Real-time Dress Recognition Paradigm

Subarna Kanti Samanta

Goal

- Person detection in every image (or snap of a video taken at entrance of a mall) and recognizing the ethnic wear they have- whether Formal Shirt/ T-shirt/ Saree/ Kurti

Approach

- The whole project is split into four parts:
  - Data collection and preprocessing
  - Person Detection
  - Wearing Classification
  - Using final deep leaning model for detection and classification

Data Collection

- Non availability of the labeled dataset was one of the biggest problem
- There are dataset that are having dress with labels:
  - deepfashion & deepfashion2
- The task requires ethnic wear which were not available
- An image scrapper using selenium had been made that downloaded images from the “google image search”
- Here encountered another problem: images were of small size, much less than (416,416) which is the default that Yolov3 accepts.
- Later on extracted both resolution images but tried to keep high resolution images more and low resolution image less
- The low resolution images were zero padded later which introduces diversity in dataset

Data Preprocessing

- Preprossessing was done manually as dataset was not too large and outliers need to be deleted
- Example of outliers:
  - Images which didn’t belong to any one of the categories were also collected in dataset
  - Occlusion: unable to see person distinctly or the wearing
- A small subset of whole file was kept aside for final testing- 150 images per class
- After preprocessing of the train images, they were passed through YOLO V3 deep learning model to get Region of Interest.

YOLO V3

- “You Only Look Once” is an algorithm that uses convolutional neural networks for object detection.
- Though it’s not the most accurate object detection algorithm, but it is a very good choice when we need real-time detection, without loss of too much accuracy.
- It not only classifies the image into a category, but it can also detect multiple Objects within an Image.
- This Algorithm applies a single Neural network to the Full Image. It means that this network divides the image into regions and predicts bounding boxes and probabilities for each region.
- These bounding boxes are weighted by the predicted probabilities.

