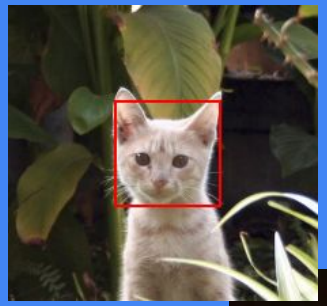
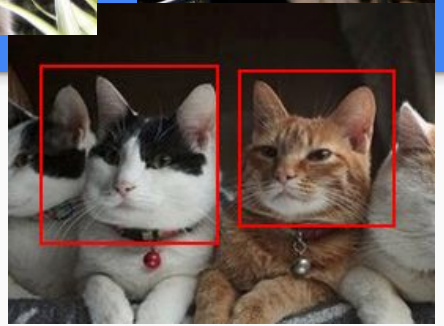
Answering the Most Important Question on the Internet

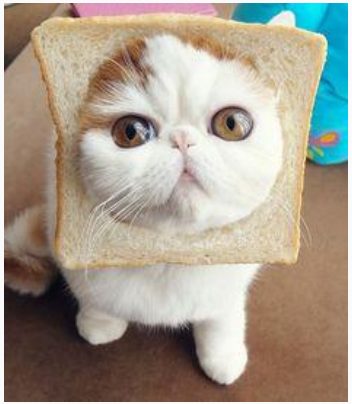
Casey Pore
CS535
12/8/2016

Does this Picture Contain a Cat?

Object Detection





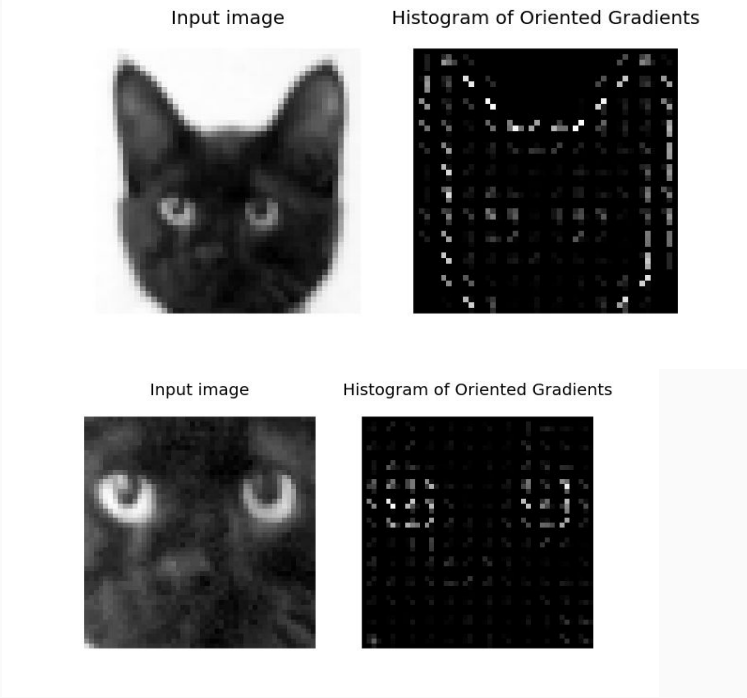


- Goal: To recognize an object in an Image or video
  - In our case: kitties in images!
- A blend of Machine Learning, Computer Vision, and Image Processing
  - high-dimensional data
  - Processing intensive
  - Still a challenging and widely researched area
- Cats are just the beginning
  - ML techniques can be applied to other objects, such as human faces, by extracting the right features.
- Project inspired by “Cat Head Detection - How to Effectively Exploit Shape and Texture Features” - Weiwei Zhang, Jian Sun, Xiaoou Tang
- Fun Fact: A similar problem has been studied here at CSU:
  - Bruce A. Draper, Kyungim Baek, Jeff Boody, “A Biologically Plausible Approach to Cat and Dog Discrimination,”

Approach

Preprocessing

- **Goal: Convert image data to feature vectors**
  - HOG descriptors - describe change in brightness between a group of pixels and adjacent pixels
  - Extract features for shape and face (texture)
  - 48x48 ROI
  - Choose ‘HOGDescriptor’ parameters (more on this later)
- **Dataset**
  - Positive Examples: Zhang et al Dataset
    - 10,000 Images - 2.0GB, zipped
    - Annotated with 9 points: eyes(2), nose(1), ears(6)
  - Negative Examples: PASCAL VOC 2007 challenge
- Resulting Dataset
  - Each feature vector 2904 dimensions
    - Values range 0-1
  - 9000 pos/neg examples each
  - 1.1 GB total for shape and texture features.
- Done offline, only takes a couple minutes
  - Could be done in Spark - could try different parameters for multiple datasets



