Plankton Classification Using VGG16 Network

Lucas Tindall
UCSD
ltindall@ucsd.edu

Cuong Luong
UCSD
c3luong@ucsd.edu

Andrew Saad
UCSD
aabdelme@ucsd.edu

Abstract

Deep learning pipelines have become a very popular method for various image recognition and classification tasks. Over time, these network architectures have grown to include many layers and are capable of producing robust networks for classification tasks. In this paper we specifically explore the effectiveness of the VGG16 convolutional neural network architecture on a 12 class subset of the WHOI Plankton dataset. We examine the benefits of transfer learning by using VGG network weights trained on the ImageNet dataset. In the end, we are able to achieve an test accuracy rate of 85%. We also explore several visualization techniques in order to make sense of what convolutional neural networks "learn".

1. Introduction

Plankton are important to marine ecosystems because of their role in the food web and biogeochemical cycles. Recently, imaging methods have assisted scientists in studying plankton ecosystems. However, plankton classification is tedious so efficient automatic classification systems based on machine learning techniques have been developed [8]. Specifically, convolutional neural networks trained on large volumes of annotated data have been capable to achieving state of the art results. This paper will begin with a discussion of the VGG16 architecture. We will analyze the results from performing transfer learning with VGG16 and also explore several visualization techniques. The data used in this paper is a subset of Oregon State Universitys Hatfield Marine Science Center plankton dataset from the National Data Science Bowl. The data was collected using In Situ Ichthyoplankton Imaging System (ISIIS) and the data is composed of 12 classes, 11868 training images and 2772 test images.

2. Convolutional Neural Network Classification

2.1. VGG16

VGG16 is a 16-layer network used by the Visual Geometry Group at the University of Oxford to obtain state of the art results in the ILSVRC-2014 competition. The main feature of this architecture was the increased depth of the network. In VGG16, 224x224 RGB images are passed through 5 blocks of convolutional layers where each block is composed of increasing numbers of 3x3 filters. The stride is fixed to 1 while the convolutional layer inputs are padded such that the spatial resolution is preserved after convolution (i.e. the padding is 1 pixel for 33 filters). The blocks are separated by max-pooling layers. Max-pooling is performed over 22 windows with stride 2. The 5 blocks of convolutional layers are followed by three fully-connected (FC) layers. The final layer is a soft-max layer that outputs class probabilities. The complete model is shown in figure 1.

3. Methods

We implemented our model in Keras using ImageNet weights provided by the library. The plankton dataset was provided by Peter Gerstoft [5]. All of our preprocessing, training, testing, and visualization code can be found on GitHub [2].

4. Results

We used the VGG16 network instead of training a custom CNN from scratch for the task of plankton classification. Our model used VGG16 with pretrained ImageNet weights but without the final default 1000 class softmax. We attached our own 12 class softmax for plankton classification and only trained that layer. The VGG16 network is essentially used as a feature extractor for a simpler softmax classifier. Experiments were done with both RMSprop[1]/Adam [7] optimizers and plots of loss/accuracy vs iterations are shown in figures 2 and 3 respectively. After performing transfer learning with a fine

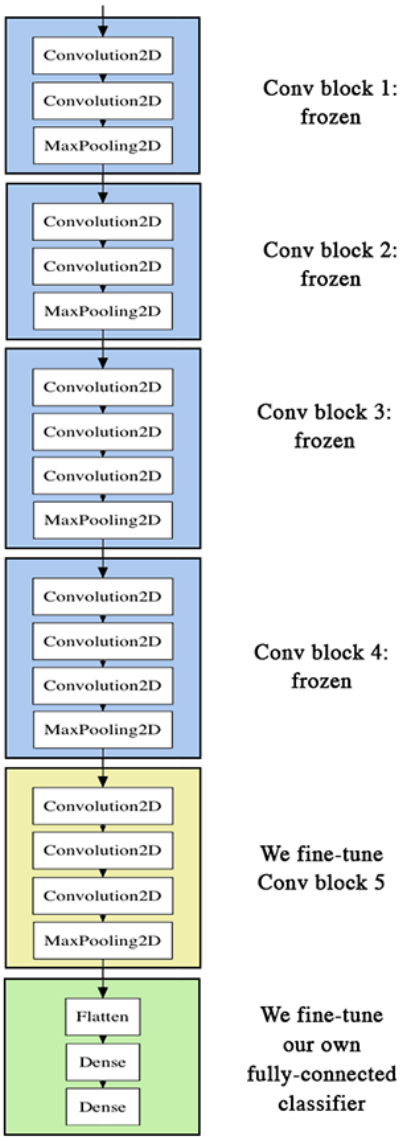

Figure 1: VGG16VGG16 Architecture

Optimizer	Accuracy	Loss
RMSprop	0.85	0.42
Adam	0.85	0.41

Table 1: VGG16 Plankton Transfer Learning Results

tuned final activation layer we were able to achieve a top accuracy of 85% on the 12 class subset of the plankton images. The best results for each optimization method can be found in table 1.

