Image Classification

Group 2

Jerome Kafrouni, Ayano Kase, Joo Kim, Chunzi Wang, Chuyuan Zhou

Project Description

- Classify images of cats and dogs
  - Feature extraction
  - Model training

- Conduct model evaluation and comparison to find the optimal method

Training Data

Unbalanced data with more dog images than cat images

- 1,358 dogs vs. 652 cats
- Unbalanced data can create biased models that better predict the more common class
- We undersampled to develop a balanced dataset:
    - 652 dogs and 652 cats

Features

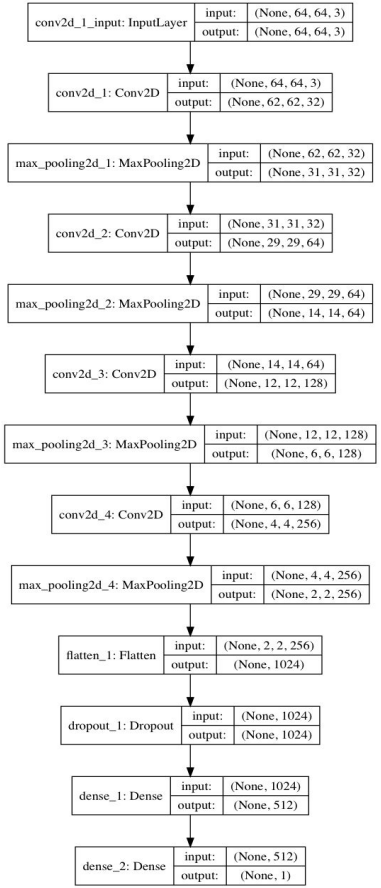
SIFT	Aggregated the 128 columns of SIFT values of each keypoint from training images into a big matrix, then used k-means clustering to put each keypoint to clusters. Then did data manipulation to see for each image how many keypoints fall into certain clusters.
HSV	(Hue, Saturation, Value) Models the way paints of different colors mix together.
HoG	Divides an image into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions (distribution of intensity gradients) is compiled.
RGB	Splits the intensity range [0,1] into 12 bins, for each channel, getting 1728 bins that are a partition of all possible colors

Models

GBM	Produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.
Random Forest	Decorrelates the trees by forcing it to consider only a subset of variables when performing a split.
SVM	Plots each data as a point in n-dimensional space (n is the number of features) and perform classification by finding hyperplane that differentiate the two classes.
Logistic Regression	Simple but effective classifier commonly used in two-class classification.
CNN	

Final Model Used: *Convolutional Neural Networks*

- 4 convolutional layers + 1 Dense layer
- Dropout on the last layer to avoid overfitting
- Trained on GPU (Gcloud - Tesla K80) in 30 min
- ImageDataGenerator to generate artificial training images



Neural net without augmentation

- Validation accuracy limited by the size of our dataset
- The NN just overfits the training data...

