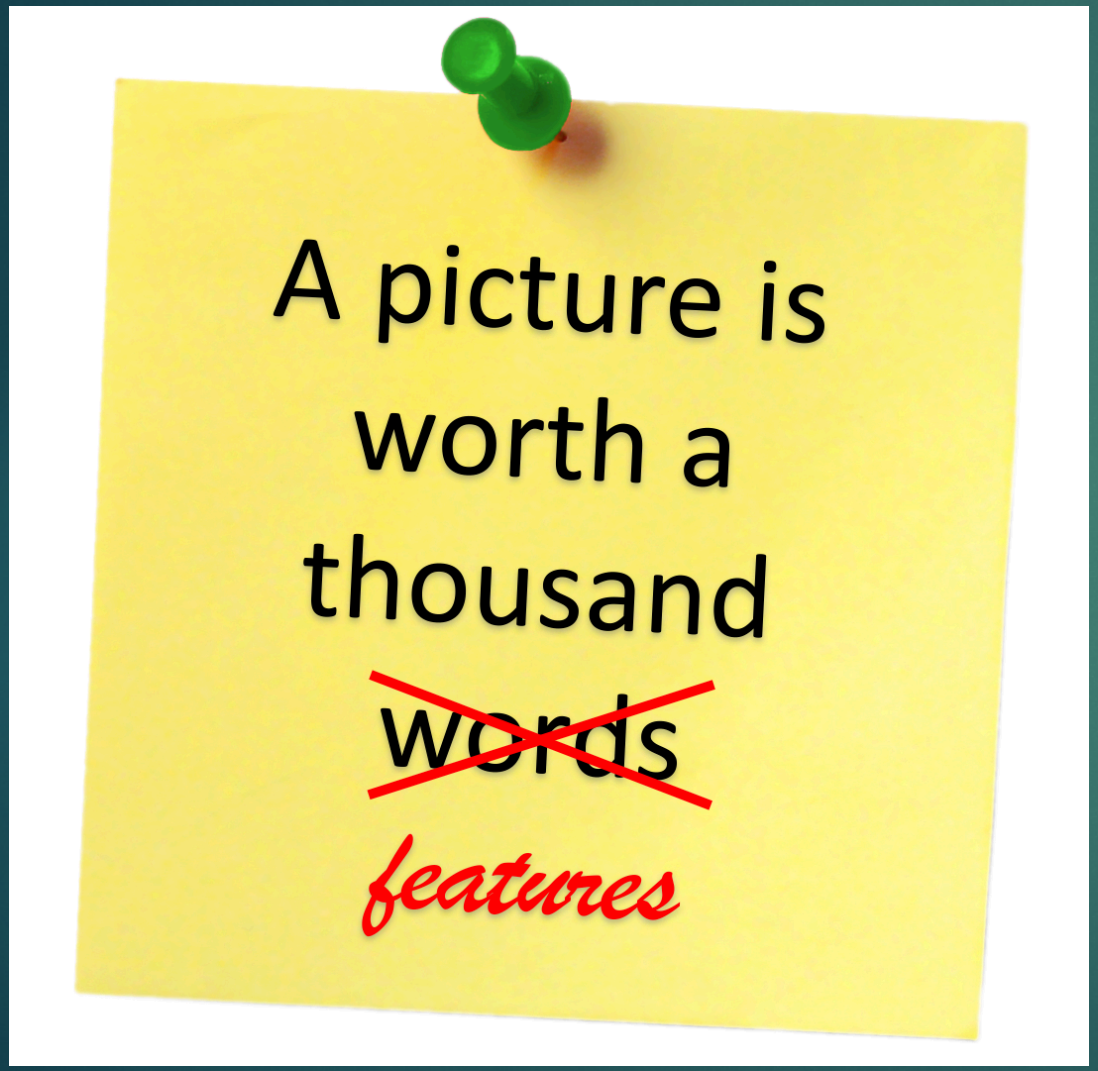
CIFAR-2 Image Recognition

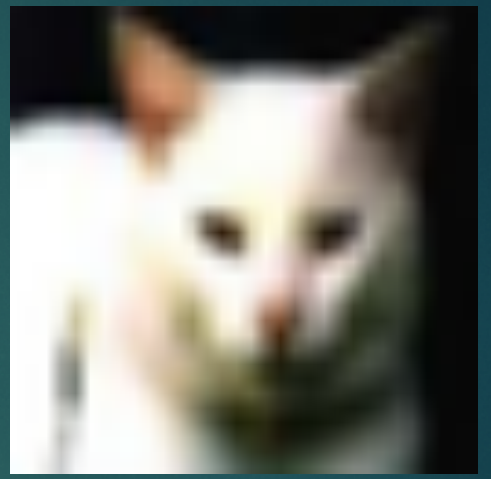
ANIL SIMON
ASHIM BHATTARAI
SARDHENDU MISHRA
RODRIGO TEJEIDA ESTRADA

Image Recognition



- Whale Tracking.
- Identify Distracted Drivers.
- Virus Classification.
- Detect Diabetic Retinopathy.
- Detect Lung Cancer.
- And more.

CIFAR-2 - Object Recognition in Images



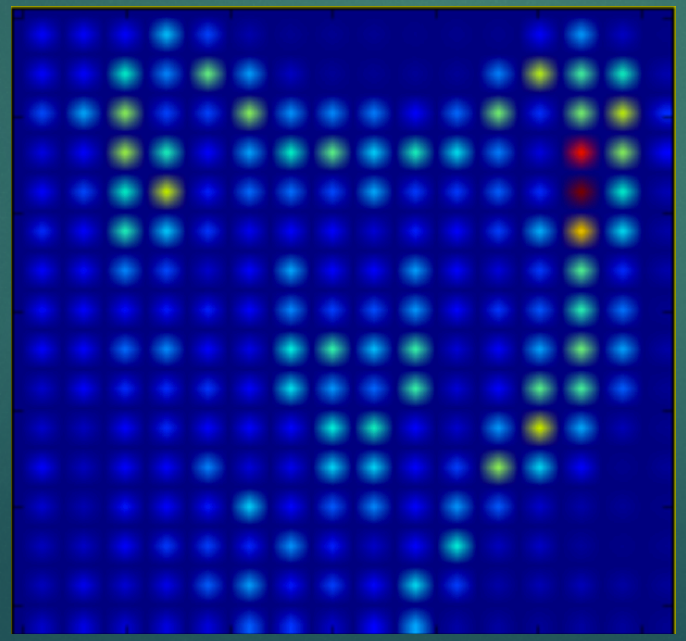
