The Nature Conservancy Fisheries Monitoring:

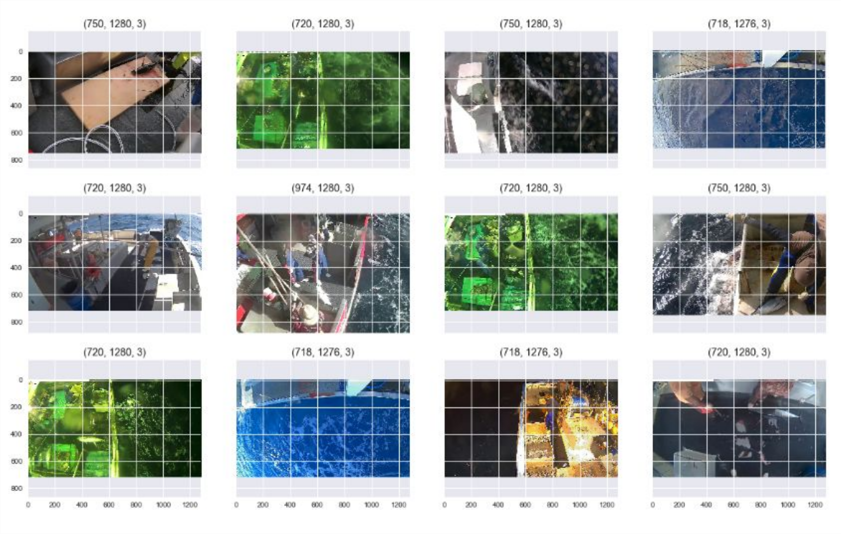
Supporting Automated Fish Classification to Help Maintain Ocean Biodiversity

The problem

- **Goal:** Detect and categorize fish species based on images.
- **Data source:** The Nature Conservancy
- **Target categories:** Albacore tuna, Bigeye tuna, Yellowfin tuna, Mahi Mahi, Opah, Sharks, Other (meaning that there are fish present but not in the above categories), and No Fish (meaning that no fish is in the picture)

