

The ability to distinguish between individuals of the same species is a fundamental tool for modern animal welfare. To ensure the protection of individuals, it is crucial to identify their whereabouts and movement patterns. Because sea turtles are a powerful indicator of overall ecosystem health, accurate identification serves to enhance our ecological understanding. Implicitly, this means that ensuring species conservation can be generated and optimized.

2.1 Turtle Tagging

In the past, the detection and tracking of individuals was done by attaching tags to the fins of the individuals found. This method is severely compromised by the loss of tags and thus the successful tracking of population dynamics is not guaranteed. Basically, the use of *flipper tags* was common, which are mostly made of metal and plastic. This method is very costly due to the extraordinarily long life span of the turtles, which means that the flipper tags were lost or identification was no longer possible due to wear and tear of the material.

2.2 Photographic Identification

Just as the human finger print, turtles have unique and time-stable facial scales by which they can be identified Carpentier, Jean, Barret, Chassagneux, and Ciccione (2016). Photographic identification has become the method of choice over time as it is non-invasive and low-cost. Due to the advances in machine learning and object identification, we are able to identify turtle individuals with algorithms as opposed to humans manually searching through a database of images. The goal of such machine learning algorithms is to assign a unique ID to each individual if it already exists in the database and to create a new ID if a new individual has been sighted.

3 Convolutional Neural Networks

In the field of machine learning, especially in the deep learning sector, a convolutional neural network (CNN) is a deep learning algorithm. It can take an input image and reproduce an identification of the object with a certain probability. The special feature of a CNN is that it is able to learn filters of different types (e.g. horizontal lines, vertical lines, etc.). A convolutional neural network has a characteristic structure in terms of its layers. It is structured in convolutional, pooling, and fully connected layers. The arrangement of alternating convolutional and pooling layers allows a more accurate and complex analysis of the image. The first layers focus on shapes and colors, while later layers contribute to the identification of more complex details for the recognition of the overall image.

CNNs have been around for a long time already: LeCun (1989), Fukushima's *neocognitron* (1979) **NEED CITATION**

3.1 Convolutional Layer

The convolutional layer is the key component of a CNN. It contains a certain set of filters, also called kernels. The parameters of the filter are learned over the course of the training. The filter interacts with the image and convolves it. From this convolution, an activation map is created which is calculated from the dot product between each element of the filter and the input. The weights in the filter are maintained as the filter moves across the image. However, these weights adjust during backpropagation and the associated gradient descent to achieve the most accurate results.

3.2 Pooling Layer

Since a great increase in dimension occurs through the use of a convolutional layer, a dimension reduction is required in the next step to reduce the number of parameters of a CNN. This is achieved by the pooling layer. It has the advantage that the computational cost decreases

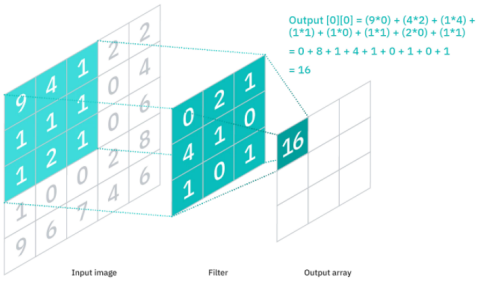

Figure 1: Convolutional Layer

drastically. Also, unnecessary details are omitted, which is helpful for the later image identification.The most commonly used pooling methods are maximum and average pooling. In figure 2 you can see how a maximum pooling is performed, leading to a dimension reduction.

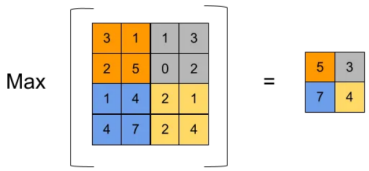

Figure 2: Max Pooling Layer

3.3 Fully-Connected Layer

Since the individual pixel values of the input image are not directly connected to the output layer, a fully-connected layer is required which is directly connected to the output layer. The layer does the classification using the collected features from the previous layers. At the end, a softmax function is applied, which outputs a classification using probabilities between 0 and 1.

