AIST4010 Project Final Report
Refined Clothing Detection with DeepFashion2 using YOLO

Hon Lam Ng (1155143298)

***Abstract*— Clothes, as everyday commodities, have huge potential for research since they can be very useful in the related fashion industry. Hence, clothes detection is also an important and interesting area to investigate. This project experiments latest YOLO models on the Deepfashion2 dataset to try to improve the detection result. In this project, 1) Clothes detection is carried out using newer YOLO models and obtain great result better than the baseline. 2) Fashion attribute classification is implemented after clothes detection for a refined prediction result using the Resnet50 model.**

I. Introduction

Nowadays, online shopping is extremely popular and the market is growing rapidly on a very large scale. In addition, clothing item takes up a significant proportion of market transactions. As customers, we may want to know what the type of clothes it is by just taking a picture of it. On the other hand, as sellers, detecting clothes from recent images may provide new insights into the current fashion trend to increase revenue. Therefore, clothes detection is indeed a notably interesting area to study.

Clothes detection is a kind of multiple object detection where given an image, all the clothing in the image will be detected, classified, and finally given a bounding box. Figure 1 shows an example of clothes detection.

In the DeepFashion2 official paper [1], Mask R-CNN [2] is used as the baseline models, which is a relatively old model proposed in 2017. Therefore, YOLO [3], as the state-of-the-art object detection model is experimented on the dataset, hoping to improve the result. Also, there exist many improved versions of YOLO models with different sizes, therefore, different versions of YOLO models will be experimented with to choose the best model for fine-tuning.

After finetuning the best YOLO model to get the final result, due to the lack of categories in DeepFashion2, the prediction is not that concrete and precise. Therefore, a further fashion attribute classification is proposed in order to predict more attributes of the clothing.

II. Related work

A. Dataset

There exists several dataset for clothes detection like WTBI [5], DARN [6], DeepFashion [7], FashionAI [8], DeepFashion2 [1]. Fig 4 shows a table of comparison of the above datasets. For example, WTBI [5] and DARN [6]have 425K and 182K images respectively. They scraped category labels from metadata of the collected images from online shopping websites, making their labels noisy [1]. In contrast,

Fig. 1: Example of clothes detection output [4]

.

mAP	mAP_{50}	$mAP_{0.75}$
0.638	0.789	0.745

TABLE I: Validation result of Mask R-CNN on DeepFashion2 Dataset [4]

CCP [9], DeepFashion [7], and ModaNet [21] obtain category labels from human annotators. Among all these datasets, DeepFashion2 [1] is the newest and most diverse dataset with a large number of labeled images for training. Hence, the DeepFashion2 dataset is selected for training the model.

B. Models

There are many standard object detection models like R-CNN [2], Fast R-CNN [10], and YOLO [3]. However, although these models are powerful, there are new models built upon them that surpass their ability. The development team of DeepFashion2 used Mask R-CNN [11] to train and detect the model and act as the baseline. As a result, they have the results as shown in Table I.

Despite the satisfactory result of the baseline model as shown in Table I, it can be improved using state-of-the-art models nowadays as mask R-CNN is a relatively old model proposed in 2017. Traditional models like R-CNN [2] or Fast R-CNN [10] perform detection using classifiers. Instead, YOLO frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. [3] Fig 2 shows the architecture of basic YOLO models.

There are several improvements to the YOLO models currently, for achieving the best result and testing out different models, YOLOv5 [12], YOLOv7 [13], and YOLOv8 [14] are

