Computer Vision Nanodegree Syllabus

Contact Info

While going through the program, if you have questions about anything, you can reach us at . For help from Udacity Mentors and your peers visit the Udacity Classroom.

Nanodegree Program Info

Version: 2.0.0

Length of Program: 69 Days*

** This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.*

Part 1: Introduction to Computer Vision

Project: Facial Keypoint Detection

Apply your knowledge of image processing and deep learning to create a CNN for facial keypoint (eyes, mouth, nose, etc.) detection.

Supporting Lessons

Lesson	Summary
Welcome to Computer Vision	Welcome to the Computer Vision Nanodegree program!
Get Help with Your Account	What to do if you have questions about your account or general questions about the program.
Image Representation & Classification	Learn how images are represented numerically and implement image processing techniques, such as color masking and binary classification.
Convolutional Filters and Edge Detection	Learn about frequency in images and implement your own image filters for detecting edges and shapes in an image. Use a computer vision library to perform face detection.
Types of Features & Image Segmentation	Program a corner detector and learn techniques, like k-means clustering, for segmenting an image into unique parts.
Feature Vectors	Learn how to describe objects and images using feature vectors.
CNN Layers and Feature Visualization	Define and train your own convolution neural network for clothing recognition. Use feature visualization techniques to see what a network has learned.

Project: Optimize Your GitHub Profile

Other professionals are collaborating on GitHub and growing their network. Submit your profile to ensure your profile is on par with leaders in your field.

Supporting Lessons

Lesson	Summary
Jobs in Computer Vision	Learn about common jobs in computer vision, and get tips on how to stay active in the community.

Part 2: Optional: Cloud Computing

Part 3: Advanced Computer Vision & Deep Learning

Project: Image Captioning

Train a CNN-RNN model to predict captions for a given image. Your main task will be to implement an effective RNN decoder for a CNN encoder.

Supporting Lessons

Lesson	Summary
Advanced CNN Architectures	Learn about advances in CNN architectures and see how region-based CNN's, like Faster R-CNN, have allowed for fast, localized object recognition in images.
YOLO	Learn about the YOLO (You Only Look Once) multi-object detection model and work with a YOLO implementation.
RNN's	Explore how memory can be incorporated into a deep learning model using recurrent neural networks (RNNs). Learn how RNNs can learn from and generate ordered sequences of data.
Long Short-Term Memory Networks (LSTMs)	Luis explains Long Short-Term Memory Networks (LSTM), and similar architectures which have the benefits of preserving long term memory.
Hyperparameters	Learn about a number of different hyperparameters that are used in defining and training deep learning models. We'll discuss starting values and intuitions for tuning each hyperparameter.
Optional: Attention Mechanisms	Attention is one of the most important recent innovations in deep learning. In this section, you'll learn how attention models work and go over a basic code implementation.
Image Captioning	Learn how to combine CNNs and RNNs to build a complex, automatic image captioning model.

Project: Improve Your LinkedIn Profile

Find your next job or connect with industry peers on LinkedIn. Ensure your profile attracts relevant leads that will grow your professional network.

Part 4: Object Tracking and Localization

Project: Landmark Detection & Tracking (SLAM)

Implement SLAM, a robust method for tracking an object over time and mapping out its surrounding environment, using elements of probability, motion models, and linear algebra.

Supporting Lessons

Lesson	Summary
Introduction to Motion	This lesson introduces a way to represent motion mathematically, outlines what you'll learn in this section, and introduces optical flow.
Robot Localization	Learn to implement a Bayesian filter to locate a robot in space and represent uncertainty in robot motion.
Mini-project: 2D Histogram Filter	Write sense and move functions (and debug) a 2D histogram filter!
Introduction to Kalman Filters	Learn the intuition behind the Kalman Filter, a vehicle tracking algorithm, and implement a one-dimensional tracker of your own.
Representing State and Motion	Learn about representing the state of a car in a vector that can be modified using linear algebra.
Matrices and Transformation of State	Linear Algebra is a rich branch of math and a useful tool. In this lesson you'll learn about the matrix operations that underly multidimensional Kalman Filters.
Simultaneous Localization and Mapping	Learn how to implement SLAM: simultaneously localize an autonomous vehicle and create a map of landmarks in an environment.
Optional: Vehicle Motion and Calculus	Review the basics of calculus and see how to derive the x and y components of a self-driving car's motion from sensor measurements and other data.

Udacity

Generated Tue Jun 11 06:37:46 PDT 2019