Object Classification of Tiny Images

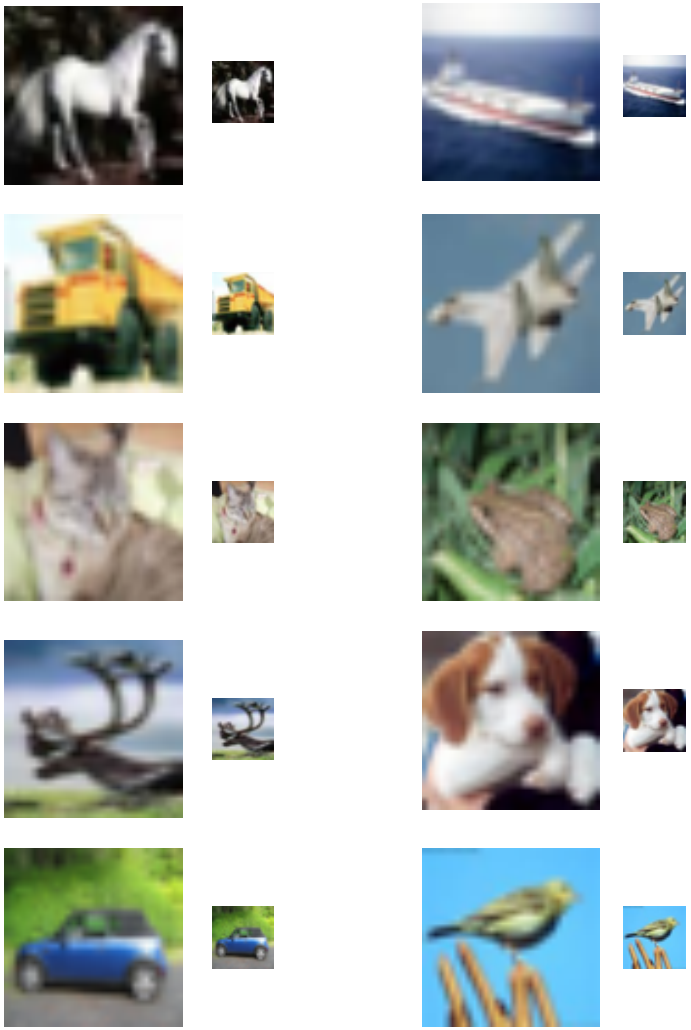
Mark Holt
GA Data Science
10 Dec 2014

The Dataset

- CIFAR-10

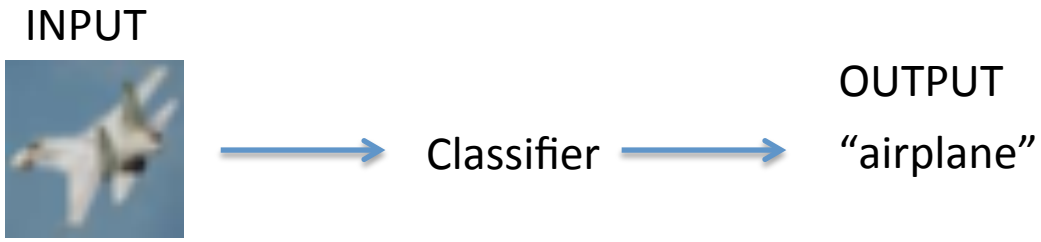
  Learning Multiple Layers of Features from Tiny Images. Alex Krizhevsky, April 8 2009.
  www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

- 60000 tiny color images
  - Tiny = 32 by 32 pixels
  - 50000 training set
  - 10000 test set
  - All hand-labeled
  - Contain a single dominant object
  - Images of 10 objects:
    - horse, ship, truck, airplane, cat,
    - frog, deer, dog, automobile, bird