Approach

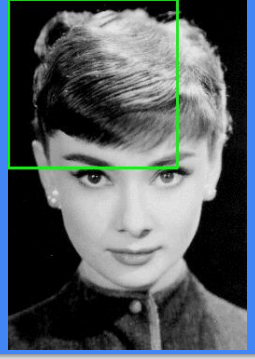
Training

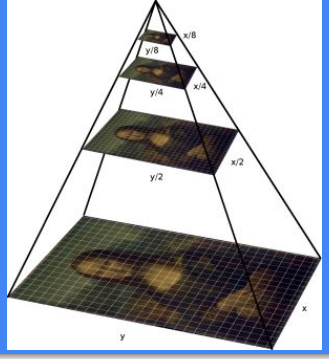
- **Framework**
  - Hadoop/Yarn/Spark stack
  - Python(PySpark) instead of Java or Scala
  - MLlib
- **Algorithms/Techniques**
  - Train model via Binary Classification
  - Linear SVM (SVMWithSGD) - Same as Zhang et al.
    - ‘train’ function distributes work
- **10-Fold Cross Validation**
  - 720 parameter sets (360 each for shape and texture)
  - Evaluated on mean % misclassified across folds
  - Records best parameter set seen so far
  - Final step train classifier with best parameter set on all data

Parameter	Values Tried	Description
iterations	100, 1000	The number of iterations
step	0.01, 0.1, 0.5, 1.0, 2.0	The step parameter used in SGD
regParam	0.00001, 0.0001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1	The regularizer parameter
miniBatchFraction	1.0	Fraction of data to be used for each SGD iteration
initialWeights	None	The initial weights
regType	"l2", None	The type of regularizer used for training the model.
intercept	True,False	Boolean parameter which indicates the use or not of the augmented representation for training data (i.e. whether bias features are activated or not)
validateData	True	Boolean parameter which indicates if the algorithm should validate data before training.
convergenceTol	0.001, 0.0001	A condition which decides iteration termination.

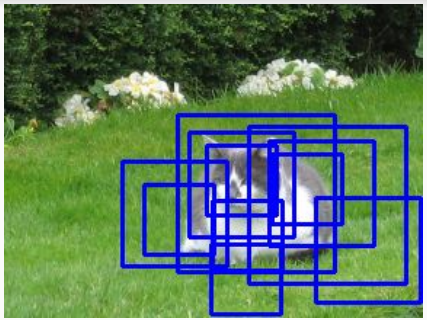
Approach

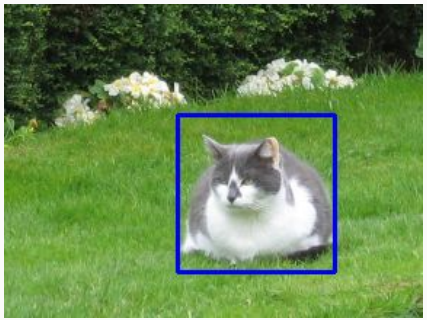
Application

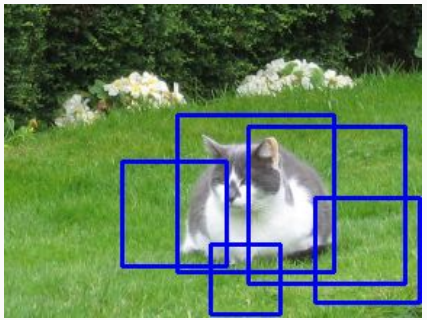




- **OpenCV**
- **Sliding Windows**
- **Image Pyramids**
- **Window Grouping**
- **All handled by 'HOGDescriptor.detectMultiScale'**
  - Returns detected rectangles and weights
  - Returns a lot of False Positives
- **Second round of window grouping**
  - Filter low weighted rectangles and apply non-max suppression
  - Select largest rectangle that contains largest weighted rectangle
  - Consider it True Positive if eyes and nose in final rectangle
- **A lot of potential for tuning this process.**





