

# Project Overview

Welcome to the Convolutional Neural Networks (CNN) project in the AI Nanodegree! In this project, you will learn how to build a pipeline that can be used within a web or mobile app to process real-world, user-supplied images. Given an image of a dog, your algorithm will identify an estimate of the canine's breed. If supplied an image of a human, the code will identify the resembling dog breed.



Along with exploring state-of-the-art CNN models for classification, you will make important design decisions about the user experience for your app. Our goal is that by completing this lab, you understand the challenges involved in piecing together a series of models designed to perform various tasks in a data processing pipeline. Each model has its strengths and weaknesses, and engineering a real-world application often involves solving many problems without a perfect answer. Your imperfect solution will nonetheless create a fun user experience!

## Project Instructions

### Instructions

1. Clone the repository and navigate to the downloaded folder.

```
git clone https://github.com/udacity/dog-project.git
cd dog-project
```

1. Download the [dog dataset](#). Unzip the folder and place it in the repo, at location `path/to/dog-project/dogImages`.
2. Download the [human dataset](#). Unzip the folder and place it in the repo, at location `path/to/dog-project/hfw`. If you are using a Windows machine, you are encouraged to use [7zip](#) to extract the folder.
3. Download the [VGG-16 bottleneck features](#) for the dog dataset. Place it in the repo, at location `path/to/dog-project/bottleneck_features`.
4. (Optional) **If you plan to install TensorFlow with GPU support on your local machine**, follow [the guide](#) to install the necessary NVIDIA software on your system. If you are using an EC2 GPU instance, you can skip this step.
5. (Optional) **If you are running the project on your local machine (and not using AWS)**, create (and activate) a new environment.
  - **Linux** (to install with **GPU support**, change `requirements/dog-linux.yml` to `requirements/dog-linux-gpu.yml`):
6. `conda env create -f requirements/dog-linux.yml`  
`source activate dog-project`
  - **Mac** (to install with **GPU support**, change `requirements/dog-mac.yml` to `requirements/dog-mac-gpu.yml`):
7. `conda env create -f requirements/dog-mac.yml`  
`source activate dog-project`  
**NOTE:** Some Mac users may need to install a different version of OpenCV `conda install --channel https://conda.anaconda.org/menpo opencv3`

- **Windows** (to install with **GPU support**, change `requirements/dog-windows.yml` to `requirements/dog-windows-gpu.yml`):
- 8. `conda env create -f requirements/dog-windows.yml`  
`activate dog-project`
- 9. (Optional) If you are running the project on your local machine (and not using AWS) and Step 6 throws errors, try this **alternative** step to create your environment.
  - **Linux or Mac** (to install with **GPU support**, change `requirements/requirements.txt` to `requirements/requirements-gpu.txt`):
- 10. `conda create --name dog-project python=3.5`  
`source activate dog-project`  
`pip install -r requirements/requirements.txt`  
**NOTE:** Some Mac users may need to install a different version of OpenCV `conda install --channel https://conda.anaconda.org/menpo opencv3`
- **Windows** (to install with **GPU support**, change `requirements/requirements.txt` to `requirements/requirements-gpu.txt`):
- 11. `conda create --name dog-project python=3.5`  
`activate dog-project`  
`pip install -r requirements/requirements.txt`
- 12. (Optional) If you are using AWS, install Tensorflow.

```
sudo python3 -m pip install -r requirements/requirements-gpu.txt
```

1. Switch Keras backend to TensorFlow.
  - **Linux or Mac:** `KERAS_BACKEND=tensorflow python -c "from keras import backend"`
  - **Windows:** `set KERAS_BACKEND=tensorflow python -c "from keras import backend"`
2. (Optional) If you are running the project on your local machine (and not using AWS), create an IPython kernel for the `dog-project` environment.

```
python -m ipykernel install --user --name dog-project --display-name "dog-project"
```

1. Open the notebook.

```
jupyter notebook dog_app.ipynb
```

1. (Optional) If you are running the project on your local machine (and not using AWS), before running code, change the kernel to match the `dog-project` environment by using the drop-down menu (**Kernel > Change kernel > dog-project**). Then, follow the instructions in the notebook.

**NOTE:** While some code has already been implemented to get you started, you will need to implement additional functionality to successfully answer all of the questions included in the notebook. **Unless requested, do not modify code that has already been included.**

## Evaluation

Your project will be reviewed by a Udacity reviewer against the CNN project rubric. Review this rubric thoroughly, and self-evaluate your project before submission. All criteria found in the rubric

must meet specifications for you to pass.

## Project Submission

When you are ready to submit your project, collect the following files and compress them into a single archive for upload: - The `dog_app.ipynb` file with fully functional code, all code cells executed and displaying output, and all questions answered. - An HTML or PDF export of the project notebook with the name `report.html` or `report.pdf`. - Any additional images used for the project that were not supplied to you for the project. **Please do not include the project data sets in the `dogImages/` or `lfw/` folders. Likewise, please do not include the `bottleneck_features/` folder.**

Alternatively, your submission could consist of the GitHub link to your repository.