Preprocessing and Initial Analysis

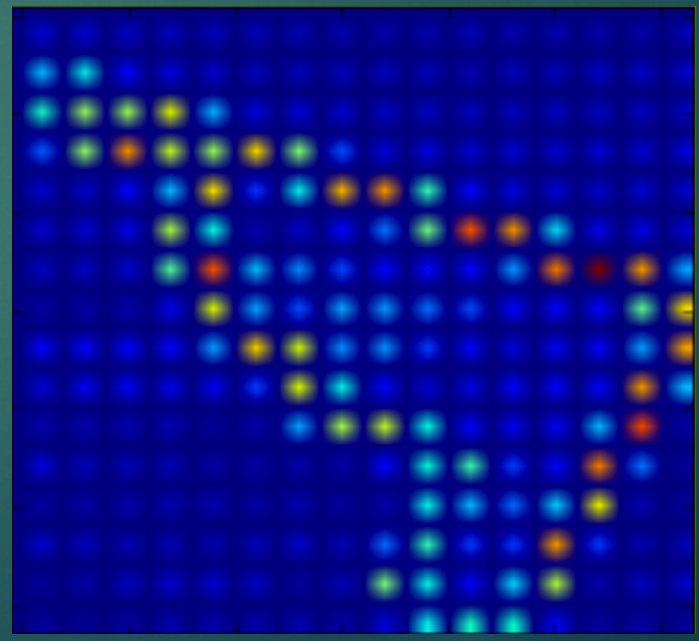
- File Handling:
  - Convert images into a single NumPy nd-array (matrix)
    - Image is 32 x 32 x 3 array of RBG channels
    - Each row in the nd-array represents one image
    - Input Matrix: 10,000 x 3072; Outcome Array (labels): 10,000 x 1
    - Basically each pixel is a “feature”

Feature Extraction:

