28 September 2017

PROJECT PLAN

55 Machine Learning for Automated Insect Classification

PREPARED BY: TEAM 11

David Arnold

Francisco Bencomo

Sadai Sarmiento Carneiro

Aaron Hulseman

Larry Preuett

Table of Contents

[**Project Description**](#_gjdgxs) **2**

[Overview (CO-5):](#_30j0zll) 2

[Key Requirements (CO-5):](#_qa5fn5lbz3wl) 2

[Deliverables (CO-5, CO-6):](#_3znysh7) 2

[Acronyms and abbreviations (CO-7):](#_2et92p0) 3

[**Design and Architecture (CO-1, CO-3)**](#_tyjcwt) **3**

[**Implementation Strategy**](#_3dy6vkm) **4**

[High-level Work Breakdown Structure (CO-2):](#_sitdn8qsoypj) 4

[Schedule / Timeline (CO-2):](#_4d34og8) 6

[Required Hardware (CO-2):](#_2s8eyo1) 7

[Third party content (CO-2):](#_17dp8vu) 8

[Quality (CO-2):](#_3rdcrjn) 8

[Other Special considerations (CO-7, Co-3):](#_26in1rg) 8

**P**[**rocess**](#_lnxbz9) **9**

[Process Description and justification (Co-2)](#_35nkun2) 9

[Tools (CO-2):](#_1ksv4uv) 9

[Roles and Responsibilities (CO-2):](#_44sinio) 9

[Location of Project Artifacts (CO-2):](#_2jxsxqh) 9

[Sponsor communications (CO-7):](#_3j2qqm3) 9

[**Risk management**](#_1y810tw) **10**

[identified Potential risks (CO-2):](#_4i7ojhp) 10

[mitigation strategies (CO-2, Co-3):](#_2xcytpi) 10

# Project Descriptio**n**

## Overview (CO-5):

This project is a data science project that aims to develop a system to automatically identify insect specimens. This system would primarily help researchers studying biodiversity, behavior, and ecology of ants but also provide means to support amateur entomologists in the field as well as to generalize to other insect species. The project consists in the development of a system composed by: 1) a deep learning image classification model, 2) a back-end software running on a server to automatically classify pictures of insect specimens using the model, and both 3) a web-based interface and 4) an app-based interface to upload pictures and visualize classification results.

## Key Requirements (CO-5):

* Description:
  + Deep learning model(s) must be able to identify ant genus from an image with 80% accuracy and provide a reasonable suggestions about insect species/higher category.
  + Web based interface should be able to upload an image file to the backend back-end server.
  + Android and iPhone apps should also be able to upload images to the back-end software.
  + All software should be open-source.
  + Back-end software should be written in TensorFlow.
  + In general, all software should be written in as modular a manner as possible, but especially the back-end server so that it might be used in later machine learning projects.
* Notes:
  + Frameworks for Android and iPhone interfaces are TBD.
  + Both mobile interfaces, and the web-based interface should be very similar in functionality.
  + Requirements are expected to change as more is learned about TensorFlow and Machine Learning.