Aims

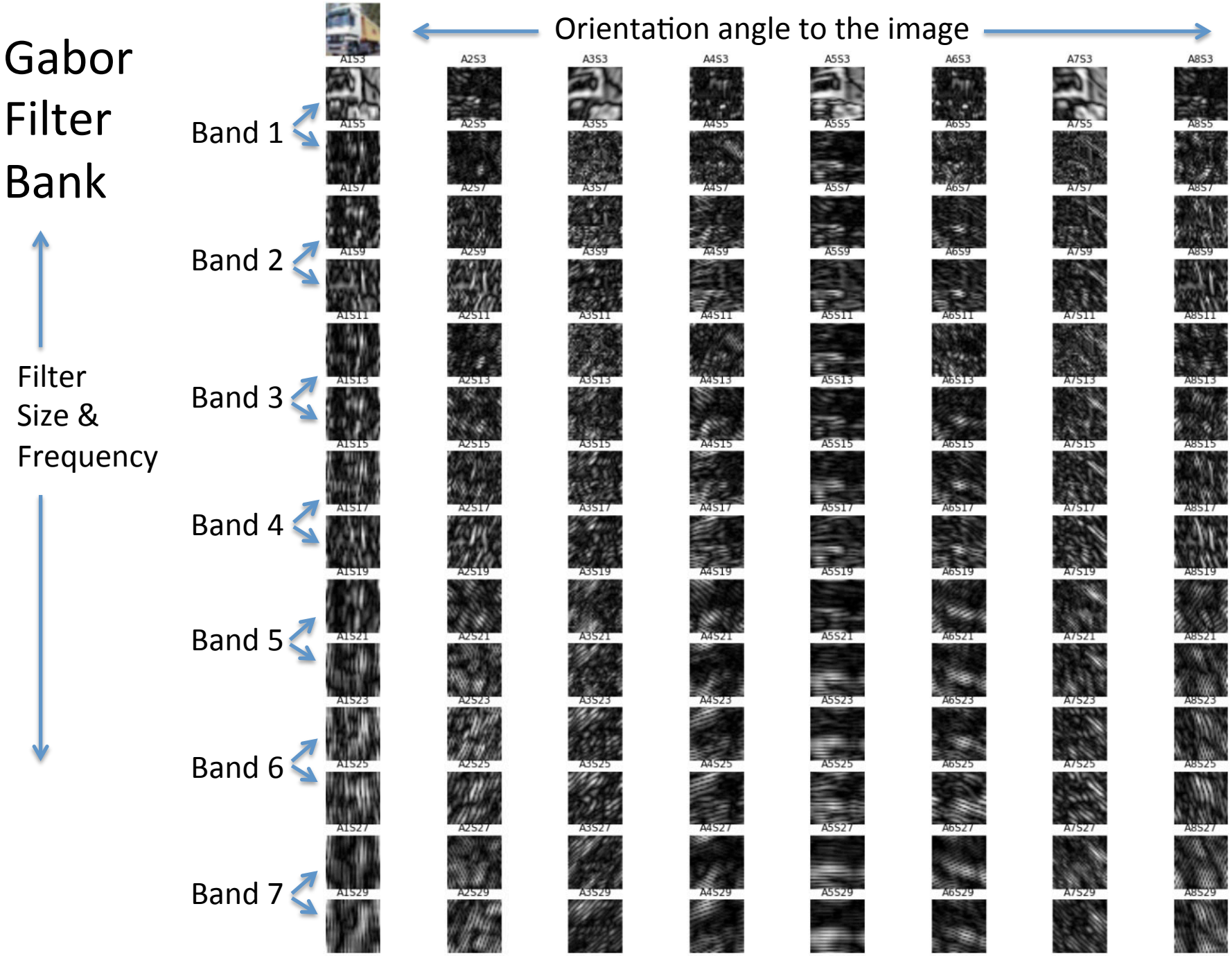
- Use machine learning techniques to build an automated image recognition system