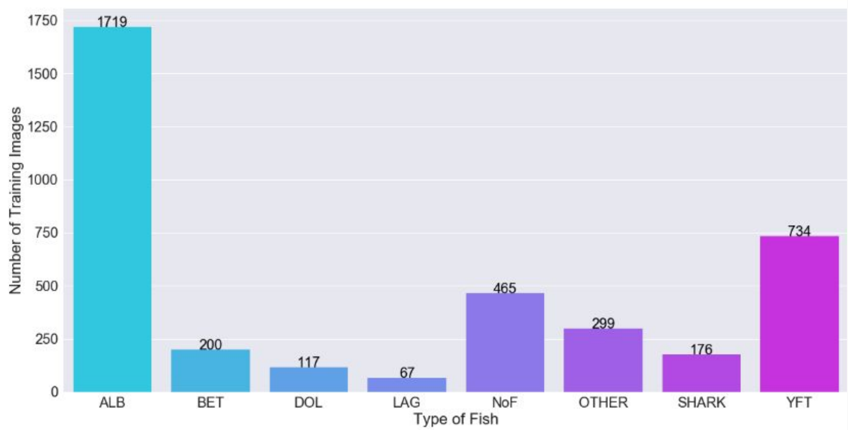
Exploratory Data Analysis

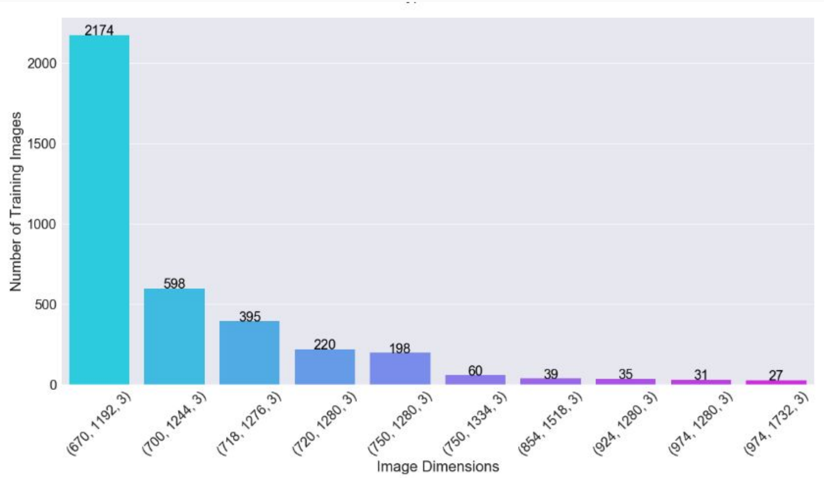
- Sample Images with Pixel sizes
- Color images (RGB) with pixels and channels as features



Training data set

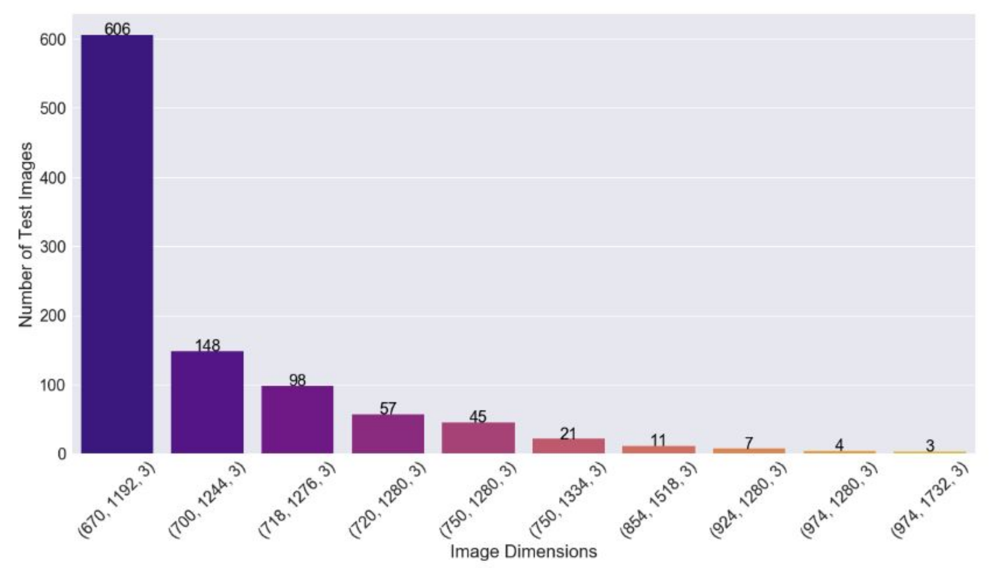
- 3777 Training images with various dimensions and types of fish





Test data set

- 1000 Test images with various dimensions



Data Preprocessing

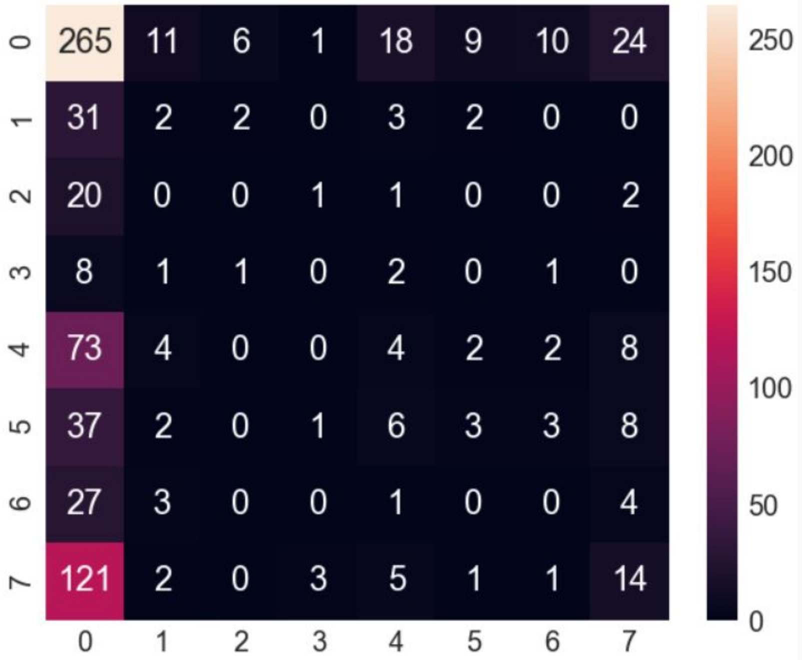
- OpenCV used for reading image files
- CNNs require all image sizes and aspect ratio to be constant over all input images
- Through trial and error, we decided 70x124 (row x col) yielded the best results based on accuracy
- One-hot encoding used based on the categorical nature of our dataset
- Convert data from int8 format into float32
- Dimensionality reduction used for our grey-scale model

