Data Description

You are given two sets of data: one labeled training set, and one unlabeled test set. You have to train your models using the labeled training set (you are strongly advised to subdivide the training set with a good validation scheme), and then use your finetuned models to predict the labels of the test set. When you submit your predictions of the test set (according to the format specified in the 'Overview/Evaluation' section), you will immediately get your multi-class log loss score calculated on a portion of the total test set. If this score surpasses your previous submissions, it will be displayed on the public leaderboard, determining your ranking there. It is important to realize that your public score is only calculated on a small portion of the test set, so avoid overfitting on the leaderboard scores!

We, the hosts, can see the private scores based on the full test set, and at the end of the competition they will become public for everyone. It is not uncommon to see large shifts in the rankings when the private leaderboard scores become public, so you must definitely take this into account when building your model to avoid large rank drops.

You may assume that the distribution of the classes in the complete test set is similar to the distribution of classes in the train set. To get some bonus points on your report, I want you to briefly write down your answer to this question at the end of your report:
What would you change to your model if the distribution of the test set was uniform over the 12 classes? Would you use the same loss function, performance metric, train/validation/test splits etc. as you did before?

Files

`train.zip` contains the labeled training data. All animals are sorted under their respective labeled folder.
`test.zip` contains the unlabeled test data. Don't change the labels of these images, since the order in which you have to submit your solutions depends on the numbering here. Don't worry, in your notebook you can load everything in the right order and after your model predicts the outputs, you can use the submission helper function to create a valid submission.

`starterskit` folder containing

- a notebook on the feature extraction and VBOW approach + making a submission
- a tutorial notebook on classification using the logistic regression classifier from sklearn
- a tutorial notebook on bias/variance and making a learning curve
  (disclaimer: use as is, it is copied over)
- a notebook on the feature descriptors
- a helper file and a file with functions creating the features
- a folder called 'features', consisting of large pickle files which contain the precomputed feature descriptors for your convenience (created with settings equal to the ones in the 'create_VBOW' notebook)