Classifying radio galaxies with deep learning

Vesna Lukic, Marcus Brüggen
Hamburg Observatory,
University of Hamburg

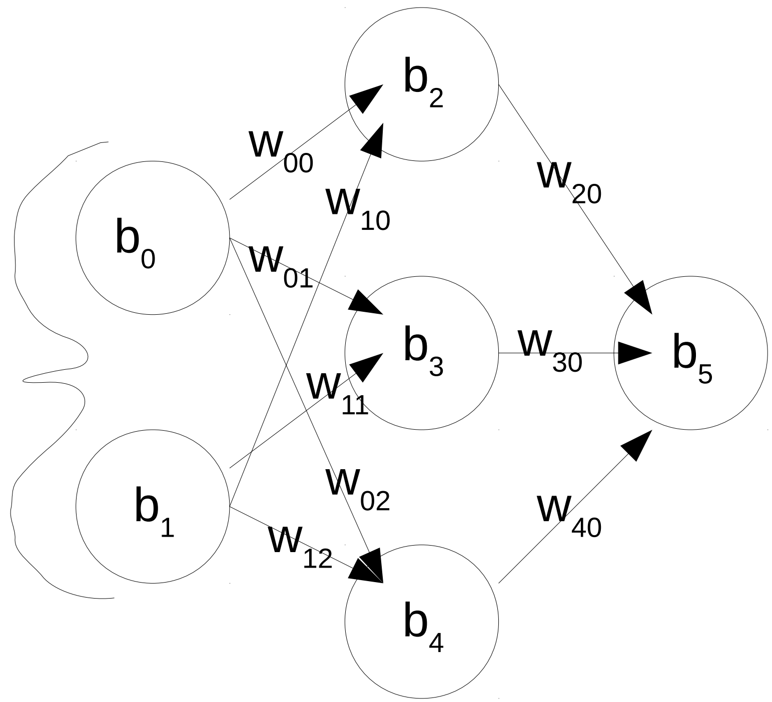
Background

- Galaxy classification is important as it allows us to learn more about the evolution of the universe
- Manual classification is time-consuming and tedious
- Aim to achieve automatic classification (machine learning)
- This has mainly been done with optical galaxy images, eg Kaggle Galaxy Zoo

Neural network basics

1. Input x

This sets input layer activations



3. Error in final layer

$$\delta^L$$

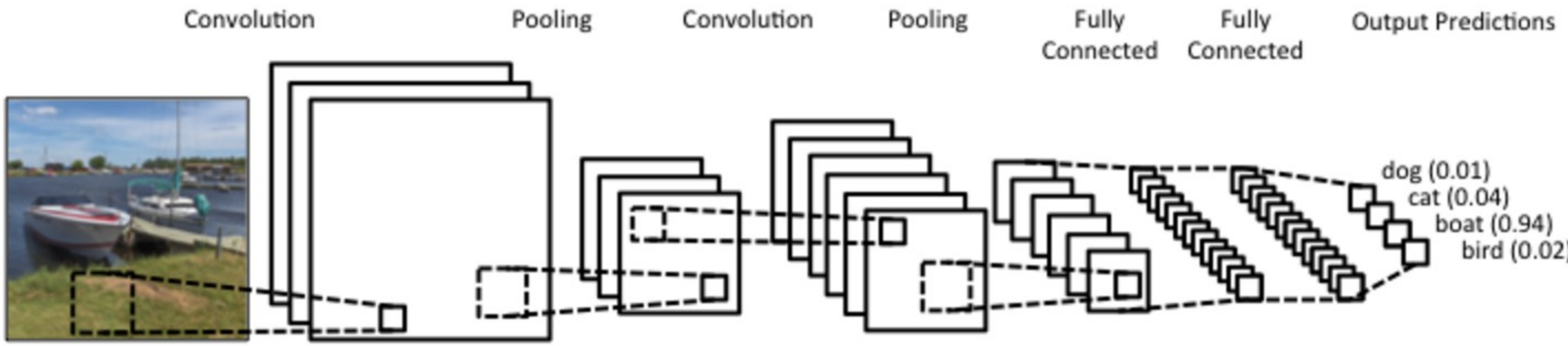
Backpropagate this error back through all layers

