Automation of Garment Measurement

Pranava Seth
Thapar Institute of Engineering and Technology,Patiala,India
email

Dr. Deepak Panghal
National Institute of Fashion Technology,Delhi,India
deepak,panghal@nift.ac.in

Dr. Prashant Singh Rana
Thapar Institute of Engineering and Technology,Patiala,India
prashant.singh@thapar.edu

Abstract

When, we talk about large scale Garment measurement, it can be extremely or in some cases almost impossible to measure each Garment and find out the defect. So here, we introduce an Automated solution for this task, a system that work on the Aggregate Landmark Detection,classifies a certain cateogory of Garments and extracts the Keypoints of the Garment , with the help of Pose Estimation and measure the distance between the Keypoints. System is highly accurate and returns the coordinates of the keypoints and the distance between them both in still image and in real time. For the landmark detection part, we have used https://github.com/lzhbrian/deepfashion2-kps-agg-finetune pipeline and integrated the distance measurement part with it.

1 Introduction

Large Scale Garment Measurement can be a tedious task and requires large amount of man power. This will not only result in loss of accuracy but also increase the expenditure. This is one of the major problem of the Garment Industry and requires an automated solution.

Landmark Detection has been a technique which has seen major success in recent years, especially in pose estimation. Our System is basically an extension of HRNET - Human Pose Estimation Network to design a complete real time system that extracts the coordinates of the Keypoints in real time and measures the distance.

We trained our model on Deep Fashion2 Images Dataset, that contains images of 10 Cateogories of Image that gave us the accuracy of around 70 percent. We used the Aggregation as our training method, which resulted quite better than existing state of the art model for Clothes Landmark Detection.

We extracted the coordinates of the keypoints and plotted distance matrix using it, which returns the distance of each keypoint to another keypoint.

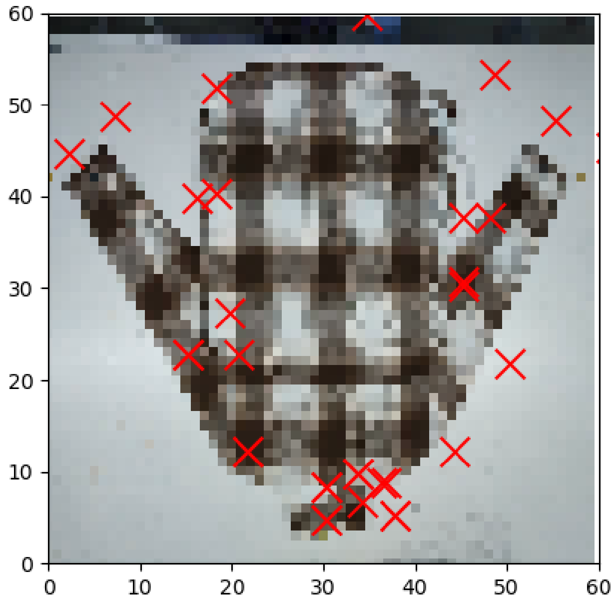

Figure 1:

2 Camera Calibration

Camera Calibration is a technique for calculation of the Extrinsic(Environment) Parameters and Intrinsic(Internal Parameters) of the Camera. It is extremely necessary, when the input is taken in a real time, as if the camera is not camera is not calibrated and the specs are wrong, image as well measurement will be distorted. The inverse Extrinsic Parameters are used to calculate the real world distance from the Pixel Distance.

This section describes the importance of Camera Calibration. Various types of parameters in regard to Camera Calibration and the technique used to find these parameters:

Extrinsic Parameters: These parameters describes the surroundings where the camera is placed. There parameters are the transformation matrices for converting real world coordinates into Camera Matrix. These constitutes the Translation Matrix and Rotation Matrix of the Camera.

Instrinsic Parameters: These parameters the matrix of the camera. These matrices are responsible for transforming the Camera Coordinates to Pixel Coordinates and the distortion coeffecients.

Here, we have used the Zhang's method of Calculating these parameters by using a chessboard of size 7 x 5.The method is as follows:

1. Take a Chessboard of a particular size, 7 x 5, in our case.
2. Click around 100-120 Images of the Chessboard, from the Camera that has to be calibrated.
3. Find the Corner Points of the Chessboarusing Opencv and designate them as the world coordinates.
4. Refine the Corner Points and extract the Instrinsic and Extrinsic Parameters of the camera using cv2.calibratecamera.
5. Finally, we will get the Parameters in the form of numpy array.

3 Garment Landmark Detection

3.1 Keypoints Extraction

In this section I basically describe the HRNET pipeline that have been used in this system for classifying the Garments and extracting the keypoints. Here, it is a two level system. First is the classification pipeline tat classifies the garments and have used ResNeXt-101-64x4d as our object detection model,

Second is the Keypoint Extraction Network, I used aggregation approach Conventionally, one would treat each of the 294 landmarks independently, and a deep learning model is often designed for generating 294 pieces of heatmaps for each landmark and have used HRNet-w48 as our keypoints detector.

3.2 Data

DataSet: Here we have 13 classes that include short sleeved shirt,long sleeved shirt, short sleeved outwear, long sleeved outwear, vest, sling,shorts, trousers, skirt,short sleeved dress,long sleeved dress, vest dress, sling dress. Model was trained on QaudroP6000, 6 GPU running simaltaneouly keeping the batch size per GPU as 4 on around 191961 images.

3.3 Validation

Coordinates: After, we are done with the training part and we get our weights, we come to the inference part. We created a a validation set of various garments and extracted the coordinates of each garment's keypoints. We did so by creating a heatmap using the predicted keypoints and automated the plotting of the extracted keypoints on the the input image passed.

We also visualize the the keypoints on the the Input Image to monitor the overlapping and disturbed position of the predicted keypoints.

4 Distance Calculation

4.1 Distances Matrix

Distance Matrix: It is a matrix that returns the distance from a pair of the coordinates to all the pairs of coordinates available.

In this case, we have extracted the coordinates of the predicted keypoints and combined the abcissa and ordinate into one data frame, and that frame is used to create the distance matrix.

4.2 Implementation of Distance Matrix

Here the distance matrix was implemented using scipy's spatial distance function. It took around 975 values of keypoint coordinates return the distance matrix containing the distance of of each coordinates to one another.

5 Results

1. In the aggregation technique of landmark detection, we trained a model on DeepFashion Dataset containing around 191000 Imaages.

2. We got around 70 percent accuracy for the Keypoint Prediction.

3. After the Keypoint Prediction, the heatmaps of the preds were used to extract the coordinates.

4. Looping on the extracted coordinates was done to extract the coordinate of eack keypoint.

5. Then the distance matrix was plotted for the extracted coordinates.

6. We inferened the predicted distances to the actual distances.

6 Future Works

1. Next Job, in our case will be to optimize the prediction of keypoints, mainly focussing on prevention on overlapping of the Keypoints.

2. After the filtering, we plan to introduce the matching of Keypoints opposite to each other. We plan to do this to reduce the computation time and make it more user friendly as the distances will be more specific to the Region of Interest.

7 References