Exploring State-of-the-Art Techniques for Object Detection and Instance Segmentation for Apparel

LIU, Yuting
yliuhs@connect.ust.hk

LEE, Hsin-ning
hleeau@connect.ust.hk

1. Introduction

Choosing the right set of outfits can be challenging for the general public as they usually don't have stylists or exceptional fashion tastes. Though the salespersons might be there to assist, people often prefer referencing or even imitating celebrities' outfits. To approach this problem, we will build a model that can segment apparel and classify them into different categories.

2. Problem Statement

This research aims to tackle the fashion selection dilemma faced by the general public by developing a computer vision-based system that can segment and classify apparel items worn by celebrities. The objective is to analyze these segmented apparel items and identify different combinations of clothes that exhibit visual harmony and align with the current fashion trends, empowering the general public to make informed fashion choices.

2.1. Data Set

We will leverage the provided dataset from Kaggle, including a taxonomy built by fashion experts and the product descriptions available online. This dataset comprises over 40,000 celebrity images for training, 3000 images for testing, 300,000 annotations, and 45 apparel categories. On average, each image has 8 annotation records that include ImageId, EncodedPixels, Height, Width and ClassId.

2.2. Expected Results

Since we will be using the pre-trained model from MMDetection as the backbone for our own model, we can expect a slightly lower mean Average Precision (mAP) of approximately 30. This decrease in performance can be attributed to the smaller size of our training dataset compared to the dataset used to train the pre-trained model. However, we are taking steps to improve the performance of our model. We plan to fine-tune the existing model by incorporating augmented data and introducing new layers. With these enhancements, we aim to achieve a higher mAP of 40 by the conclusion of our research.

2.3. Evaluation Metrics

To evaluate the performance of our model, we will employ two quantitative metrics and one qualitative method. First, we will compute each segmentation's Intersection over Union (IoU) and compare it with a predefined threshold, typically set above 0.5. This threshold will help us determine the accuracy of the segmentations as it informs us of the acceptability of the overlapped ratio. Additionally, we will utilize the mAP to measure the accuracy of object detection and segmentation, which provides an overall assessment of our model's performance. In addition to the quantitative evaluation metrics, we will incorporate visual inspection to assess the effectiveness of instance segmentation. As apparel tends to overlap with each other and such circumstances do not follow a general pattern, we will visually examine the segmented outfits and compare them with ground truth. By obtaining subjective feedback from users, we can improve our model's performance further.

3. Technical Approach

We will employ MMDetection, an open-source object detection toolbox developed by OpenMMLab, as the foundation of our model. MMDetection boasts impressive performance, with an average precision of 52.8 for object detection tasks and 44.6 for instance segmentation. The model consists of five distinct categories of components, which are as follows:

1. **Backbone**: A Fully-Connected Convolutional Network to extract feature maps

2. **Neck**: The connection bridge between backbones and heads

3. **Head**: Body for computing bounding box prediction and mask prediction

4. **RoI Extractor**: Extract RoI features from feature maps

5. **Loss**: Loss Calcuation in head

Given the flexibility of the *mmdet* library, we intend to tailor each component to our specific needs based on the initial results obtained from the base model. Following this approach, we aim to introduce new components, particularly the Double Head R-CNN (also referred to as Cascade R-CNN). By incorporating an additional stage of refinement through an additional bounding box head, this variant surpasses the capabilities of the standard Region-based Convolutional Neural Network (R-CNN). We anticipate that this enhancement will significantly enhance object detection performance, resulting in higher accuracy levels.

In terms of evaluating the performance of our model, we can utilize the evaluations functions available in the *mmdet* library to assess the mAP on the testing dataset and measure the recall based on IoU for segmentation. To gain deeper insights into the model's tradeoff between recall and localization accuracy, we will analyze the Recall-IoU curve. This curve allows us to observe how different IoU thresholds impact the model's recall performance. Moreover, it facilitates our understanding of the model's ability to generalize across varying levels of object overlap, which poses a particular challenge in apparel object detection and segmentation tasks due to the frequent occurrence of overlapping instances.

4. Intermediate Results

4.1. Data Pre-processing

As we are integrating MMDetection into our model pipeline, we must prepare our data in the appropriate Common Objects in Context (COCO) format. This entails deriving segmentation masks, area, and bounding box attributes from the Encoded Pixels provided in the original dataset, which are encoded in the Run-Length Encoding (RLE) format.

