

Plankton Classification using Caffe

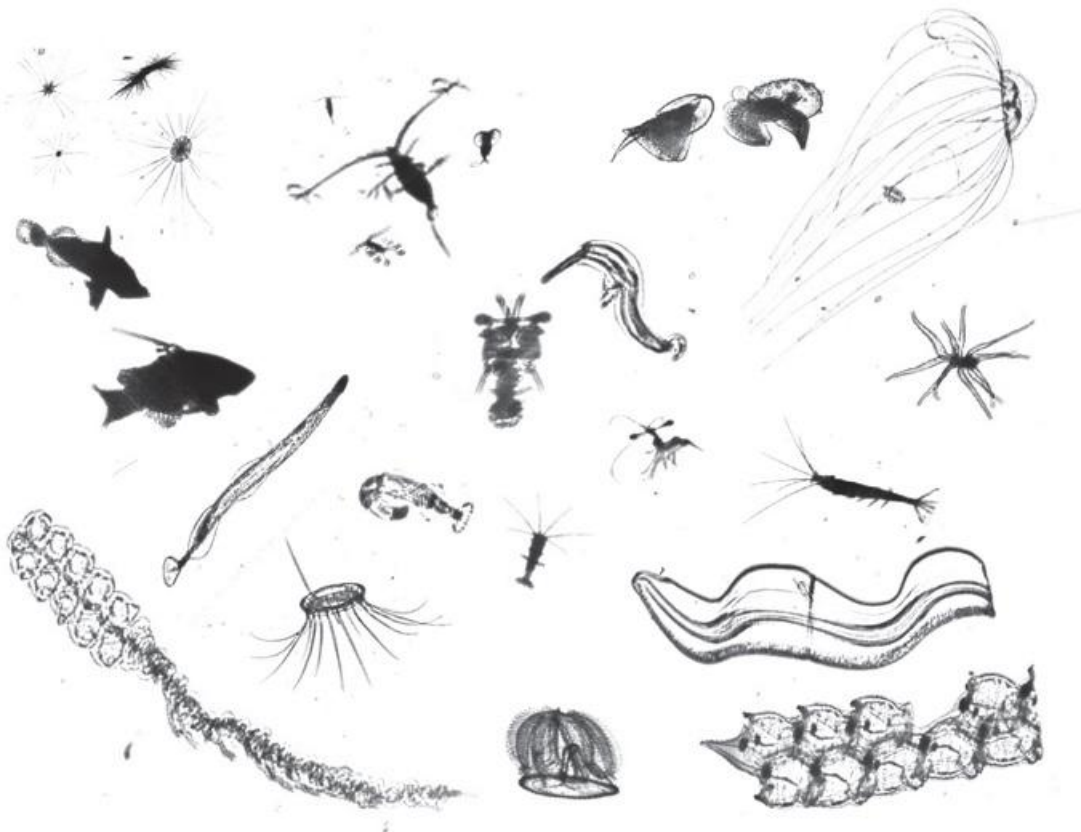
National Data Science Bowl

Dataset

- 30k Training Images
- 130k Test Images
- 121 Classes

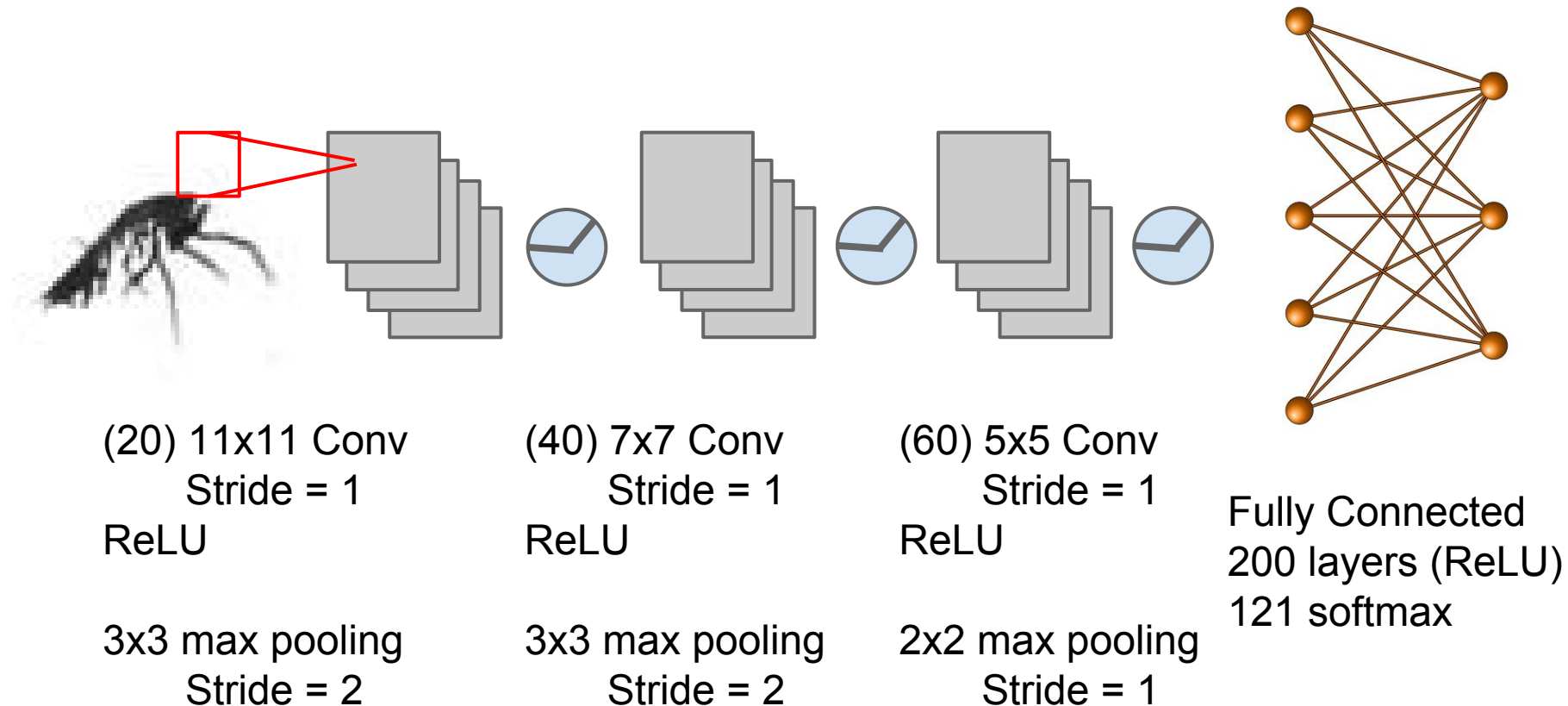
Preprocessing

- Center image and resize to 64x64
- Remove image mean



<https://www.kaggle.com/c/datasciencebowl/data>

Architecture



Architecture in Caffe / Proto

Describe Architecture in Proto

- Layers are described and connected in Proto

Convolutional Layers

```
layer {  
  name: "conv1"  
  type: "Convolution"  
  bottom: "data"  
  top: "conv1"  
  convolution_param {  
    num_output: 20  
    kernel_size: 11  
    stride: 1  
  }  
}
```

ReLU Layers

```
layer {  
  name: "relu1"  
  type: "ReLU"  
  bottom: "conv1"  
  top: "conv1"  
}
```

Max Pooling

```
layer {  
  name: "pool1"  
  type: "Pooling"  
  bottom: "conv1"  
  top: "pool1"  
  pooling_param {  
    pool: MAX  
    kernel_size: 3  
    stride: 2  
  }  
}
```

Architecture in Caffe / Proto

Describe Architecture in Proto

- Architecture and Data described using ~250 line text file

Fully Connected Layers

```
layer {  
  name:    "ip1"  
  type:    "InnerProduct"  
