University of Michigan
EECS 442: Computer Vision
Fall 2023. Instructor: Andrew Owens.

Project proposal information

Posted: Wednesday, October 11, 2023 **Due:** Monday, November 13, 2023

Please submit your proposal to Gradescope as a PDF file. Please only submit one proposal per group, and add the other members of your group to your submission.

1 Proposal guidelines

For your final project, you'll study an area of computer vision a bit more deeply – for example, by reading and reimplementing a computer vision paper, or by applying computer vision to a task that's important to you.

- You may work in a group of up to 4 people. If you'd like to work in a larger group, that's fine, but please chat with us; our expectations for will be a bit higher.
- This project is open-ended: you can either choose from a list of project ideas from the list below, or you can propose a topic of your own. *We highly encourage you to propose your own project!* You'll probably learn more, and you'll have more fun!
- If you are proposing a new project, you should say: 1) what problem you are addressing, 2) why it is interesting or important, 3) what methods you'll use to solve it, 4) how you'll evaluate your work.

  If you select a project idea from the list, please describe in more detail what you'll do. For example, if you choose an idea from the list that involves reimplementing a paper, please: 1) briefly describe the algorithm in the paper, 2) describe what you will implement, and 3) say how you'll evaluate your model.
- Your project proposal should be short (less than one page is encouraged).
- Projects that overlap with research, side projects, etc. are generally fine. Please note this in your proposal, though.
- You don't necessarily need to use visual data for your project. You can also apply techniques described in the course to other vision-like signals (e.g. audio, LIDAR, medical imagery, etc.).
- If you'd like to significantly change the focus of your project after submission, you may revise your Gradescope submission. For example, if a topic covered later in the course grabs your attention, you may update your proposal to address it instead. We will then review the updated proposal.

- We unfortunately cannot provide GPU resources to groups, beyond what is available on Google Colab. Please keep this in mind when proposing a project. You may simplify your models (e.g. if you are reimplementing a paper, by training them on less data) to help address this.
- We encourage you to come to office hours to discuss your project ideas, so that we can help steer you toward relevant research literature. We may also point you to materials when we grade your proposal.
- The proposal will be worth 5% of your final project grade (and recall that your full final project is worth 30% of your overall class grade).
- Since the project will be due at the very end of the semester, you cannot use late days for it.
- We'll ask you to provide a description of the contributions that each group member made.
- There will be a *rolling deadline* for the project proposal, beginning November 4. You can turn it in as late as November 16.
- The final project writeup and presentation will be due during the final exam period.

2 Project ideas

To help you think of projects, we've provided a few ideas below. Please note that these projects only cover a very small portion of the possible things you can do — most involve reimplementing and extending a paper. We encourage you to propose your own, creative project ideas, and to use these as a starting point! We may also add new project ideas to this list in the coming weeks.

Applications of vision. Apply computer vision to a task that's important to you! We highly encourage this option if it appeals to you, since it's often the most fun option. For example, students in previous EECS 442 classes have applied computer vision to Settlers of Catan, measured the volume of liquid that could be held by a teacup, and analyzed the coffee coming out of an espresso machine. Often these projects will involve applying a few different computer vision models to a task, and analyzing the results.

Image synthesis. Implement a (small) version of a generative image model, such as VQ-GAN [1] or a diffusion model [2].

Extend an existing image synthesis model. Extend an existing image synthesis model, such as Stable Diffusion [3] in an interesting way (see here [4] for architecture details).

Video magnification. Implement a motion magnification algorithm, such as the method of Wadhwa et al. [5]. Try running it on your own videos, too.

Stereo. Implement a system that can estimate depth from a collection of photos using stereo. An easy-to-implement reference point is Goesele et al. [6].

Structure from motion. Implement a simple structure from motion system that can estimate the camera pose of each photo in a video sequence. As a point of reference, consider studying Bundler [7] or the simple SFMedu system.

Reconstructing historical scenes. Recently, Luo et al. [8] obtained a dataset of "antique" binocular stereo pairs, recorded using cameras from the 19th century. For this project, follow [8] and reconstruct the 3D structure of these historical scenes using a stereo depth estimation algorithm, and use a view synthesis algorithm to simulate new viewpoints of these scenes. Extend their work in some way (e.g. by trying to improve the depth estimation, inpainting, or view synthesis methods).

Reconstruct Martian stereo images. Similar to the above, but reconstruct stereo images recorded on Mars![1]

Background subtraction. Zoom's background subtraction algorithm (i.e. the "virtual background" feature) often has a lot of artifacts. Can you do better? See [9] for a state-of-the-art background subtraction method.

Single-image depth estimation. Train a model to predict depth from a single image (see [10] for an example model).

Predict soundtracks from videos. Generate audio to go along with a video [11].

Semantic segmentation. Implement a semantic segmentation method, such as [12], and compare the results to other approaches that use a different network architecture (e.g. compare dilated convolution models to skip connection models).

Self-supervision. Implement a self-supervision method, such as MOCO [13] or CMC [14]. Experimentally evaluate how different design decisions affect performance on downstream classification tasks.

Object detection. Extend your object detection system from PS7. For example, consider converting it to a two-stage model like Faster R-CNN, or have it output instance segmentation masks.

Unpaired image translation. Implement CycleGAN [15]. Extend the model using techniques from GAN literature, and experimentally evaluate how these changes affect performance.

Addressing bias. Analyze bias in computer vision models or datasets, or apply a method to reduce bias. See [16, 17, 18] for examples of related work.

Detecting deepfakes. Train a method to detect fake images and videos. See [19].

Image captioning. Implement a neural captioning model, such as [20, 21].

Transformers. Implement a simplified version of a vision transformer (ViT) [22].

[1] This project idea is from Bill Freeman.

References

[1] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[3] Stable diffusion. https://huggingface.co/spaces/stabilityai/stable-diffusion.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[5] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Phase-based video motion processing. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.

[6] Michael Goesele, Brian Curless, and Steven M Seitz. Multi-view stereo revisited. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2402–2409. IEEE, 2006.

[7] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Siggraph 2006 Papers*, pages 835–846. 2006.

[8] Xuan Luo, Yanmeng Kong, Jason Lawrence, Ricardo Martin-Brualla, and Steve Seitz. Keystonedepth: Visualizing history in 3d. *arXiv preprint arXiv:1908.07732*, 2019.

[9] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2291–2300, 2020.

[10] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[11] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016.

[12] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[14] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[15] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[16] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

[17] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

[18] Hao Liang, Pietro Perona, and Guha Balakrishnan. Benchmarking algorithmic bias in face recognition: An experimental approach using synthetic faces and human evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[19] https://www.kaggle.com/c/deepfake-detection-challenge.

[20] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[21] Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. *arXiv preprint arXiv:2306.07915*, 2023.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.