Error analysis

Error analysis:

- Get 100 mislabeled dev set examples
- Count up how many of them are of one particular class

Knowing that you can take decisions on which class mislabelling you should tackle. This process gives you an estimate of how worthwhile it might be to work on each of the different categories of errors and it five syou a sense of the best options to pursue (maybe blurry images, images of one particular category being mislabeled as another one, etc)

Incorrectly labeled data: It turns out that deep learning algorithms are quite robust to random errors in the training set; if the errors are reasonably random, then it's probably okay. Nevertheless, deep learning algorithms are less robust to systematic errors.

Correcting incorrect dev/test sets: It might only be worthwhile doing it after evaluating the impact using error analysis. If you decide to do it some guidelines are:

- Apply same process to dev and test sets to make sure they continue to come from the same distribution.
- Consider examining examples the algorithm got right as well as the ones it got wrong.
- Train and dev/test data may now come from slightly different distributions. (this might be okay)

Build the first system quickly and then iterate: Set up dev/test set and metrics. Build an initial system quickly and then use bias/variance analysis and error analysis to prioritize next steps.

Mismatched training and dev/test sets

Bias and Variance with mismatched data distributions When training and testing data comes from different distributions you can no longer draw the conclusion that the algorithm is not generalizing if your dev error ir much bigger than the train error.

For this problem is important to create a *training-dev* set that has de same distribution as the training (but it's not used for training). Looking at this error you can see if you have a generalization problem (your training-dev error is similar to the dev error) or a distribution problem (you training-dev is small so the problem is because of the data mismatching)