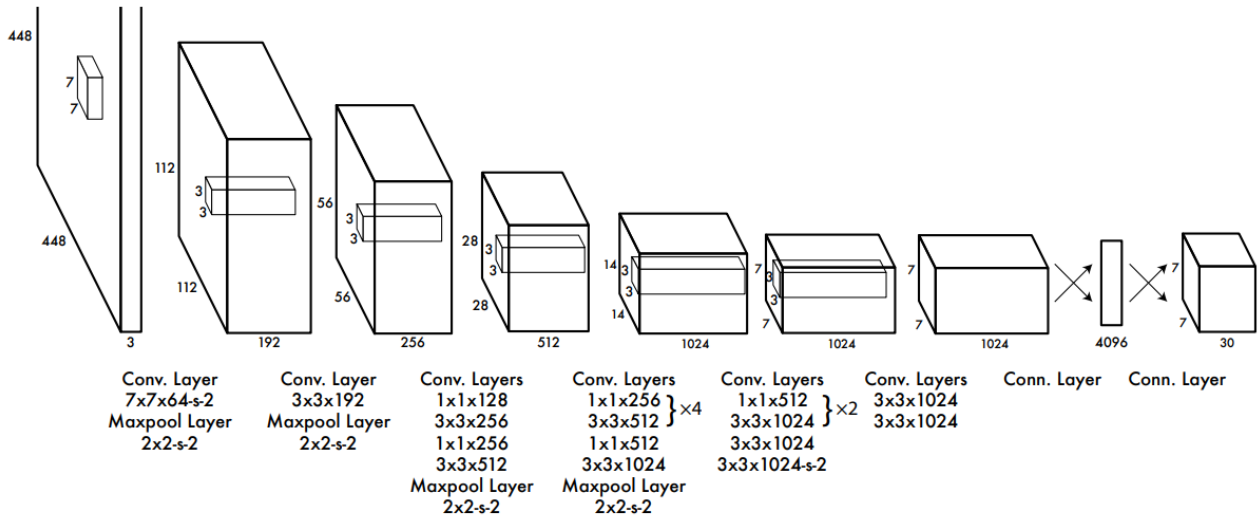

Fig. 2: Model Architecture of basic YOLO network [3]

.

used since they have different architectures and may provide different results. Fig 3 shows the comparison of different YOLO models on the COCO [15] dataset, in which all show great results.

III. Clothes Detection

A. Data

In this task, DeepFashion2 is used. It is developed by The Chinese University of Hong Kong and SenseTime Group Limited which can be downloaded at [4]. It contains 491K diverse images of 13 popular clothing categories, shown in Table II. Moreover, DeepFashion2 has 801K clothing items where each item has rich annotations such as style, scale, viewpoint, occlusion, bounding box, dense landmarks (e.g. 39 for 'long sleeve outwear' and 15 for 'vest'), and masks. There are also 873K Commercial-Consumer clothes pairs. [1] The annotations of DeepFashion2 are much larger than its counterparts such as 8× of FashionAI Global Challenge. [1] Fig 5 shows the statistics of DeepFashion2. Fig 6 shows some samples of images in the DeepFashion2 dataset.

However, the DeepFashion2 dataset downloaded at [4] has the following distribution in Table III, which is lower than what the paper suggests. Therefore it may affect the training result significantly.

1	Short sleeve top	2	Long sleeve top	3	Short sleeve outwear
4	Long sleeve outwear	5	Vest	6	Sling
7	Shorts	8	Trousers	9	Skirt
10	Short sleeve dress	11	Long sleeve dress	12	Vest dress
13	Sling dress				

TABLE II: Categories of Deepfashion2 with index number

B. Approach

1) Data Preprocessing

DeepFashion2 has its own format of annotation but YOLO models require labels to be in YOLO format [16], as shown in Fig 7. Fig 8 shows an example of a json label file provided. Therefore, additional preprocessing is performed and the labels are converted into YOLO format for proper training.

	Train	Validation
Number of image	191K	32K

TABLE III: Distribution of Deepfashion2 Dataset downloaded

2) Models Preparing

There is open source code of the implementation of YOLOv5[12], YOLOv7[17], and YOLOV8[14], which are downloaded to the CSE server. An anaconda virtual environment is set for each version of the YOLO models, for independent training on the CSE server.

Each version of the YOLO model exists different types of model architecture with various sizes. Due to the limited GPU usage, only smaller models are used for testing. Table IV shows the models used in this project. In addition, new config yaml files for Deepfashion2 are added for training on the dataset. After testing the dataset on all models, the best model will be selected for finetuning to improve the performance.

model	size (pixels)	Layers	params(M)	FLOPs(G)
YOLOv5s	640	214	7.2	16.1
YOLOv5m	640	291	21.2	48.4
YOLOv7-tiny	640	263	6.2	13.3
YOLOv7	640	415	36.9	104.7
YOLOv8s	640	225	11.2	28.7
YOLOv8m	640	295	25.9	79.1