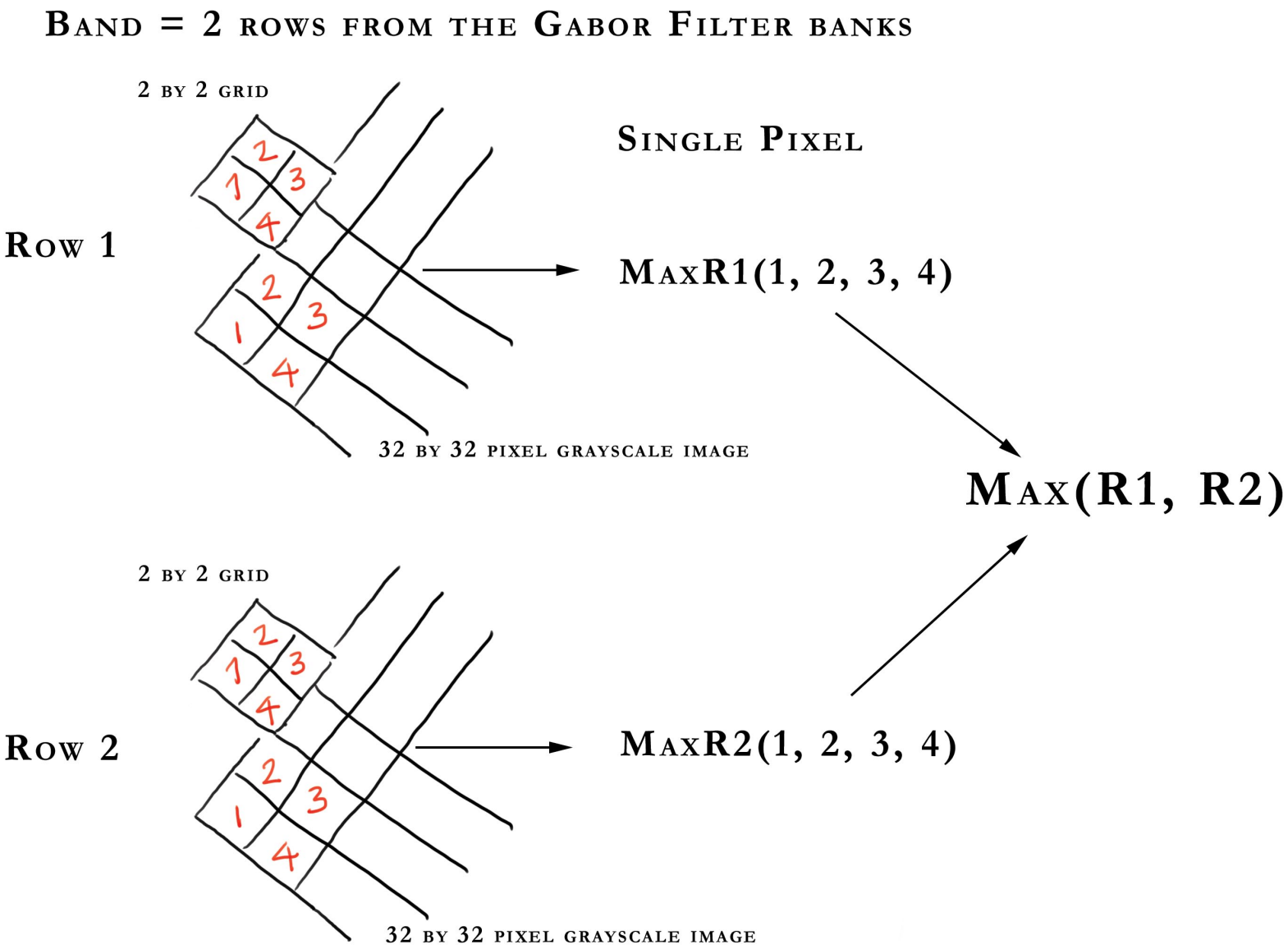
- Learn some familiarity with Python image processing tools
- Investigate image pre-processing
- Investigate 2-D Fast Fourier Transforms
- Experience the challenges of working with images
- Acquire some experience with decision trees

Pre-processing & Feature Extraction

- Convert to grayscale
- Equalize the histogram
- Gabor Filter Banks
  - Mimic simple receptor cells in the visual cortex
  - Edge detector
  - Tunable for:
    - Orientation to the image (0, 22.5, 45, 67.5, 90, 112.5, 125 degrees)
    - Frequency (0.8 – 3.2)
    - Filter size (5 -15 pixels)
  - Resistance to image rotation, reflection
- Filter Bank Pooling
  - Resistance to image scale, and small translations


