Classifying Plankton with Deep Neural Networks

Kewin Dereniewicz & Tom van den Bogaart

The Problem

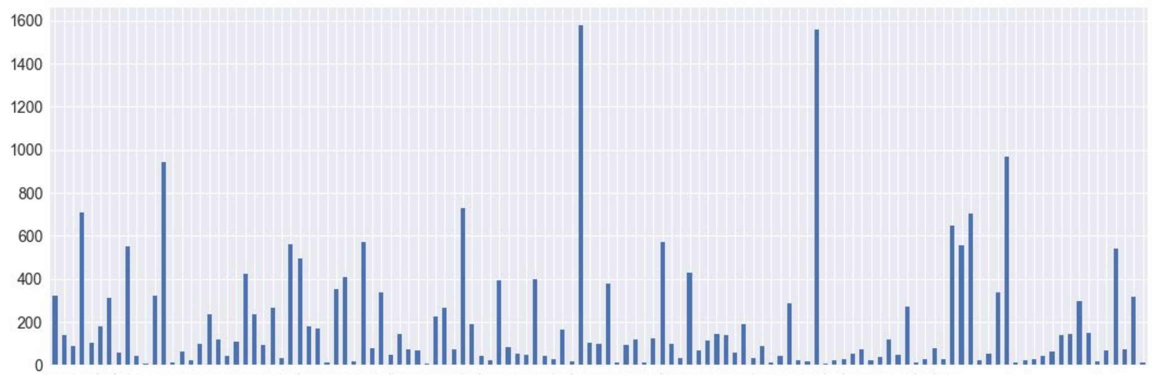
Build an algorithm to automate plankton identification process. The goal is to classify grayscale images of plankton into one of 121 classes.

Exploring the Data

- Training data - 24204 images with 121 different labels
- Test data - 6132 images
- Images have different dimensions
  → Set fixed image size to 64x64
- Species with least amount vs highest amount

Class	Occurrences
ephyra	7
hydromedusae_haliscera_small_sideview	7
heteropod	7
...	...
copepod_cyclopoid_oithona_eggs	965
trichodesmium_puff	1555
chaetognath_other	1580