TABLE IV: List of YOLO models used

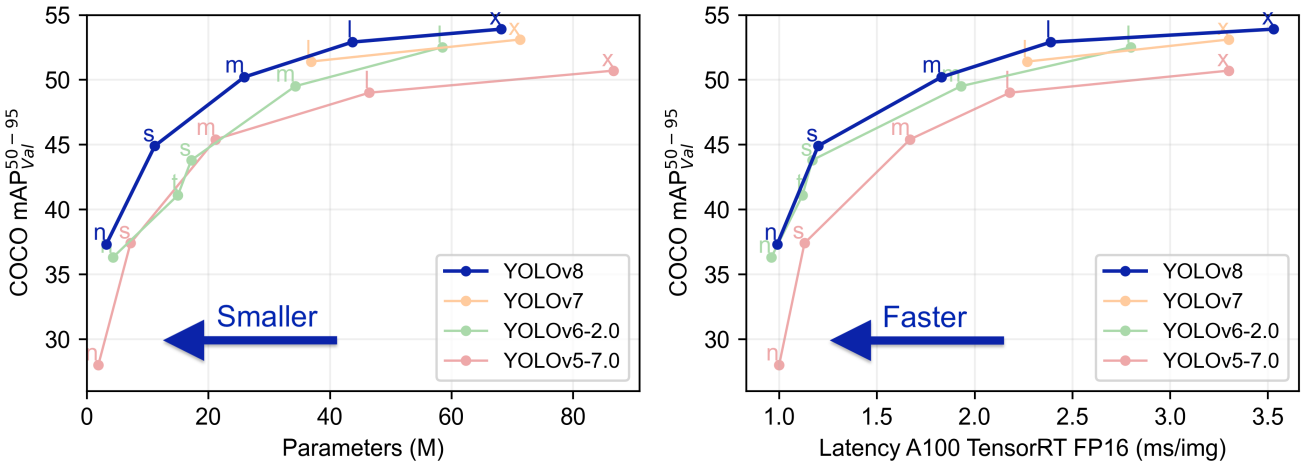

Fig. 3: Comparision of different YOLO models [14]

.

	WTBI	DARN	DeepFashion	ModaNet	FashionAI	DeepFashion2
year	2015[5]	2015[7]	2016[14]	2018[21]	2018[1]	now
#images	425K	182K	800K	55K	357K	491K
#categories	11	20	50	13	41	13
#bboxes	39K	7K	×	×	×	801K
#landmarks	×	×	120K	×	100K	801K
#masks	×	×	×	119K	×	801K
#pairs	39K	91K	251K	×	×	873K

Fig. 4: Comparisons of DeepFashion2 with the other clothes datasets. The rows represent number of images, bounding boxes, landmarks, per-pixel masks, and consumer-to-shop pairs respectively. Bounding boxes inferred from other annotations are not counted. [1]

.

