



# Classifying Plankton Species with Computer Vision and Deep Learning

*An ANC Seminar Talk by*  
**Scott Lowe**

**24<sup>th</sup> March, 2015**

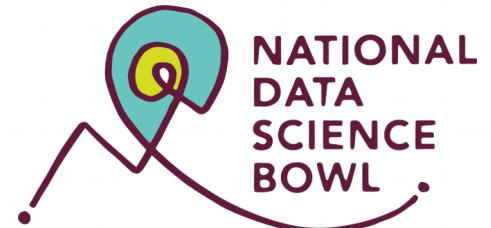


# Why am I studying plankton?!



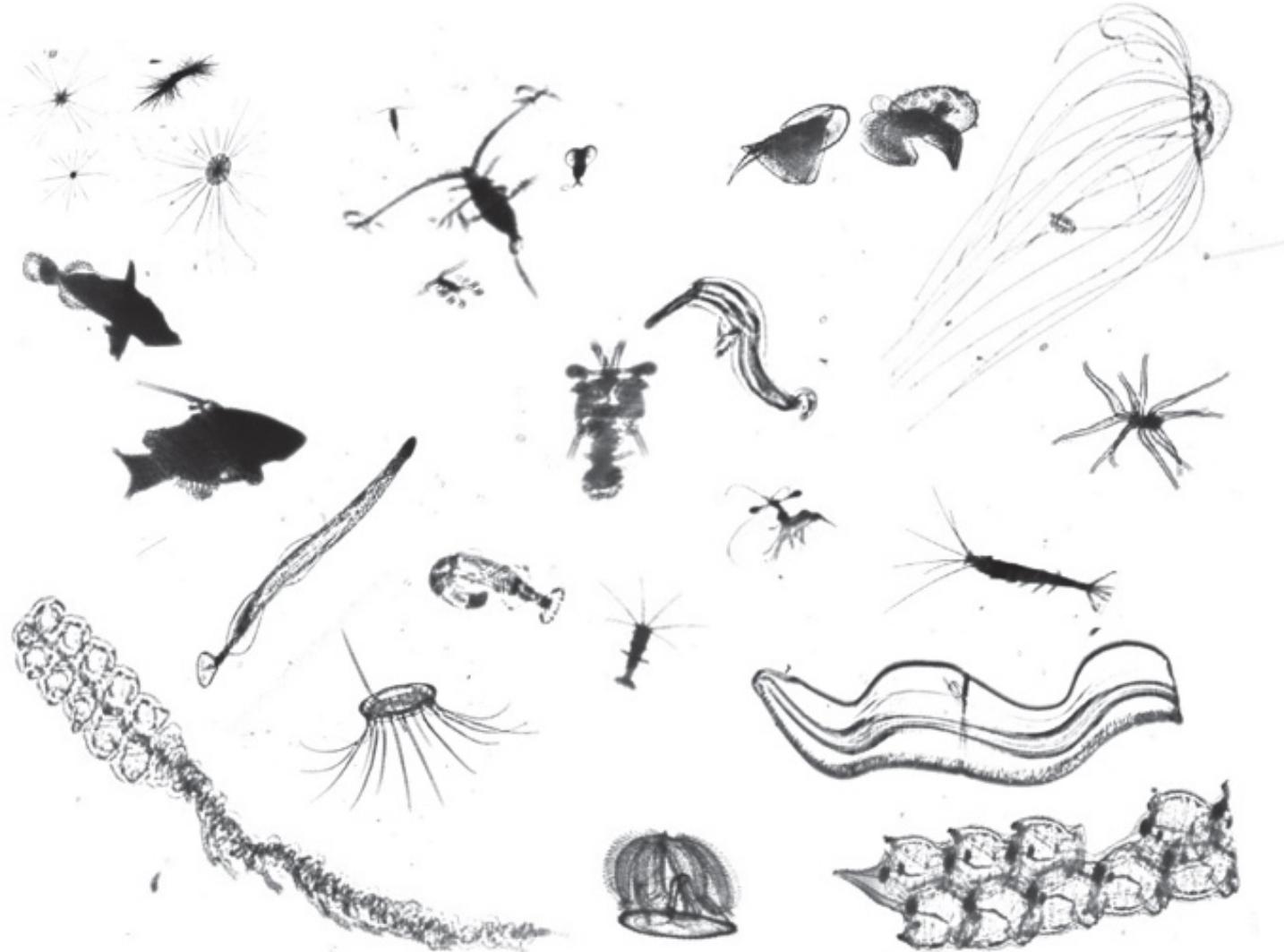
- Kaggle competition, with 1<sup>st</sup> prize \$100,000
- Good way to learn some stuff about Convolutional Neural Networks and Computer Vision
- Fun project
- The team was
  - Matt Graham
  - Gavin Gray
  - Scott Lowe
  - Finlay Maguire
  - Alina Selega
  - Dragos Stanciu

