Apparel Detector

ABSTRACT: Object detection technology has various applications, especially in the field of fashion where applications can take the form as apparel recognition, fashion recommendation, and online shopping. This project seeks to detect and correctly identify images using Tensorflow Object Detection API and evaluate the process in doing so. Looking at the some of the difficulties such as memory and time constraints as well the detecting accuracy with the range of characteristics of apparel and human bodies, detection is not an easy task.

Object detection is an extremely beneficial technology in the field of computer vision that I personally use on a regular basis via the Google Images and Pinterest image detecting software. These platforms take an image and search for other images that resemble it, usually providing me with information about the image's origin. For Pinterest, my primary use for the app is to find pieces of clothing or outfits that I find inspiring, and so the ability to uniquely identify clothing in a photo and being able to match it to another photo with the exact items or ones that are similar is a very practical use of fashion detection. Furthermore, the platform will usually trace the store which the clothing came from along with a price tag, showing its applications for online search and shopping. With this tool in mind, I thought I could create something similar and make a detector that would detect the category of clothing in generally any photo with humans in it. Of the many categories that exist I decided to narrow it down into common Western style fashion groups; shirts, pants, shoes, jackets and purses. I believed most outfits belong into the five groups, so these were reasonable choices to create the detector for. This project ended up being more complicated than I had anticipated, a few of the challenges being creating a workable dataset, as well as troubleshooting with downloading the required files for training the dataset.

Some background for my project was the works of Chu-Hui Lee and Chen-Wei Lin[1] who conducted a very similar project, with the task of identifying apparel in the categories; jacket, top, pants, skirt and bag. Their goal was to create a faster, more

---

[1] Lee, C.-H.; Lin, C.-W. A Two-Phase Fashion Apparel Detection Method Based on YOLOv4. Appl. Sci. 2021, 11, 3782. https:// doi.org/10.3390/app11093782

efficient detector, which was implemented using an algorithm called YOLOv4-TPD (YOLOv4 Two-Phase Detection). Mainstream algorithms are based on convolution neural networks, or CNN, which can come in a one-stage or two-stage detection task. The first task is region proposal, while the second is classification, and one-stage detections attempt to perform both tasks in one-stage in order to reduce time. Two-stage detections are more accurate than one-stage because the tasks are split, however have a high time complexity and so can be inefficient. YOLO is a one-stage detection algorithm which only needs to perform one convolutional calculation on the image to determine the target category and location. Apart from the advantage of faster detection speed, this algorithm in later volumes would have optimization features that enhanced data, such as normalization methods, data imbalance processing, replacing activation functions, and optimizing bounding box regression problems. It even reduced requirements for hardware with the optimization of networks. To improve on YOLOv4, Lee and Lin incorporated another stage in YOLO used for transfer learning, which takes information learned in the first step, or source domain and applies it to the second, or target domain. The process was divided into two phases where the five categories; jackets, tops, pants, skirts and bags were simplified into three; clothes, bottoms, and goods. Images were first labelled into the three categories, and then enhanced to reveal more clear contours and improve contrast in images. The images were then trained, and then the process of relearning occurred where image model weight was saved and used as the new pre-trained model for the stage 2 model training. Images were relabeled except the clothing category was divided into tops and jackets, whereas the bottom category was shorts and skirts. This way the system would only have to differentiate between tops and jackets instead of all five categories. In the end, their results proved to be successful, with the YOLOv4-TPD model being more precise than the YOLOv4-TL, YOLOv4-CLAHE and YOLOv4.

My methodology for this project was rather a tedious and overly time-consuming process. As I quickly realized, the five different categories that I had chosen for detection had a large amount of variety, meaning collecting data for the set of images to train and test would also have to be varied in order to account for the different styles. I also had no reference for what an appropriate number of images in a dataset would be, so I really just had to guess and manually download what I deemed enough images with little style changes in each category. For example, with the pants category I had to manually search for skinny jeans, straight jeans and baggy jeans for the dataset. I also had to account for the different angles and positions the person could be standing or sitting in, as I wanted my detector to work on practically any image. I carefully chose specific images that covered a small range of poses. Once I downloaded the images I wanted to use for training, I used an open source image labelling software called LabelImg where I was annotating category and bounding box information for all instances in every image. During the labelling process I ran into the instance where the categories were not overly visible or the shape of the item was not abundantly clear, for example someone's shoes could be mostly hidden in their pants or their shirts were mostly hidden behind a cardigan or sweater. I had to make the decision whether or not I should label these, and this decision could drastically alter my results. After the images were labelled, XML file formats were exported for each image through LabelImg, which contained information about the image file name, path, size and the coordinates for the bounding box(s). I divided my dataset into 20% of the data as test labels and 80% as train labels. Then I converted the XML files in each group into two csv files and created a label map for the labels. For the project I was following instructions for setting up the Tensorflow Object Detection API, by Gilbert Tanner[2] with an accompanying YouTube video, but ran into many troubleshooting issues that he did not have. These issues were always caused by missing files or non-existent commands which were not always clear how to resolve them. This required me to search online for ways to install packages manually, which sometimes required other packages or commands to be installed as well. At one point I even ran out of storage completely on my computer, requiring me to delete some files and restart the training process. Other YouTube videos would not have any of the problems I was having either, and they were generally specific to the Windows operating system, while I was using Mac OS. This made it difficult to import the libraries I needed for tensorflow to work and I finally decided that there had to be another way around this. Fortunately I came across a Google Colab by YouTuber

[2] Gilbert Tanner. (2020, August 25). Tensorflow Object Detection with Tensorflow 2: Creating a custom model [Video]. YouTube. https://www.youtube.com/watch?v=cvyDYdI2nEI&t=312s

techzizou[3] which allows users to execute script commands online without taking up space on their computers. This made the process much easier and I was finally able to train my images and test them without the hassle. For the detection model, I used the SSD-MobileNet V2 Trained on MS-COCO Data, which is a one-stage object detection model. I used this model because it was one-stage, meaning it was faster than two-stage models and I didn't want a model that would take a long time to train. For the batch-size in the model config file, I set it to 1 due to the limitations of the GPU, and the steps parameter was set to 10000 steps. I finally loaded the tensorboard and began the training, and attempted to stop the process when I noticed the loss reached below 0.1.

