Plankton Classification with Deep Convolutional Neural Networks

Hendrik Weideman, Chuck Stewart

Department of Computer Science
Rensselaer Polytechnic Institute

April 2015

Problem Statement

The distribution of different plankton species in a lake serves as a key indicator to determine the health of an aquatic ecosystem. Unfortunately, collecting and classifying these organisms is a challenging task, even for highly-trained experts. We plan to make this process easier by developing a system that automatically can classify organisms into different classes of plankton. Ideally this would help biologists to

- study the distribution of millions of organisms across a lake,
- study how these distributions change at different depths and locations, and
- build a spatial-temporal map of species across a lake.

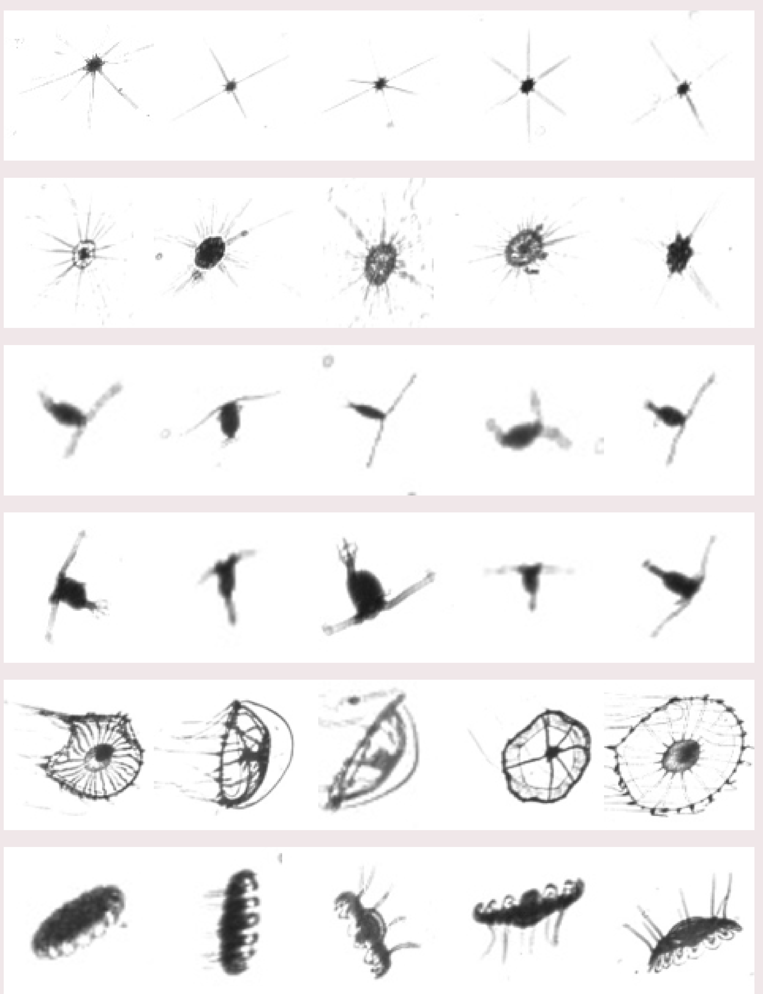
Project Goals

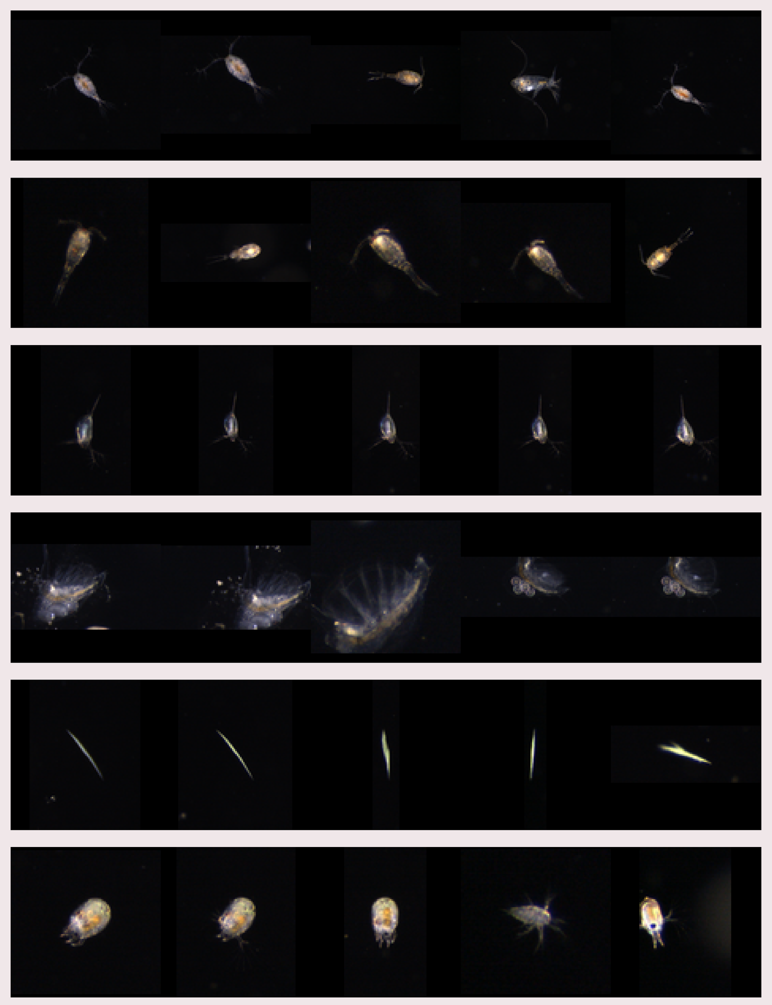
Currently, such a study would be infeasible because of the sheer effort required to classify organisms on this scale. Thus, we aim to streamline this process to the point where such a study could be practical, by developing a learning model that can

- automatically classify plankton into different pre-defined classes,
- be used to assist a human operator to accelerate the process of manual annotation, and
- be extended to recognize new classes without prohibitively expensive re-training.

The Data: Background

We are currently collecting data to build an annotated plankton dataset from the species present in Lake George. Eventually this data will form the core of our training data. Until this dataset is complete, however, we are working with a very recent dataset made available through a Kaggle competition to develop our classification algorithms. Below we show images from both datasets. On the left are images from the Kaggle dataset, and on the right are some images from the in-progress Lake George dataset. All the organisms in the same row belong to the same class.

