**Image Dataset Generation**

Working on images and media files for a data science is always a very exciting prospect. When it comes to using images, most of us retort to using an annotated dataset or use a set of images for manual annotation. Whilst this is a good approach, we have to keep in mind that Machine Learning and Deep Learning algorithms need a very high a volume of input training data to train a model. As easy as it id to find simple images, there would be a few cases where the dataset of images is not available or is very scarce, leading to lack of training data. And even if similar images were to be available, manually annotating them is a tedious and time-consuming task.

The idea of the project is to propose a systematic method to generate images that suit specific requirements. We focus highly on creating images that have all visual components that is expected of it, and use algorithmic methods to obtain the results. Since the end goal is to generate dataset and not artistic images, we used royalty free image vectors for all the image components. The image components are chosen randomly and used, but since we know exactly which components we are adding to these images, we obtain the annotation details without additional effort.

We use the python package Pillow for image manipulation. The flowchart below shows the entire process briefly.

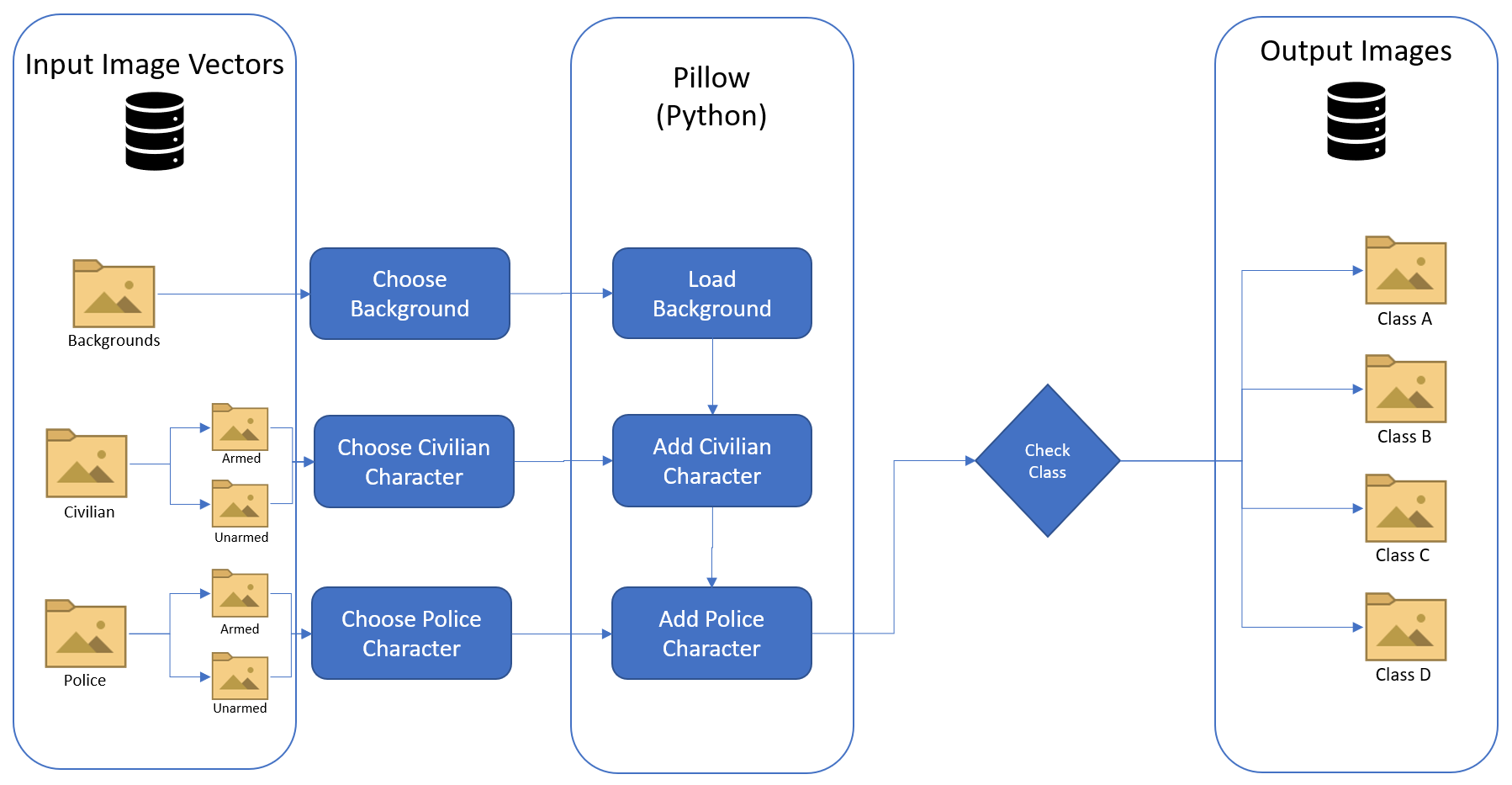


Fig: Image Generation Process Flowchart

The attributes we used three types: the background, the civilian character and the police character. For each of them we set a probability distribution in the form of a percentage to define how common each of the features will be. To ensure the balance a dataset, it is ideal to have equal probabilities for all the different options since we want our final model to predict all different types of cases, i.e., all the images in a particular attribute will have the same chance of being picked by the algorithm for image generation and all the generated classes of images will be equally distributed as well to ensure Equally likely events. In our project, we generated 20000 images in total.

With the above points in mind, we created 4 classes in our dataset:

* Class A : Civilian armed, Police armed (5000 images)
* Class B : Civilian unarmed, Police armed (5000 images)
* Class C : Civilian armed, Police unarmed (5000 images)
* Class D : Civilian unarmed, Police unarmed (5000 images)

We created the images on a local machine using Python multithreading for parallelization. We created a csv file containing the annotations for all the file names for future reference. Using these images, we trained the models that we chose by splitting them into test and train dataset. Following is a brief set of specifications of the image generation module:

* Image Resolution: 1920x1080
* No. of images: 20000
* No. of classes: 4
* Total Execution Time: ~6 minutes
* Packages used: Pillow, NumPy, tqdm, os, threading