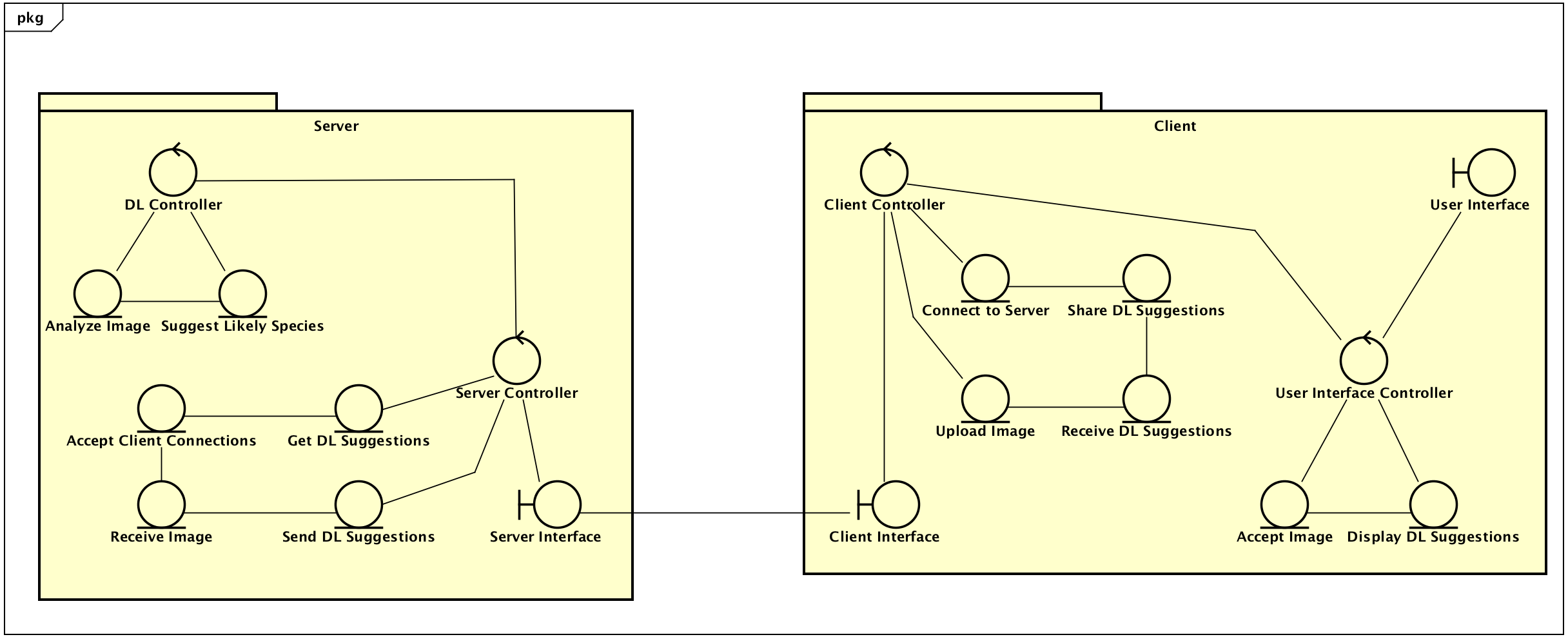
## Deliverables (CO-5, CO-6):

* Web-scraped datasets consisting of images from the public databases AntWeb.org and BugGuide.net. The datasets will be curated and annotated with the help from project sponsors. The datasets need to be augmented using standard techniques in order to increase their size, improve accuracy of the resulting model, and reduce chances of overfitting.
* Deep learning image classification model(s) (Convolutional Neural Networks) trained and tested to recognize insect taxa on the labeled AntWeb and BugGuide datasets as well as appropriate scripts to incrementally retrain the model given new image data. Web interface that allows users to upload images to be classified by the model and to visualize the results of the classification process.
* Back-end software running on a web server that allows to query the deep learning model after uploading a new image using previously mentioned interfaces and to return the results of the classification process. The back-end software should come with a clear API explaining the querying protocol.
* Front-end interface Android and iPhone apps that allow users to upload images to be classified by the model and to visualize the results of the classification process.
* Documentation for all software delivered.

## Acronyms and abbreviations (CO-7):

* DL - Deep Learning
* ML - Machine Learning
* CNN - Convolutional Neural Network
* TBD - To Be Determined
* SQL - Structured Query Language
* CO - Course Objective
* API - Application Programming Interface
* UI - User Interface
* SDLC - Software Development Life Cycle
* CI - Continuous Integration
* IDE - Integrated Development Environment

# Design and Architecture (CO-1, CO-3)



Client:

An abstraction of the three client applications our team will develop.

* The User Interface
  + The boundary between the system and the user.
* The User Interface Controller
  + Handles input from user
  + Displays system output
* Client Controller
  + Submits user data to Server
  + Receives Server response
* Client Interface
  + Interface the Server utilizes to communicate with the Client

Server:

An abstraction of the server application our team will develop.