Gabor Filter Bank
Orientation angle to the image
Band 1
Band 2
Band 3
Band 4
Band 5
Band 6
Band 7
Filter Size & Frequency
A1S3
A2S3
A3S3
A4S3
A5S3
A6S3
A7S3
A8S3
A1S5
A2S5
A3S5
A4S5
A5S5
A6S5
A7S5
A8S5
A1S7
A2S7
A3S7
A4S7
A5S7
A6S7
A7S7
A8S7
A1S9
A2S9
A3S9
A4S9
A5S9
A6S9
A7S9
A8S9
A1S11
A2S11
A3S11
A4S11
A5S11
A6S11
A7S11
A8S11
A1S13
A2S13
A3S13
A4S13
A5S13
A6S13
A7S13
A8S13
A1S15
A2S15
A3S15
A4S15
A5S15
A6S15
A7S15
A8S15
A1S17
A2S17
A3S17
A4S17
A5S17
A6S17
A7S17
A8S17
A1S19
A2S19
A3S19
A4S19
A5S19
A6S19
A7S19
A8S19
A1S21
A2S21
A3S21
A4S21
A5S21
A6S21
A7S21
A8S21
A1S23
A2S23
A3S23
A4S23
A5S23
A6S23
A7S23
A8S23
A1S25
A2S25
A3S25
A4S25
A5S25
A6S25
A7S25
A8S25
A1S27
A2S27
A3S27
A4S27
A5S27
A6S27
A7S27
A8S27
A1S29
A2S29
A3S29
A4S29
A5S29
A6S29
A7S29
A8S29