- Idea: Extract "useful" feature from images
- We extract edges, HOG features, and RGB (above)
- Classifier uses features to compare images

Canny Edges



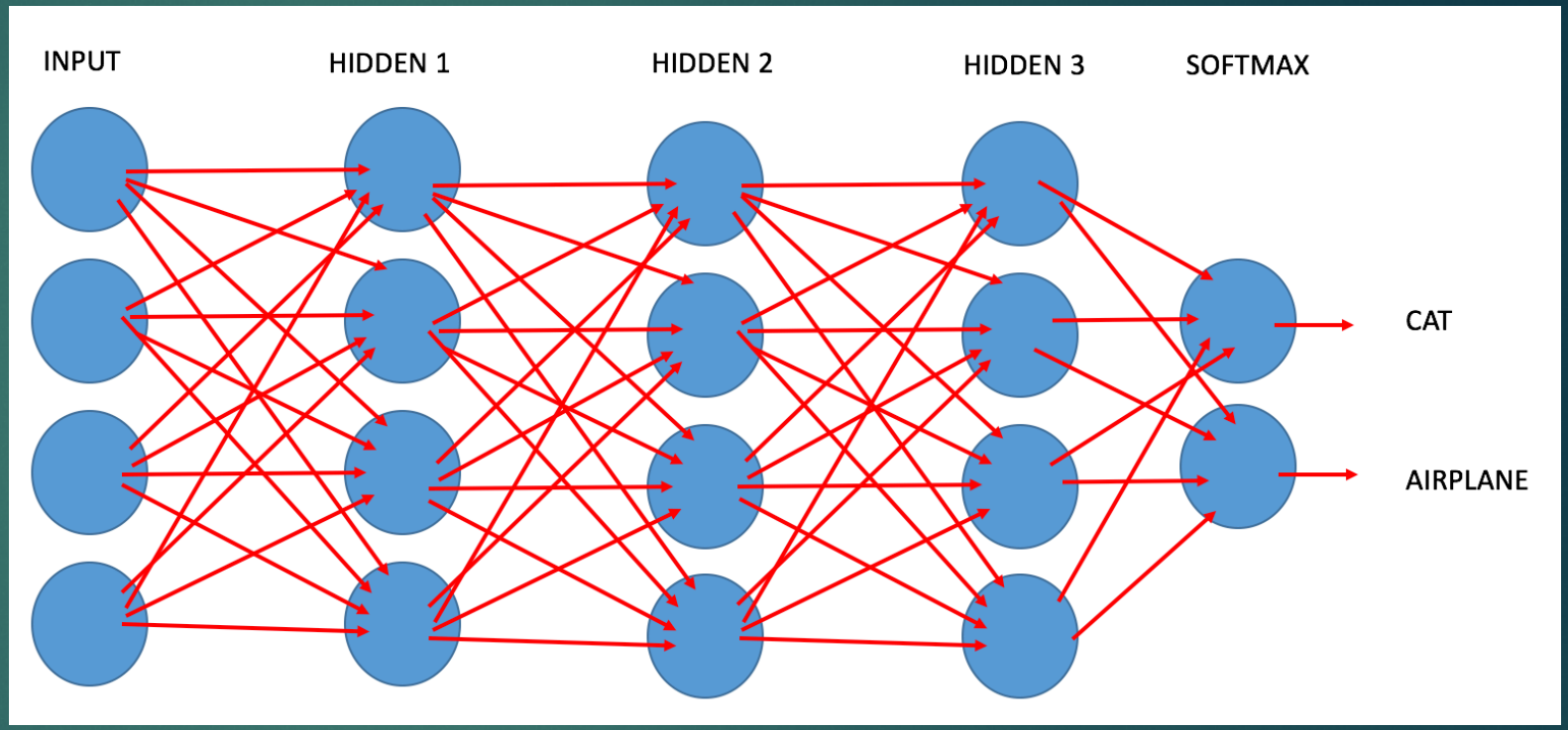


HOG Features