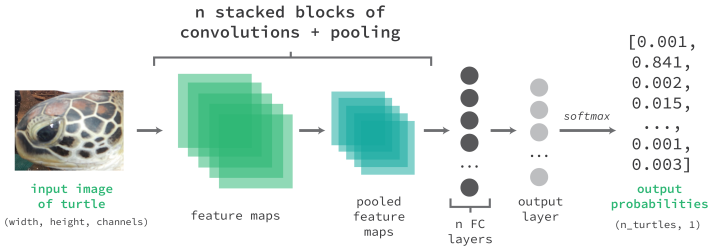

Figure 3: Turtle Image using CNN

4 Data

The dataset is provided with the *Turtle Recall: Conservation Challenge*[1] and consists of images of turtle faces. The images are labelled, each turtle has a unique ID. Besides the angle

[1] `https://zindi.africa/competitions/turtle-recall-conservation-challenge/data`

from which the image was shot (top, right, left) and the ID, no further info is provided about the images.

The images themselves have three colour channels, come in different sizes, and many have timestamps or handwritten identification tags in the background. They are split in three parts: a set of images for training, one for testing, and a large set of extra images.

The training set consists of 2145 images from 100 unique turtles. The set with the extra images comes with an additional almost 11000 images from 2231 different turtles, some already contained in the train set. This yields a total of about 13000 images and 2265 turtles. The test set only comes with images but without the annotation of turtle IDs, as it is meant to be used for model evaluation in the priced competition. However, this means that the test images are without use for our purpose.

Because the training set is rather small, we also make use of the extra images. A quick exploratory analysis shows that the dataset is both hugely unbalanced and that there are only less than 6 images per turtle on average, with a median number of 3 images per turtle. Because such a small amount of data per class will likely lead to very poor approximation results, we decide not to use the entire available data.

4.1 Preprocessing

Before actually using the images and feeding them into a neural net, we need to perform some necessary as well as some optional preprocessing steps in order to bring the image data into a more helpful format.

As a first step, we get rid of any turtles with less than a specified number of images in the dataset. We require each turtle to be represented by at least 10 images in the dataset. This reduces the total number of turtles to 253 and the total number of images to roughly 5000. Turtles now have an average of 20 images and a median of 14 images. There is still a considerable imbalance in the data, as can be seen in figure 4.

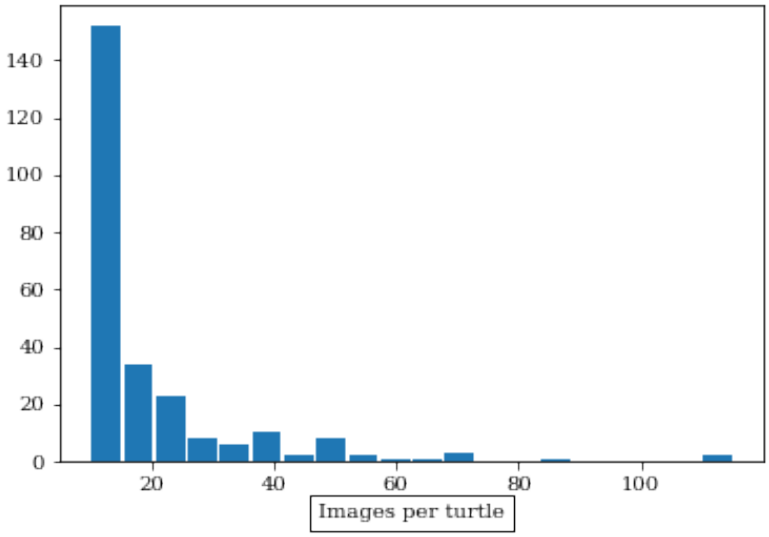

Figure 4: Number of images per turtle. The left skew of the distribution means there are many images with the minimum amount of images, while only a few have considerably more.