kaggle



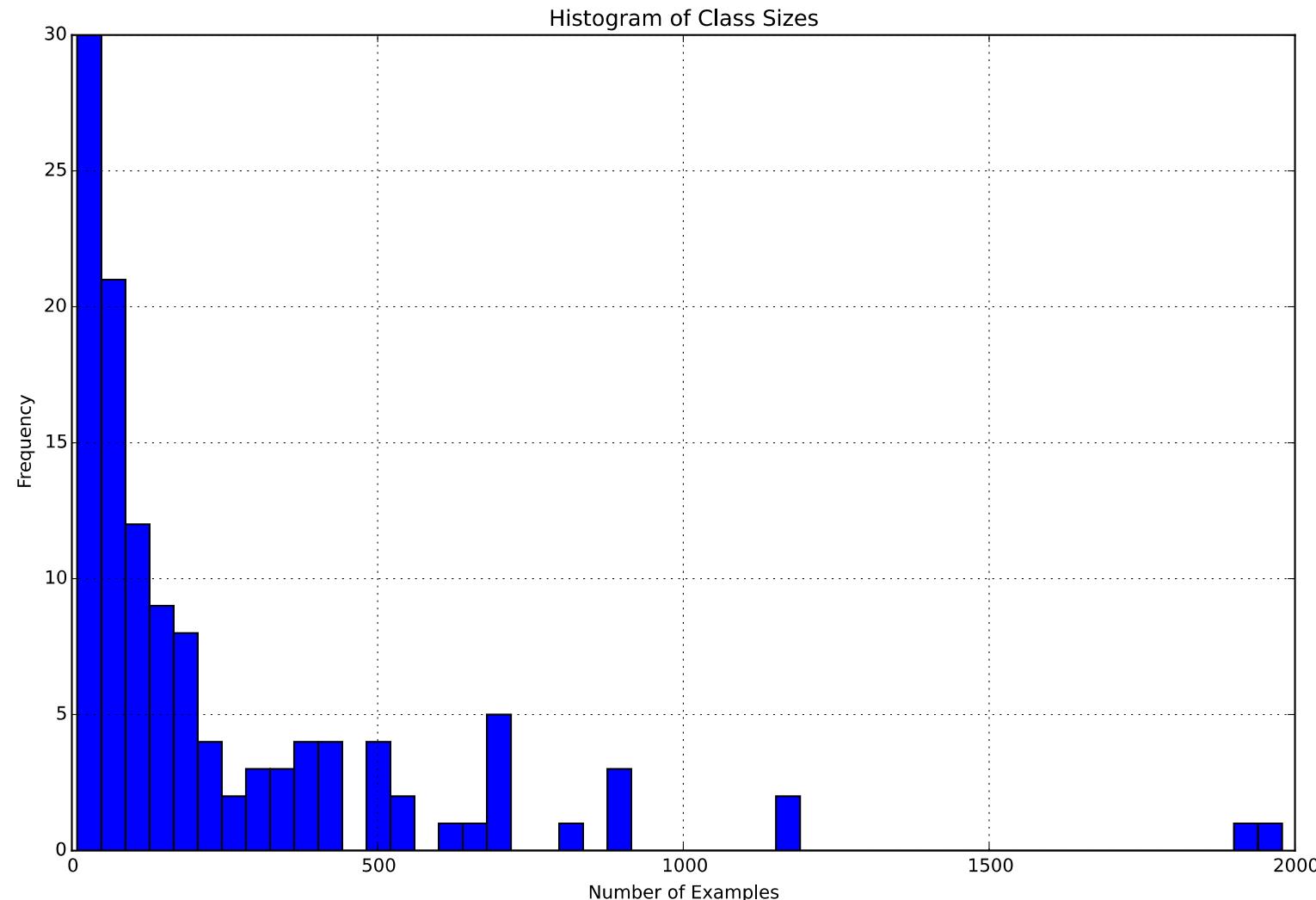


# Dataset examples

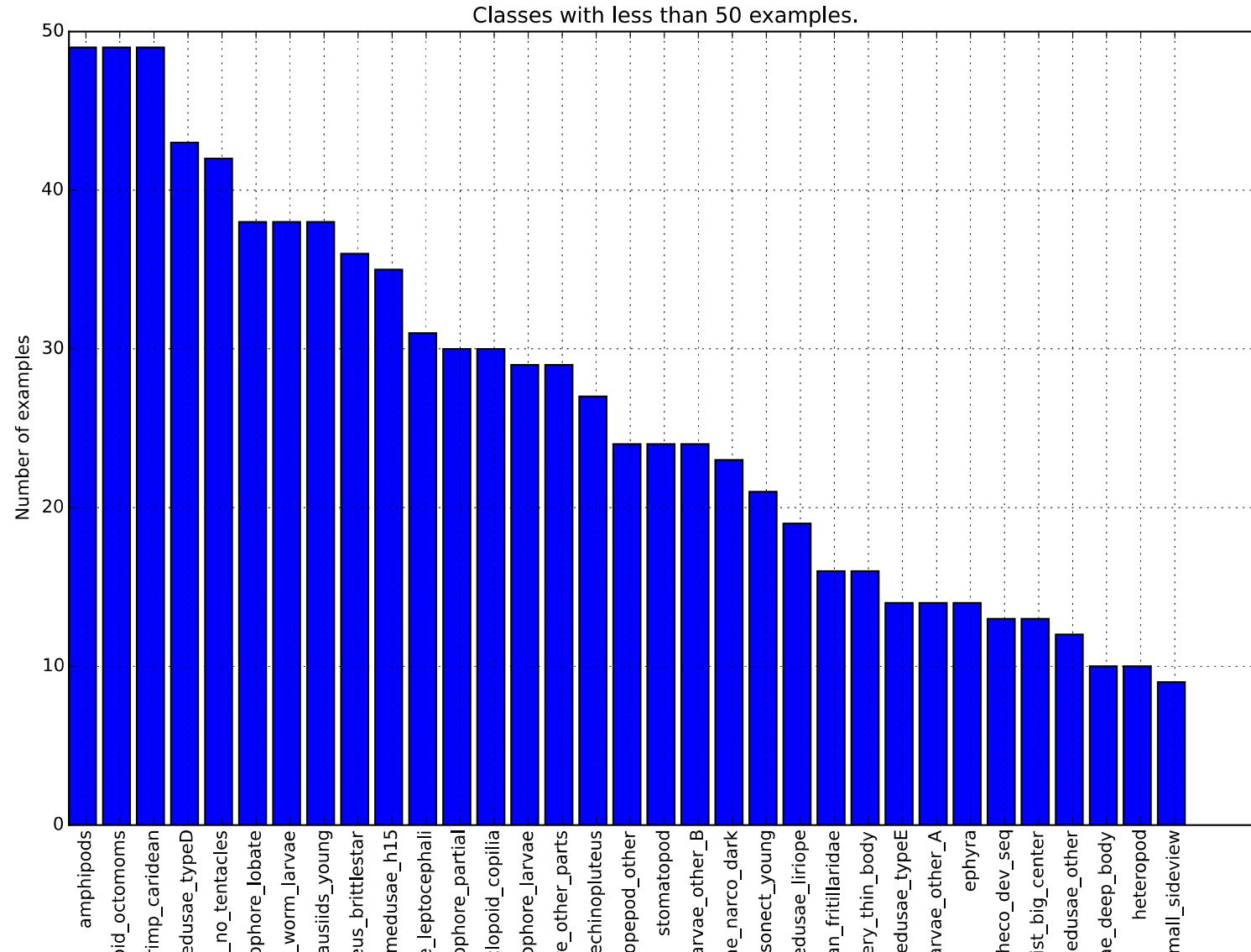




# Distribution of class cardinality

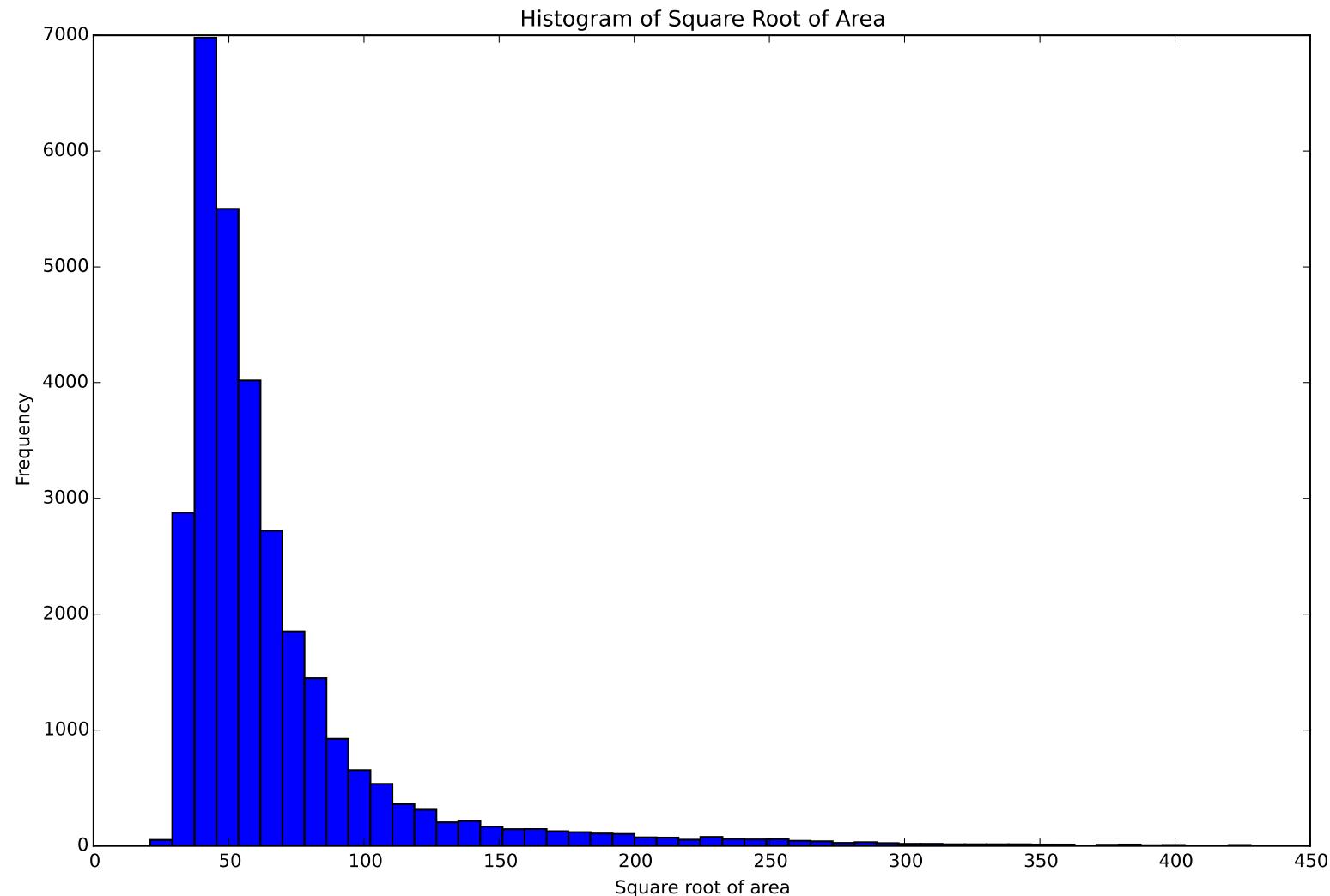


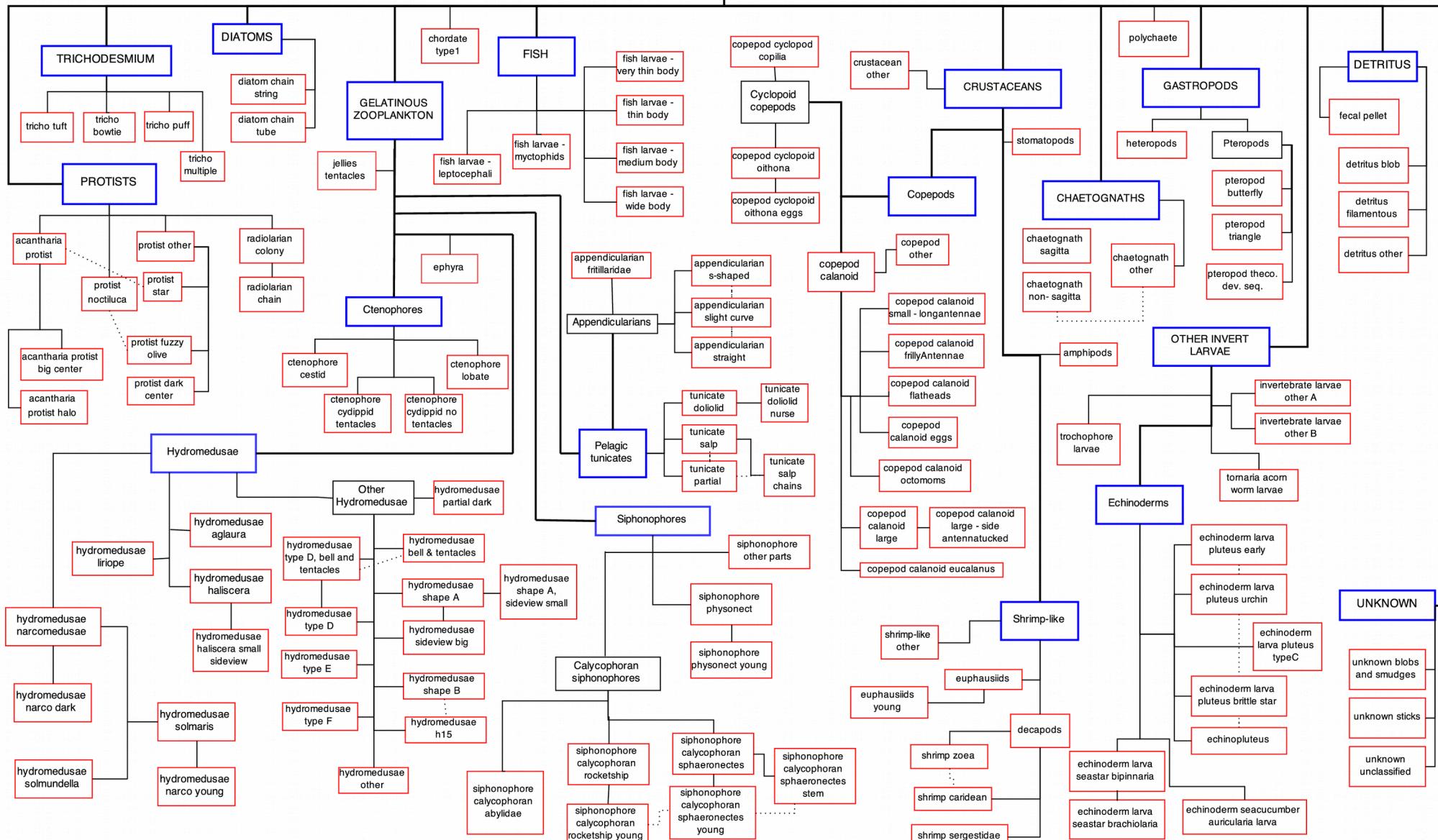
# Distribution of class cardinality





# Distribution of image sizes







# Our approach



- Two general routes for solving image classification problem
  - Convolutional neural networks (“convnets”)