Baseline Results

```
Baseline Accuracy: 0.1204
Baseline Log Loss: 15.07955286364076
##--##--##--##--##--##--##--##--##--##--##--##--##--##
Classification Report for Baseline Accuracy
              precision    recall  f1-score   support

           0       0.43      0.42      0.42       344
           1       0.04      0.05      0.04        40
           2       0.00      0.00      0.00        24
           3       0.00      0.00      0.00        13
           4       0.14      0.14      0.14        93
           5       0.10      0.07      0.08        60
           6       0.00      0.00      0.00        35
           7       0.23      0.24      0.24       147

   micro avg       0.27      0.26      0.27       756
   macro avg       0.12      0.11      0.12       756
weighted avg       0.27      0.26      0.27       756
 samples avg       0.19      0.26      0.21       756
```



Training Model 1: RGB

```
Layer (type)                 Output Shape              Param #
=================================================================
cropping2d_1 (Cropping2D)    (None, 66, 120, 3)        0
_________________________________________________________________
activation_3 (Activation)    (None, 66, 120, 3)        0
_________________________________________________________________
activation_4 (Activation)    (None, 66, 120, 3)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 66, 120, 32)       896
_________________________________________________________________
activation_5 (Activation)    (None, 66, 120, 32)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 66, 120, 64)       18496
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 66, 120, 128)      73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 33, 60, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 33, 60, 128)       0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 33, 60, 256)       295168
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 16, 30, 256)       0
_________________________________________________________________
dropout_3 (Dropout)          (None, 16, 30, 256)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 122880)            0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               31457536
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 2056
=================================================================
Total params: 31,848,008
Trainable params: 31,848,008
Non-trainable params: 0
```

Training Model 2: Grayscale

```
Layer (type)                 Output Shape              Param #
=================================================================
cropping2d (Cropping2D)      (None, 66, 120, 1)        0
_________________________________________________________________
activation (Activation)      (None, 66, 120, 1)        0
_________________________________________________________________
activation_1 (Activation)    (None, 66, 120, 1)        0
_________________________________________________________________
conv2d (Conv2D)              (None, 66, 120, 32)       320
_________________________________________________________________
activation_2 (Activation)    (None, 66, 120, 32)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 66, 120, 64)       18496
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 66, 120, 128)      73856
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 33, 60, 128)       0
_________________________________________________________________
dropout (Dropout)            (None, 33, 60, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 33, 60, 256)       295168
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 30, 256)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 30, 256)       0
_________________________________________________________________
flatten (Flatten)            (None, 122880)            0
_________________________________________________________________
dense (Dense)                (None, 256)               31457536
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 2056
=================================================================
Total params: 31,847,432
Trainable params: 31,847,432
Non-trainable params: 0
```

Testing Model 1: RGB

```
756/756 [==============================] - 44s 58ms/step
Validation Log Loss: 0.2976813446213345
```

