

# A Guideline of the Project Write-Up for CS223C

Bangpeng Yao  
Computer Science Department, Stanford University  
bangpeng@cs.stanford.edu

## 1. General Guideline

The purpose of the write-up are three folds. First, you can summarize what you have done in the project and think deep into the project to get ideas about what else you can do. Second, you need to show to the professor and TA how well you have understood the method, why did you choose a specific technique detail, what contributions or extensions have you made when you implement the algorithms. Third, the write-up should also be helpful for the professor and TA to understand your code, i.e. how to generate the results that you reported in your write-up, where to start to read your code, where to find the most important functions in your code, etc.

Please keep the three purposes in mind when you create the write-up. The write-up might affect how many points you can get in the other aspects of the project, such as experiment evaluation.

Your write-up should be formatted using the CVPR L<sup>A</sup>T<sub>E</sub>X template: <http://cvpr2011.org/sites/default/files/cvpr2011AuthorKit-v2.zip>. Unless you have a strong reason (e.g. you have made significant extensions to the original algorithm), **your write-up should not be longer than five pages**.

Please see below for a brief outline of a sample write-up. You do not need to strictly follow this outline. But your write-up should cover most of the contents described below.

## 2. Object Detection

In this section, you need to give a brief overview and your understanding of the problem, state-of-the-art algorithms, etc. Prof. Li's lecture and the papers that we read in class will be useful here. Please cite related papers. You can also read more related papers but this is not required.

## 3. Algorithm

### 3.1. Deformable Part Model

In this section, give a brief description of the algorithm that you are implementing. Instead of simply copying from the original paper, you need to describe your understanding

of the algorithm with your own words.

### 3.2. My Algorithm

Is your implementation exactly the same as the algorithm described in the paper? If they are not exactly the same, why did you make those changes? (The reason might be: (1) you think they can lead to better performance; (2) one component of the original algorithm is very complicated and difficult to implement, so that you used a simpler approach; (3) any other reasons.) Did you make any extension to the original algorithm? There might be some parameters to set in the code. How did you set these parameters? In many cases, the original paper misses some implementation details. How did you address this problem?

In this section, you need to address the above questions. Please note that you do not have to purposely make your approach different from that in the original paper - if you can implement the original algorithm exactly and perfectly, you have done an amazing job.

## 4. Code

This section is the README file of your code. From this section, the professor and TA should be able to know how to run your code to achieve the performance reported by you. You should also give a brief explanation of the key files and functions in your code, so that the professor and TA can look into the details of your code.

## 5. Results

Use this section to give both quantitative and illustrative results of your approach. As for quantitative results, you need to generate the number and curve that evaluate the performance of your implementation using the evaluation metric specified by us. You can vary some of the parameters or change some small components of your method, and then compare the performance of different settings. As for illustrative results, please pick some images and show your results on them. It is encouraged to give your insights about why your method can do well on some of the images, but do bad on some other images.