CSCI 4830/5722 Computer Vision – Spring 2021
Instructor: Fleming
FINAL PROJECT
Due Tuesday, April 20th, by 10:00pm MDT (implementation)
Due Friday, April 23rd, by 10:00pm MDT (report)

Choose your own Computer Vision adventure

The objective of the final project is to make use of the algorithms and tools learned during the semester, and to develop some creative new applications of computer vision. You and your team will write a detailed report and will also demo the project for your fellow classmates during the last week of classes.

You can work in **groups of at most three students**. We recommend you divide the work, share ideas, algorithms, tools, and choose a collaborative development tool such as GitHub in order to minimize the time needed to meet in person. (Note: disregard advice about meeting in person if you and your teammates live in the same residence). Because of the physical limitations, we understand that one member of the group might not be fully aware of what a partner has implemented.

- The code/implementation part of the assignment can be one submission from only one of the team members, with a clear note/comment of who the team members are.
- The report can have some common parts, but it should be an individual assignment, which will be turned in individually; there will be a separate submission link on Canvas.
- Your final demo should showcase the contribution of each member of the group.

You need to choose one of two project areas:

A. The Photo Sorter project
B. An interactive project of your own choice that includes either
    - An object (or face) recognition/classification task, or
    - An implementation of an algorithm from literature.

Grading will be based on the completeness of the project, the level of functionality attempted, the clarity and thoroughness of the report, and quality of the demo. Further specifics on grading are given for each project below.

Project Details – Option A. Photo Sorter

Over the last decade, digital cameras have overtaken traditional film cameras. As a result, we are amassing ever larger consumer archives of photos as well as large online repositories such as Instagram, Facebook, and Google Photos. Facebook is currently the largest photo online storage site, with over 300 million photos updated every day (300 million users use Facebook Stories, 500 million users use Instagram Stories every day – source:https://zephoria.com/top-15-valuable-facebook-statistics/).

Digital cameras have tended to alter people's photo-taking behavior. Since images are now "free", most people take more pictures. They also take many duplicates in the hope of getting one "really good" shot. But, as these photos come off the camera, they tend to pile up on the drive without any useful sorting or clustring. This is where you come in.

The project's objective is to develop several photo sorting tools. The project ground rules are as follows:

- You are supplied with a folder of digital images. https://drive.google.com/open?id=1YFxLAJx1V2tFRFQ4xOceLGIF9waBa-Zc

  Your job is to detect common attributes among photos and use them to create "clusters" of similar photos. Optionally, you can also apply labels to the clustered photos or save them into subfolders.

- You will create an interactive application interface that can be demonstrated. For example, you can present the user with a menu/choice driven application with functionality such as:
  - load the images set
  - sort images
  - select only:
    - outdoor scenes
    - portraits (of one or a group of people)
    - text documents
    - images containing cars
    - images containing sky
    - images containing flowers
    - images containing buildings, etc.
  - add image to the set and re-sort, etc.

  Note: these are only ideas and suggestions; you are free to come up with your own functions and tasks.
- You can use the programming language of your choice.

There are three levels of grading for this project:

1. *Baseline*: The baseline project is to detect **near-duplicate** photos and to group them. *Near-duplicates* may differ in small details, but the overall composition would be the same. A project that does this well will get 75% credit.
2. *Preferred*: The preferred project will perform baseline operations, but also be able to detect common portions of images and group images based on those. Examples from the menu above are all valid choices. Your project could detect that scenes contain the same people or objects (but might differ in composition)

and will group based on those objects. This does not need to be a general-purpose functionality. That is, it suffices, for example, to recognize some small set of people, and it is ok to assume the people have not altered their appearance. A project that exhibits 2-3 of these additional capabilities would get 100% credit.

3. *Deluxe*: A deluxe project would add additional functionality, a polished user interface, interactivity, or interfaces to other tools (Facebook, Flickr, Instagram etc.). Example: implement interactive tools where, for example, the user can click or circle a part of the image, and then use the properties of that image patch later in your algorithm. Credit for deluxe items will be given on a case-by-case basis.

Project Details – Option B. Interactive project of your choice

Goals: Over the course of the semester we have become familiar with many tools and techniques used in Computer Vision today. With this set of tools you can actually solve some interesting and challenging problems. The final project is your chance to apply some of these tools to a problem that interests you.