  - Computer vision heuristic techniques
- So combine the two techniques and beat though who use only one!



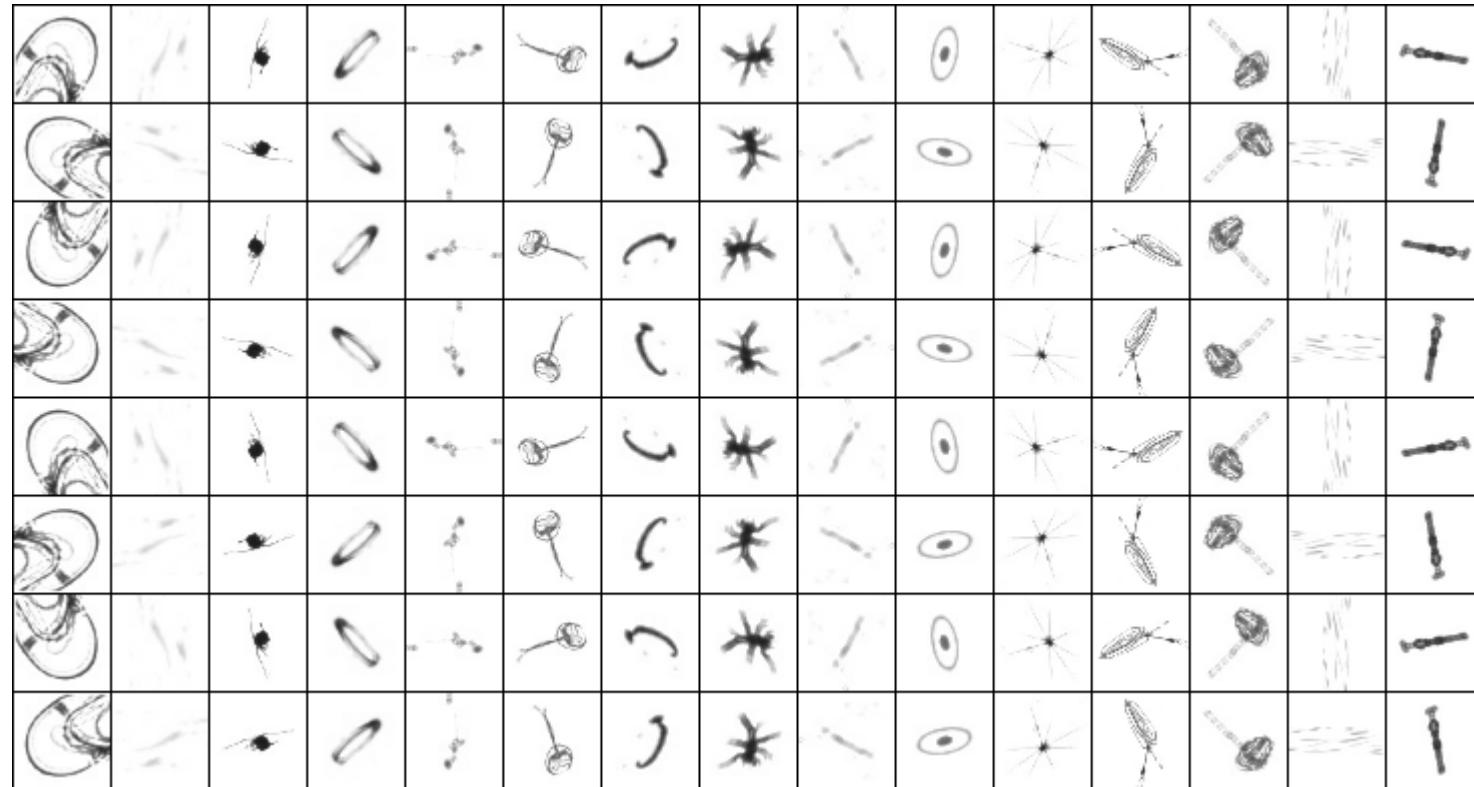
# Offline Image Augmentation



- Offline augmentations
  - Rotation: [0,90,180,270] or [0,45,90,...] degrees, etc
  - Reflection: [no, yes]
  - Crop: [none, 20% top, 20% left, bottom, right]
  - Others implemented but not used
- Constrained by lack of CPU RAM