* Server Interface
  + Interface utilized by the Client to send and receive messages
* Server Controller
  + Submits user data to the Deep Learning model
  + Sends Deep Learning model response to Client
* Deep Learning Controller
  + Controls access to the Deep Learning model
  + Accepts images to classify
  + Returns set of suggestion species/genus.

# Implementation Strategy

## High-level Work Breakdown Structure (CO-2):

* Description:
  + Deep learning classification model
    - Deep learning model can be broken down into several tasks.
    - The learning algorithm must be developed first to feed data into.
    - The learning model is then fed the data sets and tested with test data.
    - If the model meets an 80% success rate, it will be deemed ready for market. If 80% success rate is not achieved, the learning algorithm will be revisited and trained until an 80% success rate is met.
    - The model will accept an image and return the taxonomic classification of the best match along with a link for additional information.
    - In addition to the best match, a list of 3 other potential matches in order of likelyhood will be returned to the client.
    - The skill set required for this task is experience in Python, and Machine Learning algorithms.
    - This deliverable will likely take the longest amount of time. Our team will need time to become familiar with new ML frameworks, building datasets, and techniques to train a CNN.
  + Back-end server
    - Server software should be able to create client-server connections with iPhone, Android and web interfaces.
    - It should also be able to accept image files from them and run them through the DL model.
    - The server will use this data to access an insect database and return insect bio to the client.
    - Knowledge of server-client interfaces and SQL will be necessary for this deliverable.
  + Front-end iPhone, Android, and Web interfaces
    - These applications provide an interface for the user to take a picture of an insect and upload it to the server for analysis.
    - The app should allow the user to select an image on their phone or computer, likely from within the device’s photo gallery.
    - It should create a client-server connection to the server and upload the image file.
    - Application should display results from server to the user.
    - This will require familiarity with iPhone, Android, and Web development. There is experience on the team for web and mobile development, and most are enrolled in courses covering this topic.
    - It is expected to take two sprints to complete these deliverables.
  + Documentation
    - Design documentation will occur during the life of the project. It will be worked on by all team members to help with completeness.
  + Datasets
    - Source of most data will be ImageNet, AntWeb, and BugGuide.
    - It should first be decided what data needs to be saved about each image and how it should be formatted in code and classified. e.g. An ant might be classified as one species, but later, re-classified as a different species. Therefore, a data-repository for ant classifications needs to be located and used.
    - Skillset required - Data Augmentation techniques so to increase the size of the dataset by creating multiple different images from the same image.
    - It then needs to be decided how much data needs to be labeled for the neural network or if the model will be trying to learn and label some of the data as it is being input into the learning model. The appropriate amount of data should then be labeled.
    - Help will be needed from the sponsors to help annotate and curate images to label data for the neural network.
    - Data should also be separated into learning data and test data. The general ratio according to preliminary research suggests this is about an 80/20 split.
    - Building data sets will be time consuming as the team will need to web scrape images, augment images, and annotate and curate images for both an ant model, and a general insect model.