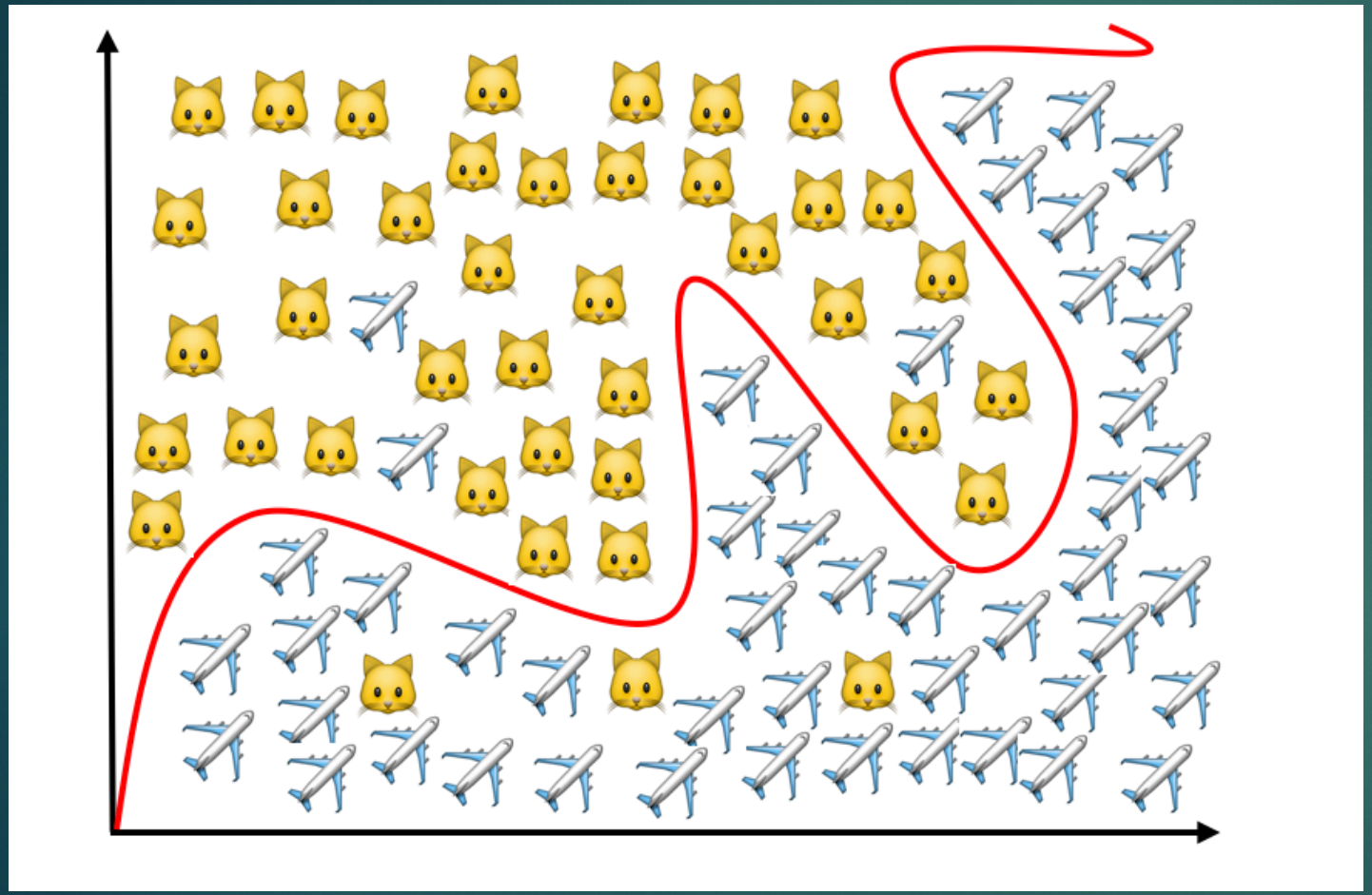
Models

Neural Network

- Four layer dense network: 1024, 1280, 1280 units
- Learning Rate = 0.006
- Dropout: 0.75, 0.5,0.5
- Average: 87.2% Accuracy over 10-fold CV