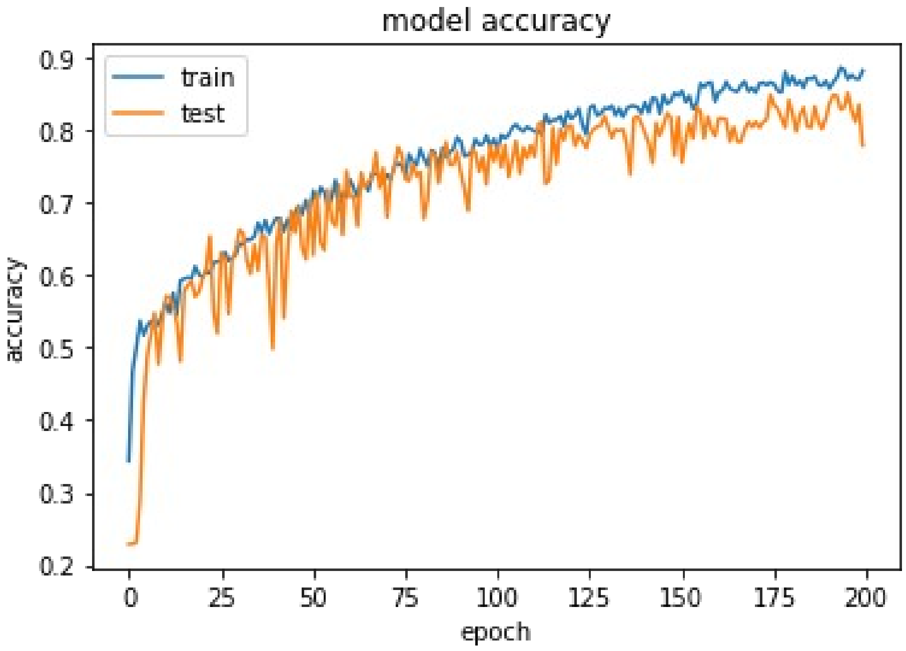
Detection using YOLO V3

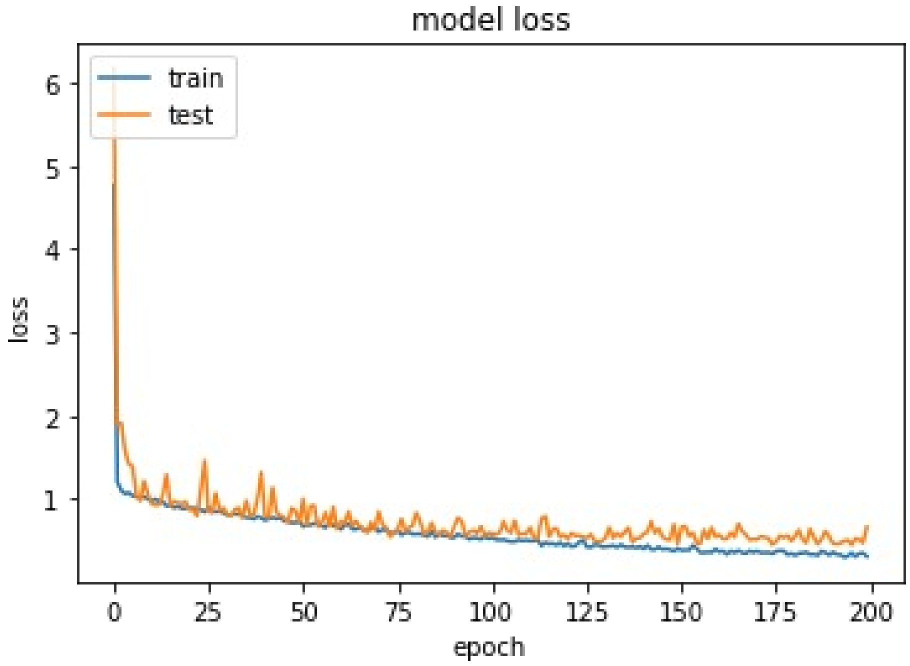
- Downloaded the model weights which were pretrained on MSCOCO dataset using the Darknet code.
- Define a Keras model that has the right number and type of layers to match the downloaded model weights.
- The model architecture is called a “DarkNet” and was originally loosely based on the VGG-16 model.
- Hence YOLO V3 was made and loaded with pretrained weights of MSCOCO dataset, which was used to detect only label “Person” from 80 classes of the dataset
- A threshold had been set to collect only those images having classing probabilities more than 60%
- After detection, got arrays which consists bounding boxes and class labels but are encoded
- Which are later decoded, resized in order to fit the original image
- Non maximum suppression was done to keep the most significant detected boxes
- Finally all detected images are saved with their respective labels

Classification

- There were many datasets created from detection which were merged and randomly shuffled.
- Further diversity is brought by Data augmentation: first used many features but later only keep one or two approaches like horizontal flip, pixel shifts because too many features lead to system crash
- CNN model was prepared with eight convolutional layers with in between MaxPool layers and Dropout layers followed by Fully Connected layers, dense layers and softmax on top, analogous to VGG 16 network, optimizer used: Adam
- After quite fine tuning able to remove overfitting and got satisfied results
- The training data set consists 7272 images from which 300 are separated for final check and 23% of remaining training images for validation