  bottom:  "pool3"  
  top:     "ip1"  
}
```

SoftMax

```
layer {  
  name:    "prob"  
  type:    "Softmax"  
  bottom:  "ip2"  
  top:     "prob"  
}
```

Loss Function

```
layer {  
  name:    "loss"  
  type:    "SoftmaxWithLoss"  
  bottom:  "ip2"  
  bottom:  "label"  
  top:     "loss"  
}
```

Training

Second proto file to describe training

- Adjust learning parameters
- Save model states every N iterations
- Perform tests every M iterations
- Switch between CPU and GPU to train model

Final Accuracy: 61%

```
# The train/test net protocol buffer definition
net: "./kaggle.prototxt"
# test_iter specifies how many forward passes the test
should carry out.
test_iter: 62
# Carry out testing every 500 training iterations.
test_interval: 500
# The base learning rate, momentum and the weight decay
of the network.
base_lr: 0.001
momentum: 0.9
weight_decay: 0.0005
# The learning rate policy
lr_policy: "inv"
gamma: 0.0001
power: 0.75
# Display every 100 iterations
display: 100
# The maximum number of iterations
max_iter: 2000000
# snapshot intermediate results
snapshot: 500
snapshot_prefix: "./arch6/kaggle"
# solver mode: CPU or GPU
solver_mode: GPU
```