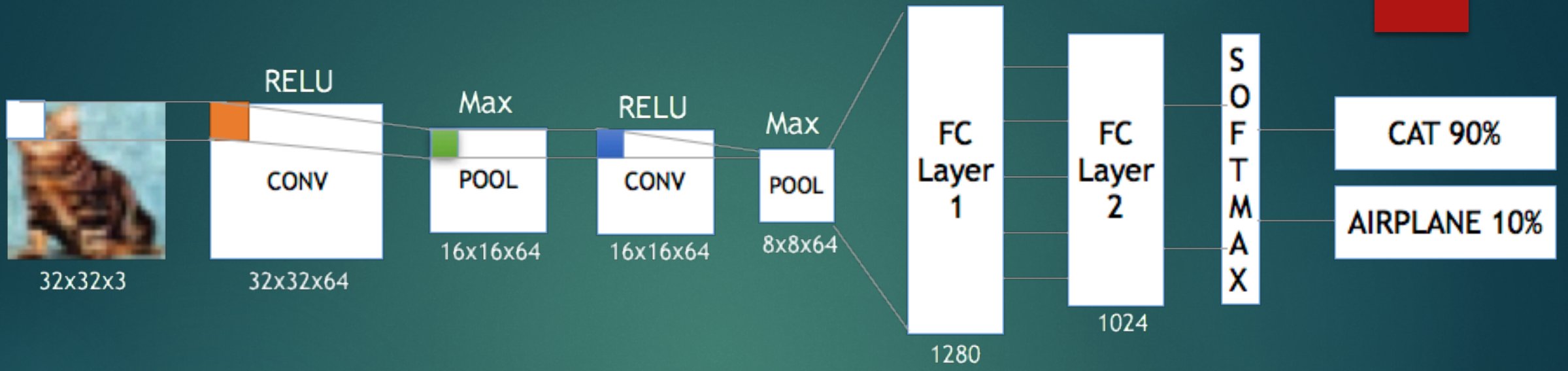


SVM with RBF kernels



(4x4 kernel)	C = 10 Gamma = 0.1
Average Airplanes Number	444.2
Average Cats Number	445.4
Average Airplane Accuracy	0.8884
Average Cat Accuracy	0.8908
Average Total Accuracy	0.8896

Convolutional Neural Networks (CNN)



- Convolution Filter1 3x3x64
- Max Pool Filter1 2x2
- Convolution Filter2 5x5x64
- Max Pool Filter2 2x2

MODEL CONFIGURATION

- Feature: RGB, contrast, brightness, Standardize
- Data Augmentation - Flip Image
- Regularization: Dropout {0.5}
- Optimizer: RMSPROP

MODEL EVALUATION

Accuracy		
Avg - 10 Fold cv - CAT	Avg - 10 Fold cv - Airplanes	Avg - 10 Fold cv - Overall
87.06	93.26	90.16

Logistic Regression

Feature Stacking → Feature Selection **Randomized Logistic Regression** → Logistic Regression

