# Computer Vision Nanodegree Syllabus

*Become a Computer Vision Expert*

---

Welcome to the Computer Vision Nanodegree program!

## Before You Start

**Educational Objectives**: In this program, you'll learn the underlying math and programming concepts that drive pattern recognition, object and image classification tasks, and object tracking systems. This course will cover the latest in deep learning architectures used in industry, and you'll combine current computer vision and deep learning techniques to power a variety of applications. With the practical skills you gain in this course, you'll be able to program your own applications, extract information from any kind of image and spatial data, and solve real-world challenges.

**Prerequisite Knowledge**: In order to succeed in this program, we recommend having significant experience with Python, and entry-level experience with probability and statistics, and deep learning architectures. Specifically, we expect you to be able to write a class in Python and to add comments to your code for others to read. Also, you should be familiar with the term "neural networks" and understand the differential math that drives backpropagation. If you feel you need to add to your Python and statistics skills, we suggest our Machine Learning program. If you'd like to learn more about neural networks and backpropagation, consider our Deep Learning program.

**Length of Program**: The program is comprised of 1 term, lasting 3 months. We expect students to work 10 hours/week on average. Make sure to set aside adequate time on your calendar for focused work.

**Instructional Tools Available**: Video lectures, Jupyter notebooks, personalized project reviews.

## Contact Info

While going through the program, if you have questions about anything, you can reach us at cvnd-support@udacity.com.

# Nanodegree Program Info

This program is designed to enhance your 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 you for a specific career or role, rather, it will grow your deep learning and computer vision expertise, and give you the skills you 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 you've learned and each project will contribute to an impressive professional portfolio that shows potential employers your mastery of computer vision and deep learning techniques.

**Length of Program**: 120 Hours*
**Number of Reviewed Projects**: 3

***** 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. Actual hours may vary.

# Projects

Throughout this Nanodegree program, you'll master valuable skills by building the following projects:

- Facial Keypoint Detection
- Automatic Image Captioning
- Landmark Detection and Tracking

In the sections below, you'll find a detailed description of each project along with the course material that presents the skills required to complete the project.

# 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 your knowledge of image processing and feature extraction techniques that allow you to programmatically represent different facial features. You'll also use your 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.

## Introduction to Computer Vision

| Lesson Title | Learning Outcomes |
|---|---|
| **INTRODUCTION TO COMPUTER VISION** | • Learn where computer vision techniques are used in industry.<br>• Prepare for the course ahead with a detailed topic overview.<br>• Start programming your own applications! |
| **IMAGE REPRESENTATION AND ANALYSIS** | • See how images are represented numerically.<br>• Implement image processing techniques like color and geometric transforms.<br>• Program your own convolutional kernel for object edge-detection. |
| **CONVOLUTIONAL NN LAYERS** | • Learn about the layers of a deep convolutional neural network: convolutional, maxpooling, and fully-connected layers.<br>• Build an CNN-based image classifier in PyTorch.<br>• Learn about layer activation and feature visualization techniques. |
| **FEATURES AND OBJECT RECOGNITION** | • Learn why distinguishing features are important in pattern and object recognition tasks.<br>• Write code to extract information about an object's color and |

| | |
|---|---|
| | shape.<br>• Use features to identify areas on a face and to recognize the shape of a car or pedestrian on a road. |
| **IMAGE SEGMENTATION** | • Implement k-means clustering to break an image up into parts.<br>• Find the contours and edges of multiple objects in an image.<br>• Learn about background subtraction for video. |

# Project: Automatic Image Captioning

Combine CNN and RNN knowledge to build a deep learning model that produces captions given an input image.

Image captioning requires that you 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. In this project, you will implement these cutting-edge deep learning architectures.

## Advanced Computer Vision and Deep Learning

| Lesson Title | Learning Outcomes |
|---|---|
| **ADVANCED CNN ARCHITECTURES** | • Learn about advances in CNN architectures.<br>• See how region-based CNN's, like Faster R-CNN, have allowed for fast, localized object recognition in images.<br>• Work with a YOLO/single shot object detection system. |
| **RECURRENT NEURAL NETWORKS** | • Learn how recurrent neural networks learn from ordered sequences of data.<br>• Implement an RNN for sequential text generation.<br>• Explore how memory can be incorporated into a deep learning model.<br>• Understand where RNN's are used in deep learning applications. |
| **ATTENTION MECHANISMS** | • Learn how attention allows models to focus on a specific piece of input data.<br>• Understand where attention is useful in natural language and computer vision applications. |
| **IMAGE CAPTIONING** | • Learn how to combine CNNs and RNNs to build a complex captioning model.<br>• Implement an LSTM for caption generation.<br>• Train a model to predict captions and understand a visual scene. |

# Project: Landmark Detection and 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 your knowledge of localization techniques that are widely used in autonomous vehicle navigation.

## Object Tracking and Localization

| Lesson Title | Learning Outcomes |
|---|---|
| **OBJECT MOTION AND TRACKING** | • Learn how to programmatically track a single point over time.<br>• Understand motion models that define object movement over time.<br>• Learn how to analyze videos as sequences of individual image frames. |
| **OPTICAL FLOW AND FEATURE MATCHING** | • Implement a method for tracking a set of unique features over time.<br>• Learn how to match features from one image frame to another.<br>• Track a moving car using optical flow. |
| **ROBOT LOCALIZATION** | • Use Bayesian statistics to locate a robot in space.<br>• Learn how sensor measurements can be used to safely navigate an environment.<br>• Understand Gaussian uncertainty.<br>• Implement a histogram filter for robot localization in Python. |
| **GRAPH SLAM** | • Identify landmarks and build up a map of an environment.<br>• Learn how to simultaneously localize an autonomous vehicle and create a map of landmarks.<br>• Implement move and sense functions for a robotic vehicle. |