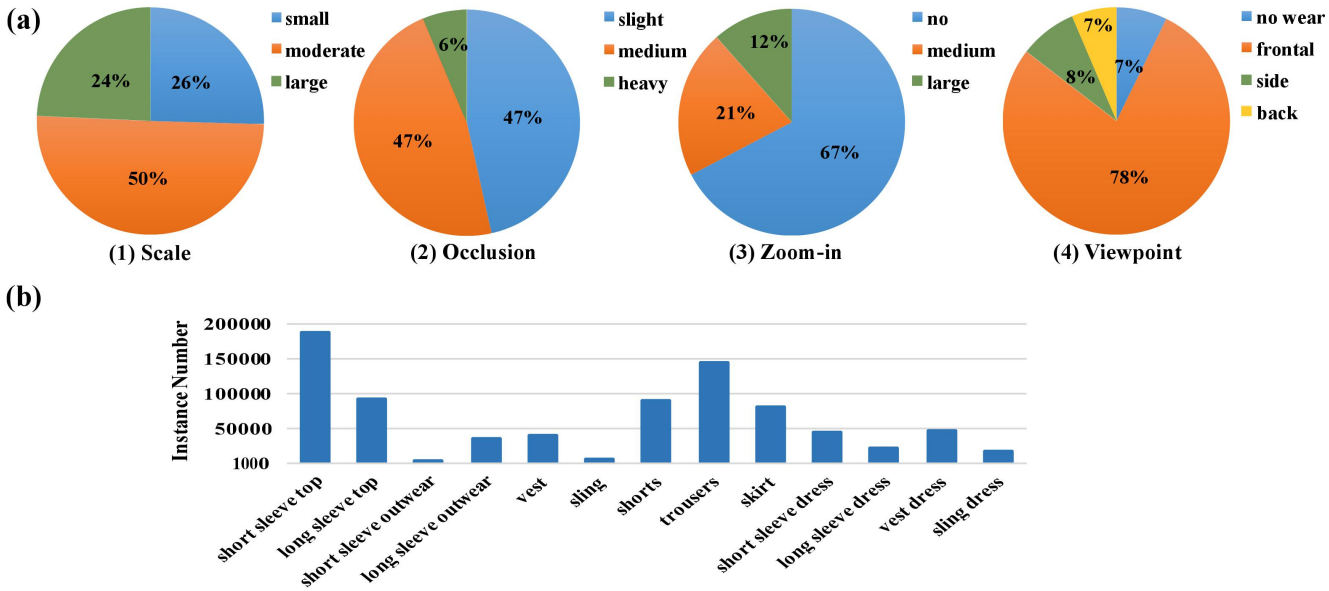

Fig. 5: The statistics of different variations in DeepFashion2. [1]

.

Fig. 6: Sample of DeepFashion2 image after detection [4]

.

```
<object-class> <x> <y> <width> <height>
```

Fig. 7: YOLO labels format[16]

.

3) Training

Because of the limited GPU memory and training time in the CSE server, the batch size is set to 8. The number of epochs is from 5 to 10 depending on the model's size. SGD is used as the optimizer for all models since it has a good performance on image data.

4) Evaluation

For performance evaluation, Mean Average Precision (mAP), Mean Average Precision with Intersection over Union with threshold 0.5 (mAP_{50}) will be used, as they are used in the baseline provided in Table I also.

Mean Average Precision (1) is calculated following COCO [15] by averaging over mAP with IoU threshold from 0.5 to 0.95 with step size 0.05. Also, mAP_{50} is calculated by only counting detections with the bounding box having IoU (2) over 0.5 respectively.

The results are compared to the baseline in Table I.

$$\text{mAP} = \frac{\text{mAP}_{50} + \text{mAP}_{55} + ... + \text{mAP}_{95}}{10} \tag{1}$$

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2}$$

C. Preliminary Results

The following Table V shows the evaluation result on the validation set, all metrics are averaged over all classes.

model	epochs	mAP	mAP_{50}
YOLOv5s	10	0.453	0.601
YOLOv5m	5	0.353	0.499
YOLOv7-tiny	10	0.391	0.547
YOLOv7	5	0.227	0.38
YOLOv8s	10	0.588	0.705
YOLOv8m	5	0.494	0.603
YOLOv8m	**10**	**0.629**	**0.738**

TABLE V: Result of YOLO models on validation set

From the above Table V, it can be observed that Y0LOv8m performs the best and have a similar performance compared to the baseline. Therefore, it will be chosen as the final model for finetuning.


```
{"source": "user", "pair_id": 2, "item1":
{"segmentation": [[126.9, 309.75, 79.29, 311.32, 52.0, 318.0, 39.0, 337.0, 3
"scale": 3,
"viewpoint": 2,
"zoom_in": 2,
"landmarks": [207, 336, 1, 145, 339, 1, 188, 375, 1, 239, 394, 2, 262, 361,
"style": 1,
"bounding_box": [0, 324, 466, 831],
"category_id": 11,
"occlusion": 3,
"category_name": "long sleeve dress"}}
```


Fig. 8: Deepfashion2 labels format.

D. Finetuning

The YOLOv8m is further finetuned for 50 mroe epochs. Due to the GPU and time limit, the model is trained for 10 epochs each time and for a total of 5 times. In each time, the best model from the last training is selected and continued to train on it.

Epoch	mAP	mAP_{50}
10	0.629	0.738
20	0.695	0.795
30	0.712	0.810
40	0.721	0.818
50	0.725	0.821
60	0.725	0.822

TABLE VI: YOLOv8m Result after more training epochs

Fig. 9 shows the training result from epochs 50-60. It can be seen that both the training loss and validation loss are decreasing, the mAP decreases originally and increases afterward. From Fig. 11, the performance of the model is improving gradually with more training epochs. However, due to the limited time and resources, the model training is stopped, it may have a better result with more training. Fig. 10 shows the precision-recall curve.

E. Result

Fig 12 shows an example of detected images for the validation set using the fine-tuned YOLOv8m. Most clothes in the image are detected correctly.

F. Problem

Although the YOLOv8m model can detect the categories accurately, due to the limited number of 13 categories (as shown in Table II), the prediction result would not be very precise. Fig. 13 shows an example in which two clothes have the same categories, but they are actually very different clothes. Therefore, a further classification be can applied to address this issue by providing more labels to the detected clothes to differentiate them.

