The Computer Vision Project (draft 1.0)

Anthony P. Reeves

Abstract

Computer vision is concerned with the extraction of meaningful information from image data. Projects on computer vision are often concerned with the development of computer algorithms for specific applications. In general we are interested in demonstrating that new or novel ideas represent an advance over traditional naive methods. The organization of the computer vision project needs to be carefully considered such that it meets requirements for scientific reporting and review. To meet this objective project must include topic research, experimental design, reporting and discussion of experiment results. Further, careful documentation is required for the computer algorithm development. the experiment procedure and the image data documentation. *Perhaps the greatest challenge in designing and conducting a project is not to show that some idea or algorithm under consideration works some of the time, as almost any algorithm can appear to do this on a limited data set, but to show that the algorithm or method has (or does not have) some advantage over simple classical or traditional methods.*

1. Introduction

Projects are modeled after studies that are conducted to advance the state of the art in computer vision. Such studies often focus one of two major goals: (a) to show that a proposed design or algorithm for solving a specific problem is an advance over previous approaches, or (b) to show that a new idea or algorithm has an advantage over previous methods. Most class projects will form into the former, where the specific task is a priori defined; this is considered in detail below. For the “new idea” project the methodology is similar; however, rather than a focus on a single application, testing is typically done with images selected from several different applications that have the desirable property to illustrate the benefit of the proposed method.

This document covers both multi-semester independent projects and class projects which are completed in a much shorter time frame. See Appendix B for explicit class project information.

1.1 The goal of the design project

Computer vision is, in general, an ill-defined problem. There may be images for which a correct outcome may be impossible to determine. Further, in many practical situations the “true value” of the outcome may be determined by human evaluation, which is subject to error. Therefore, to evaluate the performance of a computer vision method it is necessary to employ an appropriate statistical analysis to the outcomes of applying the computer vision method to a representative set of sample input test cases. It is not sufficient to show that it works on a single image.

Therefore, in addressing a computer vision task we do not generally assemble a solution from a set of well-defined rules (as one might do for example in designing a bridge or a building) and claim victory (and have a reasonable expectation that the building when built will serve its designed function and will not fall down). Instead we must construct an experiment that involves a statistically significant number and range of different inputs (images) to quantitatively establish the quality of the proposed solution and to directly compare this solution with other existing approaches. In this way, we can scientifically establish that the new approach represents an advance in the state of the art. Further, by using the quantitate experiment structure we have a vehicle to modify and optimize the performance of our solution through either algorithm modifications or by adjusting algorithm parameters.

1 .2 Project Milestones

The following are the main milestones in the progress of a project:

1. Project Topic Selection
2. Background Research and algorithm selection
3. The Project Proposal
   At this stage the following needs to be clearly specified:
   a. Algorithm design
   b. Experiment design
   c. E1: Identify a data set
   d. E2: Determine how the image data will be documented
   e. E3 Select a quantitative performance evaluation function
4. Project Execution
5. Final Report
6. Final Presentation

2. Background Research: Published Research Papers

Once you have selected a project topic the next step is to research what has already been done in the area of the topic. Quality research papers are often the project of several years of effort. Do not expect that you will just think up a better solution than has already been published. Identify the most recent research in your topic and write up a background section for the final report. This exercise should educate you on the real issues involving in addressing your topic and provide insight into what makes a good solution. Care is needed in searching for algorithm information on the web. Beware fake news and consider very carefully the source of articles that you review. Always seek multiple supporting sources of information for any idea that you wish to explore and carefully consider the quality of these sources as follows:

2.1. Text Books

The high-quality textbooks have usually received extensive reviews and updates. Most of the algorithms that contain have some evidence for utility and in many cases some implementation details are given. However, in general, the description is very brief and for full details you should read the original source articles referenced in the textbook and search for more recent updates. Textbooks may be five years out-of-date when published.

2.2. Journal Papers

Journal papers usually are well reviewed and should contain reasonably biased evaluation of algorithm utility and critical details for algorithm implementation. However, there is a wide range of quality between high-quality journals such as IEEE and, at the other end of the spectrum, journals that are pay to publish. Journal papers are frequently two or more years out-of-date when published.

