# **III. Method**

### **Data collection procedure**

The following tools were leveraged to generate the dataset needed for training the model.

1. **Beautiful Soup**

A python library that is designed mainly for web scraping objects such as texts and images. In this project, Beautiful Soup was used in scraping (1) product images and (2) product URLs from Canadian Tire website.

1. **Selenium**

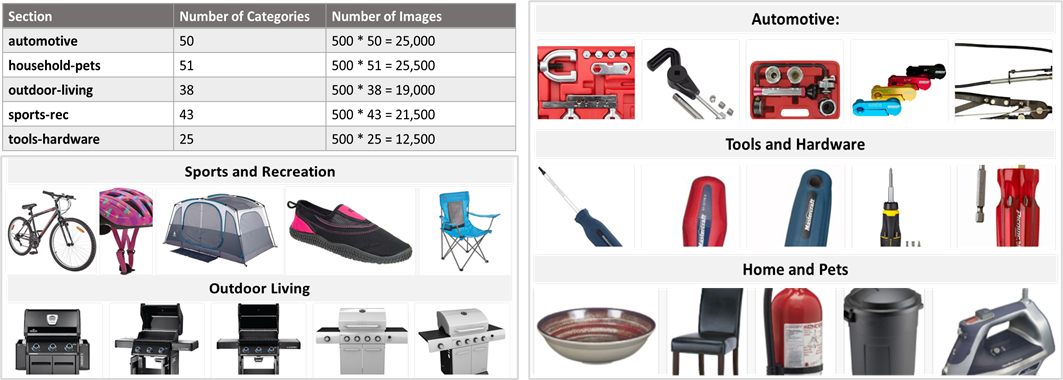
Selenium.webdriver module functions as navigator to a page given by the URL. This was used alongside Beautiful Soup and headless browsers (Chromium and Firefox) to automate the selection and click actions required to extract the objects needed from Canadian Tire.

1. **Python**

Python codes for each five sections were created separately to extract samples.

**Dataset – Canadian Tire images**

Dataset consist of 103,500 train images designated across five product categories. Overall, categories or class used for this multi-class classification model is: 207



***Figure 1*** *– Sample images extracted per section*

### **Data pre-processing steps**

Image or digital pre-processing was performed to improve the image features of the dataset. Pre-processing is vital to computer vision models as it allows any algorithm to comprehend the input data better. Below are the steps performed on this section:

1. **Read images**

In this step, all the collected images were stored to designated folders in Google Drive, where each folder pertains to their class/category names. Python script was created to count the number of original images and the augmented images that needs to be generated on the next step.

Background pattern

Description automatically generated with medium confidence

***Table 1 –*** *Sample data from read\_img table*

1. **Augment the images**

In Convolutional Neural Network, the main objective of data augmentation is to avoid overfitting. Overfitting happens when the model is trained with images with high variance, such as those images scrapped from Canadian Tire. Therefore, augmentation was performed as part of pre-processing phase.

*AugLy* was utilized to perform random transformations on the train images. This is a newly released data augmentation library by Facebook and supports four modalities including image. These functions were used primarily to transform the dataset.

* Scale – randomly resize the image
* Saturate – increase the intensity of image colors
* Shuffle – shuffles the pixels of the image
* Pixelization – pixelates an image
* Rotate- randomly rotates the image to a certain degree angle
* Blur – changes the sharpness of an image
* Change Aspect Ratio – randomly changes the height and width of an image
* Sharpen – changes the sharpness of an image
* Perspective Transform

1. **Label the dataset**

In this step, dataset features (X) and labels (y) were saved as pickle files. Pickle is a python module that allows user to save an array into disk by serializing the object structure before saving it.

### **CNN Model Architecture**

### **Web application design**

A web application made with Streamlit was created as user interface. Streamlit is an opensource framework that can be used to run machine learning models in a graphical user interface such as web browser. Based on the several trials made using different platforms (Flask, Django) - Streamlit was found to be the most efficient tool for this project.

**Front-end process**

The web application workflow is straightforward. Initially, user needs to select the category of their product, then upload a photo, and wait until the app detects the correct product name.



***Figure 2*** *– Web UI workflow*

**Back-end process**

1. Once the user uploads a photo, application calls the model for the selected section. The uploaded image will then be pre-processed through these steps:
   1. Converting to grayscale
   2. Resizing to 256 x 256 (trainset dimension)
   3. Converting to array and reshape to (-1, 256, 256, 1)
2. Prediction returns a set of arrays and will be parsed to get the class with highest prediction percentage.

Graphical user interface, application

Description automatically generated

***Figure 3*** *– Prediction output*

1. The predicted class will be mapped against the datasource table (major\_category\_id). Datasource contains a set of data with category names and Canadian Tire URL for each product.



***Table 2 –*** *Preview of datasource table*

### **Deployment to cloud**

After the completion of development phase, deployment to cloud was performed. The objective of this project is to build a simple, straightforward, and well-structured application where any changes can easily be implemented. As the web application was effectively created using Streamlit framework – proper installation and port set-up needed to be considered during deployment. Additionally, five separate model files with approximately 7GB size needed to be stored into the web server. Hence, virtual server with suitable specifications was leveraged to host the web application.

Initially, the app was deployed to Heroku, a cloud platform offering free-tier hosting with Dynos and 512MB memory. It was effective during development and testing phase; however, it will not be sufficient to perform as production server due to free-tier limitations. Thus, another cloud provider was considered.

*Graphical user interface, application

Description automatically generated*

***Figure 4 –*** *Web application architecture*

**Azure Virtual Machine**

Azure is a cloud service provider by Microsoft. It offers free-tier credit of $200 on the first month and a wider range of instance types to choose from compared to Heroku. The virtual machine used in this project has standard 4 virtual CPUs with 16GiB memory and 30GB storage OS disk. The server runs in Linux (CentOS 7.9) operating system in which can be accessed via SSH using a PEM (Privacy Enhanced Mail) key file.

**Docker**

Docker was leveraged to containerize the application. It enabled the team to easily compile, deploy and migrate the whole set-up within instances or servers. Docker is the leading container platform service in the market today. It has a central meeting place for container developers which is called Docker Hub where compiled images can be uploaded to be stored and be shared to community as well.

The application has been published to Docker Hub with public setting and can be pulled by running this command in a Docker engine:

Graphical user interface, text, application

Description automatically generated