## 

## Schedule / Timeline (CO-2):

* Milestones:
  + Detailed product plan based on Project proposal, and Team discussions with Sponsor.
    - Timeline: Sept 24 - Oct 6, 2017.
    - Deliverable: Product plan delivered on Oct 6th.
  + Understanding of TensorFlow and applying DL processes through self study, and from creating simple models by following examples found online.
    - Timeline: Oct 4 - Oct 31, 2017.
  + Completed data set to train DL model for classification of ant species.
    - Timeline: Oct 31 - Dec 15th, 2017
    - Deliverables:
      * Web crawler for scraping images. Delivered Nov 10, 2017.
      * Data set with augmented images. Delivered Dec 10, 2017.
  + Trained DL model that classify ant images with minimum 80% accuracy.
    - Timeline: Dec 16 - Dec 31, 2017
    - Deliverable: DL ant classification model. Delivered Dec 31, 2017.
  + Back-end Server
    - Timeline: Jan 1 - Feb 7, 2018
    - Deliverable: Functional server that receives image, process image, and return results. Delivered Feb 7, 2018.
  + Data set to train DL model for classification of North American insects.
    - Timeline: Feb 7 - Feb 21, 2018
    - Deliverable: Data set with augmented images. Delivered Feb 21, 2018.
  + Trained DL model that classify North American insects with minimum 80% accuracy.
    - Timeline: Feb 21 - Mar 7, 2018
    - Deliverable: DL North American insect model. Delivered Mar 7, 2018.
  + Web application with user interface for submitting images for classification.
    - Timeline: Mar 1 - Mar 31, 2018
    - Deliverable: Web application. Delivered Mar 31, 2018
  + Android and iOS mobile application with user interface for submitting images for classification.
    - Timeline: Mar 1 - Apr 21, 2018
    - Deliverable: Android and iOS application. Delivered April 21, 2018.
  + Design documentation to aid future evolution of system.
    - Timeline: Sept 24, 2017 - Apr 21, 2018
    - Deliverable: Design and Operational Manual. Delivered Apr 21, 2018

## Required Hardware (CO-2):

* Deep Learning Machine: Spectrum TXN003-0128N
  + DL machine provided by Sponsor.
  + DL machine is a shared system permanently located at Sponsor’s site.
  + Remote access granted to Development Team.
  + DL machine specifications:
    - CPU Intel Xeon E5-2630
    - RAM 256 GB
    - GPU Nvidia TITAN XP \* 4
    - SSD 500GB \* 2
  + HDD 10TB \* 3
* MacBook Pro 2016 with Touch Bar
  + CPU 2.6 GHz Intel Core i7
  + RAM 16 GB
  + SSD 512 GB
  + MacOS Sierra 10.12.6 (High Sierra)
* Lenovo Yoga 3
  + CPU 2.2 GHz Intel Core i5
  + RAM 8 GB
  + SSD 103 GB
  + Windows 10 version 10586
* HP ENVY
  + CPU 2.6 GHz Intel Core i7
  + RAM 16 GB
  + Windows 8.1
* Desktop (Custom Build)
  + CPU 4.0 GHz Intel i7 2700k
  + RAM 32 GB
  + Storage
    - 1.5TB RAID 0 (2 Mechanical Drives)
    - 512 GB RAID 0 (2 MLC SSDs)
  + Windows 10
* Phones for testing mobile applications:
  + iPhone X with iOS 11
  + Galaxy Note 4 with Android 6.0.1 Marshmallow
  + LG G6

## Third party content (CO-2):

* Images used for training the DL model fall under Fair Use.
* Images used in returned results must meet the following conditions:
  + AntWeb: Content is licensed under a [Creative Commons Attribution License](http://creativecommons.org/licenses/by/4.0/). Each image must include attribution to its photographer and "from www.AntWeb.org" in the figure caption. For websites, images must be clearly identified as coming from [www.AntWeb.org](https://www.antweb.org/), with a backward link to the respective source page.
  + ImageNet: Provides original resolution of image upon request for research and educational purposes. Lower resolution images are provided freely to avoid copyright issues.

## Quality (CO-2):

* Deep learning classification model
  + The model shall classify user images properly with at least 80% accuracy.
    - Reinforced via automated validation tests triggered by model updates.
  + The model shall reflect any potential training dataset updates within 7 days of said updates being entered into the system.
* Code quality
  + All code shall live in a source control system.
  + All code shall be written following SOLID principles.
    - Reinforced via code reviews and static code analysis policies.
  + Code that handles business logic shall be unit testable.
    - Code coverage goal: 80%.
  + Unit tests shall run every time code is pushed to source control repository.
  + Unit test failures shall prevent developers from merging code into mainline.
  + Web UIs / APIs shall be tested in an automated manner via integration tests.
  + Integration test failures shall block automated production deployments.

