INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF ARTIFICIAL INTELLIGENCE

Computer Vision

Nanodegree Program Syllabus

Overview

This program is designed to enhance one's existing machine learning and deep learning skills with the addition of computer vision theory and programming techniques. These computer vision skills can be applied to various applications such as image and video processing, autonomous vehicle navigation, medical diagnostics, smartphone apps, and much more. This program will not prepare learners for a specific career or role, rather, it will grow their deep learning and computer vision expertise and give them the skills they need to start applying computer vision techniques to real-world challenges and applications.

The term is comprised of 3 courses and 3 projects, which are described in detail below. Building a project is one of the best ways to demonstrate the skills one has learned, and each project will contribute to an impressive professional portfolio that shows potential employers their mastery of computer vision and deep learning techniques.

Built in collaboration with:

Program information

Estimated Time

3 months at 10-15hrs/week*

Skill Level

Advanced

Prerequisites

A well-prepared learner should have significant experience with Python and entry-level experience with probability, statistics, and deep learning architectures. They should also have the ability to write a class in Python and add comments to their code for others to read. Lastly, learners should have familiarity with the term "neural networks" and the differential math that drives backpropagation.

Required Hardware/Software

Learners need access to a 64-bit operating system with at least 8GB of RAM, along with administrator account permissions sufficient to install programs including Anaconda with Python 3.5 and supporting packages. The network should allow secure connections to remote hosts (like SSH).

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Introduction to Computer Vision

Master computer vision and image processing essentials. Learn to extract important features from image data and apply deep learning techniques to classification tasks

Course Project

Facial Keypoint Detection

Use image processing techniques and deep learning techniques to detect faces in an image and find facial keypoints, such as the position of the eyes, nose, and mouth on a face. This project tests one's knowledge of image processing and feature extraction techniques that allow one to programmatically represent different facial features. Use one's knowledge of deep learning techniques to program a convolutional neural network to recognize facial keypoints. Facial keypoints include points around the eyes, nose, and mouth on any face and are used in many applications, from facial tracking to emotion recognition.

Lesson 1

Introduction to Computer Vision

- Learn where computer vision techniques are used in industry.
- Prepare for the course ahead with a detailed topic overview.
- Start programming one's own applications.

Lesson 2

Image Representation & Analysis

- See how images are represented numerically.
- Implement image processing techniques like color and geometric transforms.
- Program one's own convolutional kernel for object edge-detection.

Lesson 3

Convolutional NN Layers

- Learn about the layers of a deep convolutional neural network: convolutional, maxpooling, and fully-connected layers.
- Build an CNN-based image classifier in PyTorch.
- Learn about layer activation and feature visualization techniques.

Lesson 4

Features & Object Recognition

- Learn why distinguishing features are important in pattern and object recognition tasks.
- Write code to extract information about an object's color and shape.
- Use features to identify areas on a face and to recognize the shape of a car or pedestrian on a road.

Lesson 5

Image Segmentation

- Implement k-means clustering to break an image up into parts.
- Find the contours and edges of multiple objects in an image.
- Learn about background subtraction for video.

Course 2

Advanced Computer Vision & Deep Learning

Learn to apply deep learning architectures to computer vision tasks. Discover how to combine CNN and RNN networks to build an automatic image captioning application.

Course Project

Automatic Image Captioning

Combine CNN and RNN knowledge to build a deep learning model that produces captions given an input image. Create a complex deep learning model with two components: a CNN that transforms an input image into a set of features and an RNN that turns those features into rich, descriptive language. Implement these cutting-edge deep learning architectures.

Lesson 1

Advanced CNN Architecture

- Learn about advances in CNN architectures.
- See how region-based CNNs, like Faster R-CNN, have allowed for fast, localized object recognition in images.
- Work with a YOLO/single shot object detection system.

Lesson 2

Recurrent Neural Networks

- Learn how recurrent neural networks learn from ordered sequences of data.
- Implement an RNN for sequential text generation.
- Explore how memory can be incorporated into a deep learning model.
- Understand where RNNs are used in deep learning applications.

Lesson 3

Attention Mechanisms

- Learn how attention allows models to focus on a specific piece of input data.
- Understand where attention is useful in natural language and computer vision applications.

Lesson 4

Image Captioning

- Learn how to combine CNNs and RNNs to build a complex captioning model.
- Implement an LSTM for caption generation.
- Train a model to predict captions and understand a visual scene.

Course 3

Object Tracking & Localization

Learn how to locate an object and track it over time. These techniques are used in a variety of moving systems, such as self-driving car navigation and drone flight.

Course Project

Landmark Detection & Tracking

Use feature detection and keypoint descriptors to build a map of the environment with SLAM (simultaneous localization and mapping). Implement a robust method for tracking an object over time using elements of probability, motion models, and linear algebra. This project tests one's knowledge of localization techniques that are widely used in autonomous vehicle navigation.

Lesson 1

Object Motion & Tracking

- Learn how to programmatically track a single point over time.
- Understand motion models that define object movement over time.
- Learn how to analyze videos as sequences of individual image frames.

Lesson 2

Optical Flow & Feature Matching

- Implement a method for tracking a set of unique features over time.
- Learn how to match features from one image frame to another.
- Track a moving car using optical flow.

Lesson 3

Robot Localization

- Use Bayesian statistics to locate a robot in space.
- Learn how sensor measurements can be used to safely navigate an environment.
- Understand Gaussian uncertainty.
- Implement a histogram filter for robot localization in Python.

