Crash Course: Advanced Topics in Computer Vision

(Based on: https://cs280-berkeley.github.io/)

Afonso Diela

Table of Contents

Chapter 1: Geometry of Image Formation

Section I: Key Concepts

Overview: This chapter delves into the geometric principles that underpin how images are formed. We explore the relationship between 3D scenes and their 2D projections, focusing on concepts such as perspective projection, camera models, and the geometry of multiple views.

Key Concepts:

- **Pinhole Camera Model:** Understanding the basic model of image formation.
- **Perspective Projection:** Mathematical description of how 3D points map onto a 2D plane.
- **Intrinsic and Extrinsic Parameters:** Camera calibration essentials.
- **Epipolar Geometry:** Fundamental concepts in the geometry of stereo vision.

Section II: Lab Work

Project 1: Implementing the Pinhole Camera Model

- **Task:** Code the transformation of 3D points to 2D image coordinates using the pinhole camera model.
- **Implementation Example:** A Python notebook demonstrating how to project a simple 3D scene onto a 2D plane.

Project 2: Camera Calibration

- **Task:** Calibrate a camera using known 3D points and their corresponding image points to compute intrinsic and extrinsic parameters.
- **Implementation Example:** A step-by-step Python notebook guiding through the calibration process using OpenCV.

Section III: Resources

- *Books:* "Multiple View Geometry in Computer Vision" by Hartley and Zisserman
- *Papers:* “A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms” by Seitz et al.
- *Online:* OpenCV documentation on camera calibration

Chapter 2: Radiometry of Image Formation, Light Fields, Color Vision

Section I: Key Concepts

Overview: This chapter explores the interaction of light with surfaces and how these interactions are captured in images. We also delve into light fields and the complexities of color vision, covering the radiometric principles that govern image formation.

Key Concepts:

- **Radiometry Basics:** Understanding radiance, irradiance, and their role in image brightness.
- **Light Fields:** Representation of the light rays in space and their implications for imaging.
- **Color Models:** Exploring different color spaces (RGB, HSV, etc.) and their uses in computer vision.

Section II: Lab Work

Project 1: Simulating Radiometric Image Formation

- **Task:** Create a radiometric image simulation based on surface reflectance and light sources.
- **Implementation Example:** Python notebook that models how light interacts with surfaces and forms an image.

Project 2: Manipulating Light Fields

- **Task:** Capture and manipulate light field data to refocus images after capture.
- **Implementation Example:** Example using a light field dataset to perform post-capture refocusing.

Section III: Resources

- *Books:* "Radiometry and Photometry in Computer Vision" by Reinhard et al.
- *Papers:* “The Plenoptic Function and the Elements of Early Vision” by Adelson and Bergen
- *Online:* Stanford Light Field Archive

Chapter 3: Foundations of Machine Learning: Neural Networks, Backpropagation

Section I: Key Concepts

Overview: This chapter introduces the foundations of machine learning with a focus on neural networks and the backpropagation algorithm. Understanding these principles is crucial for building and training models in computer vision.

Key Concepts:

- **Neural Networks:** Basics of neural network architectures.
- **Backpropagation:** Key algorithm for training neural networks.
- **Loss Functions:** Role of loss functions in training neural networks.

Section II: Lab Work

Project 1: Building a Simple Neural Network

- **Task:** Construct and train a basic neural network from scratch using Python and NumPy.
- **Implementation Example:** Python notebook that walks through the implementation of forward and backward propagation.

Project 2: Visualizing Backpropagation

- **Task:** Create visualizations to demonstrate how backpropagation updates network weights.
- **Implementation Example:** A Python notebook that uses matplotlib to visualize gradient descent.

Section III: Resources

- *Books:* "Deep Learning" by Goodfellow, Bengio, and Courville
- *Papers:* “Understanding Machine Learning: From Theory to Algorithms” by Shai Shalev-Shwartz and Shai Ben-David
- *Online:* CS231n course materials

Chapter 4: Image Processing

Section I: Key Concepts

Overview: In this chapter, we examine the fundamental techniques of image processing, which serve as building blocks for higher-level vision tasks. Topics include filtering, edge detection, and morphological operations.