Hence, the next section discusses the workflow of fashion attribute classification to solve the issue.

IV. Fashion Attribute Classification

A. Data

Since DeepFashion2 does not provide fashion attribute labels in the dataset, so it cannot be used for this task. As a

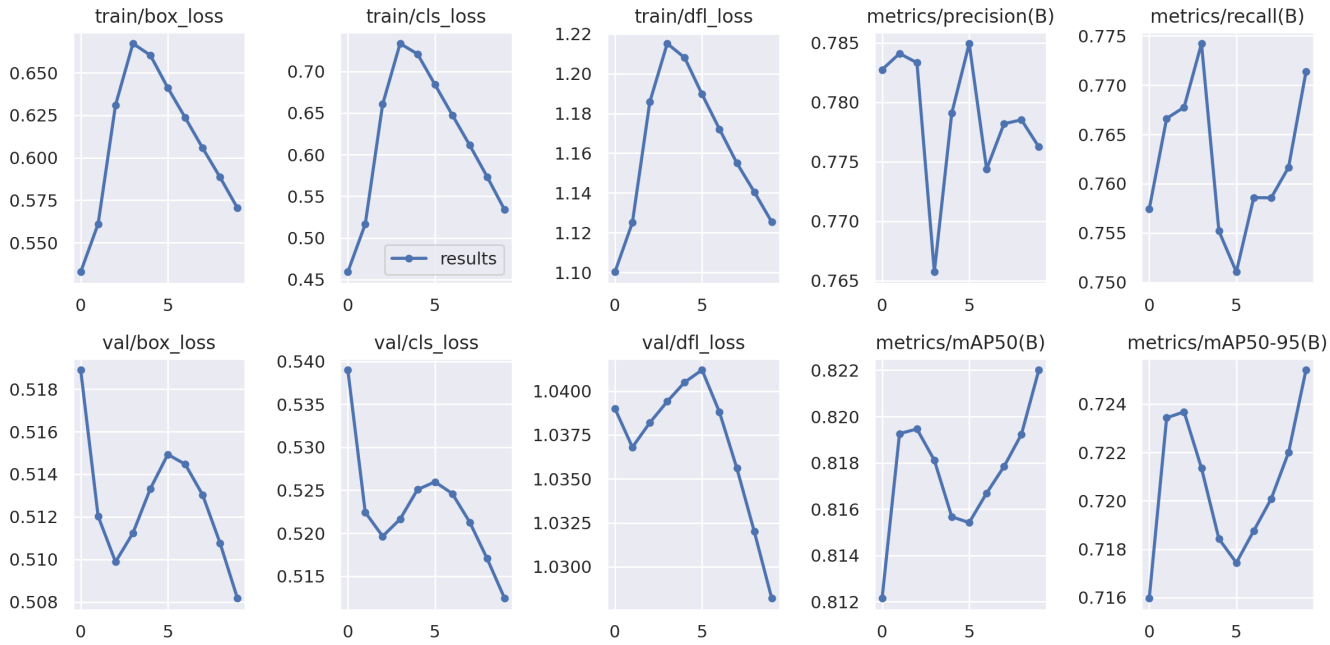

Fig. 9: Finetuning result of epoch 50-60

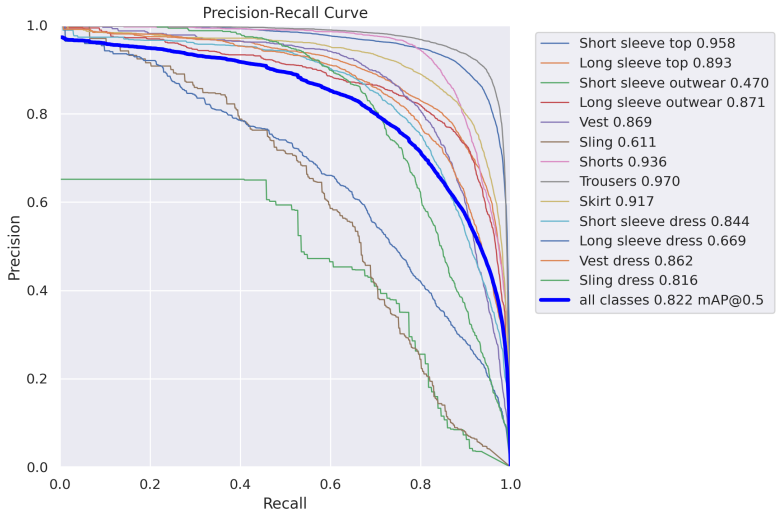

Fig. 10: Precision-Recall curve of final YOLOv8m model

result, another dataset DeepFashion is used as it provides an Attribute Prediction set.