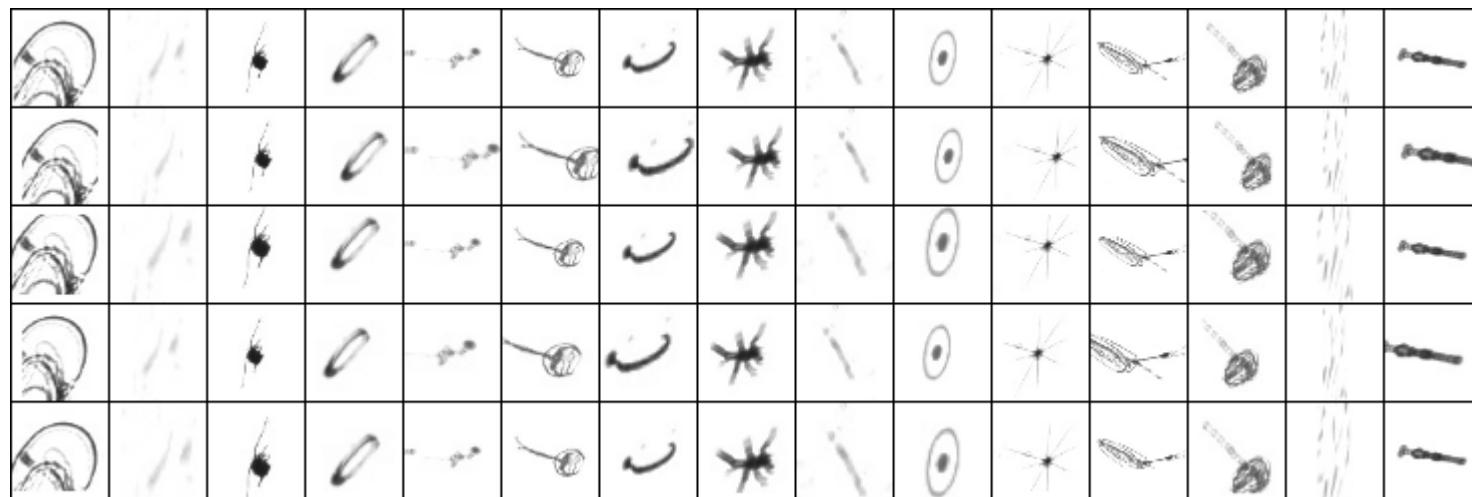


# Offline Image Augmentation





# Offline Image Augmentation





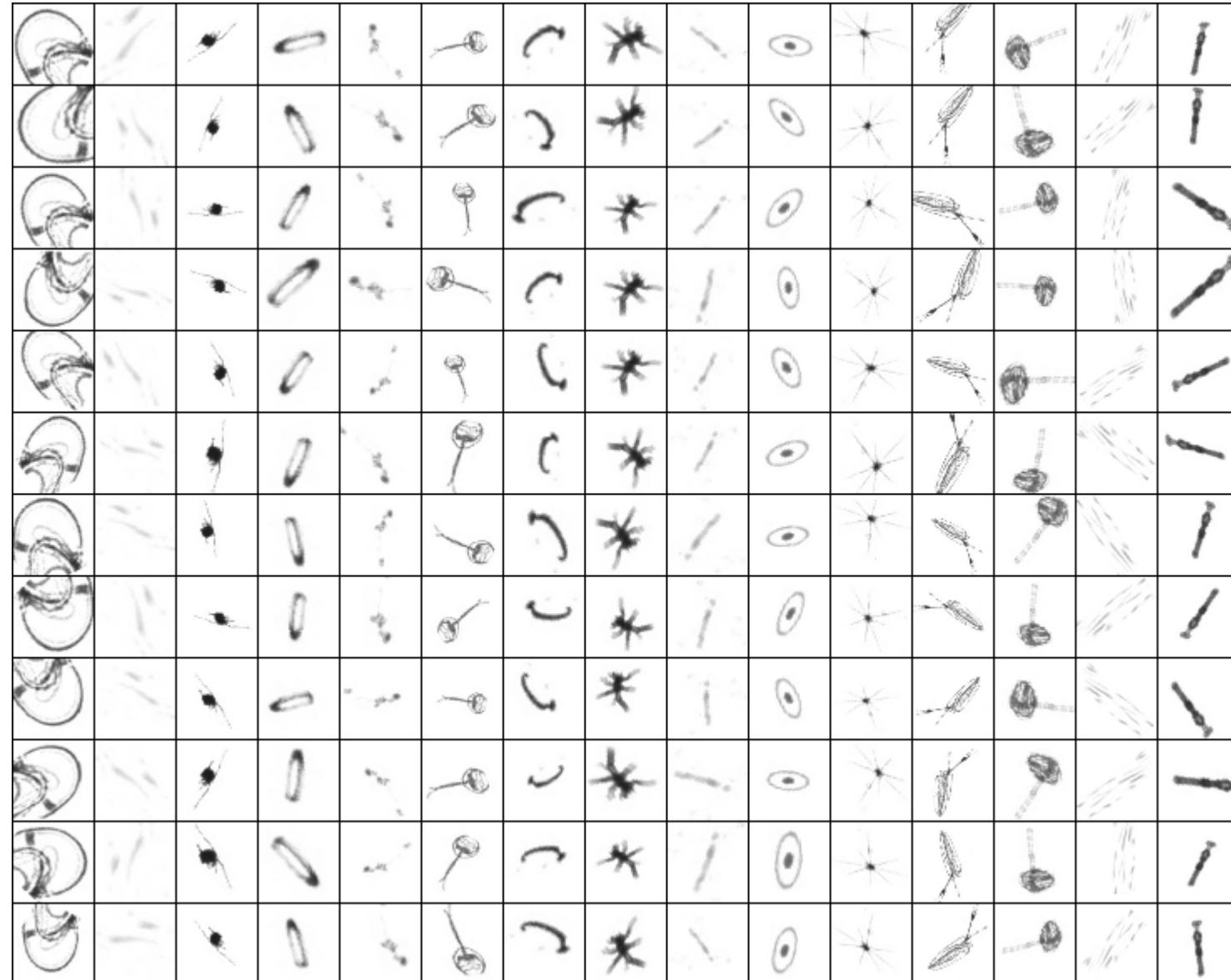
# Online Image Augmentation



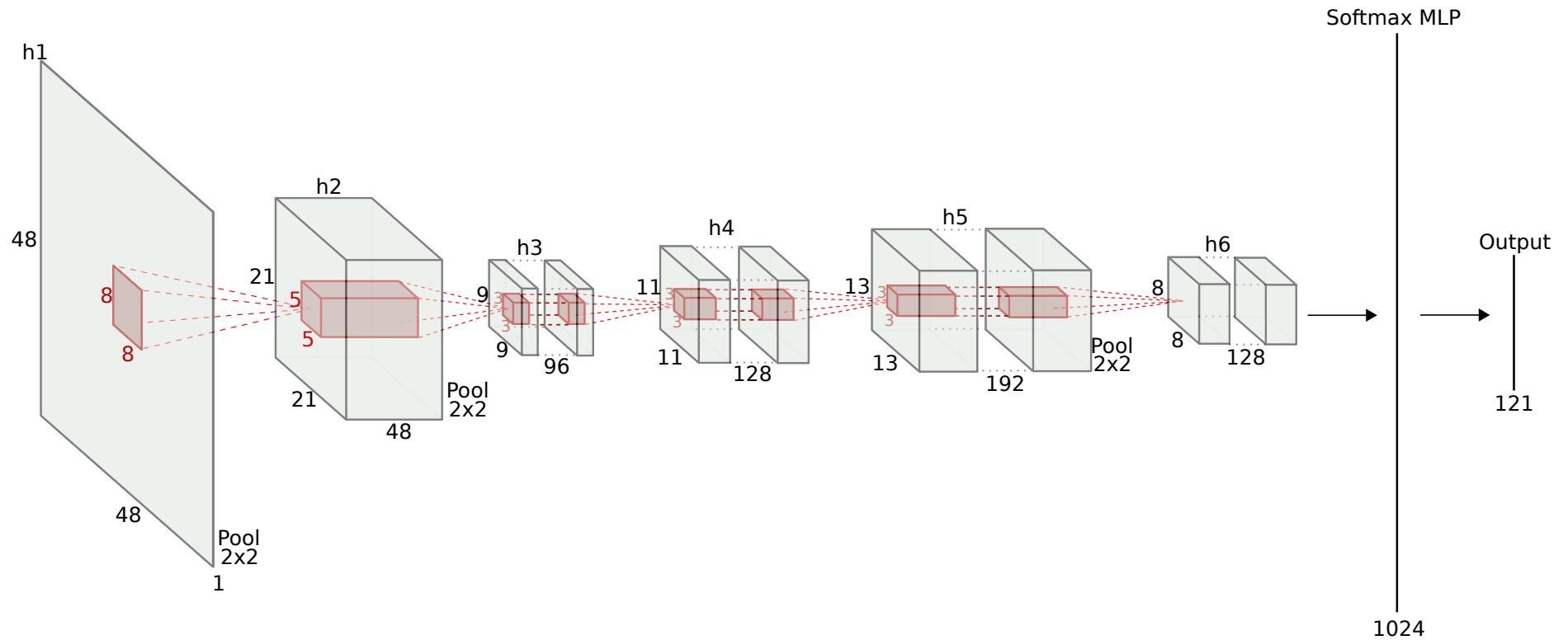
- Online augmentations
  - Rotation: Uniform across  $[0, 360)$  degrees
  - Reflection: Bernouli distributed
  - Translation: Gaussian in x and y,  $N(0\%, 5\%)$
  - Scale: Gaussian  $N(1.0, 0.1)$
  - Stretch: Gaussian in x and y individually,  $N(0, 0.01)$
  - Shear: Gaussian  $N(0, 5^\circ)$
- Augmentations generated for next batch on CPU while training on GPU



