

Team Raspberry - Image Classification

Kristian Wahlroos Ilkka Vähämaa Sean Lang

December 15th, 2017

Overview

Methods

Final System

Results

Methods

Data representation

- ▶ Treat every image as a 3D-tensor (RGB)
 - ▶ Repeat the value of grayscale images three times
 - ▶ Colorized are handled as the original tensors
- ▶ Original data has 14 labels, we used 15
 - ▶ Extra one for the unclassified images
 - ▶ One-hot encoded labels

Methods

Data processing

- ▶ Read images in batches of size 2000
 - ▶ Helps to avoid filling the RAM
- ▶ Normalize the pixel values between $[0.0, 1.0]$
- ▶ For every batch augmenting the data
 - ▶ Provided by Keras
 - ▶ Centerify, shear, zoom, rotate and flip
 - ▶ To get more variation and samples from classes with few labels

Methods

Class weights 1/2

- Classes are very unbalanced

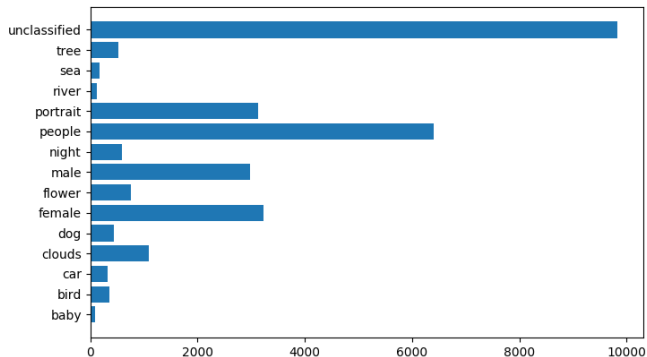


Figure : Class distribution

Methods

Class weights 2/2

- ▶ We tackled this problem by custom weights per class
 - ▶ Giving them at training phase

Class weight function

$$S(c_i; \lambda) = \ln \left(\lambda \frac{\sum_c |c|}{|c_i|} \right)$$

$$W(c_i; \lambda) = \max(S(c_i; \lambda), 1)$$

Methods

Network topology

- ▶ One network that outputs 15 classes
- ▶ Three convolution layers all followed by max pooling
 - ▶ Filters 32, 32, 64
 - ▶ Kernel size 3x3
 - ▶ Max pool size 2x2
 - ▶ ReLu as activation function
- ▶ After pooling flattening via dropout to dense layer with sigmoid activation
 - ▶ Dropout value: 0.4
- ▶ Very simple network

Methods

Loss function

- ▶ Categorical crossentropy wouldn't work as one image can be in many classes
- ▶ Binary crossentropy was suggested in many forum posts
 - ▶ Still not viable solution when there are many overlapping categories
 - ▶ Loss is too forgiving for giving 0 labels
- ▶ Solution: "custom" loss function **BP-MLL**¹
 - ▶ Actually taken directly from the paper [1]²
 - ▶ Designed for multi-label problems
 - ▶ Implementation for Keras can be found from internet
 - ▶ Punishes more from just giving 0 labels

¹Backpropagation for Multilabel Learning

²[1] *Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization*, 2006

Methods

Validation

- ▶ Thing1
- ▶ Thing2

Results

- ▶ Thing1
- ▶ Thing2

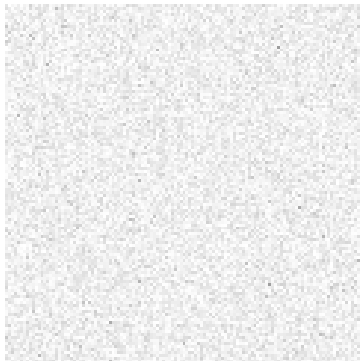


Figure : caption text