Learning Curves

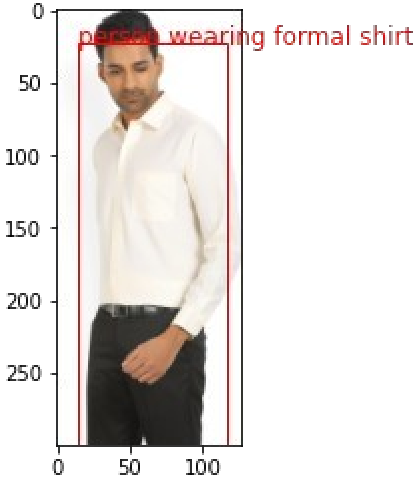


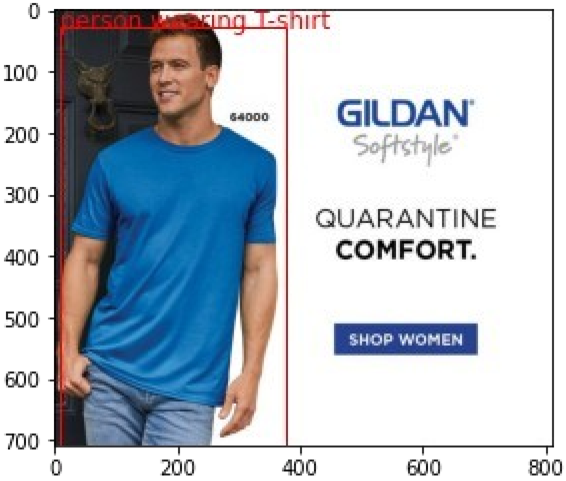


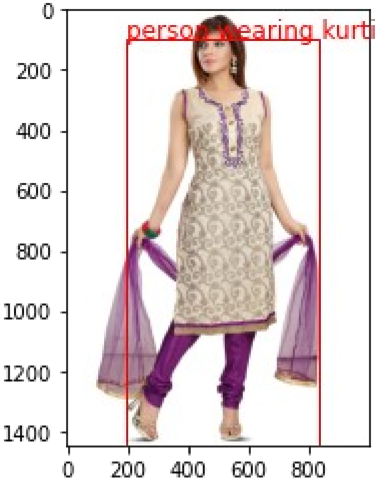
Model Testing

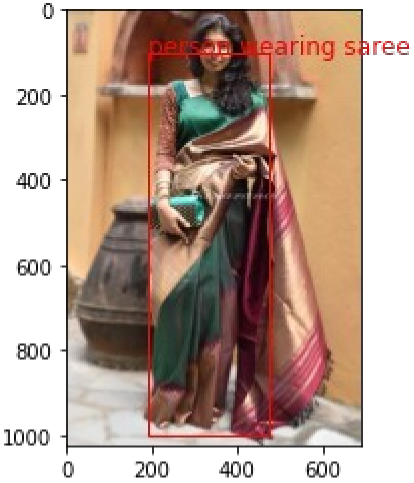
- Here both the process of person detection using YOLO v3 and classification of wearing using our CNN model prepared in classification are implemented.
- The final output is input image with bounding box, that were extracted from detection alongwith label on top of box
- Also getting time for prediction for each image which was around 4.5 Second(using GPU)
- Accuracy of test set was 82.27%

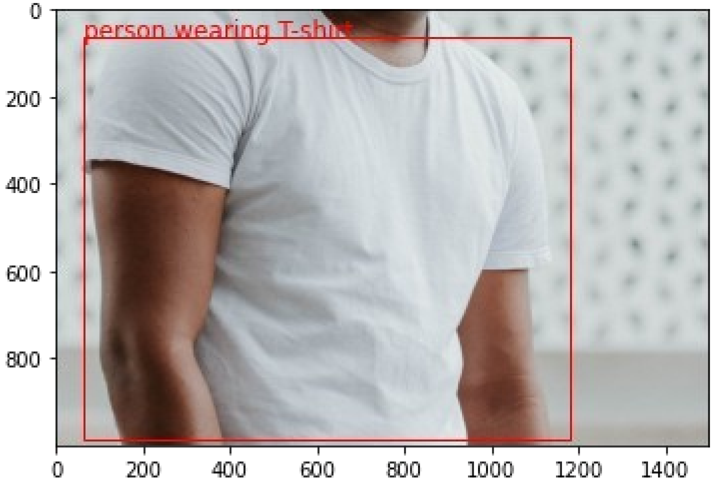
Output

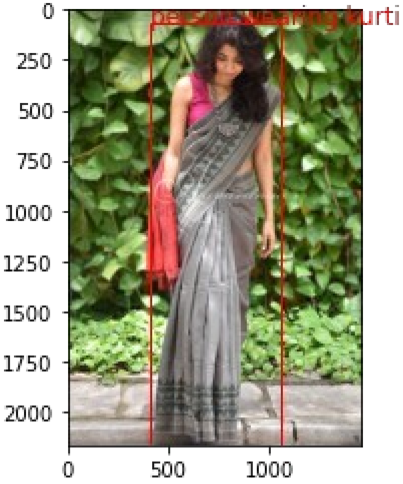

person wearing formal shirt
0
50
100
150
200
250
0
50
100

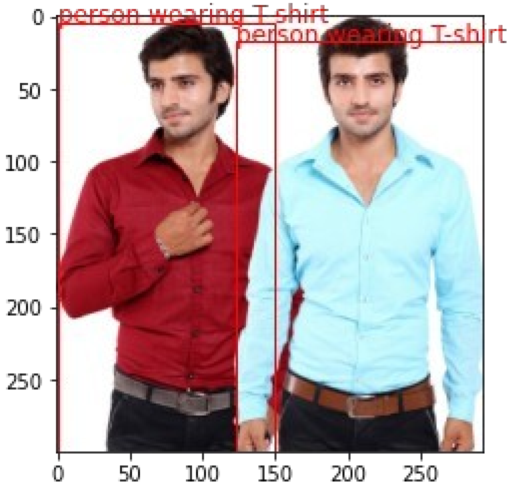

person wearing T-shirt
64000
GILDAN
Softstyle
QUARANTINE
COMFORT.
SHOP WOMEN
0
100
200
300
400
500
600
700
0
200
400
600
800