Training Model 2: Grayscale

```
756/756 [==============================] - 40s 52ms/step
Validation Log Loss Grayscale: 0.3333505303601569
```

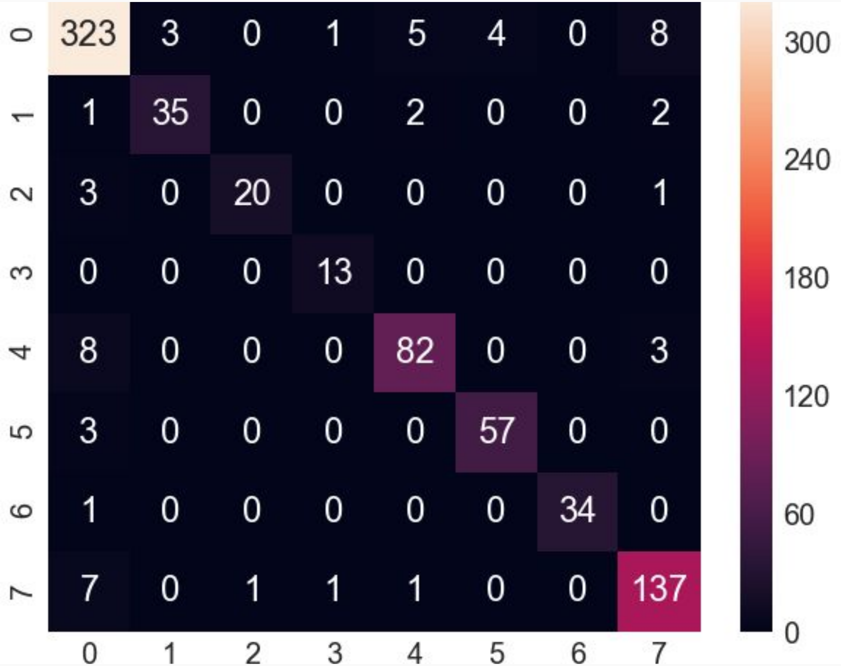
Conclusions

- There was not a huge difference between the RGB/Grayscale models in terms of log loss, time to train, or overfitting - (RGB had slightly lower validation log loss)
- Trial and error is very important with CNN - lots of settings, and not always intuitive which will work best
- If we had time to try additional improvements, we would use tfrecord to convert files to a faster format and allow for easier training and potentially deeper layering at little computational cost

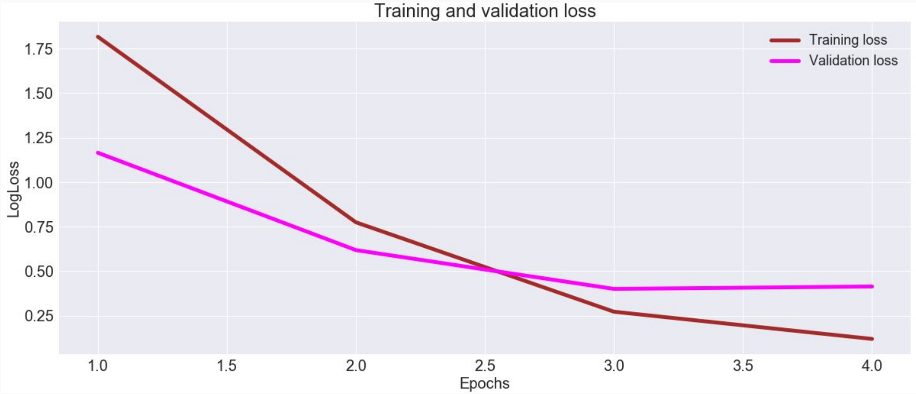
How accurate are we?