Key Concepts:

- **Image Filtering:** Low-pass and high-pass filters.
- **Edge Detection:** Techniques such as the Sobel and Canny edge detectors.
- **Morphological Operations:** Erosion, dilation, opening, and closing.

Section II: Lab Work

Project 1: Implementing Edge Detection

- **Task:** Code and apply different edge detection algorithms to a set of images.
- **Implementation Example:** Python notebook with implementations of Sobel and Canny edge detectors.

Project 2: Image Filtering

- **Task:** Apply various filters to images to enhance or extract specific features.
- **Implementation Example:** Python notebook demonstrating low-pass and high-pass filtering.

Section III: Resources

- *Books:* "Digital Image Processing" by Gonzalez and Woods
- *Papers:* “Edge Detection Techniques for Image Segmentation” by Maini and Aggarwal
- *Online:* OpenCV tutorials on image processing

Chapter 5: Linear Filters: Low-pass, Band-pass, Spatiotemporal

Section I: Key Concepts

Overview: This chapter focuses on linear filtering techniques, including low-pass, band-pass, and spatiotemporal filters, which are essential for analyzing and interpreting image data.

Key Concepts:

- **Low-pass Filters:** Basics and applications in image smoothing.
- **Band-pass Filters:** Extracting specific frequency components from images.
- **Spatiotemporal Filters:** Techniques for processing video and time-series image data.

Section II: Lab Work

Project 1: Applying Low-pass Filters

- **Task:** Implement and apply low-pass filters to images for noise reduction.
- **Implementation Example:** Python notebook demonstrating Gaussian and median filtering.

Project 2: Spatiotemporal Filtering for Video

- **Task:** Apply spatiotemporal filtering to a video to extract motion features.
- **Implementation Example:** Python notebook using 3D convolution for video filtering.

Section III: Resources

- *Books:* "The Essential Guide to Image Processing" by Al Bovik
- *Papers:* "Spatiotemporal Filtering for Video Denoising" by Patrick Pérez et al.
- *Online:* MATLAB Image Processing Toolbox documentation

Chapter 6: Sampling and Multiscale Image Representations

Section I: Key Concepts

Overview: This chapter covers the principles of image sampling and the importance of multiscale representations in computer vision. These techniques are fundamental for efficiently analyzing images at various resolutions.

Key Concepts:

- **Nyquist-Shannon Sampling Theorem:** Ensuring accurate image sampling to avoid aliasing.
- **Pyramidal Representations:** Building Gaussian and Laplacian pyramids for multiscale analysis.
- **Wavelet Transforms:** Decomposing images into different frequency components.

Section II: Lab Work

Project 1: Constructing Image Pyramids

- **Task:** Build Gaussian and Laplacian pyramids for multiscale image representation.
- **Implementation Example:** Python notebook demonstrating the construction and use of pyramidal representations in image processing.

Project 2: Wavelet Transform for Image Analysis

- **Task:** Apply the discrete wavelet transform to images to analyze frequency components at different scales.
- **Implementation Example:** Python notebook using PyWavelets to perform wavelet-based image decomposition.

Section III: Resources

- *Books:* "The Wavelet Transform: Time-Frequency Localization and Signal Analysis" by Stephane Mallat
- *Papers:* “Pyramid Methods in Image Processing” by Burt and Adelson
- *Online:* PyWavelets documentation

Chapter 7: Neural Architectures for Vision: CNNs, Recurrent Networks, Transformers

Section I: Key Concepts

Overview: This chapter focuses on advanced neural architectures that have revolutionized computer vision, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers.

Key Concepts:

- **CNNs:** The role of convolutional layers, pooling, and feature maps in image analysis.
- **RNNs:** Using recurrent connections to process sequential data like video frames.
- **Transformers:** The attention mechanism and its applications in vision tasks.

