GDLS-HIDE Progress Report

Fall 2022

Eva Muller
Michigan Tech University

HU 3120: *Technical and Professional Communication*
Instructor: Andrew Blick

Report Date: November 7, 2022

Table of Contents

Executive Summary

This report details the progress that has been made on the GDLS-HIDE project over the course of the project's length. This project is sponsored by General Dynamics Land Systems and is based around developing tools for building datasets of military images. This data will be used for training, validating and testing convolutional neural networks for combat material classification. The team first researched background information on the specified military classes, as well as best technological practices for web scraping, object detection, and CNNs. Then, tools were built for executing these tasks, including a web scraper for Google Images and a video scraper. The preliminary dataset was collected using these tools, as well as statistics to go along with the data. In the future, the team hopes to refine the dataset and focus on building tools for object detection and classification using CNNs.

GDLS-HIDE Progress Report

Introduction

This project is sponsored by General Dynamics Land Systems and is based around developing tools for building datasets of military images. These data sets will then be used for training, validating and testing convolutional neural networks for combat material classification.

By using convolutional neural networks, or CNNs, military perception technology can identify foreign military vehicles and personnel in the field. This has the potential to provide those involved in a situation, crucial information about the environment. However, these CNNs require large datasets to classify the objects with acceptable confidence. So the main issue at hand is how to design a process that is reliable, reusable, and optimizes the users time to collect the data and verify their accuracy and credibility. The HIDE student team has been tasked with the following tasks: Scrape images from the web, Catalog in a hierarchical manner, Verify labeling, Remove duplicate images, and Report Analysis.

Report Problem Statement

This report details the progress the Humane Interface Design Enterprise student team has made so far over the course of the second semester of the project. Specifically, it details the progress made by one of the team members, Eva Muller.

Project Background Research

The team, as relatively inexperienced with working with training data and image processing, decided to do preliminary research on training data collection best practices.

Automated Web Scraping
After looking at open-source solutions and applications for automated web scraping, it was evident that Python would be best suited for our purposes. Favored for applications ranging from web development to scripting and process automation, Python is quickly becoming the top choice among developers for artificial intelligence, machine learning, and deep learning projects. The benefits of this language includes the extensive selection of libraries and frameworks, the simplicity, and abundance of support. Instead of re-inventing the wheel, our team was able to use many of Python’s already developed, tested, and documented libraries.

Web Scraping

Google Images Web Scraping
The Google Images scraping allows us to collect image results from a Google Images search query. After facing some obstacles with the all-purpose scraper, we realized that most images are available publicly on Google Images. On this tab of the main scraper GUI, the user can enter the destination folder, search query, and maximum number of thumbnails to search. The GUI then opens a Chrome browser and automates the process of clicking through each thumbnail and extracting and downloading the image contents.

For technologies used, we first used a python library, google_images_download, but found that the library was limited to 100 image results by Google's default. While there were a few work-arounds, the team decided we wanted more control over the automated Chrome browser process, especially for future use. The Selenium WebDriver drives a browser natively, as a user would, either locally or on a remote machine using the Selenium server. Furthermore, it supports all major browsers on the market such as Chrome/Chromium, Firefox, Internet Explorer, Edge, Opera, and Safari. Currently, we have the GUI tailored to use with Chrome, but we plan to expand this set of browser support. For image download and processing, we are using PIL, or the Python Imaging Library. Additionally, we are using multithreading with Python's threading library, to create separate threads for the processes of each image to speed up the overall scraper process, as will be explained down below.

Dataset Results, Statistics, and Findings

Collection Process
The current dataset was collected over the span of ~5 weeks (more details can be found in the individual details.csv files about the dates). The team used solely the Google Images scraper module for collection so far, entering in various queries for each class and searching for a maximum of 10,000 thumbnail images. In all cases, there were no more than 10,000 Google results, so this covered all the current published images at those times. No other parameters were specified in these searches other than the query and safe mode off. After the base collection was performed, the team went through a manual filtering process to simply eliminate images that were completely irrelevant to the class, did not include a military vehicle, etc. In each case, those images were put into a folder, filtered_out_1 inside each class folder. It is important to note that the current structure is slightly different from the proposed/end goal in this early development.

Layout

In each class folder under Russia, there should be images that have gone through our first filtering process and passed, a filtered_out_1 directory (with the incorrect images), and a details.csv file including more detailed information about all images in the class folder (including subfolders like filtered_out_1). Any inconsistencies in the exact format of information collected can be attributed to the iterative changes made to the scraping programs over the weeks, and will be cleaned up as part of our work next semester.

Generating Statistics
To generate statistics on the dataset, we created a program statistics.py that analyzes the base folder given by the user and for each subfolder, outputs the total number of images, minimum, maximum, and average pixel x pixel size to a specified text file. This file can be found in the root directory in our current dataset, as stats.txt.

Current Statistics
In total, we have collected 7,350 filtered images (not in filtered_out_1 folders) using the process mentioned above. This includes a preliminary round of manual filtering, although another round is most definitely needed.

Conclusion/Future Plans

The next semester's work will focus on building on and refining the dataset, automating the filtering and verification process, and CNN selection and testing. For the specific classes to be collected, we will also be adding Russian military personnel to the collection and possibly Chinese military vehicles and personnel. Some of our tools should be adjusted to accommodate this addition of personnel identification. We have a lot of ideas for improving the Google Images Scraper module to specify the queries further and collecting copyright information.

The information that is collected about the images through the scraping process will then be used to migrate our manual sorting into an automated process by creating programs to identify trends and patterns in the collected image information. Manual sorting will most likely still be used in this situation to assist the learning process for these programs to make suggestions about the credibility and accuracy of the images. There is a lot of work to still be done in this area. We have a slight head start on the CNN selection and testing with the use of the YOLO object detection application. We plan on transferring to Scalable YOLO, as suggested by some members of GDLS, and identifying smaller objects in each image to identify key characteristics that may be unique to each class. The additional annotations can also provide the resulting CNN more detailed training data.

References

Selenium, Software Freedom, 2022, https://www.selenium.dev/.

“Google Search URL Parameters [Ultimate Guide].” *SEOquake*, 4 Dec. 2020, https://www.seoquake.com/blog/google-search-param/. *Selenium*, Software Freedom, 2022, https://www.selenium.dev/.