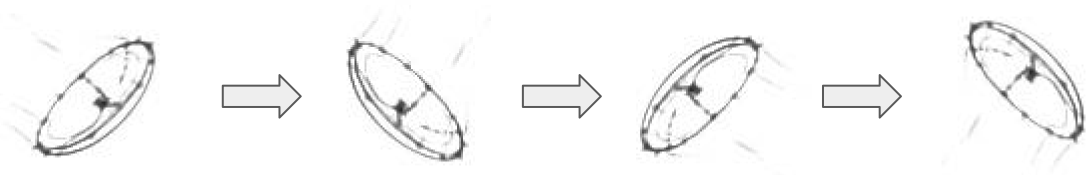
- Frequency distribution

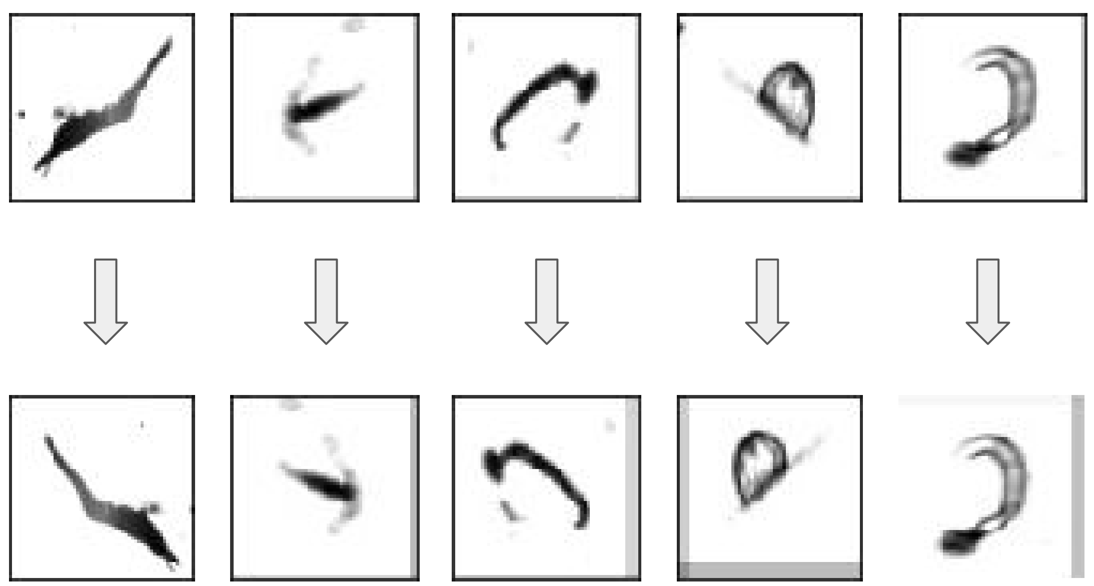


Data Augmentation

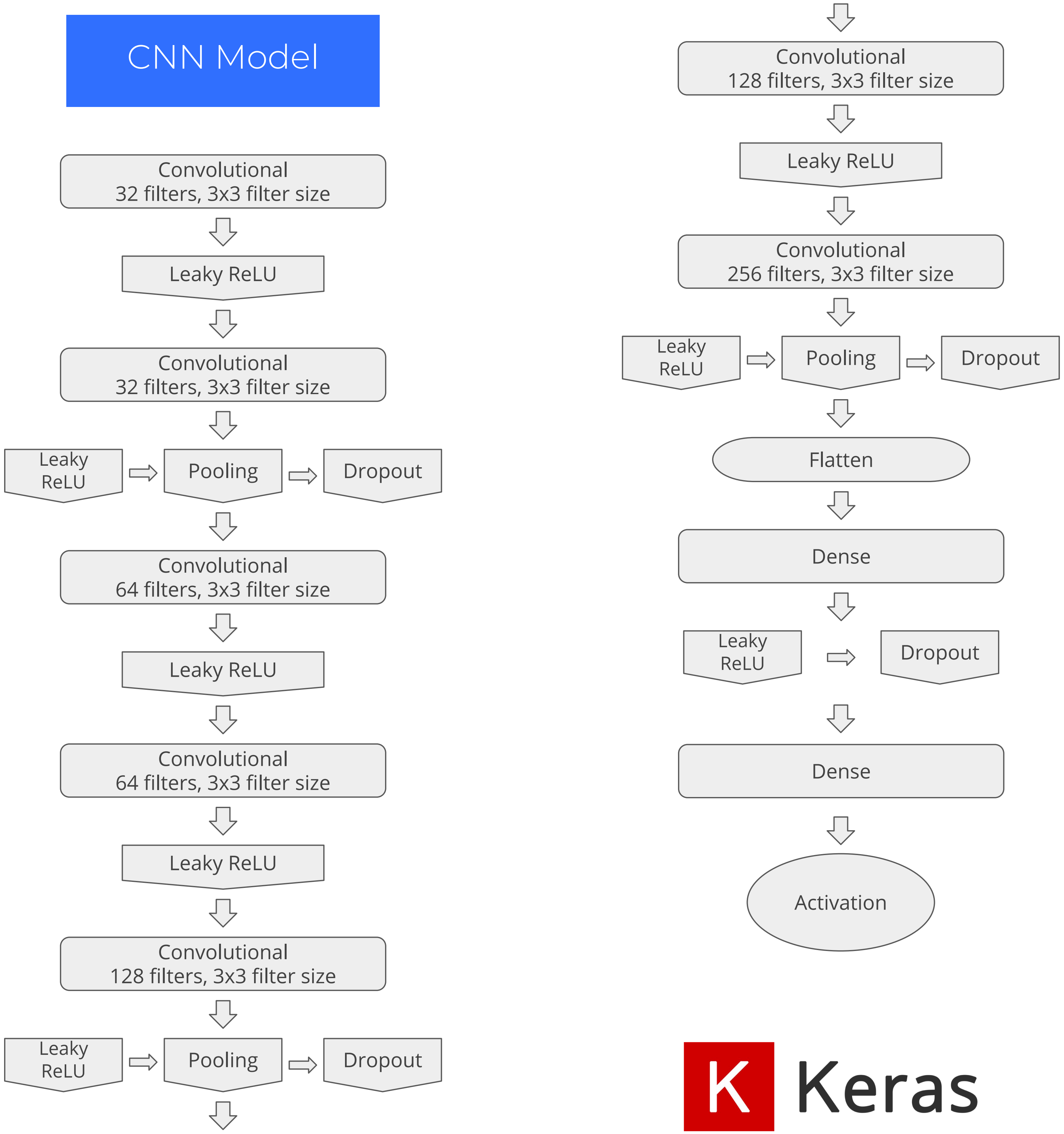
To obtain more data for training we use (*realtime*) data augmentation.