2.3, Conference Papers

Most of the papers you find in a search will likely be conference papers. Some conference papers are well reviewed but the majority is not; reader beware. Recent conference papers contain the newest ideas. Most of the ideas expressed in conference papers are not worth exploring; however, often the best ideas that will eventually published in journal papers first appear in a conference paper. Conference papers are often short, get to the point, and are easy to read; however, they frequently lack sufficient details for algorithm implementation. If you find an interesting conference paper that is more than a year old, then search for a recent journal paper on a similar topic by the same research group and go to the research groups website to find follow-up information.

2.4. Unpublished reports

This is the lowest quality of information on the web since it has not received any peer review. There may be many reasons, including, for example, contract requirements, for posting a report on-line. If you are interested in an algorithm described in a report then look for other related papers in higher quality sources to better inform your decision.

There are many places that you can start web searches for you research. Two highly recommended locations are Google scholar (http://scholar.google.com/) in general and PubMed (http://www.ncbi.nlm.nih.gov/pubmed) for medical related research.

2.5 References

When making a reference to external information in the text of your report you should provide the lead author name, year of publication, and a reference number as in Smith et al 2006 [15]. The lead author helps identify the work to the reader and the year is useful especially for a field that is rapidly developing.

3. The Experiment Design

The experiment is required to optimize the performance of the solution for the project target and to evaluate the success of that solution. Most algorithms involve a number of parameters that influence performance. Computer vision algorithms require interaction with the user have are much more difficult to analyze than those that are fully automatic; the following is focused on the fully automatic case only. The experiment design is considered in two stages. First we consider how to best evaluate the performance of an algorithm with respect to a given task. Second, we consider how to use the experiment to optimize a specific algorithm for the given task.

One very good approach to experiment design is to review designs that have been reported for this task in high-quality journal papers. These designs have generally had a great deal of detailed planning, both in the overall approach to the problem and in the specific parameters for the specific algorithm including appropriate comparison algorithms.

3.1 The hypothesis

Prior to designing an experiment, it is important to define a clear goal or endpoint. For computer vision tasks three should be a single quantitative quality measure of the algorithm outcome. Second, a level of performance for that outcome for the class of images to evaluate should be specified as a goal; without a goal, you cannot state if your experiment was a success or not. In general, this goal should be related to the state-of-the art as represented by the research background. If a paper reports on a performance level, you should be to relate your expectation to that performance. If nothing appropriate is available, then you should base your goal expectation on a performance level that would be practically useful. The hypothesis section should clearly state the quantitative quality metric and it's required minimum value to achieve the experiment goal.

3.2 Project Types

There are typically major types of project corresponding to the three main stages of image analysis: segmentation, feature extraction and classification. Some recent approaches, such as deep learning, merge these operations together which adds additional consideration.

For segmentation the objective of the algorithm is to identify the set of pixels in the image belonging to the region of interest (ROI). The usual approach is to document a set of images by indicating the through annotation the true values (i.e, the pixel set belonging to the ROI) and to measure the similarity of this to the algorithm outcome.

For feature extraction projects, there are two problems: (a) having segmented images as an input and (b) of realizing a method for evaluating the effectiveness of the features (with a classifier). It is best to have a complete analysis system available and to experiment with different feature sets to determine if better outcomes can be achieved by adding the new features to the system.

Classification tasks are challenging to conduct since they require a set of features and a corresponding set of known outcomes. Most know classifiers have robust implementations that are available on the internet for packages such as R, Python, and Matlab. The project then becomes determining a unique aspect of a specific classifier.

The deep learning approach   employs a single trainable system to "learn" how to solve the problem. Segmentation is not, in general, explicitly realized learned features are used at multiple levels. A major challenge here is that these systems require a very large number of training examples with known outcomes. A second challenge is to define the objective of the project: what method/approach are you proposing to explore? A third challenge is that training requires a large amount of computer resources and this limits the number of training cycles that can performed in the amount of time available for many projects.

3.1 The data set

In order to conduct a meaningful experiment, you will need a dataset of appropriate images than span the range of variations for the given task. The number of images depends upon the task, image availability and the complexity of documentation. For an image segmentation task 50 to 100 images may be sufficient. A classification task may require thousands of images. There are many image datasets available on the web that are adequate for many projects. For medical images the challenges posted on the web is the best resource since releasing patient data for research is not a simple process.