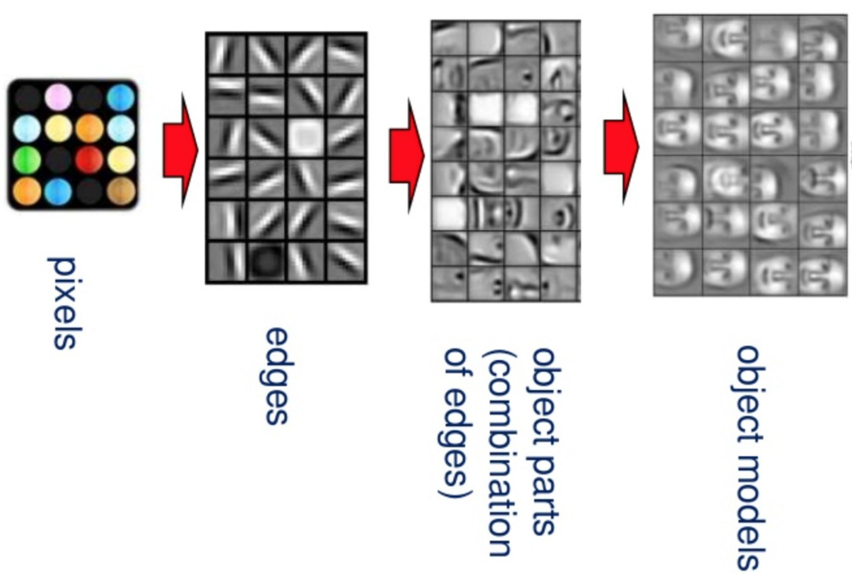
2. Feedforward through the network

4. Calculate $\frac{\partial C}{\partial w}$ $\frac{\partial C}{\partial b}$

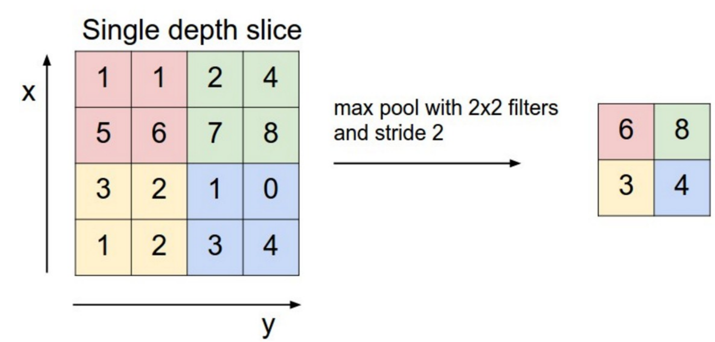
5. Update w and b

Convolutional neural networks





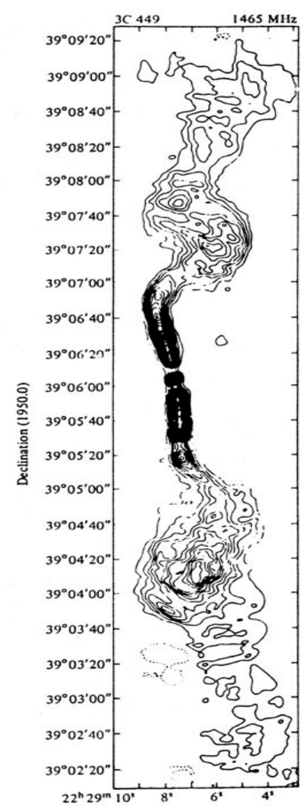
Pooling

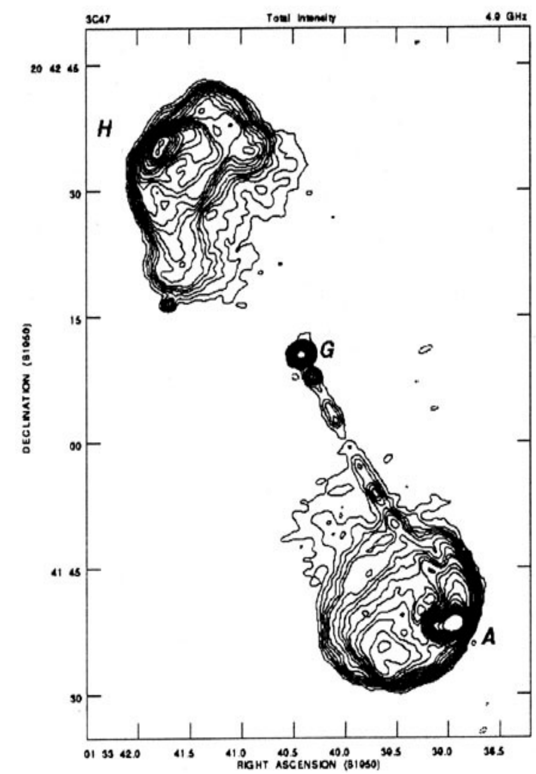


http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

Deep learning with radio galaxy images

- Motivation is to find meaningful classifications for radio galaxies
  - FRI, FRII (Fanaroff-Riley)
  - Double-double
  - Bent-tailed





Radio Galaxy Zoo (RGZ) data provided