Combining Filter Banks



Filter Bank Pooling

8 by 8 pixels

Pooled Bands

- 2 closely matching layers from the Filter Banks combined

3 by 3pixels

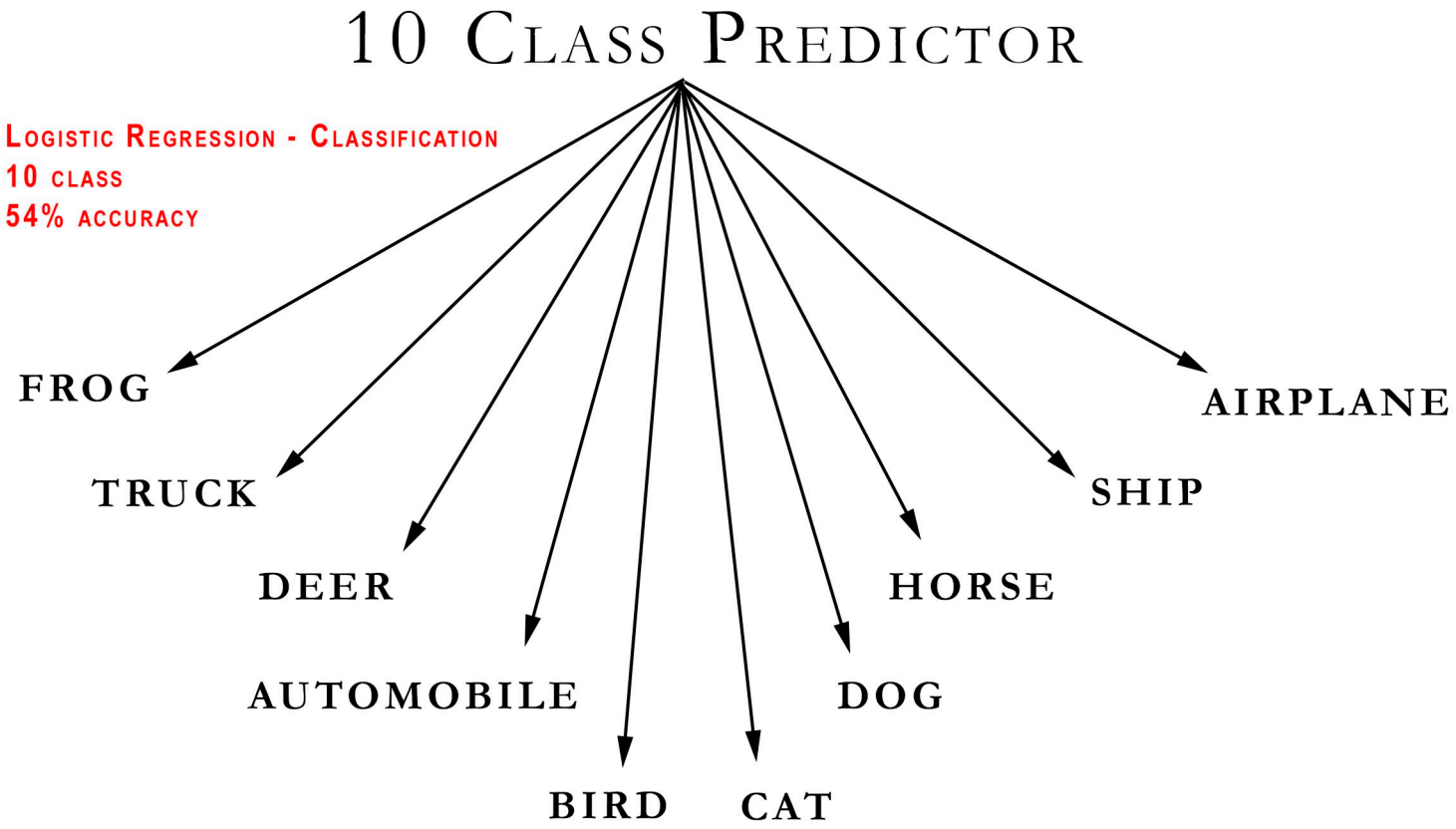
Building a Model

- Pooled Filter Banks
  - Input feature vector of length 1176
- Models investigated include:
  - Simple trees
  - Random forests
  - Extra trees
  - Stochastic gradient descent with Support Vector Machine
  - Ada Boosting
  - Logistic Regression
  - Multilayered Perceptron (MLP) – neural network
- Sheer volume of data very problematic
  - Needed to train overnight
  - Never completed
  - Kernel failed (R no better)
- Issues with python libraries
  - Opencv
  - Pybrain

