**Final Project Details**

**Requirements**

As part of CS 6476, you will be completing a final project, which is an open-ended assignment with the goal of applying skills and knowledge from the class to real-world applications or research problems. Students are expected to be in groups of 4 students. After each group submits a project proposal, you will be assigned a TA that will serve as a mentor. The final project deliverable will be a github website (e.g [template](https://github.com/pages-themes/cayman)) that will thoroughly describe the scope of your project, experiments, and results. The final project will be graded on these 5 main components:

* **Introduction/Problem Definition:** Provide a brief introduction to your project topic and describe why it's an interesting topic to investigate. This is where you want to describe the problem itself and the motivation behind tackling it.
* **Related Works:** Describe related works in your problem space (research papers, libraries/tools, etc.) for existing solutions for this problem or adjacent areas. Make sure to cite papers you reference!
* **Methods/Approach:** Indicate algorithms, methodologies, or approaches you used to craft your solution. What was the reasoning or intuition for trying each methodology/algorithm. What does the overall pipeline look like and the details behind each component? Make sure to establish any terminology or notation you will continue to use in this section.
* **Experiments / Results:** Describe what you tried and what datasets were used. We aren’t expecting you to beat state of the art, but we are interested in you describing what worked or didn’t work and to give reasoning as to why you believe so. Compare your approach against baselines (either previously established or you established) in this section.
* **Conclusion/Discussion:** Summarize your main findings or insights, and discuss what you would do to further this work? What unexpected problems did you notice in your current setup and how would you mitigate them in a future iteration?

**Example Ideas/Projects**

Below are some sample tracks we might expect final projects to fall under (taken from Stanford’s CS231A [course project page](https://web.stanford.edu/class/cs231a/project.html)):

* **Algorithm.** Is there a research problem that interests you? Maybe you’d like to expand upon the idea of an existing paper, or you have a new idea of your own. Depending on how ambitious you are, maybe this can lead to a publication.
* **Paper re-implementation.** Not all research papers provide an open-sourced implementation of their method. If that’s the case for a paper you’re interested in, you’re more than welcome to implement it yourself and try to recreate the experiment results.
* **Application.** Perhaps you’re taking the class with a specific set of interests and experience. Can you apply computer vision techniques to solve a problem in your domain of interest? If so, we encourage you to do so! (One idea is a [kaggle competition](http://www.kaggle.com/))

Example Algorithm Topics

**Limited Supervision**

* **Domain Generalization.** Given access to only a source domain, how can we train a model on this data to perform well on out of distribution target data we have not seen? Compilation of works [here](https://github.com/junkunyuan/Awesome-Domain-Generalization).
* **Domain Adaptation.** Assuming access to a source domain and now unlabeled examples of a target domain, how can we make our model do well on the target domain? Some resources [here](https://github.com/zhaoxin94/awesome-domain-adaptation).
* **Few-Shot Learning.** How can we generalize a model to a new domain given a limited number of labeled examples in our target domain. Example references can be found [here](https://github.com/Duan-JM/awesome-papers-fewshot?tab=readme-ov-file#cv).

**2D Computer Vision**

* **Image Segmentation.** Using a foundation model like Segment Anything Model ([SAM](https://segment-anything.com/)) or simply a [U-Net](https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/) allows for segmenting images into semantic parts. For a compilation of sources see [here](https://github.com/mrgloom/awesome-semantic-segmentation).
* **Object Detection**. Have you found a cool problem to employ object detection? For every repo/blog/dataset/paper on the planet related to object detection, see [here](https://github.com/amusi/awesome-object-detection).
* **Image Stitching (Panoramas).** Did you know a panorama is a collection of images that are aligned and blended together really cleverly? A list of approaches for this can be found [here](https://github.com/tzxiang/awesome-image-alignment-and-stitching).
* **Motion Tracking.** With techniques like [optical flow](https://nanonets.com/blog/optical-flow/), it’s possible to distinguish the motion of pixels between images. See optical flow sources [here](https://github.com/hzwer/Awesome-Optical-Flow).
* **Deblurring**. Want to convert a blurry image into a clear one? A curated list on image and video deblurring resources can be found [here](https://github.com/subeeshvasu/Awesome-Deblurring).
* **2D/3D Human Pose Estimation**. Want to track the skeleton pose of a human from an image/video? For a list of resources (only goes up to 2020), see [here](https://github.com/wangzheallen/awesome-human-pose-estimation?tab=readme-ov-file).
* **Facial Recognition.** Interested in any problems that require reasoning about human faces? For a comprehensive list of resources see [here](https://github.com/ChanChiChoi/awesome-Face_Recognition).

**3D Computer Vision**

* **3D Reconstruction.** It’s now possible to construct extremely accurate 3D representations of objects from just images thanks to Neural Radiance Fields ([NeRF](https://www.matthewtancik.com/nerf)) and [3D Gaussian Splatting](https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/). That means you can extract 3D meshes from just images! For a collection of sources on NeRF see [here](https://github.com/awesome-NeRF/awesome-NeRF).
* **3D Object Detection and Segmentation.**
* **Localization (and mapping).** Building a map in your environment, while also determining your location in the environment are two tasks that solving together compliments one another! We call this simultaneous localization and mapping (SLAM), which is really useful for applications in VR/AR and robotics. For a list of sources see [here](https://github.com/SilenceOverflow/Awesome-SLAM).
* **6D Pose Estimation.** Want to predict the exact position and rotation of an object? All the latest research in this can be found [here](https://github.com/ZhongqunZHANG/awesome-6d-object).

**GenAI**

* **Image Inpainting**. If you mask out a part of an image, can you fill in that region with something plausible? [Here](https://github.com/zengyh1900/Awesome-Image-Inpainting) is a collection of resources for doing this for images and also video.
* **Style Transfer.** Want to transform an image to fit a specific artistic style? A recent survey and compiled resources on style transfer can be found [here](https://github.com/ycjing/Neural-Style-Transfer-Papers).
* **Generative AI.** Isn’t it cool to create a new image/video just from a text description? Resources related to this can be found [here](https://github.com/steven2358/awesome-generative-ai).

**Multimodal**

* **Vision and Language Grounding.** Thanks to models like [CLIP](https://openai.com/research/clip), combining reasoning in language and vision works quite well! For a list of sources see [here](https://github.com/TheShadow29/awesome-grounding).

**Project Proposal Template**

Please fill out this project proposal template before the deadline (**Thursday Feb 22, 2024 11:58 pm**). Submit 1 project proposal document per group. Keep your answers concise and clear!

1. **Team Members:**

* [Member 1]
* [Member 2]
* [Member 3]
* [Member 4]

1. **Problem Definition:** What are you trying to solve or do? Why is this a problem that is useful or interesting to study or experiment with? (2-3 bullet points)
2. **Project Idea:** What do you plan on accomplishing in your project scope? How will your idea solve the problem proposed in your problem definition? (2-3 bullet points)
3. **Related Works:** Briefly describe what work has been done in this area and how they currently approach the problem. List any research papers, existing codebases, or miscellaneous articles/websites that you found that relate to your work. (minimum 5 references; at least 3 should be research papers)
4. **Datasets:** How will you benchmark your approach or theory? Describe and link each dataset you will use in your project, and explain why you selected the dataset. (min 1 dataset)
5. **Proposed Experiments:** What experiments will your team perform in your project? Roughly describe what ideas you will try. (2-3 bullet points)
6. **Compute Resources:** What compute resources (GPUs, CPUs, TPUs) does your team plan to use? The type of compute resources can limit the scope of your project, so we want teams to pick appropriate projects for the resources they have accessible.