We further crop and resize the images such that they are all in the same shape and size, namely 224×224 pixels. Before doing any computations on the image data, we also convert

them to numerical arrays, yielding us a three-dimensional array of size $224 \times 224 \times 3$ for each used image. Since the colours are in the RGB colour space, the values are now in the range 0 to 255, which we normalise to a range of 0 to 1. This has been shown to increase training speed of deep networks (Ioffe & Szegedy, 2015). We further apply a one-hot encoding to the classes, as the alpha-numerical strings the turtle IDs are supplied in are rather meaningless to a neural net. At this point, we are technically ready to feed the images into a network for training.

4.2 Data Augmentation

The dataset, unfortunately, contains very little data unevenly distributed among the classes. A low number of training examples per class can lead to the class not being learned efficiently and may even impede the entire training process (Huh, Agrawal, & Efros, 2016). To avoid this, we apply a set of basic augmentation techniques to multiply the data available for training:

Image rotation We rotate the images by 90, 180, and 270 degrees. This immediately quadruples the available data. The basic characteristics of the pattern on the head of the turtle are preserved.

Gaussian filter We also apply a Gaussian filter to introduce a blur and reduce detail in the image. Mathematically, this is the same as convolving the images with a Gaussian function.

Random HSV The HSV colour space uses hue, saturation, and value to describe colours, whereas the RGB colour model uses a linear combination of the colours red, green, and blue to describe the same. We transform the RGB values to HSV, randomly modify them within a specified range, and convert back to RGB.

Additive noise Lastly, we add random numerical values, sampled from a normal distribution with very low standard deviation, to the existing image values, followed by clipping the values within range 0 to 1 (our normalisation range).

Applying the above techniques, we have enhanced our dataset and also somewhat shifted the distribution of images per turtle to be more desirable. Further common augmentation techniques include random brightness shifts, grey level mapping, histogram equalisation, image shifting and shearing. Example images of the used augmentations can be seen in appendix A.

5 Models

5.1 AlexNet

Krizhevsky, Sutskever, and Hinton have presented their influential *AlexNet* in 2012. It consists of only five convolutional layers, some additional max-pooling layers and three fully connected layers at the end, so its structure is rather simple. Nonetheless, it is packed with more than 40.000.000 trainable parameters (the fully connected layers at the end are very large), making it not exactly computationally cheap to train. An AlexNet was supposed to serve as our baseline model from which we wanted to build things up.

5.2 EfficientNet

We then implemented a pre-trained EfficientNet with custom top layer. The EfficientNet model family was first introduced in 2019 (Tan and Le) with the idea of providing scalable CNN architectures. Multiple networks of different size for image classification purposes were introduced, many achieving better results than state-of-the-art models like MobileNet and ResNet while being significantly smaller in terms of depth, width, and image resolution (Tan & Le, 2019). Two years later, a second, even more powerful and efficient generation was introduced: EfficientNetV2 (Tan & Le, 2021). For our purposes, given the low input resolution of 224×224, the model of choice is the EfficientNetV2-B0 – the smallest model of the ensemble. The model is pre-trained on ImageNet21k and we implemented a custom fully-connected layer at the end to match our classification task.

5.3 InceptionNet

For an additional comparison, we implemented yet another supposedly very efficiently designed CNN, the InceptionV3 as introduced by Szegedy, Vanhoucke, Ioffe, Shlens, and Wojna in 2015. Its computational low cost makes the Inception architecture an attractive choice when resources are limited, e.g. in mobile scenarios. The inception network was a milestone in the development of CNN classifiers. It achieved extremely high accuracies on the ImageNet dataset for example the inception v3 network achieved an accuracy of 78,1 percent. The fundamental idea behind the inception network is the inception block. In a traditional convolutional neural network layer, the previous layer's output is the input of the next layer until the state of prediction is accomplished. The inception block takes apart the individual layers. The previous layer's output is passed to four different operations in parallel and concatenates the outputs from all these different layers. So instead of constructing a deeper network, a " wider network" is provided. The naive approach consists of a 1x1 convolution layer, a 3x3 convolution layer and a 5x5 convolutions layer followed by a max-pooling layer and a concatenation layer. Due to the high computational cost especially with the 5x5 filter, the 1x1 is first added to the naive inception module. This leads to a computational reduction of 90 percent. Additionally, the 1x1 convolutional filters allow learning cross channel patterns across the depth of the input data.