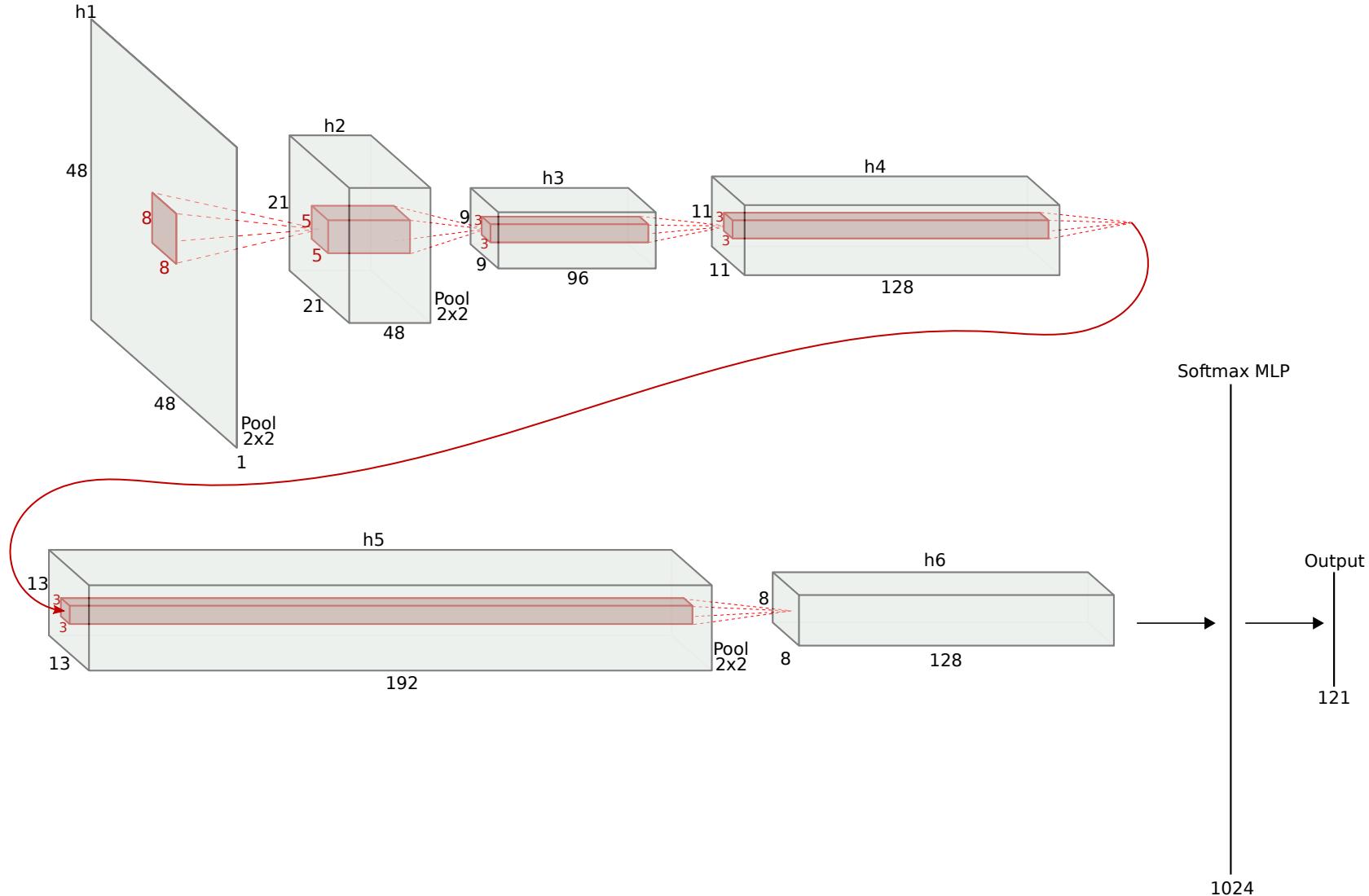
# Online Image Augmentation



# Convnet schema

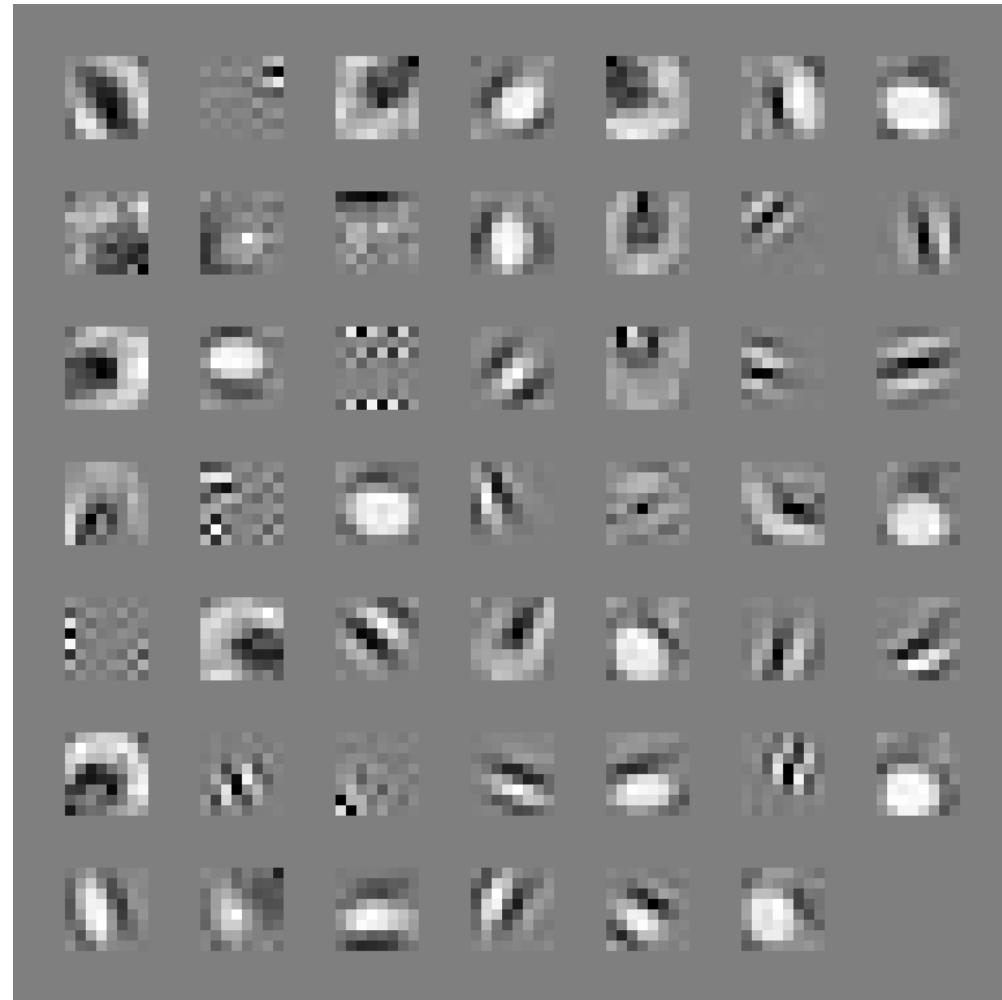


# Convnet schema

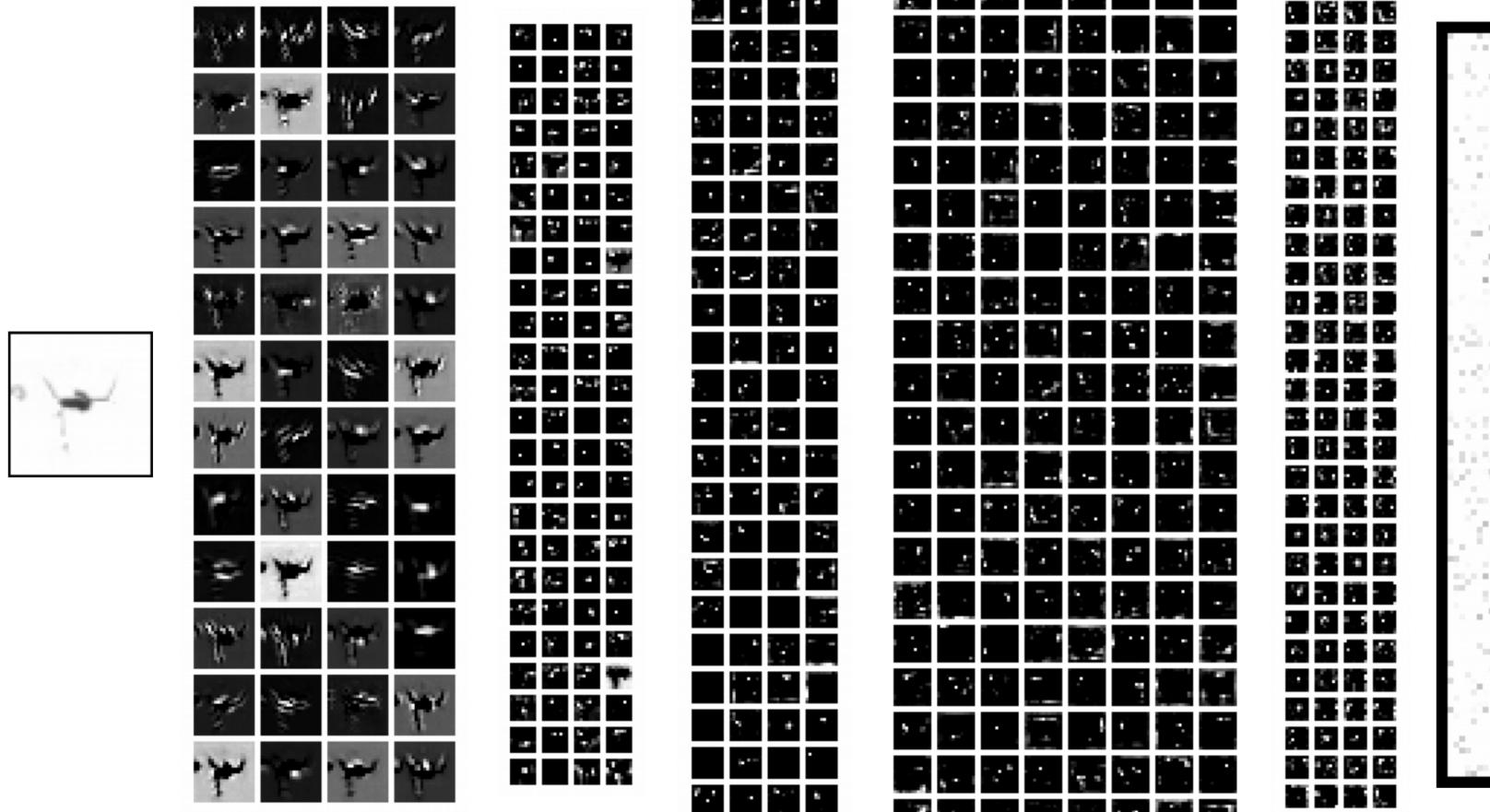




# Initial layer kernels

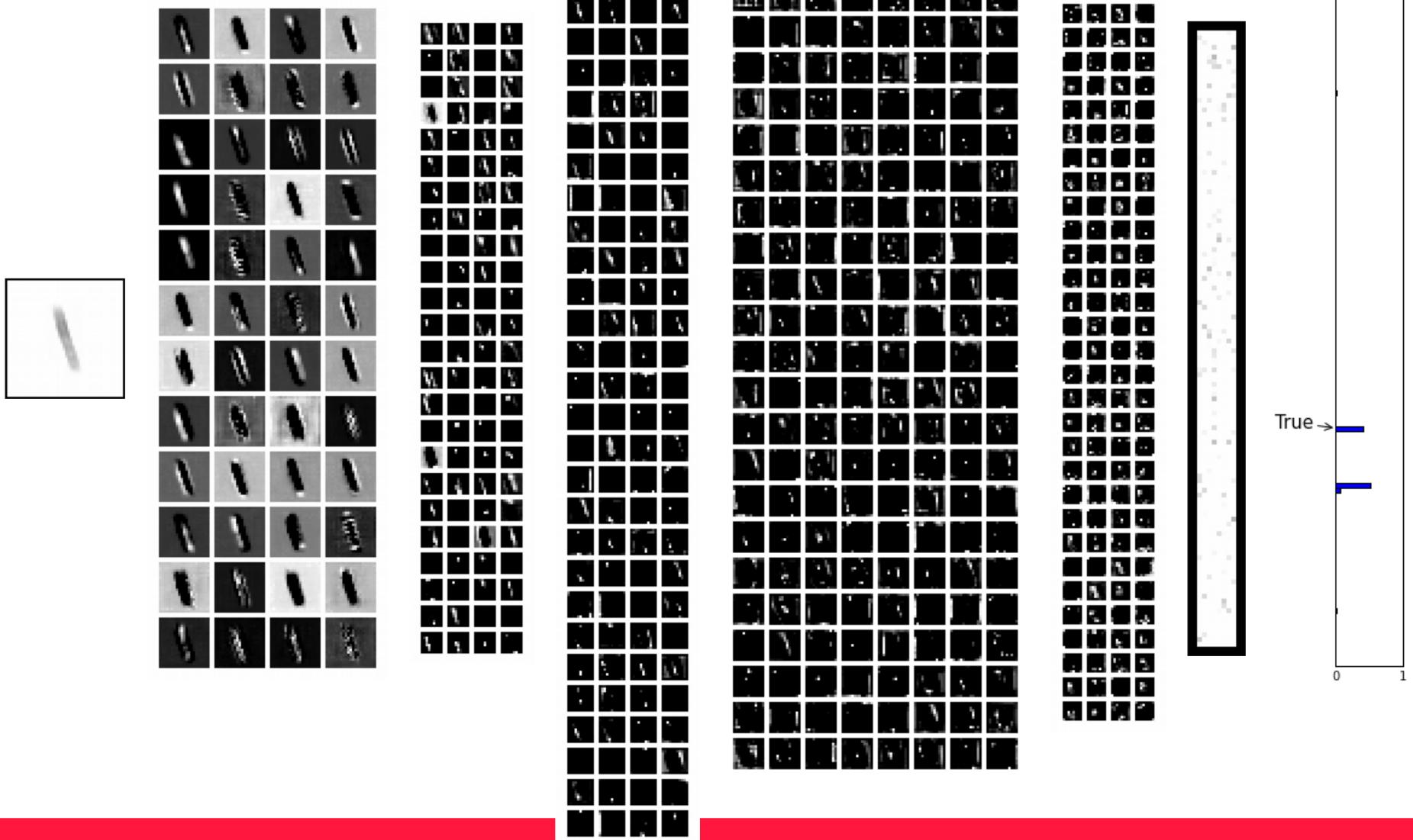


# Example stimulus response





# Example stimulus