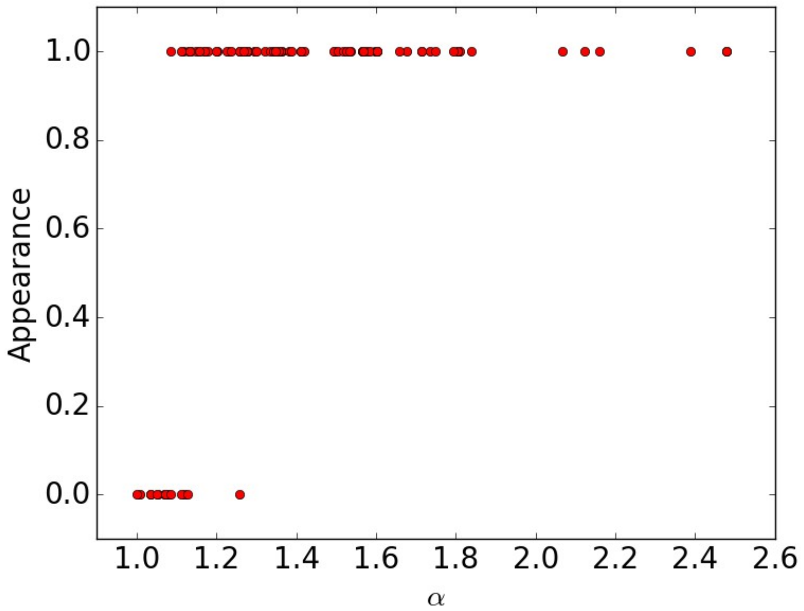
- Have been given access to image data of 206399 galaxies, from fits files
- No label data provided – needed to generate labels
- Single channel
- Typically (132,132) pixels
- Images contain different numbers of sources
- Used PyBDSF (Python Blob Detector and Source Finder) to help organise the data
- Successfully processed 175454 images

Classifying between point and extended sources

- First see if deep learning algorithm can distinguish between point sources and extended sources

Source type	PyBDSF folder	# Sources
Point source	One source*	18716
Extended source	> Three sources	17999