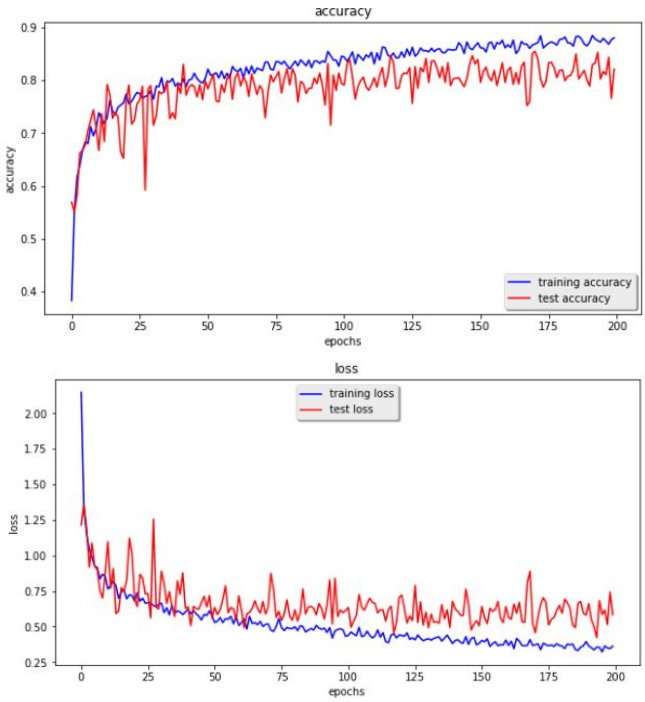

Figure 2: RMSprop Accuracy/Loss vs Iterations

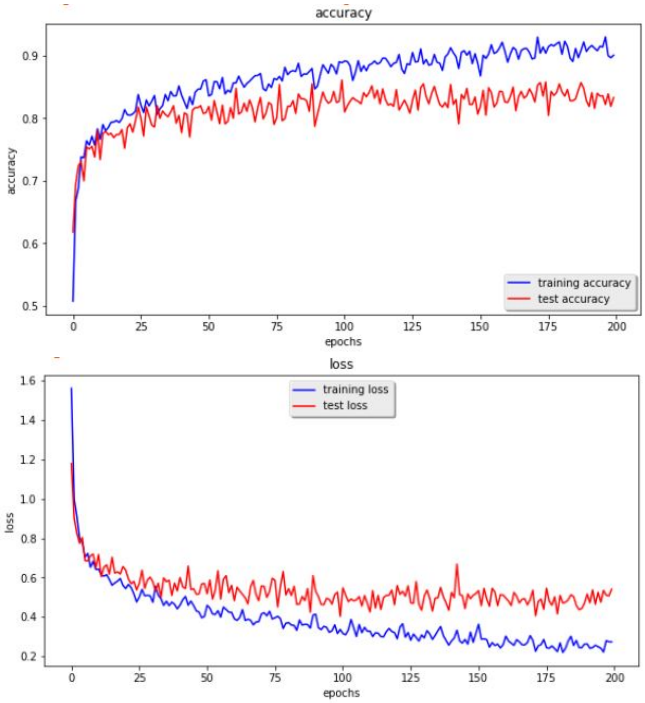

Figure 3: Adam Accuracy/Loss vs Iterations

5. Convolutional Neural Network Visualizations

CNNs are capable of extracting the relevant features from images for the task of classification as shown in the previous section. However, the black box structure of CNNs obfuscates just exactly what a CNN is learning but the following visualization tools can provide some insight. CNNs are composed of convolutional layers that are essentially groups of filters. One visualization is to find an input image that maximizes the activations of a particular filter. This provides insight into what a particular filter is learning within the CNN. This method can be extended to the final dense layer to visualize the features that are important for a particular output class. Lastly, we can draw heatmaps over input images that correspond to how important each area is to the classification decision. The following results are obtained with the help of the keras-vis library [4] according to instructions from [3].