Python / Matlab interfaces

- **Can interact with models in Python / Matlab**

Use gpu to do stuff

```
caffe.set_mode_gpu()
```

Load the network

```
net = caffe.Net(network_description_file, model_weights_file, caffe.TEST)
```

...boring stuff

Do a forward pass on all the TEST data

```
output = net.forward()
```

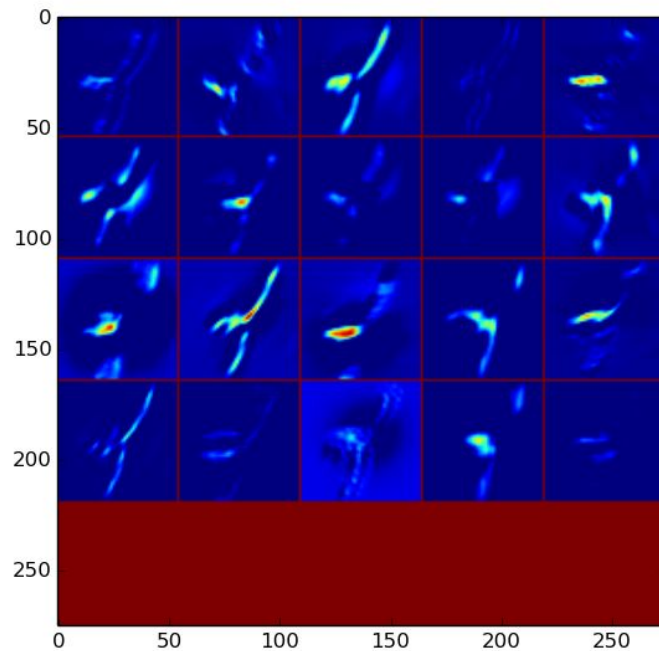
Interact with individual layers

```
probs = output['prob']
```

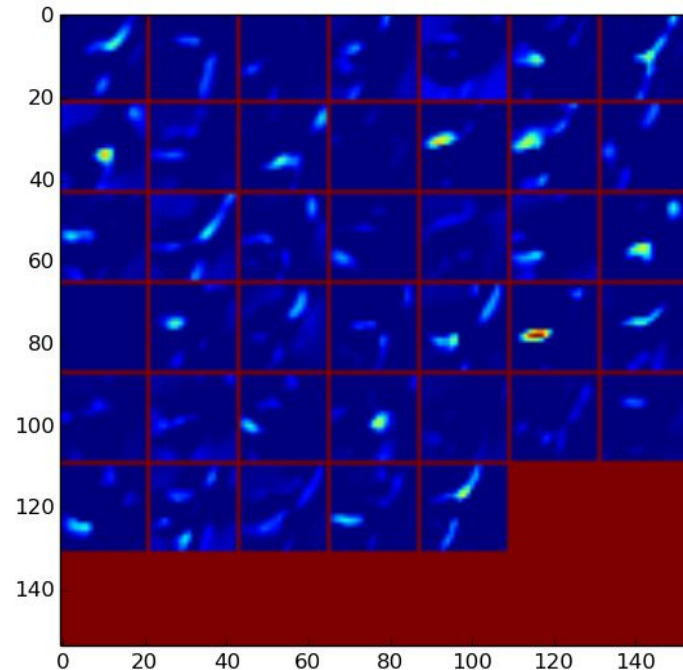
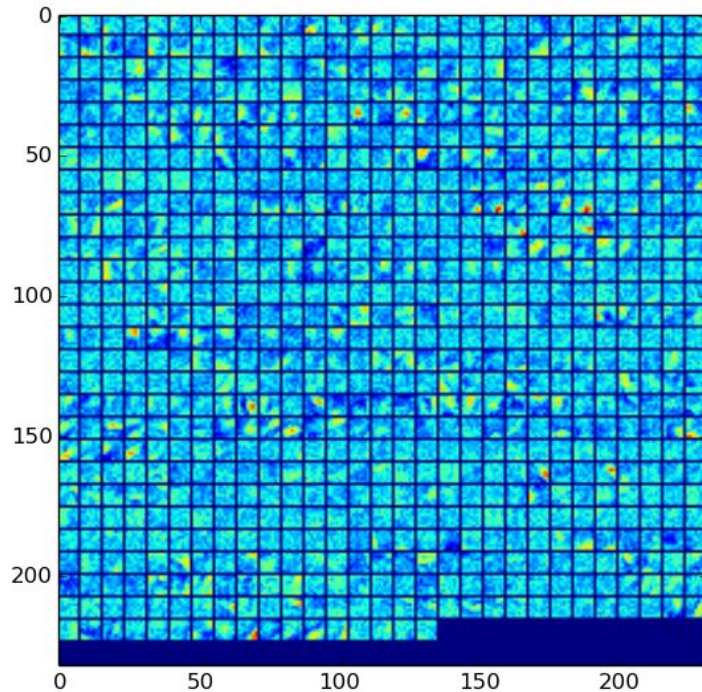
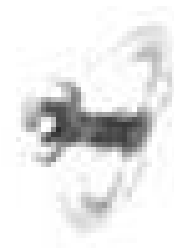
Interact with weight and bias matrices

```
filters = net.params['conv1'][0].data
```

