



# Final Project Proposal

Prof. Adín Ramírez Rivera  
adin@ic.unicamp.br

## 1 Description

The final project for the class might be the implementation and evaluation of an algorithm from a research paper (better if it is a recent one). The minimum expected work is to replicate an existing work. I encourage you, however, to do more interesting things, and enhance the results and propose an improvement over the paper. The project's theme is very open-ended; ideally, you can incorporate your research agenda in the project. However, if you go this route, you need to use results that you will develop for the project and not just present previous results as part of the course work.

## 2 Project Proposal

The proposal should be a PDF with a title for the project, your names, and two to three paragraphs describing the project idea. **No more than one page!** See § 3 for restrictions on possible papers.

Your proposal should include one to three relevant papers that deal with your problem. If you are going to reproduce a current paper, you can check the relevant references there. If you are going to do something mixed, then cite the related papers that inspired you.

If you will be collecting your data (images or video), describe any special experimental procedure (restrictions in illumination, motion, cameras, etc.). If you are not, you can get data from the author's website, or contact them to get the original data from them. In either case, explain where you will get your data from.

## 3 Project Ideas

You are strongly encouraged to find a topic that is aligned or related to your research topics and that intersect with unsupervised machine learning. In case you want to find something new or interesting, check the best conferences on Computer Vision (), Natural Language Processing (), and Machine Learning ():

CVPR: <https://openaccess.thecvf.com>

ICCV: <https://openaccess.thecvf.com> (only available for odd years)

ECCV: <https://openaccess.thecvf.com> or <http://www.eccv2016.org/proceedings/><sup>1</sup> (only available for even years)

ACL: <https://www.aclweb.org/anthology/venues/acl/> (and local versions, e.g., NAACL and EACL)

EMNLP: <https://www.aclweb.org/anthology/venues/emnlp/>

IJCNLP: <https://www.aclweb.org/anthology/venues/ijcnlp/>

ICLR: <https://iclr.cc>

NeurIPS: <https://neurips.cc><sup>2</sup>

ICML: <https://icml.cc>

---


<sup>1</sup>Previous year to 2016 are acceptable too. They are not listed here due to space constraints.

<sup>2</sup>Formerly known as NIPS at <https://nips.cc>.

Note that you **must** get a paper from one of these conferences that is related to **unsupervised learning**. Do not replicate any paper since it must have a relation to unsupervised learning. Also, consider the computational resources in case you intend to replicate a paper that uses too much resources. You can propose a version that runs on limited resources. You still need to maintain the code on Gitlab, and run the code from the Colaboratory.

## 4 Final Report

Your final report should be written like a paper (you must use the an article format) **with a strict page limit of 8 pages including your references**. You will construct it using the same setup as with previous projects within the Gitlab project.

 You should include a setup for your problem, that is, motivation and related work, as well as your work (methods, experiments, and results). You should focus on what you found and did and did not work as well as the discussion why. You should discuss what you accomplished during the project, what you learned and your application, and evidence of the effort. For the experiments execute your methods several times and produce graphics and plots to summarize your results (check the papers you are taking inspiration from to produce a similar evaluation).

## 5 Evaluation

Your grade will be defined by the following aspects:

1. Project proposal .....	10%
2. Final report .....	90%
(a) Introduction, motivation, and background .....	10%
(b) Related work .....	15%
(c) Explanation of your methodology .....	25%
(d) Experiments, discussion, and results .....	40%


Each item corresponds to the questions and requirements defined in the previous sections. **Your language usage won't be graded**, but your ability to present your results, ideas, and how they are supported will be. Each other point will be evaluated according to the completeness and correctness of the requested items.



## 6 Submission

### 6.1 Code


Your submission must be through your assigned group on Gitlab. Create a repository named `project-final` and commit all the code and report there.

Your submission must have the following subfolders:

- `input`: a directory containing the input assets (images, videos or other data) supplied with the project. Store the images only if they are asked on the project. Otherwise, set up your `Makefile` to automatically download them from a public server or repository.
- `output`: a directory where your application should produce all the generated files (otherwise stated in the problem).  This folder and all its contents must be added to the `artifacts` path on the `.gitlab-ci.yml` setup.
- `src`: a directory containing all your source code. You only need to submit files that are not derived from other files or through compilation. In case some processing is needed, prefer to submit a script that does that instead of submitting the files.

- **Makefile**: a makefile that executes your code through the docker image. An image is already built and available for use ([adnrv/opencv](#) at the docker hub registry). The code will be executed through a standard call to make, so other dependencies must be provided by you under that constraints.  Moreover, note that your code **must** run on the Gitlab executor (pipeline) on the server itself. Hence, you need to use the `.gitlab-ci.yml` configuration. You can base your work on the [demo](#) that already provides an example of such functionality.
- **report**: a directory containing the source files that produce your report.  Your report **must** be written using  $\text{\LaTeX}$  (and friends) and compiled within the pipeline of the repository in a stage named `report`. Consequently, **the PDF must not be committed**. You can use the images from [adnrv/texlive](#) to build your PDFs. **Only the PDF of the report** must be added to the [artifacts path on the .gitlab-ci.yml setup](#), and not other intermediary files of your report (e.g., images, log files, auxiliary files).

Your report must show all your work for the given project, including images (labeled appropriately, that is, following the convention given) and other outputs needed to explain and convey your work. **The constraints of this report are explained in § 4.**

 The last commit that triggered the build must be before the deadline of the submission. Do not worry about the time executing the pipeline. However, **you must ensure that your code works as no attempt will be made to patch or run broken code.**

## 6.2 Proposal

The submission of the proposal is just a PDF that complies with the instructions of § 2. **Submit this through Google classroom.**




## 6.3 Poster

The poster will be presented in the dates given in the calendar of the course. **And you should submit a PDF with your poster in Google Classroom.**

## 6.4 Final Report

The final report should be delivered as with previous projects on the repository of this project.

## 7 Notes

-  Note that there are several implementations that you can find on the internet.<sup>3</sup> This project is for **you to implement** the algorithms. Thus, **do not submit code from others**. And if you re-use code from someone for a non-restricted part, disclose it in your report and code.
-  All the submissions must be self-contained and must be executable in a Linux environment. Specifically, your code must execute in the docker image within the Gitlab pipeline, or within an automated environment that reproduce your findings (in case you need a GPU).
-  It is your responsibility to make sure your code compiles and executes correctly. No effort will be made to run your code besides executing make inside a docker.

---

<sup>3</sup>And even more with these conferences, several authors either publish their code or replicate the code of others. If you are going to replicate a paper you should do it yourself. If you are constructing on top of the code of others, then cite them properly and explain your work and how it differentiates on your report.