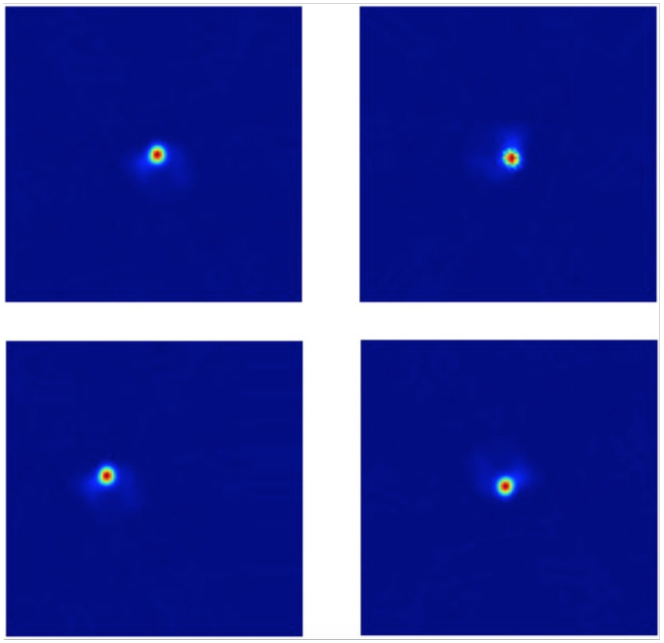
$$\alpha = \frac{1}{2}\left(\frac{\mathrm{Maj}}{\mathrm{B\_Maj}} + \frac{\mathrm{Min}}{\mathrm{B\_Min}}\right)$$



* Additionally filtered using Laplacian of Gaussian and Difference of Gaussian algorithms

Image Augmentation

- Generating more data through label-preserving transformations
- 36715 original images
- 107968 augmented images (3x original dataset)
- Done with Keras
  - Rotation
  - Flipping
  - Horizontal and Vertical shift
- No shearing or stretching



Deep learning algorithms

- Lasagne neural network, experiment with changing:
  - Number of layers
  - Using augmented images
  - Using a subset of images
- Tensorflow for Poets (‘Black box’ approach)
  - Place point and extended sources into separate folders

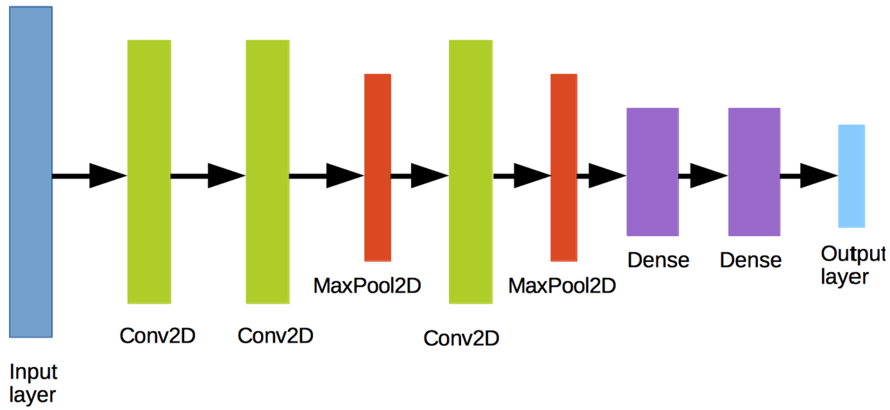
Lasagne network parameters

- Batch size 8
- Categorical cross-entropy cost function
- Train for 1000 epochs
- Mini-batch Stochastic Gradient Descent (SGD) with Nesterov momentum 0.9 and weight decay of 0
- Divide data into training, validation and test data sets