response

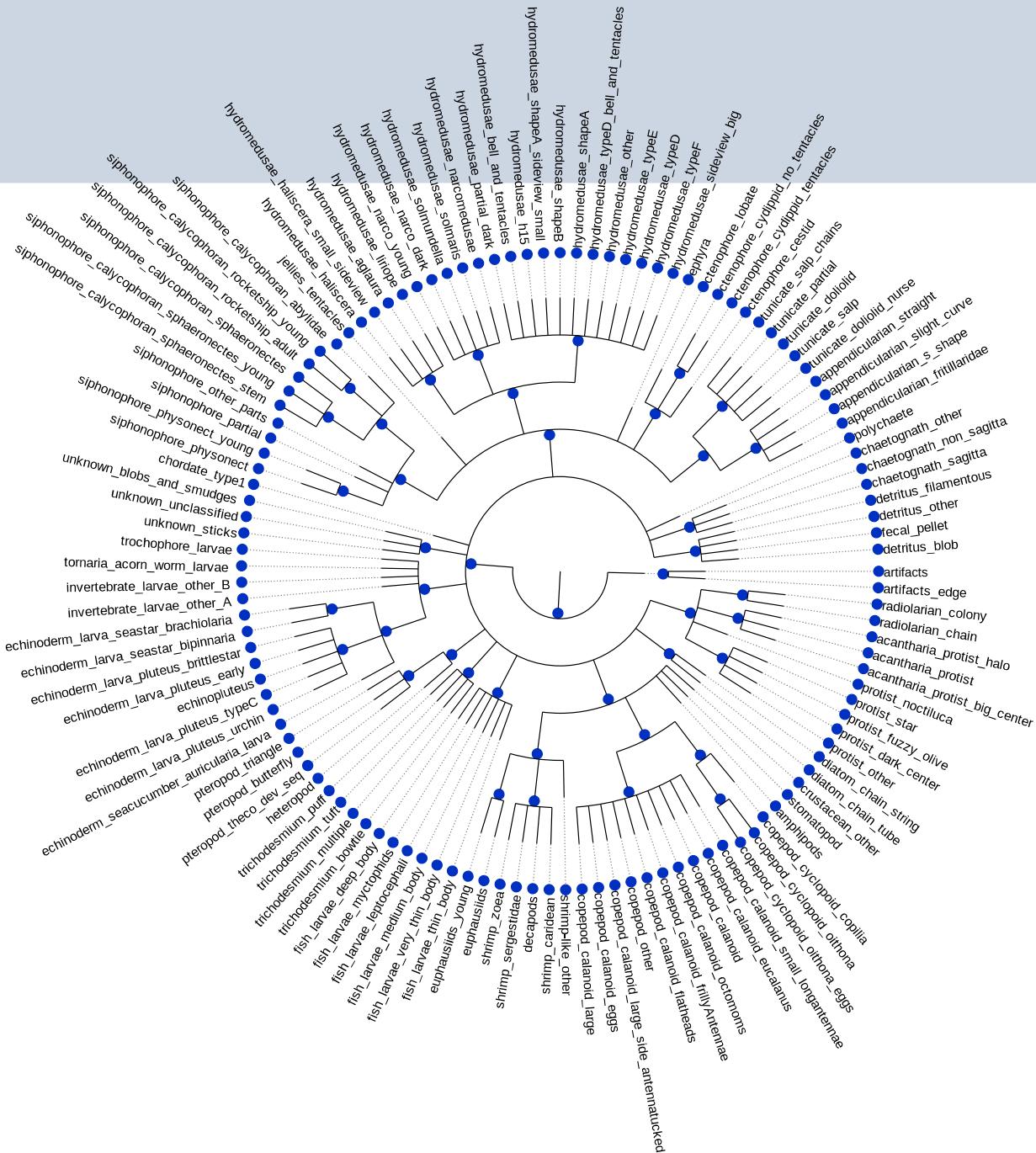


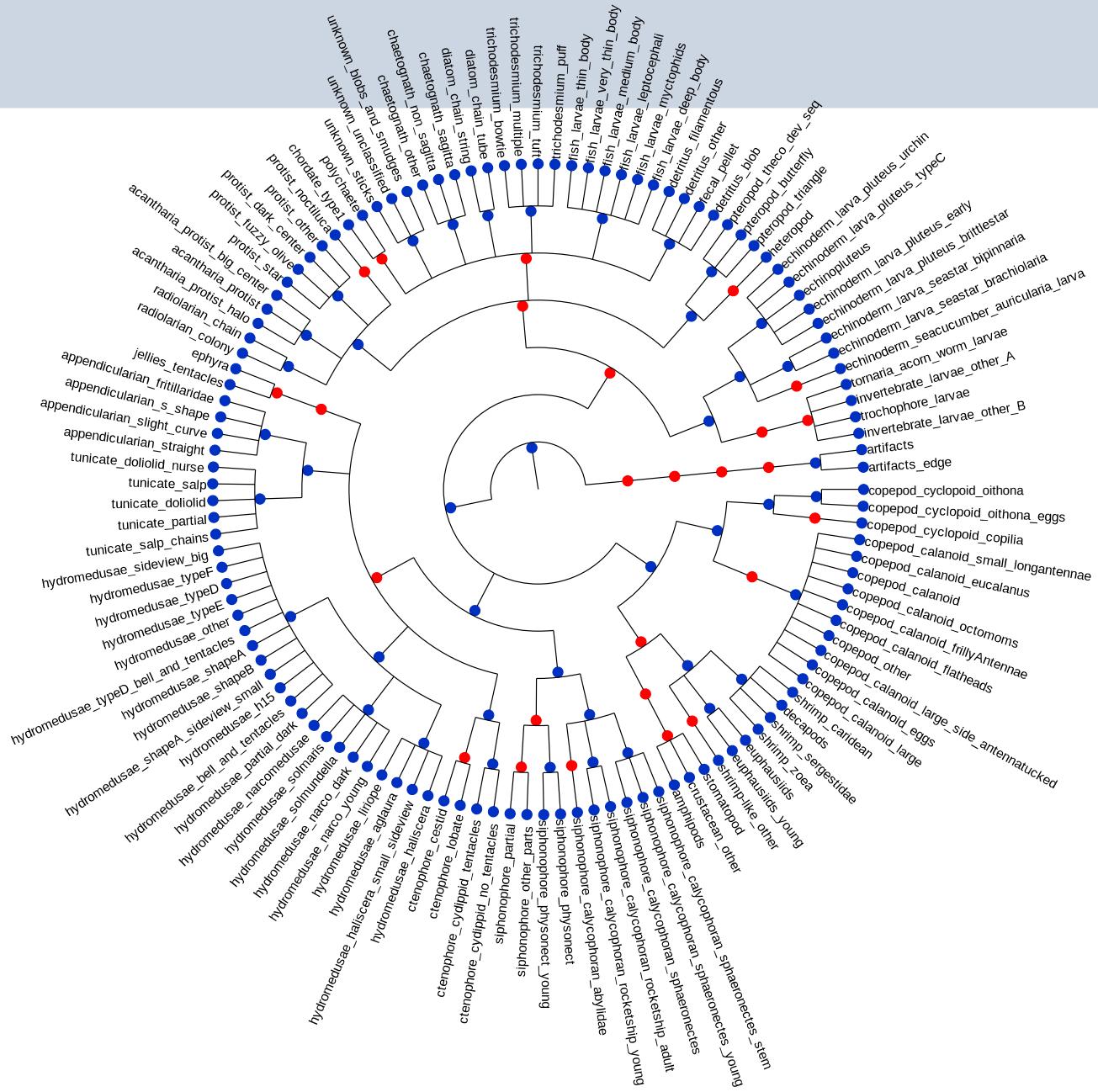


# Hierarchical Modelling

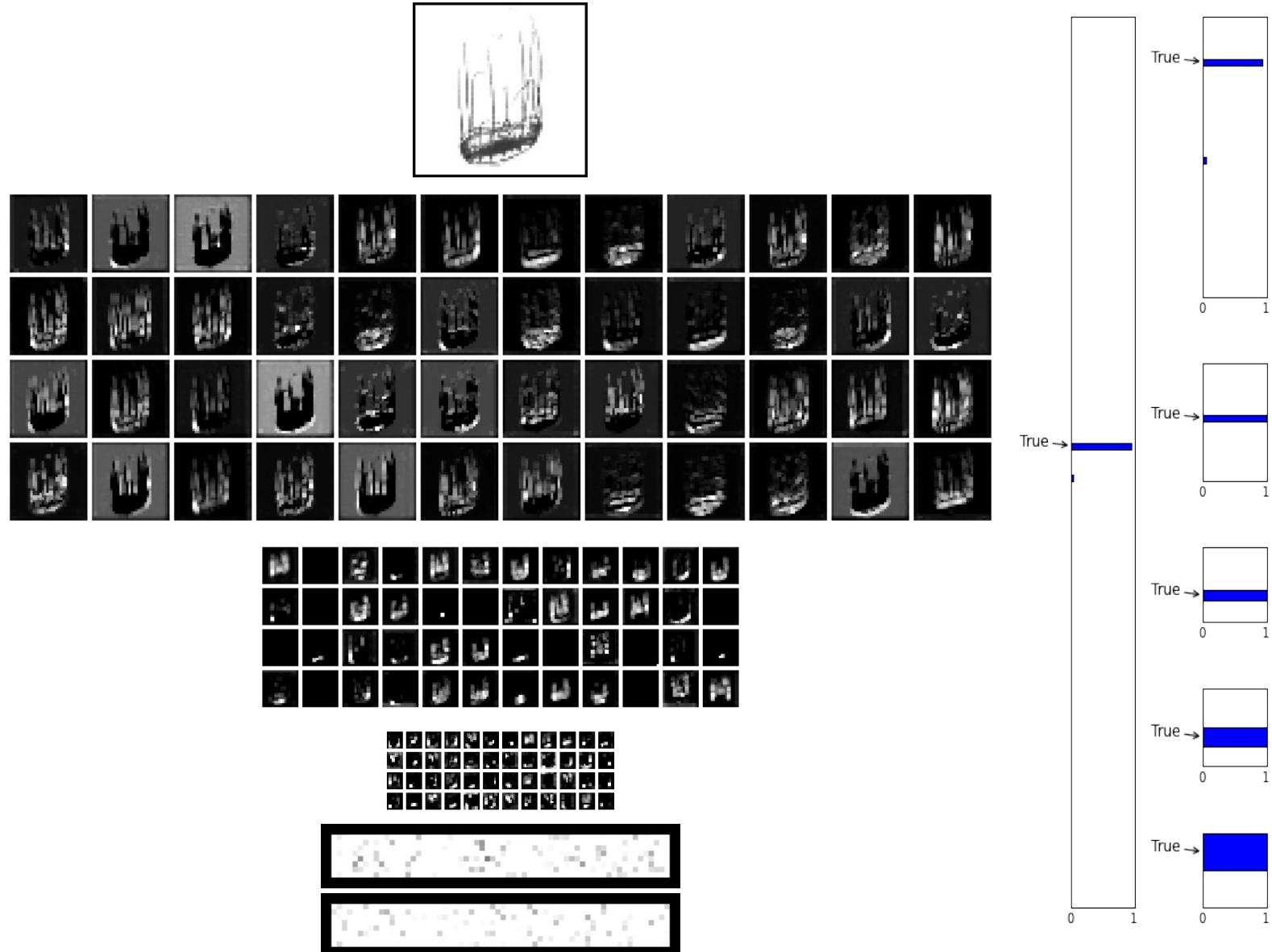


- We can also tell the network about the class structure
- Increase performance by adding this information to the output targets?