5.1. Filter Visualizations

VGG16 is composed of blocks of 3x3 filters separated by max-pooling layers. We can visualize what features each filter captures by learning the input image that maximizes the activation of that filter. The input image is initially random while the loss is calculated as the activation of a particular filter. Using gradient ascent to maximize this loss generates synthetic images that capture what a filter learns. Example visualizations after 20 iterations are shown in figure 4 to 5. The filters within the first block capture color and simple textures. Subsequent blocks take more iterations to converge but capture more sophisticated patterns. These visualizations show that CNNs "see" remarkably differently from humans but do extract and combine important features like color and texture when making predictions.

5.2. Dense Layer Visualizations

We can extend the visualization technique in the previous section to find images that maximize the final dense layer softmax for a particular category. Gradient ascent is applied until the loss reaches ≈ 1 which implies that the CNN is 100% confident that the synthetic image is that category. This technique was applied to a trained VGG16 networks with ImageNet category outputs and the images that maximize the "eagle","ouvel", and "sea snake" categories are shown in figure 8. While the generated images do not look natural, features like the eagles/ouvel wings/beak and sea snake bodies are clearly present which confirm that the CNN learns relevant features. The technique was also applied to the modified VGG16 net used for plankton classification. Convergence of gradient ascent was more difficult and only occurred for several classes of plankton. However, the resulting synthetic images also capture the relevant plankton features. In figure 9 we show the dense layer visualizations for the acanthera and helix plankton classes. These visualizations show both the spiked pattern of the acanthera class and the circular pattern of the helix class.

5.3. Attention Maps

Not every patch within an image contains information that contributes to the classification process. We can visualize the regions of an image that are most relevant with attention heatmaps. Existing methods for generating attention maps include occlusion maps and class activation maps (CAM). However, occlusion maps are inefficient and CAM is limited to certain CNN architectures ([4]) so gradient-weighted CAM(grad-CAM) as implemented in keras-vis will be used instead. Grad-CAM uses the class-specific gradient information within the final convolutional layer to generate attention maps ([9]).

Results for the 12 provided plankton classes are shown in figure 10. The heatmaps demonstrate that the network mainly focuses on the plankton bodies. However, it does make classification errors and this can be attributed to when the model applies attention to the patches that don't include the plankton.