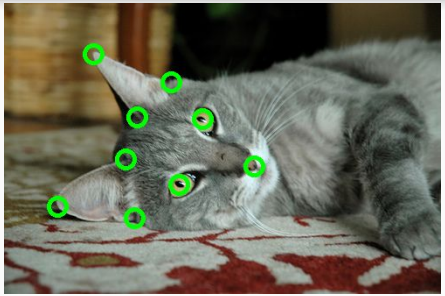


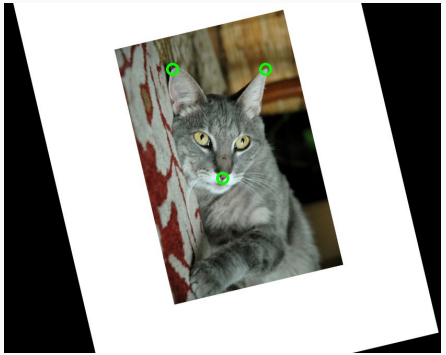
The Software

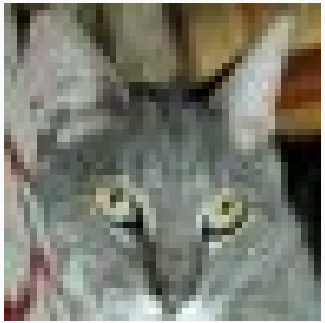
Creating Features, Training Part 1

Parameter	Value	Description
winSize	(48, 48)	Detection window size. Must match the size of the training image.
blockSize	(8,8)	Block size in pixels.
blockStride	(4,4)	Block stride. Number of pixels a block moves.
cellSize	(4,4)	Number of pixels in a cell.
nbins	6	Number of vectors in a block.

- **Feature Extraction**
  - Set Descriptor Params
  - Extract image
    - Translation - center image about the nose
    - Rotate - Ear tips horizontal
    - Scale
      - 38 pixels between ears or 30 between eyes
    - Extract
      - 48x48 imagecentered 22 pixels below ear tips, or 9 below eyes
  - Compute HOG descriptor - 2904 features
- **Training**
  - 10-fold Cross Validation (couldn't use Pyspark's CrossValidator as SVMwithSGD doesn't implement necessary interfaces)
  - Performed for each dataset, shape and texture
  - Text File loaded into RDD of LabeledPoint
  - Randomly split into 90/10 Train and Validate and Test sets
  - 90% Train and Validate set split into 80/20 Train/Validate sets
  - 90% set is split randomly 10 times







The Software

Training Part 2, Cat Face Detection

- **Training**
  - Mean Error is calculated over the 10 folds
  - If mean error < best mean error + 10%, parameters are used to train test model
  - Resulting model is trained on 90% train/validate set and tested on 10% hold-out test set.
  - If resulting error < best model seen so far, parameters are recorded
  - Process continues until all parameters are tried
  - Best parameter set found is used to train model on entire data set; This is our final model
- **Application**
  - Use ‘detectMultiScale’ and techniques described earlier
  - ‘finalThreshold’ (1.5) and ‘scale’ (1.4, 1.5) most important parameters
  - Second round window grouping set 60% threshold
    - Only weights greater than 1.5 for shape and .85 for texture make it to second-round grouping.

Results and Evaluation

Testing Process

- Training the model - Cross Validation
  - Checks mean accuracy over folds
  - Best accuracies are applied to validation set
  - Testing as it goes for best final parameter set selection.
  - ‘BinaryClassificationMetrics’ didn’t work, so no Area Under ROC Curve or Area Under Precision-Recall Curve
    - Though they *are* implemented in PySpark, gave me errors.
- Testing the applied Model
  - No usable Spark implementations
    - Could write your own. Out of scope for this project.
  - Hand-tested 100 images offline, with different parameters
  - Tried several parameters for ‘detectMultiScale’ and window grouping, final results based on best-guess parameters.
    - Lots of room for improvement - should automate to find best parameters

Results and Evaluation

Results

Metric	Parameter Values	Result
Error Rate (Training)	(iterations,step,regParam,miniBatchFraction,initialWeights,regType,intercept,validateData,convergenceTol) Texture: (1000, 0.5, 0.0001, 1.0, None, None, False, True, 0.0001) Shape: (1000, 0.5, 0.1, 1.0, None, None, False, True, 0.0001)	Texture: 1.6% Shape: 2.227%
Error Rate (Real-world)	(weight, scale) Texture: 1.3 ,1.4 Shape: 2.3, 1.5	Texture: 39% Shape: 41%
Precision (Positive Predictive Value) $PPV = \frac{TP}{TP+FP}$	(weight, scale) Texture: 1.3 ,1.4 Shape: 2.3, 1.5	Texture: 61.9% Shape: 53.7%
Recall (True Positive Rate) $TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$	(weight, scale) Texture: 1.3 ,1.4 Shape: 2.3, 1.5	Texture: 53.06% Shape: 64.44%

Results and Evaluation

Evaluation

- PySpark worked well, though took around 120 hours to process 7200 parameter sets on 1.1GB of data.
- Did good job training the classifier.
- So-so for actual detection
  - At least one rectangle found cat head, but lots of false positives
  - Precision on par with Zhang et al. (Average Precision on VOC data=.632)
  - Recall slightly worse (no average recall, but at .4 when precision is at .6 on PR curve)
  - Only detects one cat, if any.

Conclusion

- Improvements
  - Preprocessing limitations
    - What are the best parameters to create HOG descriptors?
  - Detection limitations
    - Parameters, parameters, parameters
      - Did I mention parameters?
  - Hard-Negative mining
  - Try rotations in addition to image pyramid
  - Try other classifiers (remove openCV's SVM dependency)