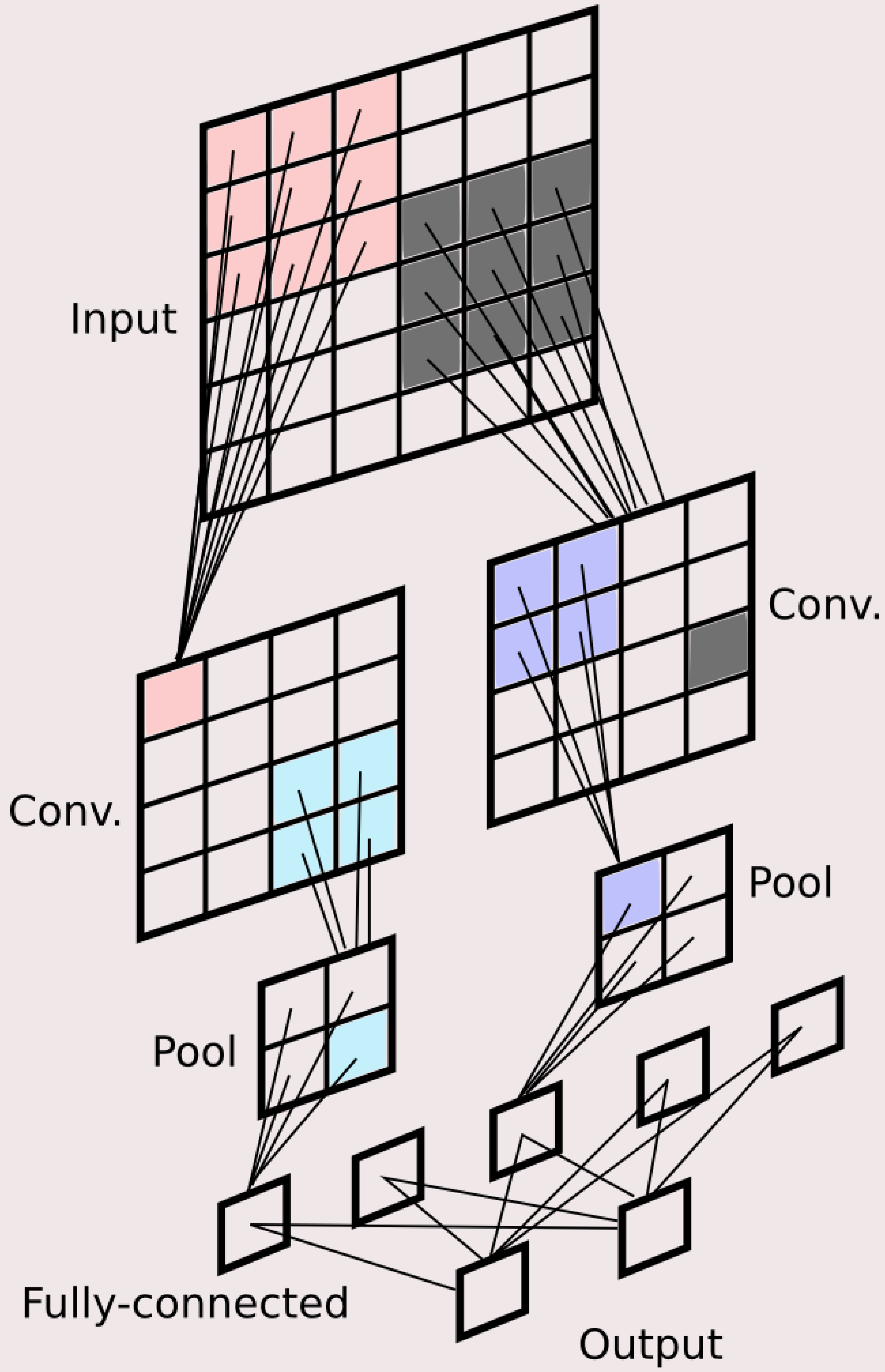




The Data: Challenges

- Two organisms from the same species do not necessarily belong to the same class. Often a species-level classification is too coarse for biologists when studying the complex dynamics in an ecosystem, and so our learning model needs to deal with some classes that are distinct, yet visually very similar.
- The images are two-dimensional representations of a three-dimensional world, and so organisms appear in various orientations around three rotational axes.

A Simple Deep Convolutional Neural Network



A Stack of Non-Linear Layers

The deep convolutional neural network shown on the left has been greatly simplified for visualization purposes, with only two feature maps per layer and only a single convolutional and pooling layer, respectively. Additionally, many connections between layers are omitted for visual clarity. A real network would typically have several convolutional and pooling layers, with anything from tens to hundreds of feature maps per layer. Additionally, the output from each convolutional and fully-connected layer typically passes through a non-linear function called the *activation function*. This stack of layers, each with its own non-linearity, is what allows deep convolutional neural networks to learn a highly non-linear mapping from input to output.

Different Layers for Different Purposes

- The **input layer** takes a gray scale image, or color image separated into channels, and processes the raw pixel data directly, with an optional pre-processing transformation.
- **Convolutional layers** consist of multiple convolutional feature maps, where each feature map has an associated convolutional kernel that is learned by means of error backpropagation. This kernel is replicated over the entire input with shared weight parameters.
- **Pooling layers** introduce a degree of translational invariance by selectively keeping outputs from the preceding convolutional layer's feature maps. When only the maximum responses are kept, this is so-called *max-pooling*.
- **Fully-connected layers** are closely related to the hidden layers of a standard artificial neural network. They disregard any spatial information from the previous layer and simply compute a function mapping the output of the previous layer to target outputs.
- The **output layer** is typically another fully-connected layer with a special non-linearity, typically a *softmax* activation function. This layer computes a probability distribution function over the target classes from the output of the previous layer.

The Future

Once our system is complete, we plan that it may be used to

- produce new annotated datasets for scientific research,
- accelerate manual annotation to allow researchers to label their own datasets,
- assist biologists in gaining a better understanding of complex aquatic ecosystems, and
- assist researchers in the simulation sciences to create an immersive visualization of plankton communities.

Rensselaer