- Randomly shift images horizontally
- Randomly shift images vertically
- Randomly flip images horizontally
- Randomly rotate image by 0, 90, 180 or 270 degrees





Model architecture



Keras

Overfitting

To prevent overfitting the model uses dropout layers

Optimizer

SGD with momentum and nesterov

Loss function

Categorical cross entropy

Activation function

Leaky ReLU yielded better results than standard activation functions

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.03x & \text{otherwise} \end{cases}$$

Early stopping

When validation loss starts decreasing the training is stopped

Tensorboard

Further evaluate the model and the training process

Model checkpoint

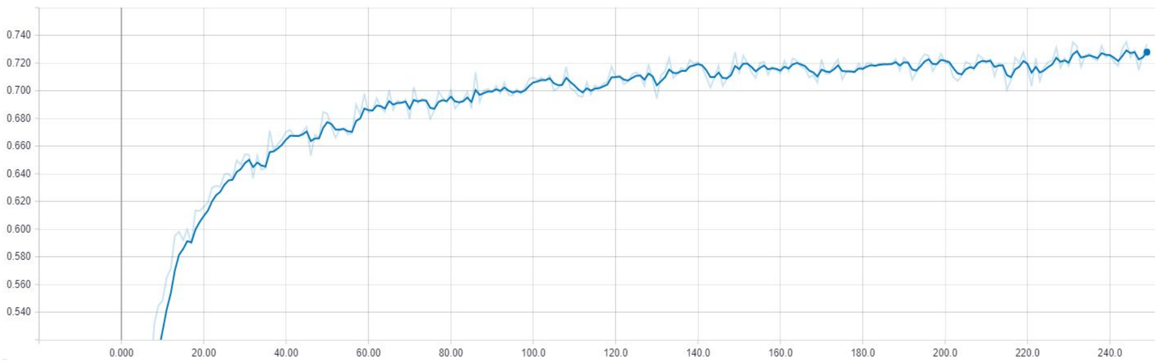
After each epoch, the model is validated and the best one is saved to a file

Results

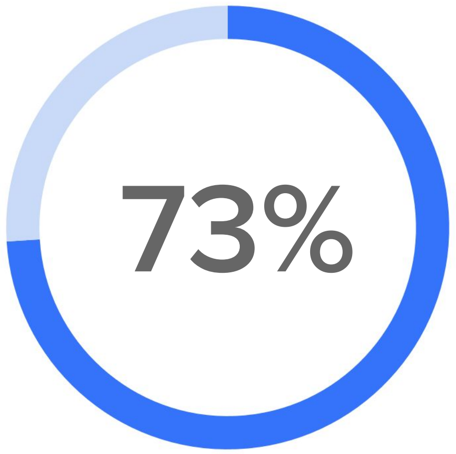
Classification report on average for validation set (20%)