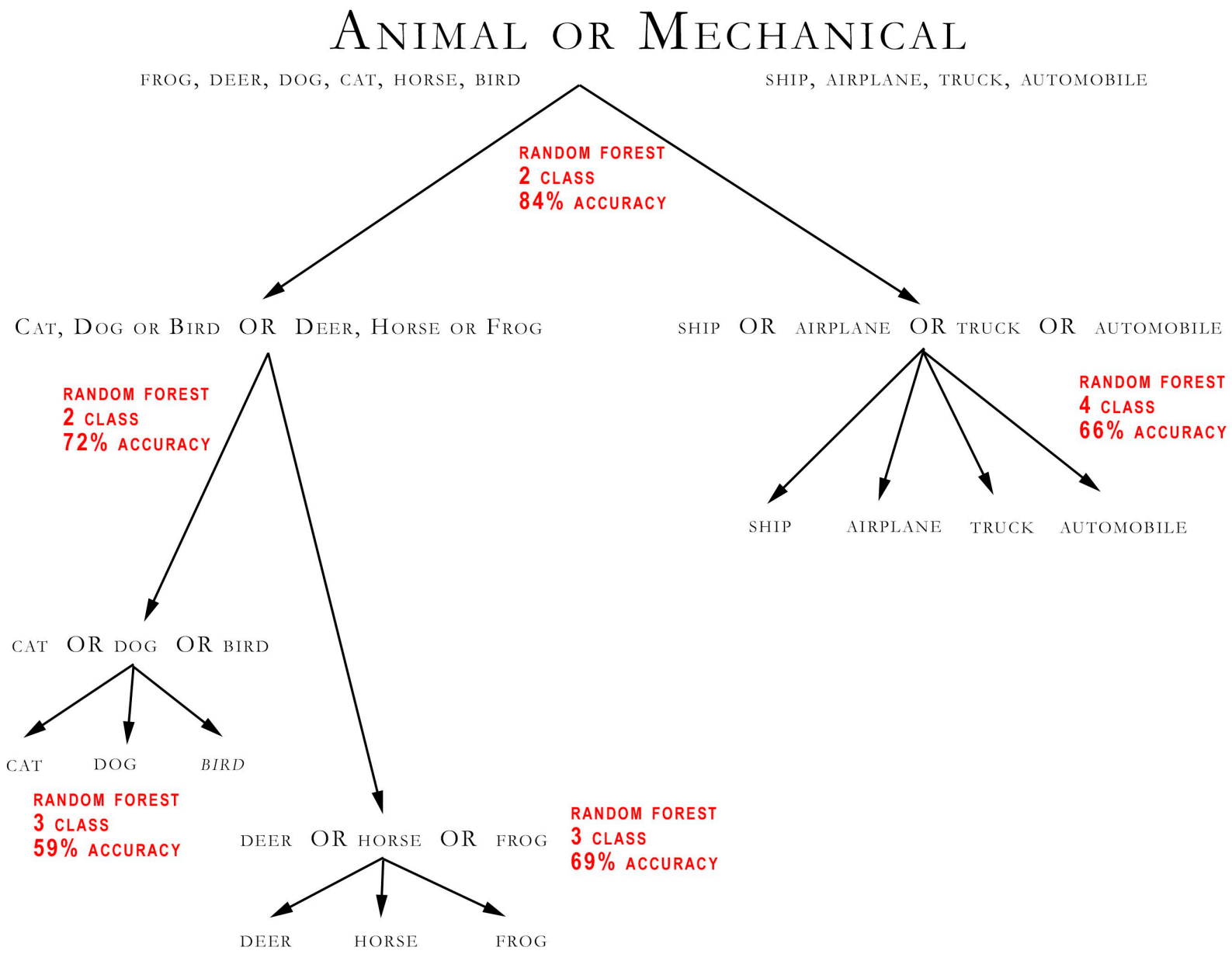
Solutions

- Utilized only:
  - Logistic Regression & Random Forests
  - Random Forests, Extra Tree implementation saved the day
    - Multi-core implementation
      - Reduced processing time to minutes
      - Speed-up > 10 times
- Two Implementations:
  - Logistic Regression provided best 10 class model
    - 54% accuracy
  - Random Forests provided the best “cascade” model
    - 47% accuracy
  - State-of-the-Art
    - 82% accuracy

Classification Using a Single Model



Classification Using a Cascade



Results in More Detail

	frog	truck	deer	auto	bird	cat	dog	horse	ship	plane
Predicted	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Actual										
frog **0**	608	12	51	29	81	76	67	19	18	39
truck **1**	10	675	20	112	21	25	14	28	54	41
deer **2**	93	22	433	35	71	86	67	127	23	43
automobile **3**	39	125	19	642	18	28	14	16	61	38
bird **4**	116	27	92	15	357	87	100	60	45	101
cat **5**	104	36	82	38	62	349	152	72	44	61
dog **6**	63	15	58	25	62	146	511	74	24	22
horse **7**	21	47	75	23	47	49	79	620	16	23
ship **8**	16	48	17	67	22	29	14	16	678	33
plane **9**	34	46	48	43	78	36	26	25	142	522

Difficult to separate:

Deer & horse
Automobile & truck
Cats & Dogs
Cats & frogs
Birds – generally hard,
Frogs, dogs & airplanes!

“Easiest” to separate:
Ships & trucks

Demonstration

- Ten classifier in action
- Cascade classifier in action

Future Work

- Additional feature extraction
  - Deep networks
  - Prototypes
  - Auto-associative networks?
- Utilize the color channels
- Implementation for generating the models
  - Fast, low-level, GPU
- Augment the training set
  - Reflections
  - Scaling
  - Rotation
- Dimensional Reduction?
  - Simple experiments suggest little impact