## Other Special considerations (CO-7, Co-3):

* Not all data provided by BugGuide.net is complete. Many images do not fully come with all the data necessary to train a DL model. It is anticipated that considerable time will be required to curate and annotate images retrieved from the website.

# process

## Process Description and justification (Co-2)

Our team has elected to follow the Agile framework Scrum. This process was selected because of the complexity of the project and our unfamiliarity with the technologies required. Utilizing the Agile process will allow our team to easily adapt to changing requirements that are likely to occur in this project. We are adapting the process by changing stand up frequency to 3 days a week to occur on Monday, Wednesday, and Friday, and cycling the scrum master the 1st of every month.

## Tools (CO-2):

* TensorFlow 1.3 - Software library for Machine intelligence (<https://www.tensorflow.org/>)
* Data resource for testing DL model: ImageNet ([www.image-net.org/](http://www.image-net.org/))
* Data resource for building Data sets: AntWeb (<https://www.antweb.org/>)
* Data resource for building Data sets: BugGuide (<http://bugguide.net/>)
* Online Scrum Board: Taiga (<https://taiga.io/>)
* Source Code Control: git and GitHub (<http://www.github.com/>)
* IDE: PyCharm (<https://www.jetbrains.com/pycharm/>)
* CI: Travis CI (<https://travis-ci.org/>)

## Roles and Responsibilities (CO-2):

* Engineer: Responsible for every facet of the SDLC from inception to delivery including, but not limited to: design, programming, quality, testing, and documentation.
* Minutes Keeper: Responsible for recording meeting minutes. This role will rotate amongst the development team on a weekly basis.
* Scrum Master: Keeps sprint on track and works with project sponsors to remove blocks. This role rotates amongst the development team on a monthly basis and begins the 1st of the month and ends the last day of the month.

## Location of Project Artifacts (CO-2):

* Github will be the platform used to house the project’s artifacts. Github was chosen due to its’ familiarity amongst the team, the tools made available for version control, and for it’s ease of use. In addition, Github is well integrated into PyCharm making it easy to perform Git-related tasks right from within the IDE. To visit the project’s repository, please [click here.](https://github.com/dmarnol2/Deep-Learning-TensorFlow-Python)

## Sponsor communications (CO-7):

* The sponsor and team will meet weekly during the planning phase of development. Meeting frequency will be reevaluated as the project progresses.
* Google Hangouts will be used for all meetings requiring video conferencing.
* Slack will be used for any communication not involving video conferencing. Used for day to day communication, sharing files, links, brief thoughts and/or inquiries.

# 

# Risk management

## identified Potential risks (CO-2):

* Technical
  + Development team may have to spend extra time revisiting the machine learning model due to lack of experience. The impact should be noticeable early on and can potentially extend the modelling phase beyond our initial timeline estimates.
* Communication
  + Team members may not be available for meetings due to scheduling conflicts.
* Infrastructure
  + We will rely heavily on 3rd party websites and APIs for our datasets. We expect these resources to be available throughout our project (at the very least during the data scraping phase.) Any down-time or rate limiting can have a major impact on the proposed timeline
  + The ML model will be trained on shared hardware resources. We will need 1) proper machine / network access and 2) we will need to work with the owners of these resources to make sure we get enough time and storage space to complete our model training iterations.

## mitigation strategies (CO-2, Co-3):

* Technical
  + Consistent effort, time management and collaboration we believe will help us overcome our inexperience.
* Communication
  + Meeting minutes will be made available for each member to review should they miss a meeting. Slack provides a quick way for members to communicate ideas or asks questions.
* Infrastructure
  + We will attempt to perform data scraping during low traffic hours so as to minimize the impact of rate limiting. We will also plan to perform our data scraping with ample time to alleviate any potential risk of the data scraping causing delays in progress of the project.
  + Our sponsor has been working on getting our team access to said resources. We plan to continue asking for updates at each meeting.