

# A Guideline for CS231B Project Write-Ups

Jonathan Krause  
Computer Science Department, Stanford University  
jkrause@cs.stanford.edu

## 1. General Guidelines

There are three purposes of the write-up. First, in the write-up you summarize what you've done for the project, which lets you brainstorm about other approaches or things you can do. Second, in the write-up you should demonstrate that you have understood the method. You should describe why you chose particular technical details and any contributions or extensions you have made while implementing the algorithms. Third, the write-up should aid the instructors in understanding your code, *i.e.* how to generate the results reported in your write-up, where to begin reading your code, which functions are the most important, *etc.*

Please keep these purposes in mind when making the write-up. Besides the points for the write-up itself, points for experimental evaluation are also influenced by the content of the write-up.

Your write-up should be formatted using the CVPR L<sup>A</sup>T<sub>E</sub>X template: <http://www.pamitc.org/cvpr15/files/cvpr2015AuthorKit.tgz>. Unless you have a strong reason (*e.g.* you made significant extensions to the original algorithm), please limit your write-up to no longer than five pages.

Below we demonstrate a sample outline for a write-up. Though you do not need to strictly follow this outline, your write-up should cover most of the content described below.

## 2. Image Segmentation

In the section, you need to give a brief overview and your understanding of the problem, state-of-the-art algorithms, *etc.* The lectures and papers covered in class will be useful. Please cite related papers covered in class. You can also read more related papers, but this is not strictly required.

## 3. Algorithm

### 3.1. GrabCut

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

the algorithm in your own words. If appropriate, use equations.

### 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 that you think they lead to better performance or one component of the original algorithm is very complicated and difficult to implement, so you used a simpler approach. Did you make any extension to the original algorithm? How did you set any parameters used in the code? Were any implementation details missing in the original paper, and if so, how did you address this problem?

## 4. Code

This section is the README for your code. From this section, the instructors should be able to know how to run your code to achieve the performance you reported. You should 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 qualitative results of your approach. For quantitative results, you need to generate numbers and curves that evaluate the performance of your implementation using the evaluation metric specified by the instructors. You can vary some of the parameters or change some components of your method and compare the performance of different settings. For qualitative results, please pick some images and show your results on them. It is encouraged to give insights about why your method does well on some images, but poorly on other images.