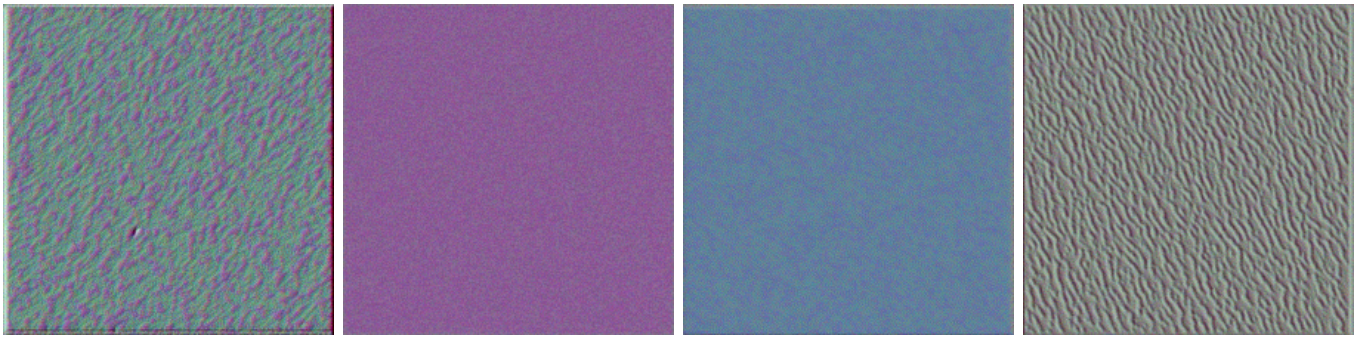

Figure 4: VGG16 Block 1 Filters

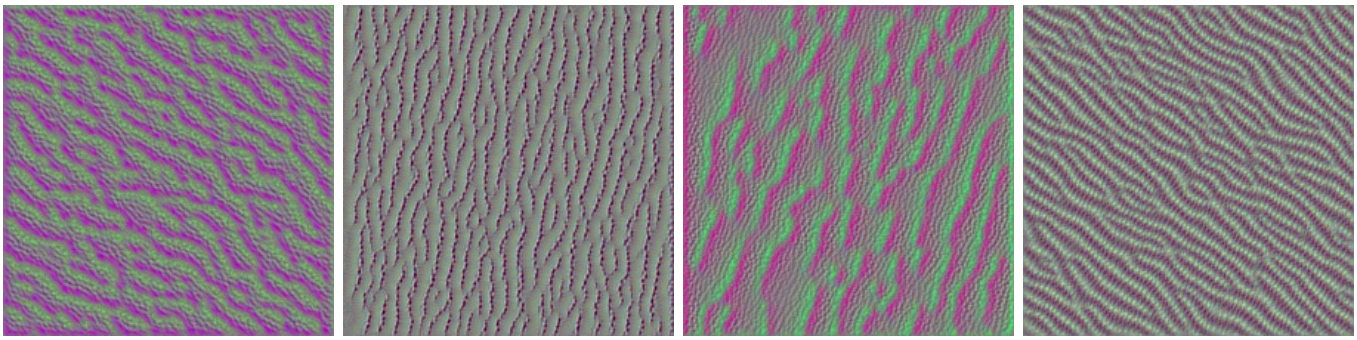

Figure 5: VGG16 Block 2 Filters

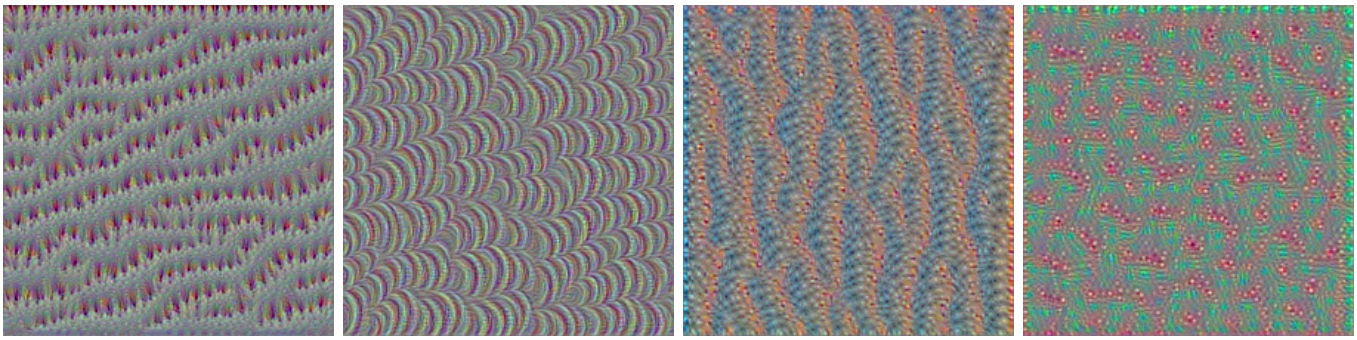

Figure 6: VGG16 Block 3 Filters

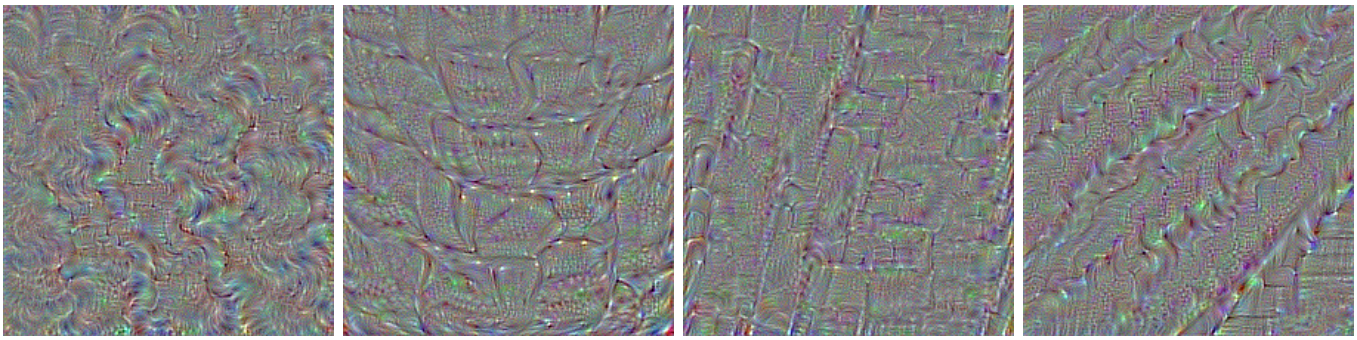

Figure 7: VGG16 Block 5 Filters

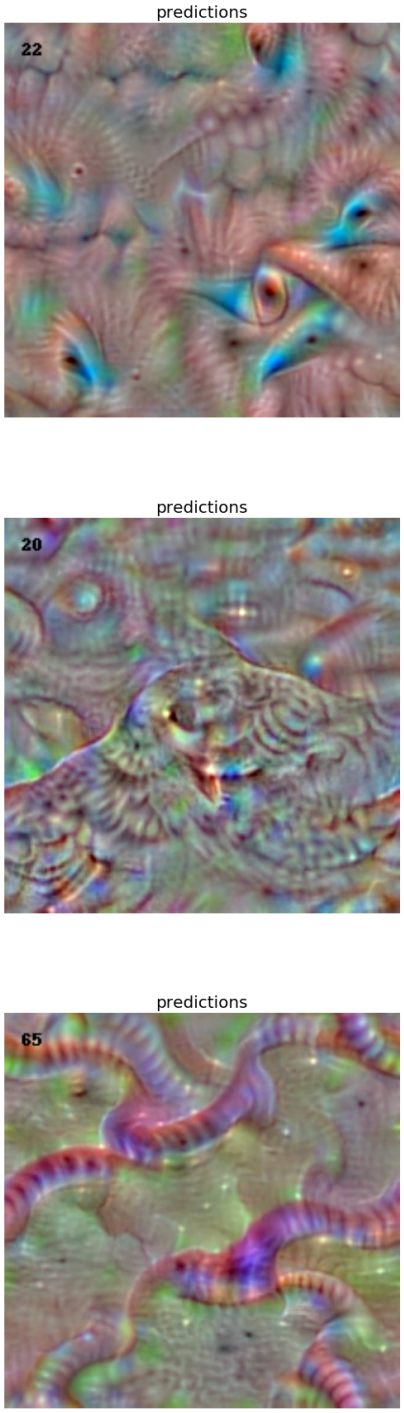