As for the results, the model was accurate in some ways but inaccurate in others. I noticed the system recognized t-shirt cropped tops to be shirts more than other types of tops. Likewise, straight pants were labelled and recognized more easily than other types of pants. Jackets were difficult to distinguish from shirts and were usually labelled as a shirt, especially if it was shorter. Trench Coats and other longer jackets were not recognized very well, if at all by the system. Shoes were recognized if they were angled horizontally, and platform boots would be labelled correctly but given multiple bounding boxes. Unfortunately, the purses were not identified at all. Overall the confidence levels in detecting pants were higher than any other category. I believe my results were due to my dataset which only totaled to around 200 images, and if I were to reconduct this project I would opt for a larger scale project. Some additional information that I was given with the Tensorboard was that the learning rate begins to decrease at around 2.5k steps if the smoothing is set at 0.6 (Figure 1). The losses are also shown to depreciate as steps increase (Figure 2). The overall detection time was roughly 9 minutes and 12 seconds.

My project covers a very small range of possibilities when it comes to computer vision as well as the future of fashion. We can already see changes to the way we shop for clothes through object detection with the introduction of online shopping where users can query what they are searching for or even search for items using images. This technology can also be seen in use with big data companies that work with retail brands

[3] techzizou. (2022, January 1). Train a Deep Learning model for custom object detection using TensorFlow 2.x (On Google Colab) [Video]. YouTube. https://www.youtube.com/watch?v=amURyS6CAaY&t=110s

to track users' information and target personalized ads based on their specific apparel preferences. Technology is fast pacing in the field of the fashion industry, so future investments in fashion technology may prove to be beneficial and fuel a new generation of technologies.

Figure 1

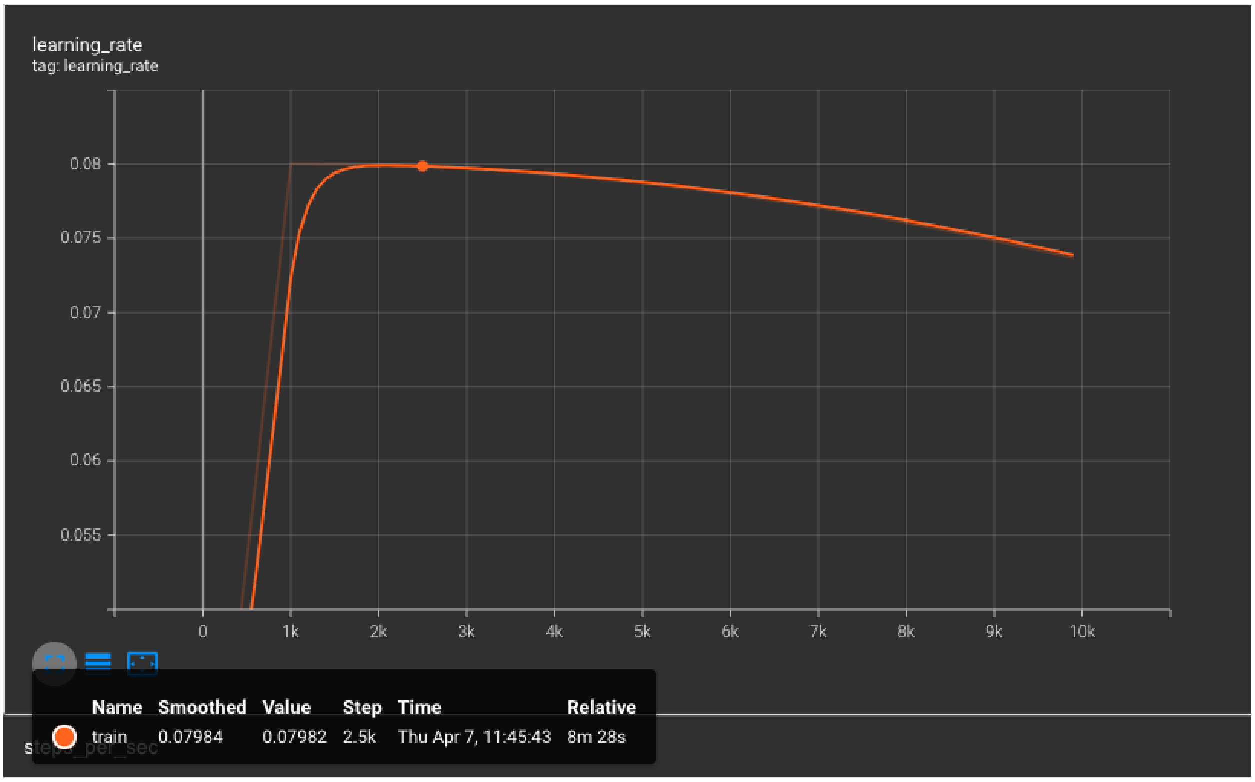

Figure 2