Division method	Training	Validation	Test
Div1	81%	10%	9%
Div2	60%	33%	6.7%

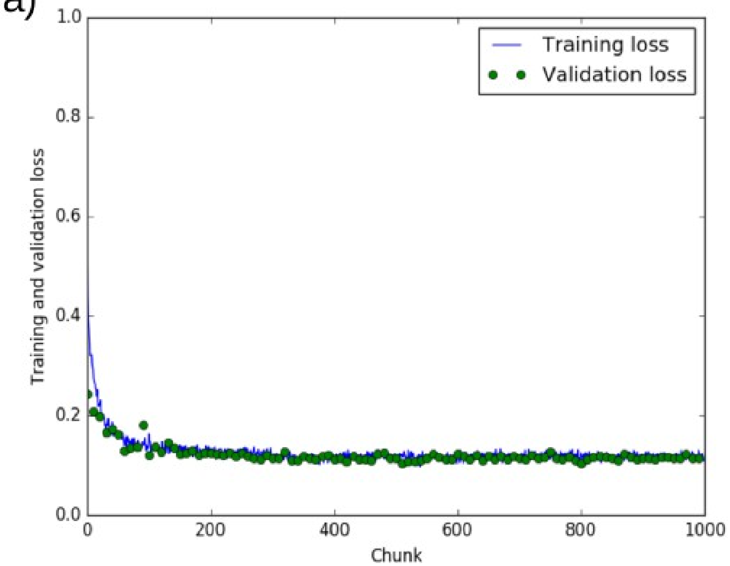
Lasagne results – Effect of using augmented images

144683 images in total

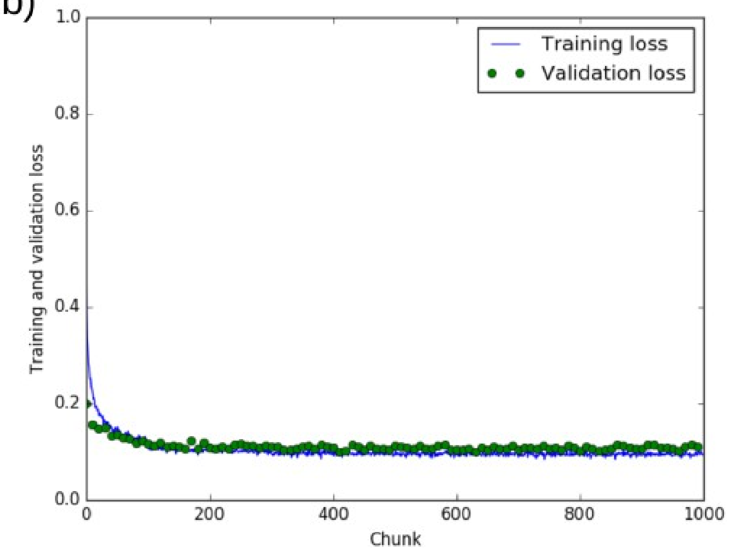


Chunk size	Precision	Recall	Accuracy
8000	0.9847	0.9379	96.22%
16000	0.9882	0.9512	97.03%

a)

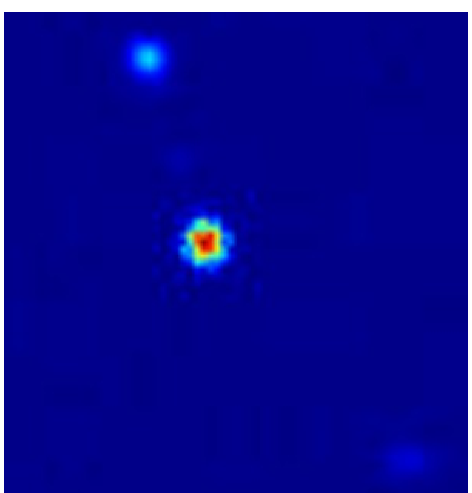


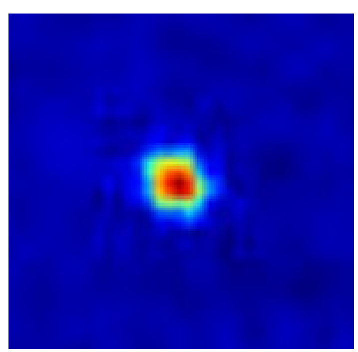
b)