3.2 Dataset Documentation

To evaluate your algorithms, you will need to know in advance what the correct outcome for each case is. In most segmentation problems, the "ground truth" is represented by a binary image in which pixels inside the region of interest are set to true. For classification problems, the correct diagnosis for each case is needed. In most cases the documentation is provided by expert image annotations for segmentations or my patient records for outcomes. In some cases you will need to do your own image annotation. It is critical to identify quality image documentation or to clearly specify how you will do the annotation and how much effort that will require before you start the experiment.

3.3 The experiment evaluation procedure

To effectively evaluate and develop an automated compute algorithm you will need to set up a script based procedure that you can execute automatically with difference image case lists. You may need more than one script; perhaps one for development and training and a second for testing. The usual practice for a project is to establish the experiment procedure first and to test it with a simple baseline algorithm then you will have the appropriate tools for developing and evaluating your main algorithm.

An important design consideration is the aggregate quantitative outcome measure that you will use for optimizing the performance of your algorithm. While you may report on a number of outcome quality measures in your algorithm analysis you must identify a single metric for indicating algorithm improvement before you can optimize algorithm performance.

3.4 Algorithm Specific Experiment Design

4. Project Reports

The body of the report for the project should be structured with the following main sections: introduction, methods and materials, Results, Discussion, and conclusion. The contents of these sections are as follows:

4.1 The Introduction

contain three main sections: a review of the project topic, a discussion of your program implementation, and a description of the experiments that you conducted with a discussion of the results obtained. The report should have several (2-4) pages for each of these sections (the experimentation section may possibly be longer).

In addition, a UNIX style manual page documentation for your program and a listing of your of your program should be included as appendices. Make sure that your program is well commented and that the documentation page contains sufficient details about the programs operation and parameters. See the ``Program Documentation'' section for more details.

In summary, the final report should contain the following:

1. A one paragraph abstract outlining what the project is and what was achieved.
2. A review of previous work done and the current issues in the project topic (including the three references).
3. An overview of the remaining sections of your report (may be one paragraph).
4. A description of the experiments that were conducted
5. A description of any programs that you developed.
6. A section on the results and observations from your experiments.
7. A conclusion, summarizing the achievements of the project and suggesting future work.
8. A listing of your program
9. A one page documentation for each of your programs (including the main scripts that you use to run your experiment. This should be in a style to that used for the UNIX "man" pages. You do not need to actually use the UNIX formatting tools.

4.2 Program Documentation

For all programs that you develop two forms of documentation are required. First there should be comments included in the program; often there are many lines of comments as there are functional lines in a program; however, be careful to use meaningful comments and not pad out the program with useless comments that just explain the program syntax. The second form of documentation is in the style of a Unix man page. All Unix, Linux and VisionX software uses this documentation style. An example of the format with a description of the required contents is given below. One of the main errors in whiting man page documentation is to make the description section too short. Usually several paragraphs are required and including a full reference if the program implements an algorithm reported in a scientific paper.

UNIX man style program documentation

NAME

A one line description with the following format:

<program-name> - <one line description>

The format is important because it is used to define an index entry for the program.

SYNOPSIS

The program name and all possible options and flags, optional parameters should be enclosed in square brackets

DESCRIPTION

A description of what the program does. If the program performs a non-standard algorithm then there should be sufficient information to describe to someone in the field ALL the relevant information. The description may vary in length from one paragraph to two pages. If more information that that is required is should be in a separate document. If the program implements a published algorithm then a complete reference to the published article should be given.

The description should attempt to describe the programs main operation without referring directly to all the options (where possible). Frequently, this can be achieved by describing the action of the program without any of the optional parameters selected. Note, the operation of the program cannot be adequately described with just one or two sentences.

CONSTRAINTS

This section describes any constraints on the data set formats etc. This section is particularly useful for VisionX documentation where a large number of different data formats exist and many programs can only accept a subset of these.

OPTIONS

This section contains a list of the programs command line options. For each option a short but informative description of the options effect and the constraints of any option parameters should be given. If the parameter is optional then its default value should be explicitly given.

AUTHOR(S)

Let the names of the guilty parties be known.