Figure 8: Dense Layer Visualizations using ImageNet Categories (eagle, ouvel, and seasnake classes)

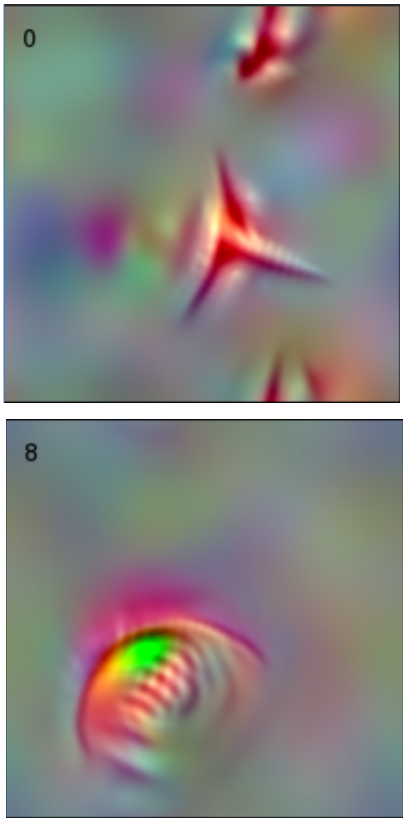

Figure 9: Dense Layer Visualizations using Plankton Categories; the top image corresponds to the acanthera class, the bottom image corresponds to the helix class