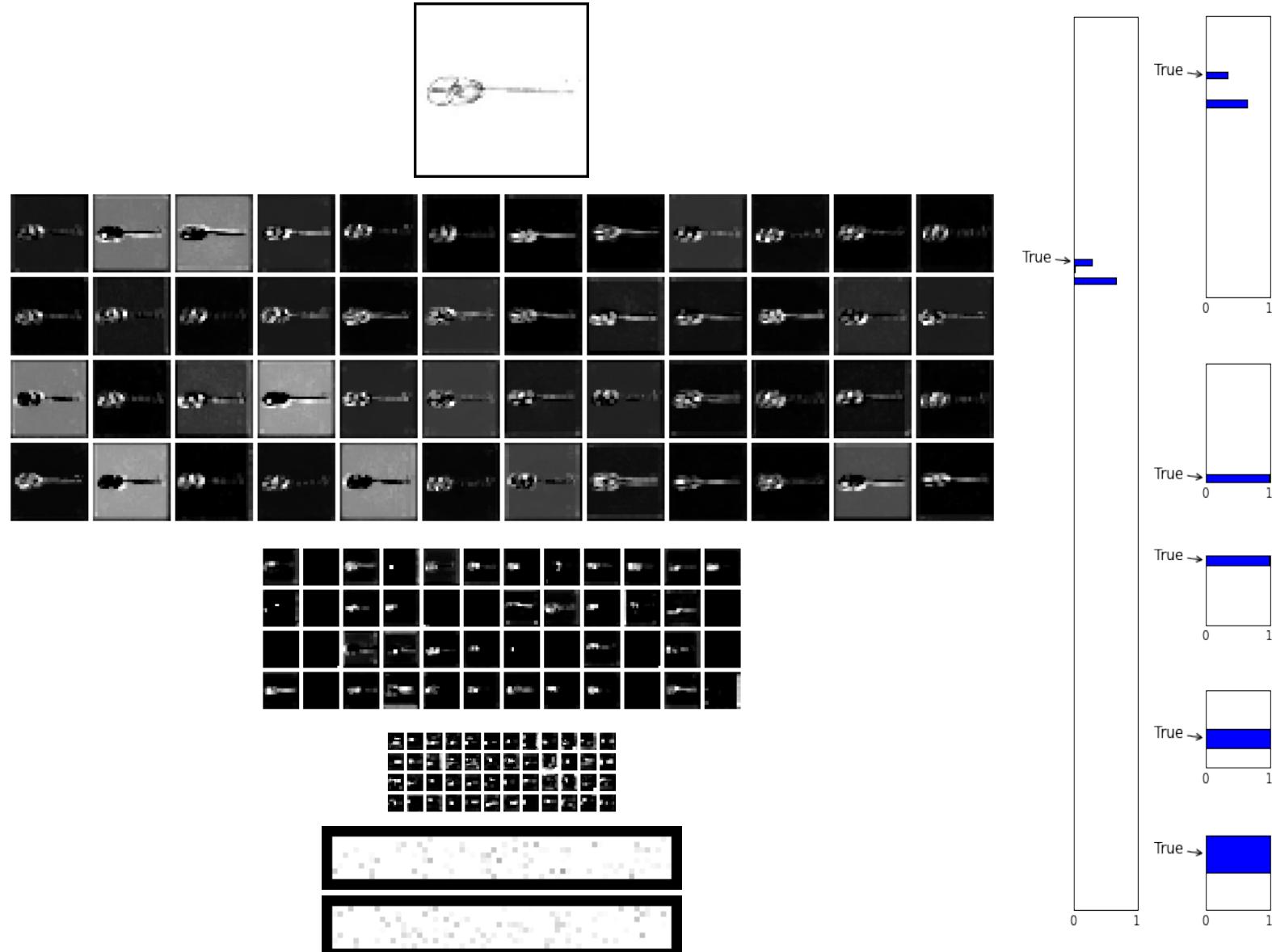




# Example stimulus response



# Example stimulus response





# Things which didn't work, but should have: **Hierarchical class encoding**



- Models trained using the hierarchical structure as the output targets
- Peak performed was similar to models without hierarchical class encoding