SEE ALSO

List here any closely related programs or programs that are dependent on this program.

Appendix A. Project 10 point checklist

1. *What is the project?*
   A precise description of what is the goal or endpoint of the project. A good project has a single goal.
2. *What has already been done or claimed to have been done for this project*?
   Research: A careful literature review is required. There is no excuse for ignorance of your chosen area. A very good strategy is to replicate an already published algorithm and evaluation methodology. (You can always add your fancy innovations at the end of the project if you are on schedule)
3. *How will I know that my program is working as intended? How can I verify the correct operation?*
   Typically, a test with one or two small (synthetic) test images with a precisely known outcome.
4. *How can I demonstrate the concept of my advanced algorithm?*
   Create a simple (synthetic) test image that clearly shows an improved outcome when compared to a simple naïve baseline method.
5. *How will I demonstrate that my claimed algorithm is an improvement over standard naive methods?*
   First implement a "simple" standard method and use this to validate your experiment and to establish baseline outcomes.
6. What data set will I use to evaluate the algorithm (training set and test set)?
7. How will this data set be documented (have the correct outcomes been pre-established)?
8. What is the evaluation metric?
   How will I know that one algorithm is better than another? a single number.
9. What is the evaluation experiment?
   How will the results be presented (graphs rather than pictures)?
10. How will I optimize the algorithm's settings and parameters through multiple executions of the experiment?

Appendix B. The Class project

Class projects are typically short independent projects that occupy a part of the semester and are a one of the requirements for completing a course. The basic format is similar to that of full independent projects and the milestones are identical. Reporting requirements and the depth of background research may be reduced to fit the limited time and effort for these small projects.

The class projects will be done in groups of two to four students. The project has three main components: topic research, program/algorithm development, and experimentation; each of these will count in the determination of a project grade.

A critical aspect of a course project is topic selection. Since the time to complete a project is just a few weeks, do not spend more than one week on selecting the topic. Also move with all haste to complete the project proposal and schedule. This is critical to identify and address any barriers to success in a timely manner.

In general, each project will require the development of at least one small program. It is to be stressed that the objective of the project is to gain experience with the issues in computer vision and not to write a long fancy program. For a class project the maximum part of the grade that is attributed to the algorithm computer program is 10%; however, complete program documentation as outlined in section XX is still expected for each program or script that is developed by the students for use in the project. The VisionX system provides many convenient features for computer vision experimentation; no credit will be given for reinventing the wheel.

A version of the final report MUST be handed in by the deadline. In some cases, an extension may be given for a report addendum that includes some new and exciting results. As with any report, the final report should include a title, an abstract, an introduction, the body of the report, and a conclusion. See section XX for more details on the project final report.

B.1 Hints for success:

1. Think big plan small
   The scarcest resource in your project is time; budget it wisely. Consider the big picture for your project but also make a solid plan for the minimum project so that if your project idea turns out not to work as expected you can redefine a reduced project goal and still do a good complete project.
2. Know when to hold em, know when to fold em.
   If your project schedule starts to slip know when to switch to a more modest but achievable goal.
3. Learn from history, don't repeat it
   One of the best ways to plan a project is to find a good paper on the topic and see how the authors addressed the issues. Given the size and time constraints of the project it is valid to replicate work that is reported in a published paper. If you achieve this in a timely manner then you may have additional time in which to explore your own variations of the idea. Most published research papers are the result of several years effort in which the authors have had the time to identify

critical issues. The best projects take advantage of the work that has been done before; read the literature FIRST before you make well-known mistakes.

4. The topic is computer vision
   The goal of the project is to provide experience in an advanced topic in computer vision; that is, the extraction of information from image data. Topics that focus on other related topics such as image coding or computer graphics are not usually acceptable for course projects
5. Do not reinvent the wheel
   The VisionX programming environment provides an extensive set of tools for solving computer vision problems; the five scheduled labs are an introduction to the use of these tools. You are expected to make good use of these tools for you project using the computing environment provided. You must get my explicit permission if you wish to use other programming tools. From past experience almost all the projects that have used different programming tools have been disasters.
6. We are on your team
   The course instructor and the TA's expect to have a number of individual meetings with each project group during which time we may be able advise you about or provide you with additional resources that can make your project successful.