The dataset contains 289,222 diverse clothes images labeled with 1000 different attributes. Fig. 14 shows an example of some attributes in the dataset.

B. Approach

1) Data Preprocessing

Since there are 1000 different attributes, therefore if all 1000 attributes are used, the performance of the model may not be very good. Therefore, only 98 attributes that correspond to style, fabric, season, and type of the pattern are selected, as provided in [18].

After filtering out images that have the selected attributes, there are 137108 images in the dataset. Then, the data is loaded with Dataloader, and the labels are transformed into One-Hot encoded vectors for proper training.

2) Models

As clothes can have more than one attribute in the dataset and real life, this task is a multi-label classification problem.

Therefore, a traditional computer vision model Resnet50 [19] pretrained on ImageNet is used to reduce the training time. The model is loaded using fastai API.

3) Training

The model is trained with 10 epochs on the dataset using RAdam [20] as provided by fastai. Also, since this is a multi-

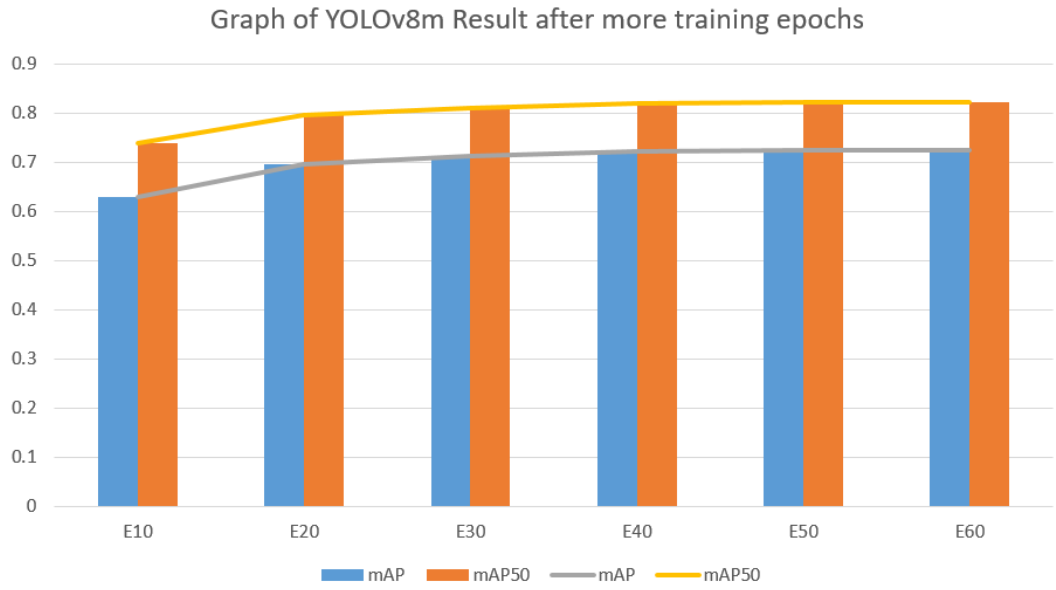

Fig. 11: Graph of YOLOv8m result after more training epochs

Fig. 12: Clothes detection result sample using final YOLOv8m model

label classification problem, and the labels are one-hot encoded, the loss function is chosen to be BCEWithLogitsLoss.

4) Evaluation

Generally, for classification problems, accuracy is used as the standard metric. However, since the labels are one-hot encoded and clothes may only have a few attributes, even a zero vector can obtain a good performance in terms of accuracy. Therefore, the F-beta score is introduced as the metric to evaluate the performance of the model.

F-beta score is the generalization of F-score with a beta parameter that controls the weight of precision and recall. Equation 3 shows the formula of F-beta score. For F-beta score,a beta value of 1 is the standard F1-score. Moreover, a smaller beta value, such as 0.5, gives more weight to precision and less to recall, whereas a larger beta value, such as 2.0, gives less weight to precision and more weight to recall in the calculation of the score.

$$\text{F-beta Score} = (1+\beta^2)\frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \quad (3)$$

In this task, F2-score (beta value of 2) is used because we want to put more focus on recall to reduce false negatives, and to classify correctly as many positive samples as possible.