Section II: Lab Work

Project 1: Implementing a CNN for Image Classification

- **Task:** Build a simple CNN to classify images from a dataset such as CIFAR-10.
- **Implementation Example:** Python notebook using TensorFlow/Keras to implement and train the CNN.

Project 2: Applying Transformers to Vision Tasks

- **Task:** Implement a Vision Transformer (ViT) for image classification or segmentation.
- **Implementation Example:** Python notebook demonstrating how to set up and train a Vision Transformer using PyTorch.

Section III: Resources

- *Books:* "Deep Learning with Python" by François Chollet
- *Papers:* “Attention Is All You Need” by Vaswani et al.
- *Online:* TensorFlow and PyTorch tutorials

Chapter 8: Probabilistic Models of Images

Section I: Key Concepts

Overview: This chapter introduces probabilistic models in image analysis, which are essential for handling uncertainty in computer vision tasks. We cover models such as Gaussian Mixture Models (GMMs) and Markov Random Fields (MRFs).

Key Concepts:

- **GMMs:** Modeling image data as a mixture of multiple Gaussian distributions.
- **MRFs:** Understanding spatial dependencies in images using Markov properties.
- **Bayesian Networks:** Applying Bayesian inference to image-related tasks.

Section II: Lab Work

Project 1: Image Segmentation with GMMs

- **Task:** Use GMMs to segment an image into different regions based on color or intensity.
- **Implementation Example:** Python notebook implementing image segmentation using scikit-learn's GMM.

Project 2: Denoising with MRFs

- **Task:** Implement an MRF-based method for image denoising.
- **Implementation Example:** Python notebook showing MRF-based denoising using Gibbs sampling.

Section III: Resources

- *Books:* "Pattern Recognition and Machine Learning" by Christopher M. Bishop
- *Papers:* “Probabilistic Graphical Models” by Koller and Friedman
- *Online:* Scikit-learn GMM documentation

Chapter 9: Generative Image Models and Representation Learning

Section I: Key Concepts

Overview: In this chapter, we explore generative models that create new image data, focusing on approaches like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). We also discuss representation learning, which is crucial for unsupervised learning in vision.

Key Concepts:

- **GANs:** The adversarial framework and its role in generating realistic images.
- **VAEs:** Probabilistic generative models that learn latent representations of images.
- **Representation Learning:** Techniques for learning useful features from unlabeled data.

Section II: Lab Work

Project 1: Generating Images with GANs

- **Task:** Implement and train a simple GAN to generate new images from a noise distribution.
- **Implementation Example:** Python notebook using TensorFlow to build and train a GAN on a dataset like MNIST.

Project 2: Image Reconstruction with VAEs

- **Task:** Implement a VAE to reconstruct and generate new images.
- **Implementation Example:** Python notebook demonstrating the construction and training of a VAE using PyTorch.

Section III: Resources

- *Books:* "Deep Generative Models" by David Foster
- *Papers:* “Auto-Encoding Variational Bayes” by Kingma and Welling
- *Online:* TensorFlow and PyTorch VAE tutorials

Chapter 10: Challenges in Learning-Based Vision: Dataset Bias

Section I: Key Concepts

Overview: This chapter addresses the challenges associated with learning-based vision systems, particularly focusing on dataset bias. We explore how biased datasets can affect model performance and generalization.

Key Concepts:

- **Dataset Bias:** Understanding how imbalances in data can lead to biased models.
- **Domain Adaptation:** Techniques to adapt models to new domains or datasets.
- **Fairness in Vision Models:** Ensuring models do not perpetuate biases present in training data.

Section II: Lab Work

Project 1: Identifying and Mitigating Dataset Bias

- **Task:** Analyze a dataset for bias and implement techniques to mitigate its impact.
- **Implementation Example:** Python notebook demonstrating bias detection and rebalancing techniques.