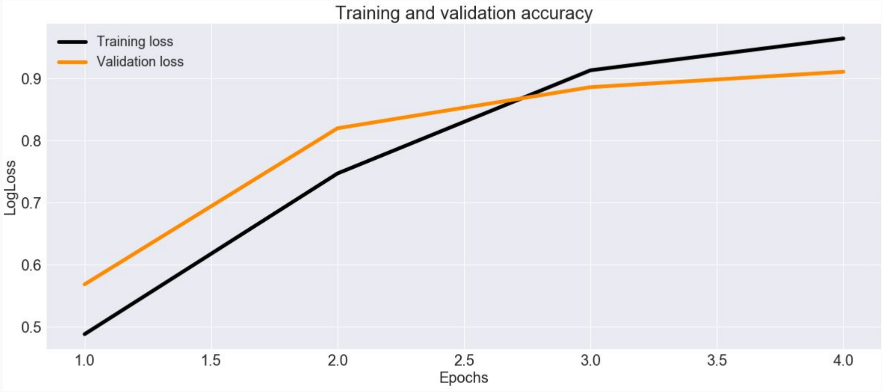
Classification Report for Grayscale Model Accuracy

	precision	recall	f1-score	support
0	0.92	0.96	0.94	344
1	0.92	0.90	0.91	40
2	0.78	0.88	0.82	24
3	1.00	1.00	1.00	13
4	0.96	0.86	0.91	93
5	0.92	0.97	0.94	60
6	1.00	0.94	0.97	35
7	0.93	0.88	0.90	147
micro avg	0.93	0.93	0.93	756
macro avg	0.93	0.92	0.92	756
weighted avg	0.93	0.93	0.93	756



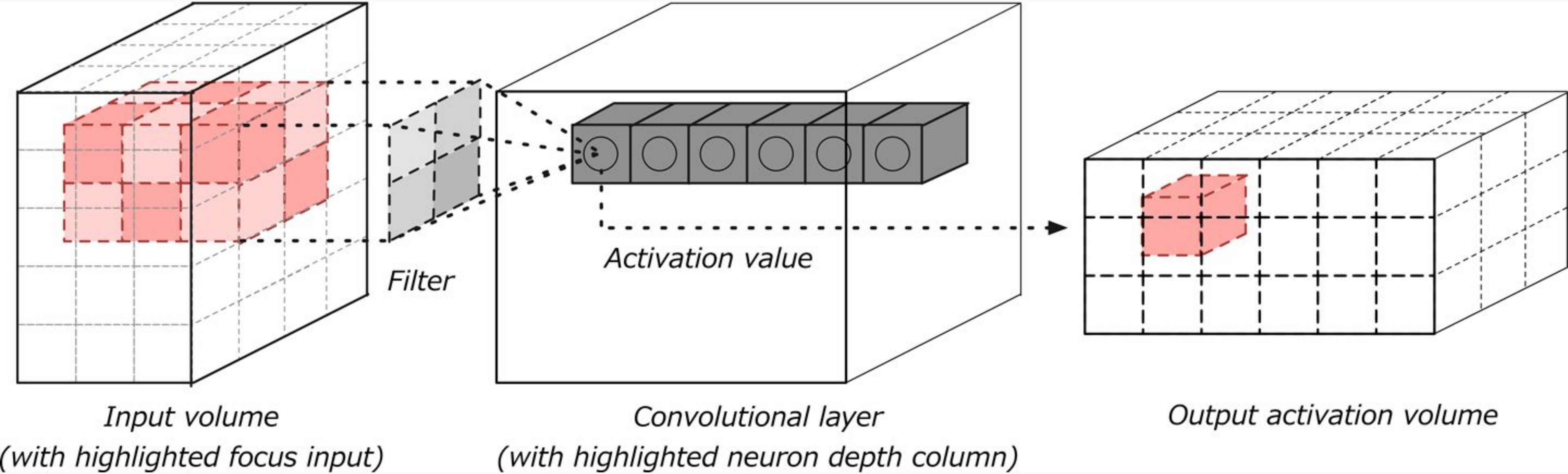
How accurate are we?