Precision	Recall	F1-Score	Support
0.73	0.73	0.72	4841

Validation accuracy over amount of training epochs



$$\mathit{Category\ Accuracy} = \frac{1}{N}\sum_{y_i=\hat{y}_i} 1,$$

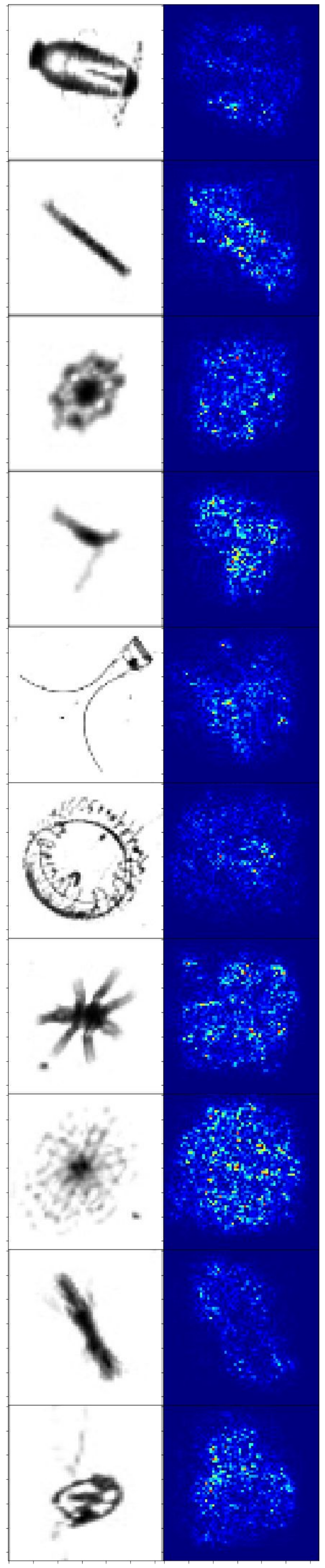


Kaggle Rank

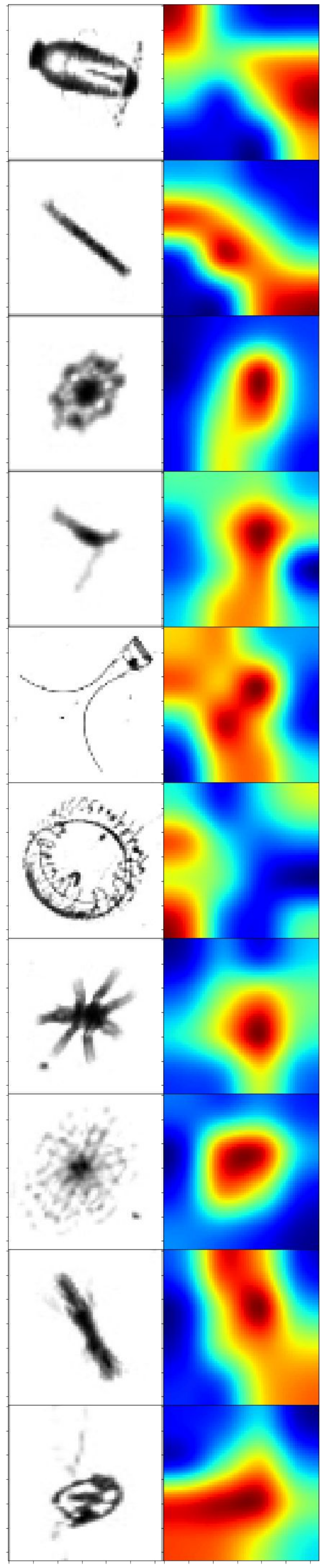
Visualizations

Attention Maps

saliency

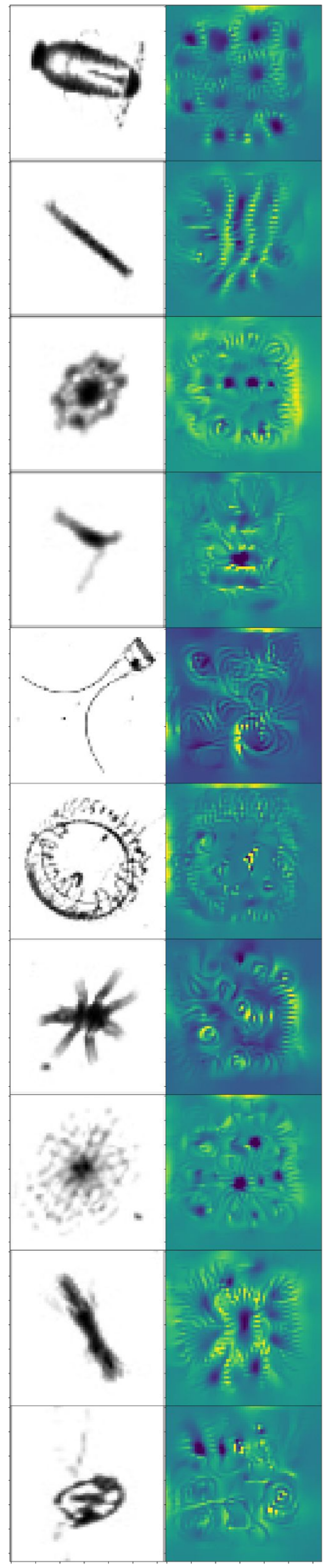


grad-CAM



Network's Perception of Species

visualization final dense layer outputs



References

1. Kotikalapudi, Raghavendra and contributors, *keras-vis: Neural network visualization toolkit for Keras*, 2017, https://github.com/raghakot/keras-vis
2. Andrej Karpathy, *CS231n: Convolutional Neural Networks for Visual Recognition*, 2017, https://goo.gl/W721rT
3. Sander Dieleman, *Classifying Plankton with Deep Neural Networks*, 2015 https://goo.gl/gPZske