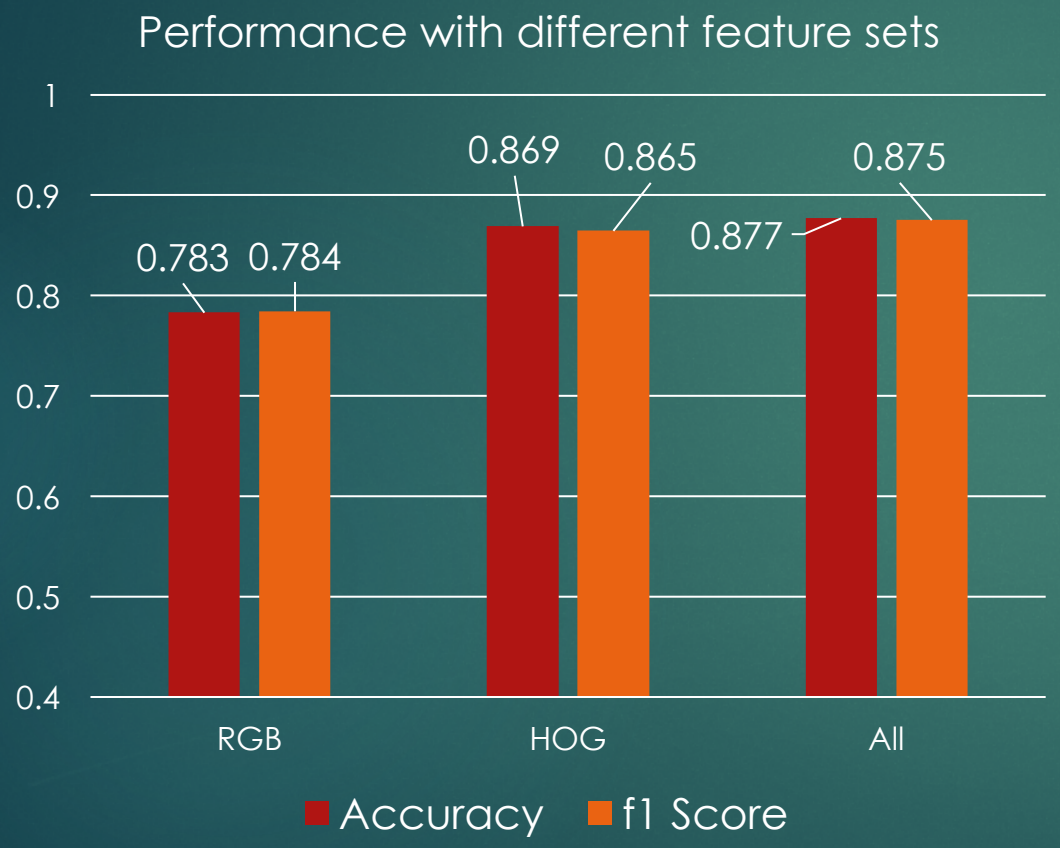
RGB + ZCA + GRAY + STD + EDG+ HOG

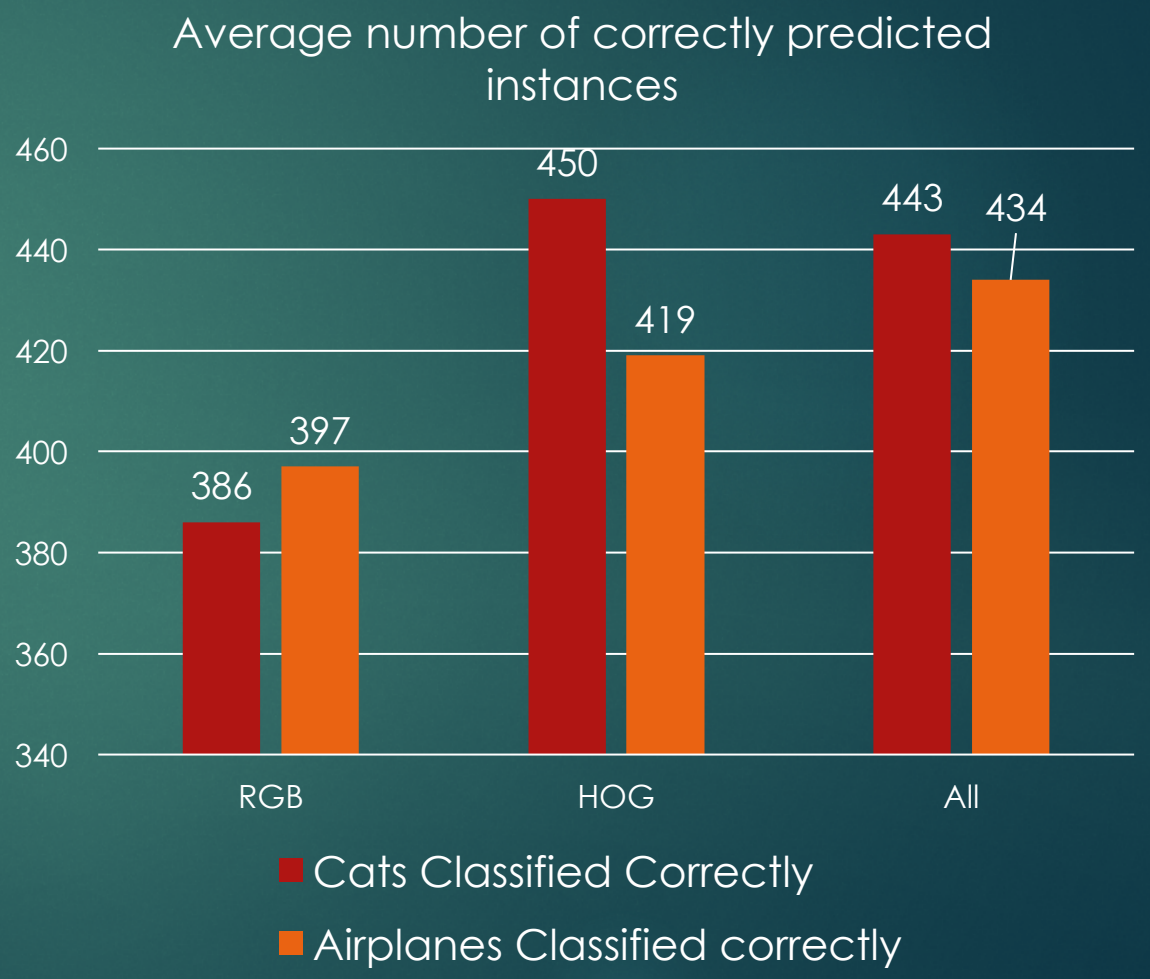
10417
- Total Features

120
- Selected features

Parameters
- Regularization = L2
- Regularization Strength = 0.1

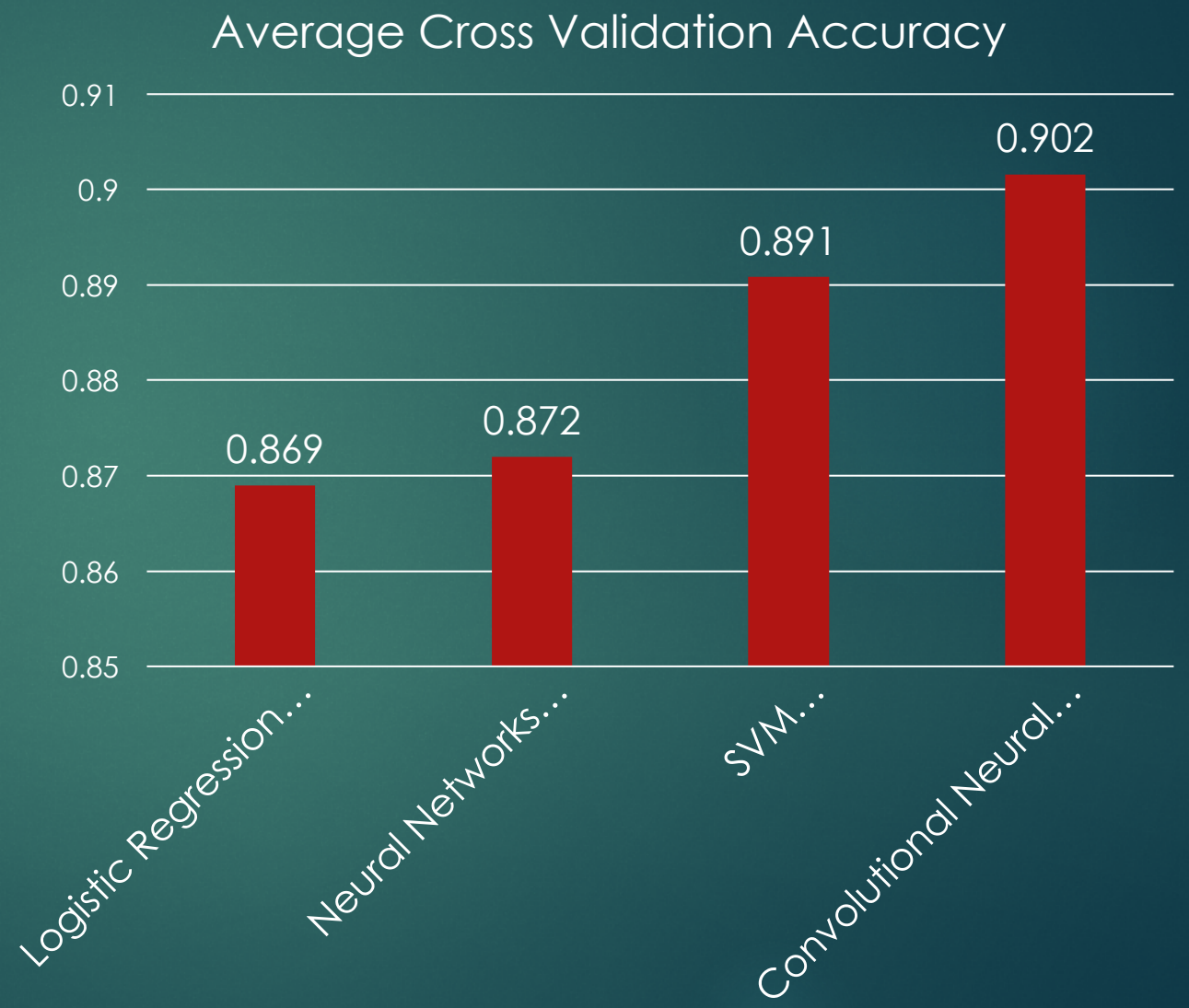
Comparison of Feature sets





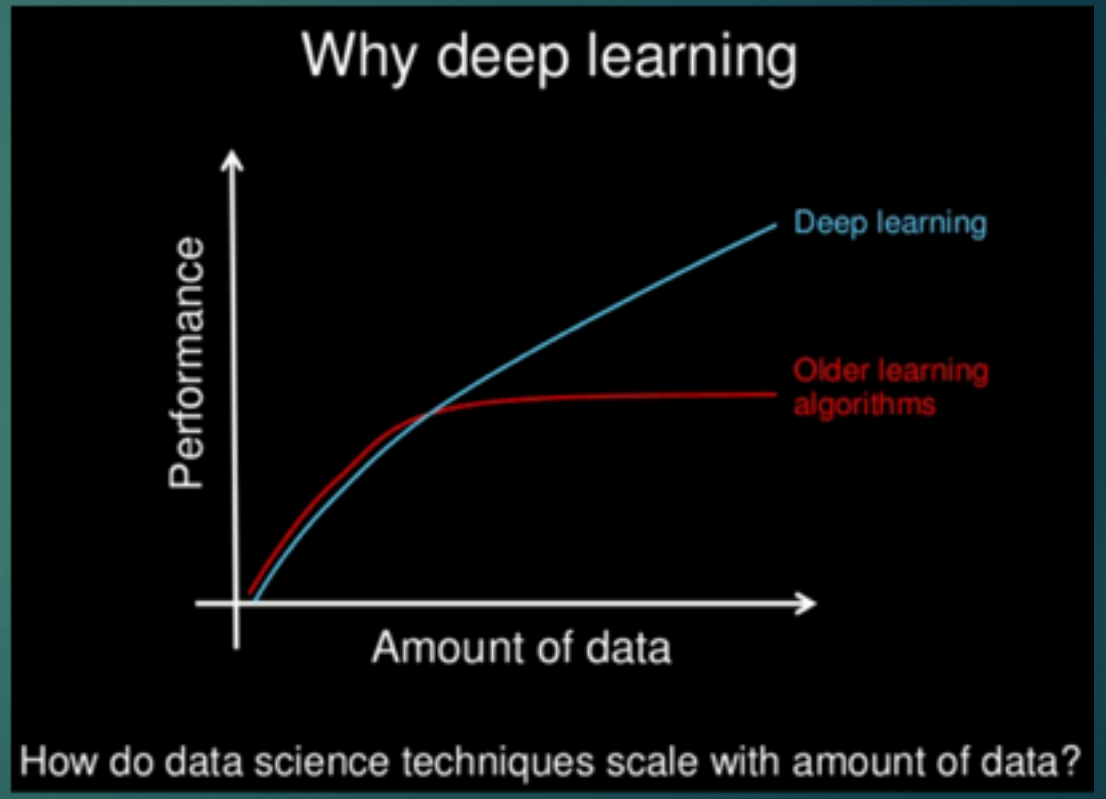
Comparison of Model Performance

- Complexity of features used
  - Time taken for Extraction
- Training time
- Accuracy of predictions
- Scope for improvement/tuning



Lessons Learned.

- Keep Calm and lower your learning rate.
- Don’t be too proud to use widely accepted model architecture.
- Preference of a Simple solution over a Complex one. (CNN - Total parameters = 6,563,586)
- Image Recognition + Big Data + GPU = Conv Nets.



Possible Next Steps.

- Identify misclassified images - check for common patterns
- Implement Inception modules for CNN
- Use Ensembles for different classes.
- Tune parameters and run the models for all the 10 class.

Thank You!!