(a) Image 1 (b) Image 2

Fig. 13: Example of detected clothes that have the same category

Fig. 14: DeepFashion attributes example

Fig. 15 shows the change of F2-score in the training. After training the model for 10 epochs, an F2-score of 0.485126 is obtained. The result is not very outstanding, but sufficient to complete the task at a proper level.

C. Results

After training the Resnet50 model, we can now classify the fashion attribute for the detected clothes. A label of the fashion attribute enclosed in a bracket will be annotated at the bounding box together with the category. Fig. 16 shows how the clothes is further classified as compared to Fig. 13

V. Conclusion

In this project, the result of clothes detection is improved by a significant amount compared to the baseline provided by using newer YOLOv8 models. The issue of unprecise labeling of clothes is alleviated by employing a fashion attribute classification using Resnet50. Some new images searched online are also tested with the models, the result is satisfactory as shown in Fig. 17.

In conclusion, the workflow of the detection system is presented below:

1) Clothes detection with bounding box using YOLOv8m
2) Use the image bounded, input to the Resnet50 attribute prediction model

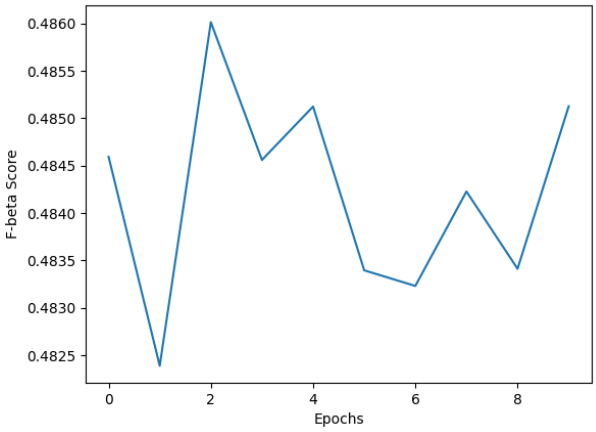

Fig. 15: F2-score of Resnet50 in training

3) Obtain categories in (1), attributes in (2)
4) Visualization

References

[1] Y. Ge, R. Zhang, L. Wu, X. Wang, X. Tang, and P. Luo, "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," *arXiv preprint arXiv:1901.07973*, 2019.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv preprint arXiv:1311.2524*, 2014.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[4] "DeepFashion2 Dataset," https://github.com/switchablenorms/DeepFashion2, 2019.

[5] M. Hadi Kiapour and Xufeng Han and Svetlana Lazebnik and Alexander C. Berg and Tamara L. Berg , "Where to buy it: Matching street clothing photos in online shops." *ICCV*, 2015.

[6] J. Huang, R. Feris, Q. Chen, and S. Yan, "Cross-domain image retrieval with a dual attribute-aware ranking network," *ICCV*, 2015.

[7] Z. Liu, P. Lu, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations." *CVPR*, 2016.

[8] X. Zou, X. Kong, W. Wong, and Y. C. Congde Wang, Yuguang Liu, "Fashionai: A hierarchical dataset for fashion understanding," *CVPR*, 2018.

[9] W. Yang, P. Luo, and L. Lin, "Clothing co-parsing by joint image segmentation and labeling," *arXiv preprint arXiv::1502.00739*, 2014.

[10] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083*, 2015.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *arXiv preprint arXiv:1703.06870*, 2018.

[12] Ultralytics, "Yolov5," https://github.com/ultralytics/yolov5, 2023.

[13] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv::2207.02696*, 2022.

[14] Ultralytics, "Yolov8," https://docs.ultralytics.com/#ultralytics-yolov8, 2023.

[15] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," *arXiv preprint arXiv::1405.0312*, 2015.

[16] "Yolo format," https://pjreddie.com/darknet/yolo/, 2022.

[17] WongKinYiu, "Yolov7," https://github.com/WongKinYiu/yolov7, 2023.

[18] tsennikova, "Fashion attribute recognition," https://github.com/tsennikova/fashion-ai/blob/main/fashion-attribute-recognition.ipynb, 2023.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[20] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

(a) Image 1 (b) Image 2

Fig. 16: Final Result of detected clothes with attribute classification

(a) Image 1 (b) Image 2

Fig. 17: Detection example of new images searched online