person wearing kurti
0
200
400
600
800
1000
1200
1400
0
200
400
600
800

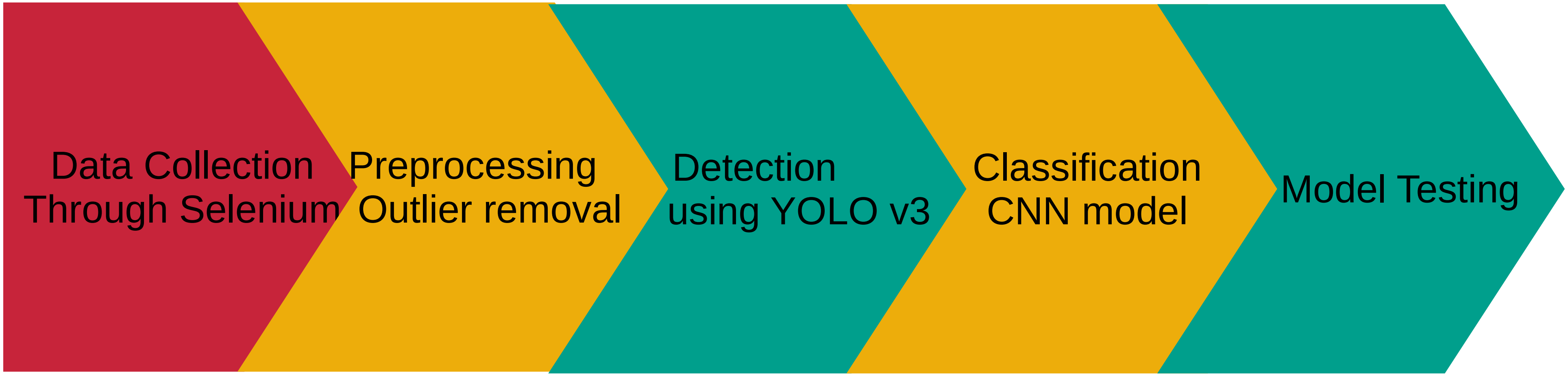

person wearing saree
0
200
400
600
800
1000
0
200
400
600


person wearing T-shirt
0
200
400
600
800
0
200
400
600
800
1000
1200
1400


person wearing kurti
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000


person wearing T-shirt
person wearing T-shirt
0
50
100
150
200
250
0
50
100
150
200
250

Pipeline



Data Collection Through Selenium

Preprocessing Outlier removal

Detection using YOLO v3

Classification CNN model

Model Testing

Conclusion

- The accuracy of the classifier was improved on the validation set high as high as 80% which was able to achieve the decent results.
- For the test run, there were approx 150 images/class and the accuracy was 82.27%.
- The other observation was, that the classifier was working even if the multiple persons were in the image.
- Diverse dataset lead to prevention of occlusion, and posture change effects.
- The model was confused between t-shirts/ formal shirts because of too much varieties in t-shirt, and formal shirt dataset is small

Problems faced

- Data Collection – images which were extracting, not all of them were required: resolution issues, without person, heavy occlusion
- Preprocessing: Didn’t able to figure out to automate method for removing unrequired images came in data collection, so manually done that.
- Runtime Error – system was crashing because of longtime running, and also because of low internet connectivity which was solved by splitting work in chunks
- Good Architecture – choosing a good architecture, and then tuning it was an time taking process, and till end didn’t achieve the most wanted accuracy. Almost tried six different models by removing and adding many features, basically followed VGG network.

Further Optimization

- Time – Model is very slow so prime focus is on decreasing the time which is now approx 4.5 sec(with gpu) for one image, this can be improved by optimizing yolo network and also fine tuning classification model
- Accuracy – There was a lot of gap in accuracy which can be improved by introducing more layers or using different architectures
- Data size – need to be increased to enhance diversity
- Multipurpose model – Enhancing the model for further use like recommending advertisements on basis of declining demand on some particular wear.