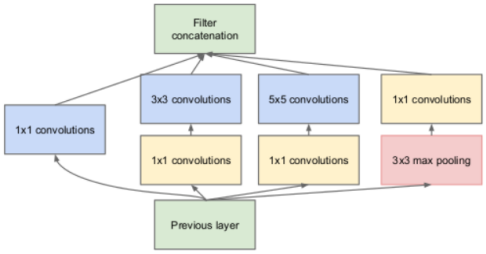

Figure 5: Inception Block

- balance of depth and width within network - aggressive regularisation - inception architecture already introduced by GoogLeNet (2014)

5.4 Hyperparameters

For simplicity, we use more or less the same hyperparameters for all networks.

Categorical cross-entropy: Cross-entropy loss is differentiable with respect to the logits and thus can be used for gradient training of deep models.

6 Evaluation

7 Open Questions + Future Outlook

References

Carpentier, A. S., Jean, C., Barret, M., Chassagneux, A., & Ciccione, S. (2016). Stability of facial scale patterns on green sea turtles chelonia mydas over time: A validation for the use of a photo-identification method. *Journal of Experimental Marine Biology and Ecology*, *476*, 15-21. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0022098115300733` doi: https://doi.org/10.1016/j.jembe.2015.12.003

Huh, M., Agrawal, P., & Efros, A. A. (2016, 8). What makes imagenet good for transfer learning? *CoRR*. Retrieved from `https://arxiv.org/abs/1608.08614v2`

Ioffe, S., & Szegedy, C. (2015, 2). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*. Retrieved from `http://arxiv.org/abs/1502.03167`

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks.. Retrieved from `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdfhttp://www.cs.toronto.edu/~kriz/imagenet_classification_with_deep_convolutional.pdf`

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015, 12). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 2818-2826. Retrieved from `https://arxiv.org/abs/1512.00567v3` doi: 10.48550/arxiv.1512.00567

Tan, M., & Le, Q. V. (2019, 5). Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*.

Tan, M., & Le, Q. V. (2021). Efficientnetv2: Smaller models and faster training. *CoRR*, *abs/2104.00298*. Retrieved from `https://arxiv.org/abs/2104.00298`

Appendices

A Augmented Images

Figure 6: Non-augmented image.

Figure 7: Image rotated by 180 degrees.

Figure 8: Image with Gauss-filter applied.

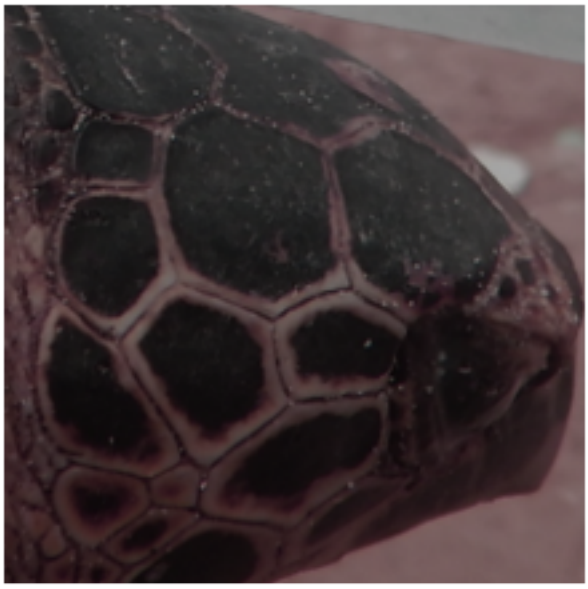

Figure 9: Image with randomly changed HSV values.

Figure 10: Image with added noise.