Lesson 4

Graph Slam

- Identify landmarks and build up a map of an environment.
- Learn how to simultaneously localize an autonomous vehicle and create a map of landmarks.
- Implement move and sense functions for a robotic vehicle.

Meet your instructors.

Sebastian Thrun

Founder and Executive Chairman at Udacity

As the founder and president of Udacity, Sebastian's mission is to democratize education. He is also the founder of Google X, where he led projects including the self-driving car, Google Glass, and more.

Cezanne Camacho

Curriculum Lead

Cezanne is an expert in computer vision with a master's in electrical engineering from Stanford University. As a former researcher in genomics and biomedical imaging, she's applied computer vision and deep learning to medical diagnostic applications.

Alexis Cook

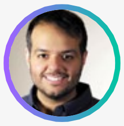
Curriculum Lead

Alexis is an applied mathematician with a master's in computer science from Brown University and a master's in applied mathematics from the University of Michigan. She was formerly a National Science Foundation Graduate Research Fellow.

Juan Delgado

Content Developer

Juan is a computational physicist with a master's in astronomy. He is finishing his PhD in biophysics. He previously worked at NASA developing space instruments and writing software to analyze large amounts of scientific data using machine learning techniques.



Jay Alammar

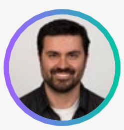
Instructor

Jay has a degree in computer science, loves visualizing machine learning concepts, and is the investment principal at STV, a $500 million venture capital fund focused on high-technology startups.

Ortal Arel

Curriculum Lead

Ortal Arel has a PhD in computer engineering, and has been a professor and researcher in the field of applied cryptography. She has worked on design and analysis of intelligent algorithms for high-speed custom digital architectures.

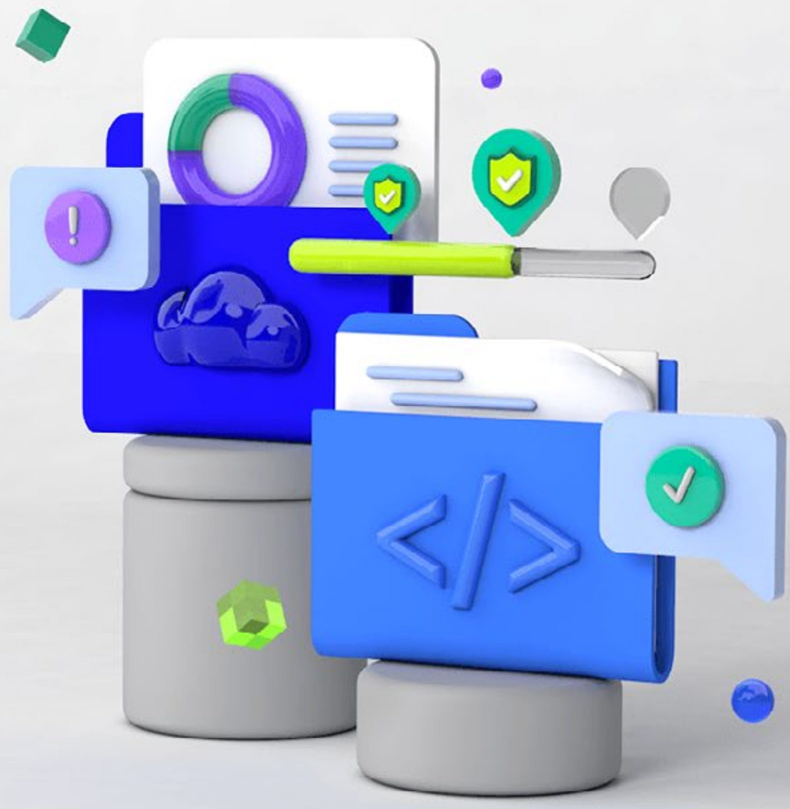


Luis Serrano

Instructor

Luis was formerly a machine learning engineer at Google. He holds a PhD in mathematics from the University of Michigan, and a postdoctoral fellowship at the University of Quebec at Montreal.

Udacity’s learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren’t just multiple choice questions or step-by-step guides, but instead require critical thinking.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.

Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

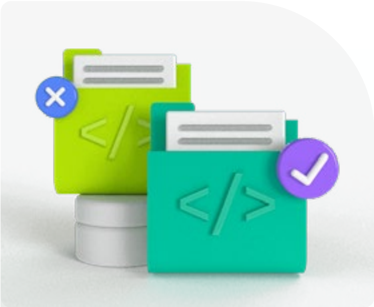
Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

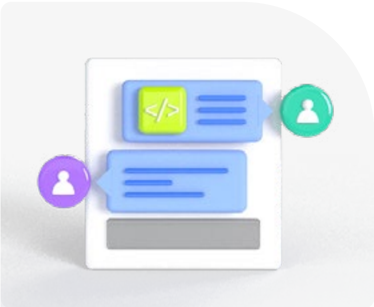
Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

Verify skills mastery.

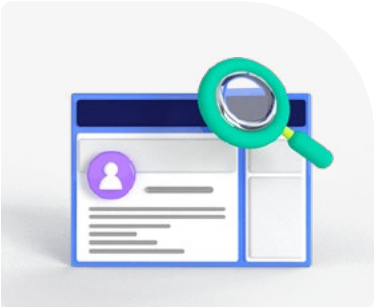
- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.



Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

UDACITY

Learn more at

www.udacity.com/online-learning-for-individuals →