To obtain the segmentation masks, we employ a process of parsing the RLE string and reconstructing the binary mask. This procedure allows us to obtain a detailed representation of the object's pixel locations and boundaries. For instance, Figure 1 below illustrates the segmentation mask for the input image. Subsequently, we utilize the segmentation mask to derive the bounding box attribute, where the x and y coordinates denote the top-left corner of the bounding box, while the width and height indicate the dimensions of the segmentation mask. Lastly, we calculate the area attribute by considering the total number of pixels encompassed by the object's segmentation mask. This attribute serves as an estimation of the object's size or extent within the image.

4.2. Model Performance & Evaluation

Before training the model on the complete dataset, which consists of 46 classes and over 300,000 annotations, we initially utilized a pre-trained model on a smaller dataset containing only one class to assess its performance visually. Based on the results depicted in the images, while the model demonstrated the capability to identify occluded balloons and provide probabilities for instances belonging to the balloon class, it exhibited misclassifications by identifying ceiling lamps and plates as balloons. Apart from the limited data, we speculate that these errors could be attributed to the inadequacy of the pre-trained model, as it may not generalize effectively to the specific balloon dataset.

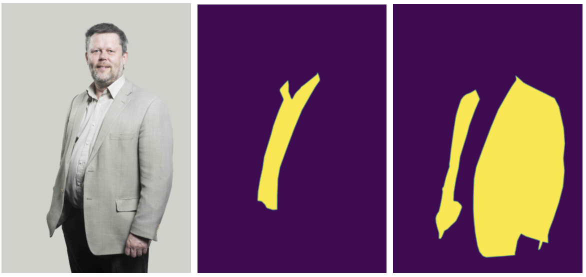

Figure 1. Segmentation masks for different classes

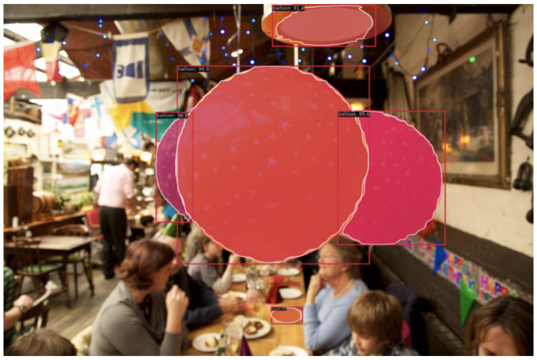

Figure 2. Segmentation result obtained from balloon dataset

4.3. Challenges

Below is a compilation of the challenges we encountered leading up to the milestone, which we ultimately managed to overcome except for the last one.

1. Gaining a comprehensive understanding of the Kaggle dataset: The dataset provided by Kaggle lacked adequate documentation and did not adhere to the desired COCO format. It required additional efforts to analyze and preprocess the data accordingly.

2. Setting up the virtual environment and installing the necessary packages: We successfully installed the required packages and configured a virtual environment to ensure compatibility and smooth execution of our project.

3. Addressing kernel crashes and memory limitations when handling a large dataset: Initially, we attempted to utilize Google Colab for our model exploration. However, we encountered frequent kernel crashes and out-of-memory errors due to the

size of the dataset. Consequently, we transitioned to local machines to overcome these limitations.

4. Overcoming the time-consuming process of generating a large JSON file: Generating a JSON file of significant size proved to be a time-consuming task. We devoted considerable effort to optimize and streamline this process for improved efficiency.

5. Identifying and resolving issues with checkpoint file generation during training: While preparing to train our model with apparel images, we encountered difficulties generating checkpoint files for evaluation purposes. This issue requires further investigation to ensure proper functionality during the training process.

4.4. Future Plan

To transition to the next stage of our research, which involves a more in-depth exploration, we will set up a GPU environment provided by the school. This environment is crucial as it enables us to leverage the computational power of GPUs and effectively utilize a larger dataset for training our model. By utilizing a larger dataset, we expect to achieve improved performance and accuracy. While the pre-trained model is based on the Mask R-CNN architecture, we plan to further investigate the capabilities of alternative architectures such as Cascade Mask R-CNN and Hybrid Task Cascade (HTC). These architectures have proven to be powerful detectors in the COCO dataset benchmark. Lastly, we will strive to improve our current model by introducing custom, self-defined layers that align with our specific objectives.

References

[1] K. Chen et al., “MMDetection: Open mmlab detection toolbox and benchmark,” arXiv.org, *https://arxiv.org/abs/1906.07155*