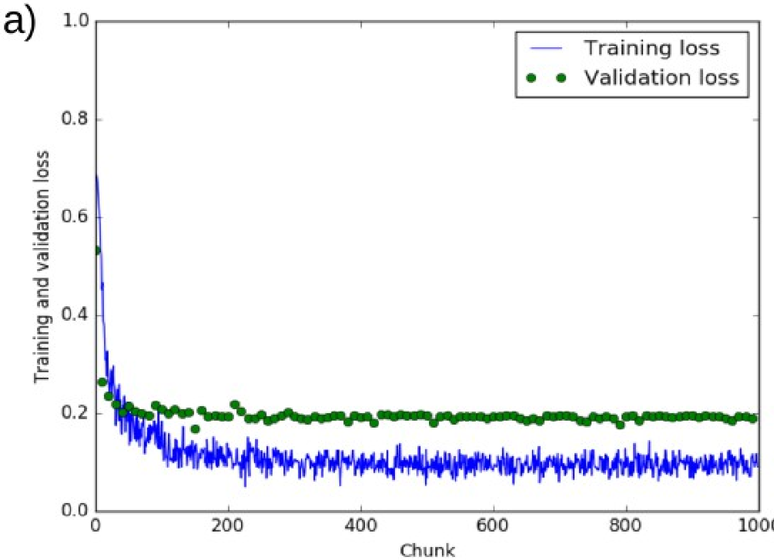
Lasagne results – Effect of using subset of images

# Images	Chunk size	Precision	Recall	Accuracy
1000	328	1.0000	0.9643	98.44%
1000	80	1.0000	0.9167	95.45%
500	160	1.0000	0.9474	96.88%
500 original, 1000 aug	400	0.9130	0.9333	92.71%

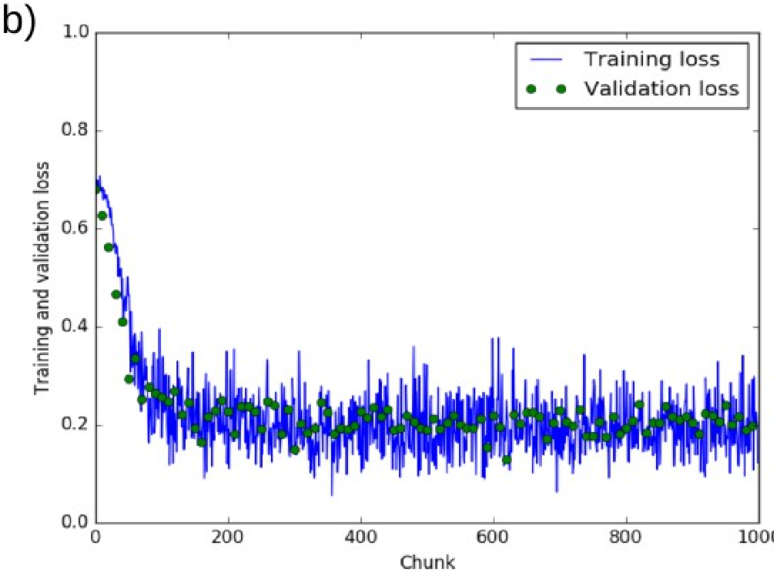




a)



b)



Tensorflow results

- Performs better when original images are used, rather than with additional augmented images
- Inferior performance to Lasagne neural network approach
  - Lasagne network enables more control over parameters

# Total images	# Compact source	# Extended source
36715	18716	17999
144683	72699	71984
# Test samples	**# Training steps**	**Test accuracy**
3603	4000	95.60%
14454	4000	94.60%

Conclusions and next steps

- Able to discriminate between compact and extended sources with typically > 96% test accuracy
- Neural network built with lasagne produces accuracies that supersede the tensorflow approach
  - 3 conv + 2 dense architecture
  - Larger chunk sizes tend to give better accuracies, at expense of overfitting
  - Make sure that enough original images are used since augmented image probably have interpolation artefacts
- Generate more specific classifications for extended sources