

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

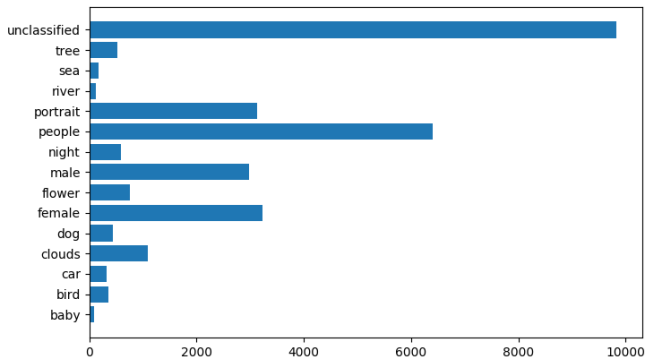


Figure : Class distribution

Methods

Class weights 2/2

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

Class weight function

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

Methods

Topology

- ▶ Thing1
- ▶ Thing2

Methods

Loss function

- ▶ Thing1
- ▶ Thing2

Methods

Validation

- ▶ Thing1
- ▶ Thing2

Results

- ▶ Thing1
- ▶ Thing2

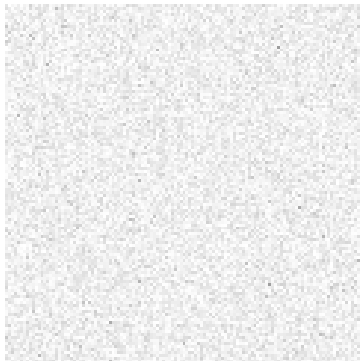


Figure : caption text