Project 2: Domain Adaptation with Transfer Learning

- **Task:** Use transfer learning to adapt a vision model trained on one dataset to a new, different dataset.
- **Implementation Example:** Python notebook showing transfer learning techniques using pretrained models in Keras.

Section III: Resources

- *Books:* "Fairness and Machine Learning" by Barocas, Hardt, and Narayanan
- *Papers:* “Unsupervised Domain Adaptation by Backpropagation” by Ganin et al.
- *Online:* Tutorials on fairness in machine learning

Chapter 11: Stereo Correspondence and Multiview Geometry

Section I: Key Concepts

Overview: This chapter covers the methods and algorithms used to infer depth from stereo images and understand the geometry of multiple views. These techniques are vital for 3D reconstruction and scene understanding.

Key Concepts:

- **Stereo Matching:** Algorithms to find correspondences between two images for depth estimation.
- **Depth Estimation:** Techniques for computing depth from stereo image pairs.
- **Multiview Geometry:** Understanding how to reconstruct 3D scenes from multiple images.

Section II: Lab Work

Project 1: Implementing Stereo Matching

- **Task:** Implement a stereo matching algorithm to generate a disparity map from a pair of images.
- **Implementation Example:** Python notebook using OpenCV to compute disparity and depth maps from stereo images.

Project 2: 3D Reconstruction from Multiple Views

- **Task:** Reconstruct a 3D model of a scene using images taken from different angles.
- **Implementation Example:** Python notebook demonstrating multiview 3D reconstruction using structure from motion techniques.

Section III: Resources

- *Books:* "Multiple View Geometry in Computer Vision" by Hartley and Zisserman
- *Papers:* “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches” by Žbontar and LeCun
- *Online:* OpenCV documentation on stereo vision

Chapter 12: Understanding Motion: Optical Flow, Learning-Based Methods

Section I: Key Concepts

Overview: This chapter explores the detection and interpretation of motion in images and videos. We cover classical methods like optical flow as well as modern learning-based approaches for understanding motion.

Key Concepts:

- **Optical Flow:** Techniques for estimating motion vectors in an image sequence.
- **Lucas-Kanade Method:** A popular algorithm for computing optical flow.
- **Learning-Based Motion Analysis:** Using deep learning to predict motion and understand dynamic scenes.

Section II: Lab Work

Project 1: Computing Optical Flow

- **Task:** Implement the Lucas-Kanade method to compute optical flow in a video sequence.
- **Implementation Example:** Python notebook using OpenCV to demonstrate optical flow estimation.

Project 2: Learning-Based Motion Estimation

- **Task:** Train a neural network to predict motion vectors from video frames.
- **Implementation Example:** Python notebook using TensorFlow to build and train a model for motion estimation.

Section III: Resources

- *Books:* "Computer Vision: Algorithms and Applications" by Richard Szeliski
- *Papers:* “FlowNet: Learning Optical Flow with Convolutional Networks” by Dosovitskiy et al.
- *Online:* OpenCV tutorials on optical flow

Chapter 13: Understanding Vision with Language

Section I: Key Concepts

Overview: This chapter examines the intersection of vision and language, covering topics such as image captioning, visual question answering, and multimodal learning.

Key Concepts:

- **Image Captioning:** Generating descriptive text from an image using deep learning.
- **Visual Question Answering (VQA):** Systems that answer questions about images.
- **Multimodal Learning:** Techniques that combine visual and textual data for richer representations.

Section II: Lab Work

Project 1: Image Captioning with Neural Networks

- **Task:** Implement a model that generates captions for images using a CNN-LSTM architecture.
- **Implementation Example:** Python notebook using TensorFlow/Keras to build and train an image captioning model.

Project 2: Visual Question Answering

- **Task:** Develop a model that can answer questions based on the content of an image.
- **Implementation Example:** Python notebook implementing a simple VQA model using PyTorch.