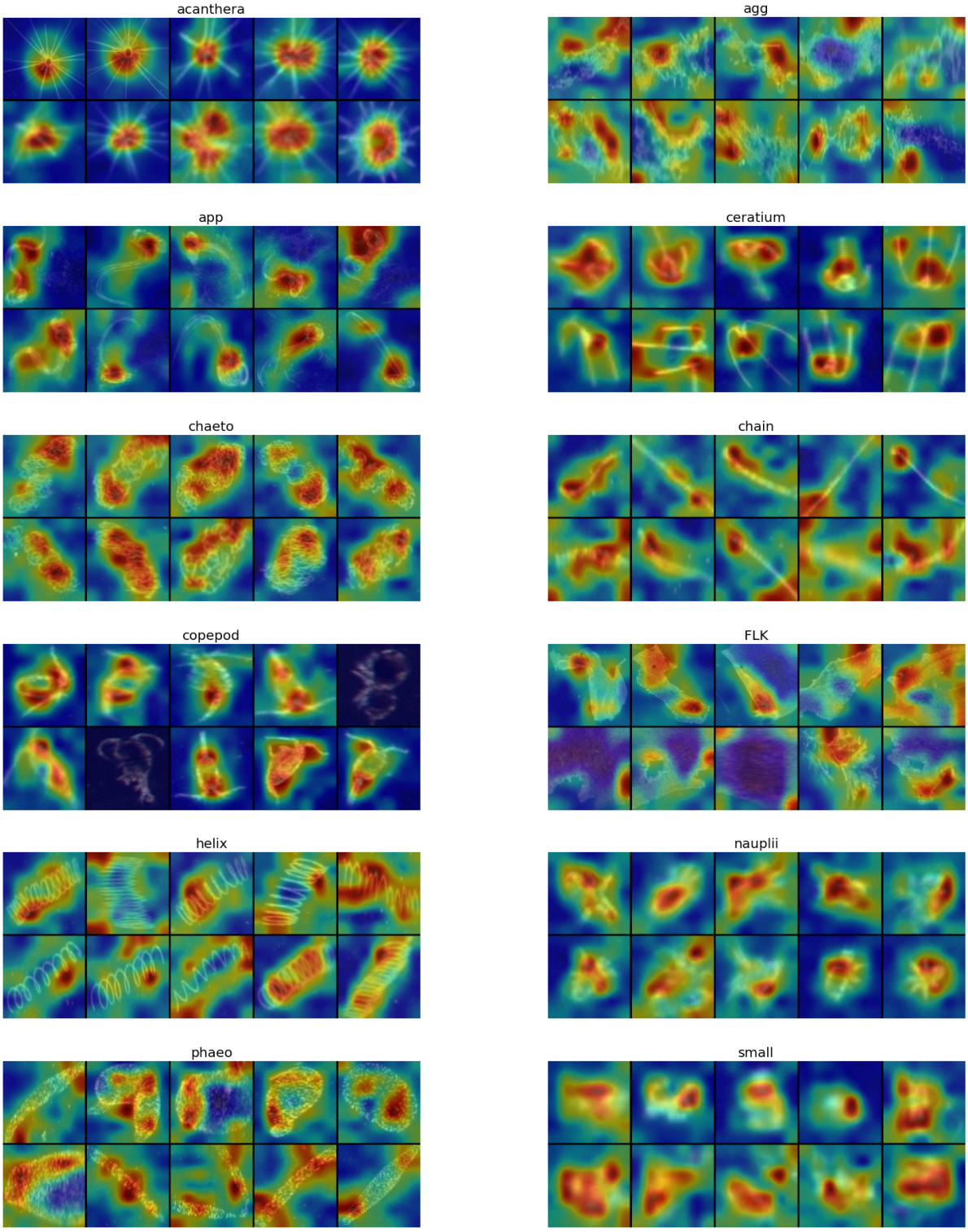

Figure 10: Plankton Attention Maps

6. Conclusion

In this paper we applied transfer learning to plankton classification. Using a VGG network pre-trained on the large ImageNet dataset we then fine tuned the last dense layer to learn features of the plankton dataset. The effectiveness of this method showcases the robust properties of the VGG network. Even though the network had been trained on different images, the many features that had been learned throughout the 16 layers of the VGG proved to be extensible to multiple classification tasks. The extensibility of these features can be seen in the filter visualizations and the attention maps. Our dense layer visualizations show how our fine tuned last layer was also able to extract distinguishing features for each plankton class.

References

[1] Geoff hinton's coursera course - lecture 6e. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[2] Github repo for report code. https://github.com/ltindall/ece285_plankton.

[3] How convolutional neural networks see the world - an exploration of convnet filters with keras. https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html.

[4] Keras visualization toolkit. https://raghakot.github.io/keras-vis/.

[5] Plankton dataset. https://www.dropbox.com/sh/o3dghc0d5dl67b3/AAC4_v3e2wayZ5sry4aTmUYWa/image_data.

[6] Visualizing what convnets learn. http://cs231n.github.io/understanding-cnn/.

[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[8] E. C. Orenstein, O. Beijbom, E. E. Peacock, and H. M. Sosik. Whoi-plankton- A large scale fine grained visual recognition benchmark dataset for plankton classification. *CoRR*, abs/1510.00745, 2015.

[9] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016.

[10] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[12] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.