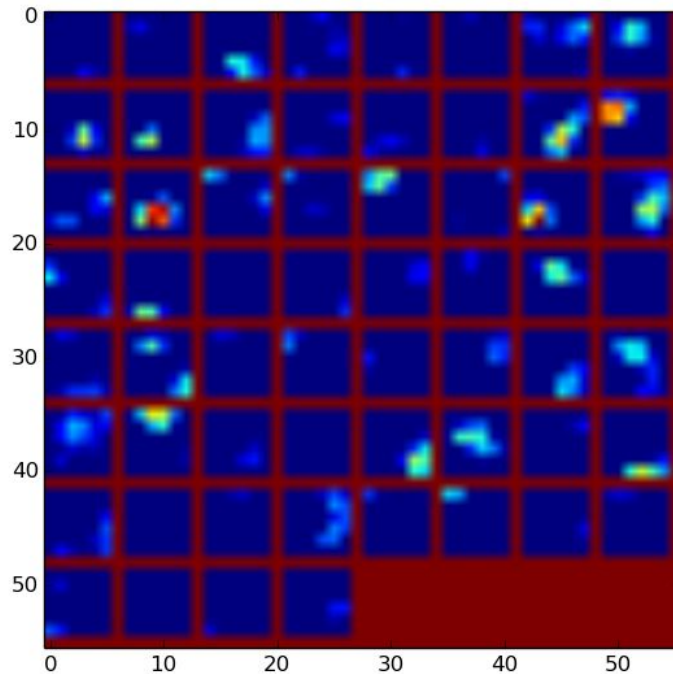
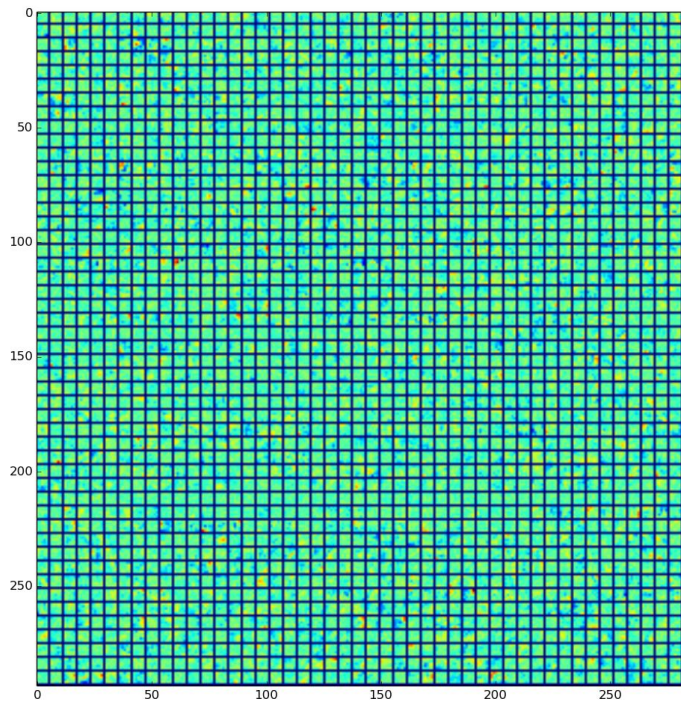
Receptive Fields - conv1



Receptive Fields - conv2



Receptive Fields - conv3



Fully Connected and Softmax