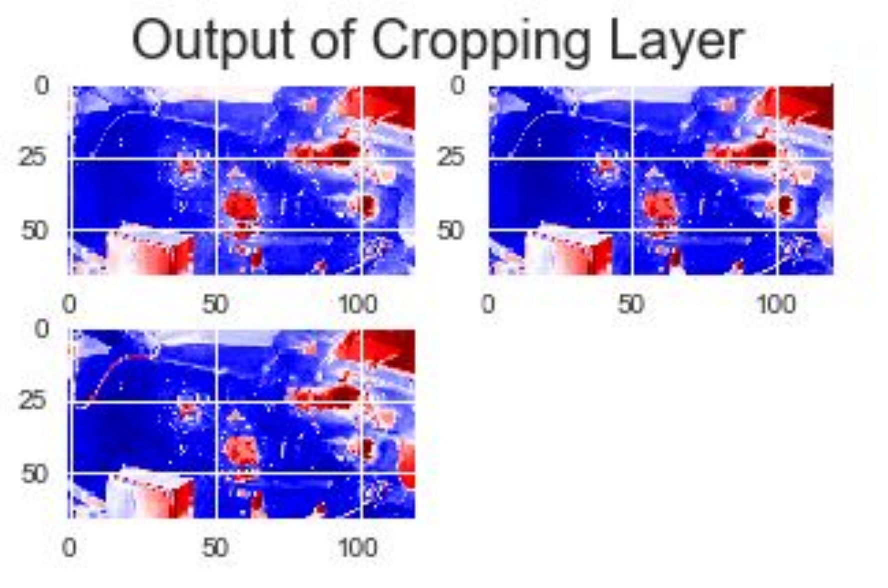
- We visualized our training loss vs. validation loss to help us evaluate the potential for overfitting or underfitting
- We also visualized the accuracy to understand en ensure we are not over fitting, or stopping training too early

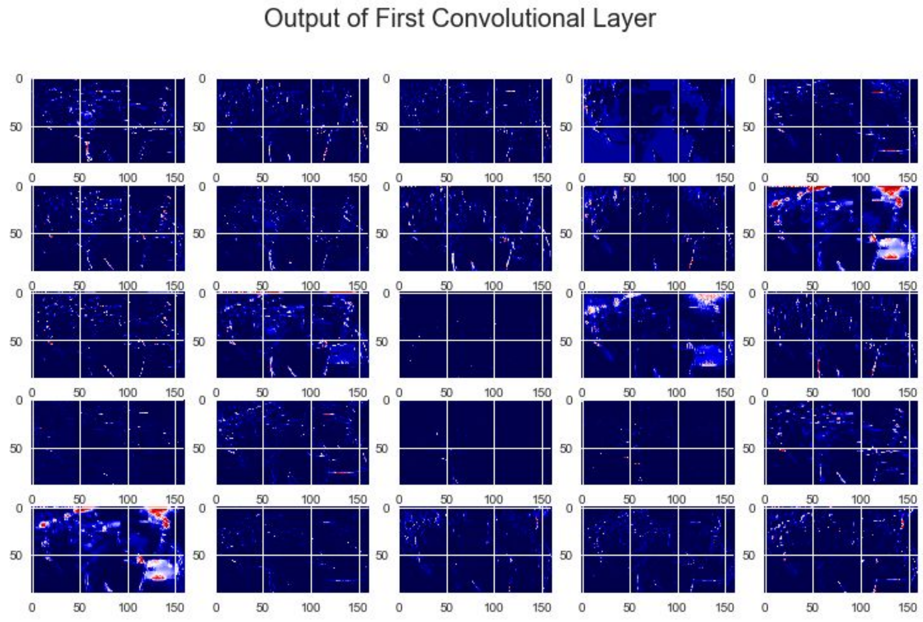
Visualizing the Network: First Convolutional Layer



