1. Actually the best is if the training set is balanced, otherwise it is possible that the model will be biased towards the most frequent classes. So make the distribution as uniform as possible. For this you can use the idea from your second question.

2. Actually the idea you described is an often used technique in ML, and it is called Augmentation. There are many different methods how you can add new artificially made images. Just a few of them:

* Slight random rotations
* Slight random translations
* Slight affine transformations
* Flipping the image (if the flipped image trivially belong to the same class)
* Adding noise (salt-and-pepper, Gauss, etc...)
* Adding random shadows
* Making the image brighter/darker
* Changing the contrast in the image
* ...

I think this article can be interesting and relevant for you:

Dealing with unbalanced data

<https://medium.com/@vivek.yadav/dealing-with-unbalanced-data-generating-additional-data-by-jittering-the-original-image-7497fe2119c3>

调参技巧：

Hello there! Very good job on this submission of the Traffic Sign Classifier project. There are some minor things to take into account, but the global outcome is splendid! Congratulations!

Please follow the indications in order to fulfill the rubric requirements. You are almost there!

Regarding your questions:

* One method of balancing a data set is through augmentation. Rotations, translations and scaling are methods used to increase the number of training elements.
* The design criteria depends on the task. If you are satisfied with the results, it is a good design! Of course, this depend on the amount of new test images to evaluate. Have you tried with real images instead of sign drawings? This can also give you a hint if your model is confident enough to be useful in real life. For purposes of this exercise, your model is good enough. It is up to you if you want to go above and beyond!

Thank you for your great effort, and keep up the hard work!

1.Here is a discussion on how to choose the [batch size](https://stats.stackexchange.com/questions/140811/how-large-should-the-batch-size-be-for-stochastic-gradient-descent) of Stochastic Gradient Decent.

<https://stats.stackexchange.com/questions/140811/how-large-should-the-batch-size-be-for-stochastic-gradient-descent>

2.Here is a discussion on the [Adam Optimizer](http://ruder.io/optimizing-gradient-descent/index.html" \l "adam).

[http://ruder.io/optimizing-gradient-descent/index.html#adam](http://ruder.io/optimizing-gradient-descent/index.html" \l "adam)

3.For hyperparameter optimization, you can refer to this [source](http://cs231n.github.io/neural-networks-3/" \l "hyper).

[http://cs231n.github.io/neural-networks-3/#hyper](http://cs231n.github.io/neural-networks-3/" \l "hyper)