# Things which didn't work, but should have: Siamese networks



- Resizing images loses detail in large images, and loses the absolute size of the plankton
- Convnets are frequently applied to patches of a larger image
- Why not feed the image through with and without resizing and show both to the network?
- Did not help - performed worse than single resized input



# Things which really underperformed: CV feature-based decoder



- Detected global texture features
  - Haralick features
  - Grey-Level Co-occurance Matrix properties
  - Contour moments and Hu moments
  - Grey-level intensity histogram (within contour)
  - Parameter Free Threshold Adjacency Statistics
- Local feature detection and description
  - ORB
  - BRISK (but super slow, so not used)
  - MSER
- Fit classifier with RF, SVM or Logistic Regression (similar performances)
- Performance was better with global than local features
- Constructed a hierarchical model, but got no benefit (for global features)
- All models were worse than the simplest convnets



# Things which didn't work, but should have: CV feature integration



- Tried to integrate offline-augmented CV features with convnet
- Joined CV features into the network after convolutions
- Got worse performance than other models



# Final submission



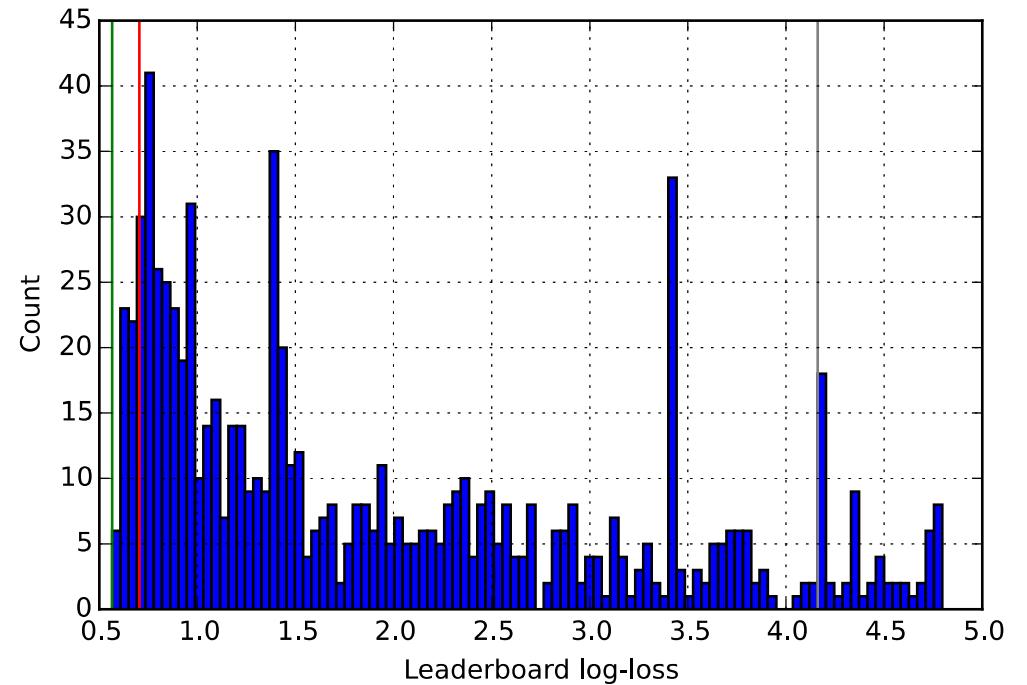
- Average probabilities from lots of augmented versions of test stimulus
- Weighted average of 3 models



# Competition Results



- 57<sup>th</sup> out of 1049 teams (top 5.4%)
- Log-loss 0.704582
  - Winner: 0.565971
  - Uniform distribution: 4.795791
  - Stratified prior: 4.161789





# Toolkit



- Git repository for code sharing on Github
- waffle.io for issue tracking
- Unit tests (but only at the beginning...)
- Tests performed on every push using Travis
- Python 2.7
  - Convnets built with Pylearn2 (Theano under the hood) running CUDA
  - skimage, mahotas, OpenCV
  - sklearn
  - HoloViews for plotting



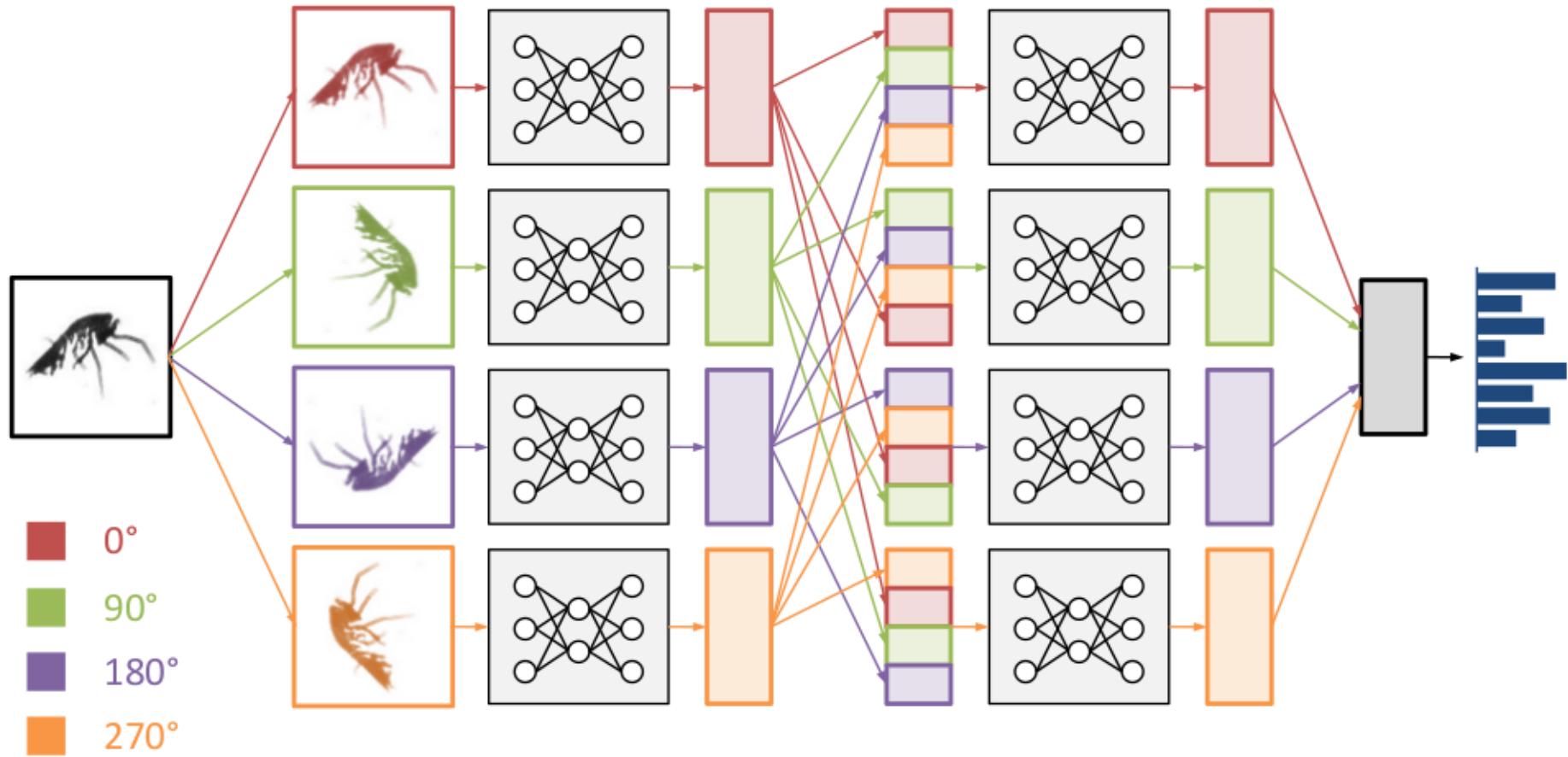
# What did the winning team do?



- Everything we did but better
- More convolution layers (10), and smaller convolutions
- Cyclic pooling & maxing across 4 rotations simultaneously
- Leaky rectified linear units



# Winning team's Cyclic pooling





# Conclusions



- Unit testing is essential to know your code works the way you think it does
- Running experiments testing only one thing at a time is essential
- Pylearn2 is poorly documented and probably not work using
- Convnets more powerful technique than Computer Vision feature extraction
- Hierarchical modelling did not work for us



# Acknowledgements



- The team was
  - Matt Graham
  - Gavin Gray
  - Scott Lowe
  - Finlay Maguire
  - Alina Selega
  - Dragos Stanciu
- Our thanks to Bob Fisher, Amos Storkey and Chris Williams for their helpful advice.



# What next?



- Diabetic Retinopathy Detection challenge on Kaggle, could be tackled with a similar approach. Prize is \$50,000