Important: If you are a graduate or undergraduate student and you are interested in using machine learning or deep learning, please get approval for your project. Your project must have a not-negligible "Classic Computer Vision" component. Projects that focus solely in training a neural network will not be accepted.

There are a number of components to a good project:

- Choosing the project: It's hard to choose something that is both interesting and doable in 3.5 weeks. Ask for help with this process.
- Formalizing the problem: How do you take an idea like "detecting mouths in a single image" and formalize it and then turning it into something that can be computed. This is connected with the algorithm development part.
- Data collection: You will need data with which to test your algorithm. The safest bet is to use data sets that others have already gathered and made available and that have been used for a similar task to yours. Ask us if you need help. It's not always the best idea to collect new data because it can be time consuming and increase the level of challenge for your project.
- Algorithm development: You can use the programming language of your choice. Set incremental goals. Test your software in stages. Use good programming practices (don't let Matlab/Python make you lazy). Try to break your code! If it works the first time don't believe it; try strange/edge cases.
- Experimentation: You want to show us experiments that illustrate the ideas and the behavior of your method. Don't overwhelm us with results but rather choose good illustrative examples (of success and failure).

- Write-up: A key part of any research project is effective communication of the results. Unlike the initial homework assignments in the course, your grade will be determined in a big part by your write-up.

Tips: Don't get carried away. Choose something that seems small and then set intermediate goals. You want to have a fallback strategy if you don't meet your ultimate goals. These projects should not be basic research. If nobody has ever done this before then the risks are too high for a final project. Choose something from the literature that seems doable and implement that.

Note: the Web is a great source of ideas, papers, data, and even code. If you do get data, code, or ideas from the web you MUST disclose this. Become a fanatic about appropriately citing work from any source. Your write-up should specify in detail your sources and your personal contribution to every task in your algorithm, and every piece of your pipeline. If we find any portion of your submitted solution code on a website, it means you have violated the Honor Code rules.

Suggestions for everyone:

- Start early: This assignment involves plenty of implementation and multiple inter-connected components can make it hard to debug.
- Start with small data: Verify your system on small data before massive processing.
- By aware of the running time: try to optimize your implementation using vector-based operations and avoid to loop for every pixel.

Submission Guidelines (for all projects):

The **Final report** will represent 50% of your grade. Regardless of which project you choose, you should document the following:

- Your initial project idea and project goals.
- Your algorithm/approach and your hypothesis (what will work, what do you think you'll be able to achieve).
- The data analysis and the results: which goals you managed to achieve.
- The computer vision methods you used to achieve these goals.
- If things did not work out, try to offer a reason for it.
- Known limitations of the methods you have used and possible future extensions.
- Your contributions to the project. Because this part of the report should be different for each team member, **we expect individual report submissions for each team member!**
- List at least 3 things you learned during this project.

- *What advice you would give to next year's vision students?* Don't skip this part. We are looking forward to reading your recommendations.

During the last week of classes each team is required to make a demo or presentation of their project. Focus on what your goals were at the beginning, the challenges encountered and your final results. We will create a scheduler on Canvas for teams to present during the live lecture, using the Screen Sharing option in Zoom. We are looking forward to finally meeting everyone.

For teams who are in the Distance section of the course: record and submit a video/screencast, and drop your video file in this shared folder:

> https://drive.google.com/drive/folders/1YGJmu0_w3fKlo-goZU7k65wGDpLjg_88?usp=sharing

Note1: If your project will make use of a **neural network** (CNN or otherwise), it is not sufficient to present just your results. You need to argument your choice of parameters for your network and explain how the training was done. Training from scratch won't work very well and otherwise you'll have to fine tune a pre-trained existing network. Keep in mind the purpose of this project is to put together everything you learned this semester in order to build a computer vision application pipeline.

Note2: If you are combining your course project with a project from another class, you must receive permission from both instructors, and clearly explain in the report the exact portion of the project that is being counted for CSCI 4831/5722. In this case you must prepare separate reports for each course and submit your final report for the other course as well.

The submission for each student should include:

1. Solution code, as a zip file, via Canvas before Tuesday, April 20th, by 10:00pm MDT
2. Written report, as a pdf file, via Canvas before Friday, April 23rd, by 10:00pm MDT
3. (If in the Distance section) A video, via Google Drive folder before Sunday, April 25th, by 11:55pm MDT