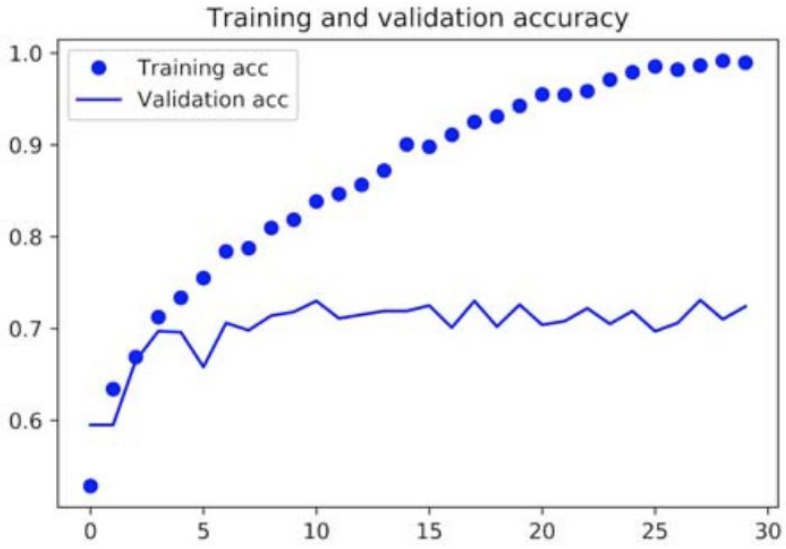


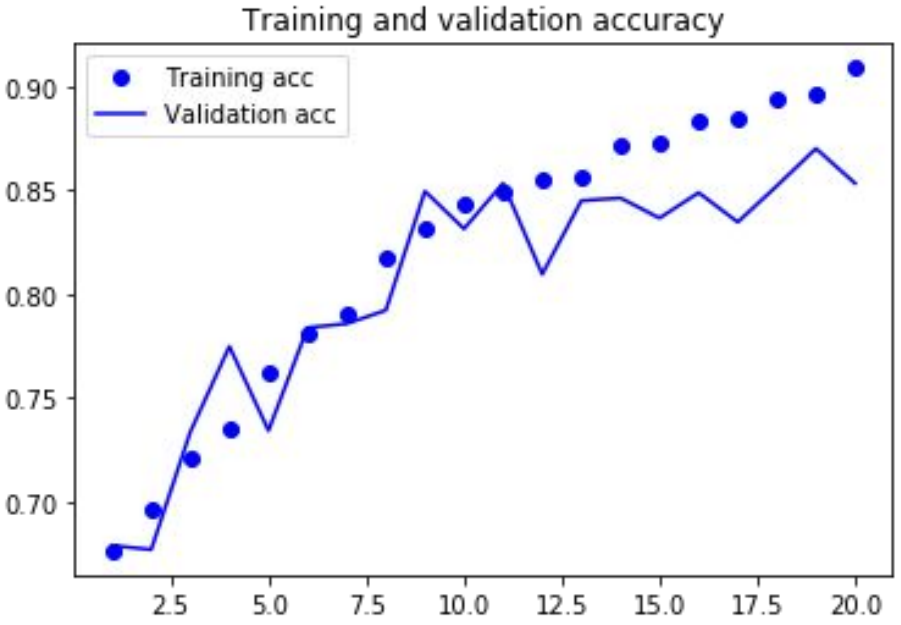
Image data generation

- Random shifts / flips / rotations applied to the data

  => Virtually infinite number of training samples

Neural net with augmentation

Up to 90 % validation accuracy!



Results

		GBM	Random Forest	Logistic Regression	SVM rbf	CNN
RGB	*best accuracy (%)*	65	50	52	51	
	best parameters	depth = 5	number of trees = 400 number of variables = 60		Cost = 5 Gamma = 0.001	
	time for training on whole dataset (s)	20.1	16.3	23.1	7.2	
HSV	*best accuracy (%)*	63	64	53	51	
	best parameters	depth = 3	number of trees = 400 number of variables = 70		Cost = 0.5 Gamm = 0.00001	
	time for training on whole dataset (s)	8.3	12.8	349.3	2.6	
HOG	*best accuracy (%)*	66	70	50	54	
	best parameters	depth = 5	number of trees = 300 number of variables = 90		Cost = 1 Gamma = 0.00001	
	time for training on whole dataset (s)	1.7	27.4	0.81	2.4	
SIFT	*best accuracy (%)*	72	79	74	55	
	best parameters	depth = 10	number of trees = 500 number of variables = 90		Cost = 0 Gamma = 0.000001	
	time for training on whole dataset (s)	80.0	24.5	0.44	1.5	
Raw image	*best accuracy (%)*					87
	best parameters					
	time for training on whole dataset (min)					30.00

Recommendations for Future Work

- Better ways to combat unbalanced dataset for classification
- Cropping approach: 3-way cropping
- Resizing images
- Neural networks:
  - Use a pre-trained neural network for “warm start” and train it further on our data
- The trade-off:
  - Neural network demonstrates high accuracy of 90%, but 30 minutes to train with GPU
  - SIFT took a long time to compute clusters but faster than other models
  - Accuracy vs. computational efficiency

Thank you!