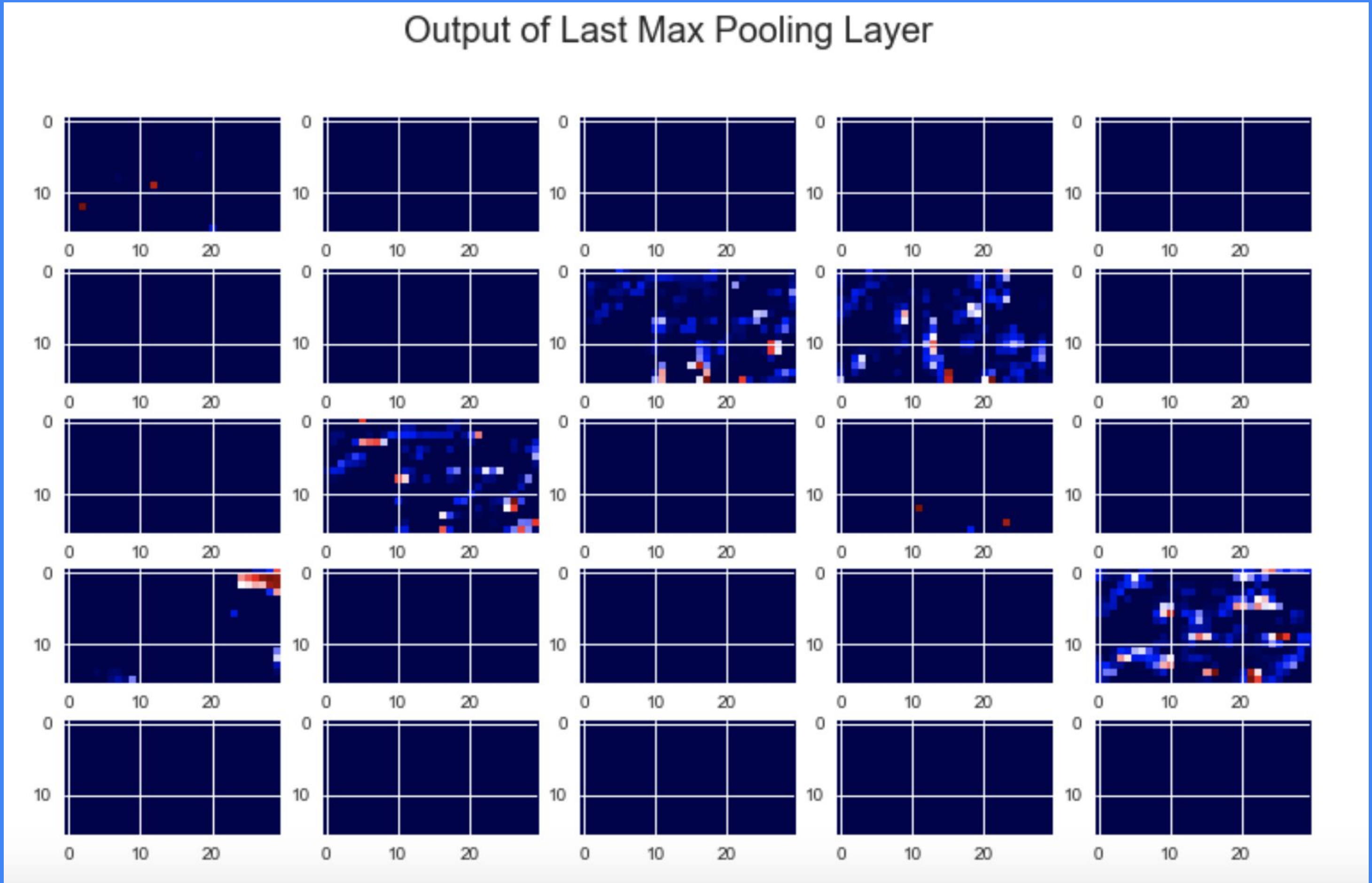
Source: Getting Started with Deep Learning, O'Riely

Visualizing the Network: First Convolutional Layer





Visualizing the Network: Last Max Pooling Layer




Computer Vision helping
to save the oceans