Section III: Resources

- *Books:* "Deep Learning for Computer Vision with Python" by Adrian Rosebrock
- *Papers:* “Show and Tell: A Neural Image Caption Generator” by Vinyals et al.
- *Online:* COCO dataset and related VQA challenges

Chapter 14: Failures of Computer Vision and Where We Should Go Next

Section I: Key Concepts

Overview: In this final chapter, we reflect on the limitations and failures of current computer vision systems and explore directions for future research and development.

Key Concepts:

- **Common Failures:** Understanding scenarios where computer vision models often fail.
- **Ethical Concerns:** The ethical implications of deploying vision systems in real-world scenarios.
- **Future Directions:** Emerging trends and technologies that could address current limitations.

Section II: Lab Work

Project 1: Analyzing Failure Cases

- **Task:** Collect and analyze cases where vision models fail, particularly in edge cases or biased datasets.
- **Implementation Example:** Python notebook documenting failure cases and exploring potential reasons for failure.

Project 2: Proposing Improvements

- **Task:** Develop and propose a method or architecture that could address a specific limitation in current computer vision systems.
- **Implementation Example:** Conceptual Python notebook outlining a new approach or improvement based on recent research.

Section III: Resources

- *Books:* "Artificial Unintelligence: How Computers Misunderstand the World" by Meredith Broussard
- *Papers:* “Why Are Deep Networks Reversible: A Simple Theory, with Implications for Training” by Behrmann et al.
- *Online:* Discussions on AI ethics and failure analysis from leading AI conferences

Appendix: Additional Resources and References

1. **Geometry of Image Formation**
    - *Books:* "Multiple View Geometry in Computer Vision" by Hartley and Zisserman
    - *Online:* OpenCV Camera Calibration
2. **Radiometry of Image Formation**
    - *Books:* "Radiometry and Photometry in Computer Vision" by Reinhard et al.
    - *Online:* Stanford Light Field Archive
3. **Foundations of Machine Learning**
    - *Books:* "Deep Learning" by Goodfellow, Bengio, and Courville
    - *Online:* CS231n materials
4. **Image Processing**
    - *Books:* "Digital Image Processing" by Gonzalez and Woods
    - *Online:* OpenCV Tutorials
5. **Linear Filters**
    - *Books:* "The Essential Guide to Image Processing" by Al Bovik
    - *Online:* MATLAB Image Processing Toolbox documentation
6. **Sampling and Multiscale Representations**
    - *Books:* "The Wavelet Transform: Time-Frequency Localization and Signal Analysis" by Stephane Mallat
    - *Online:* PyWavelets documentation
7. **Neural Architectures for Vision**
    - *Books:* "Deep Learning with Python" by François Chollet
    - *Online:* TensorFlow and PyTorch tutorials
8. **Probabilistic Models of Images**
    - *Books:* "Pattern Recognition and Machine Learning" by Christopher M. Bishop
    - *Online:* Scikit-learn GMM documentation
9. **Generative Models and Representation Learning**
    - *Books:* "Deep Generative Models" by David Foster
    - *Online:* TensorFlow and PyTorch VAE tutorials
10. **Challenges in Learning-Based Vision**
    - *Books:* "Fairness and Machine Learning" by Barocas, Hardt, and Narayanan
    - *Online:* Tutorials on fairness in machine learning
11. **Stereo Correspondence and Multiview Geometry**
    - *Books:* "Multiple View Geometry in Computer Vision" by Hartley and Zisserman
    - *Online:* OpenCV documentation on stereo vision
12. **Understanding Motion**
    - *Books:* "Computer Vision: Algorithms and Applications" by Richard Szeliski
    - *Online:* OpenCV tutorials on optical flow
13. **Vision with Language**
    - *Books:* "Deep Learning for Computer Vision with Python" by Adrian Rosebrock
    - *Online:* COCO dataset and related VQA challenges
14. **Failures of Computer Vision**
    - *Books:* "Artificial Unintelligence" by Meredith Broussard

- *Online:* AI ethics discussions from leading AI conferences