RopaNet: Fashion Detection and Classification

Oghenetegiri Sido
Stanford University
osido@stanford.edu

Abstract

We introduce a new fashion classification model called RopaNet. Like many of its predecessors, RopaNet builds upon a standard convolutional neural network to classify clothing images into fifty distinct categories. However, unlike prior models, RopaNet employs a novel mask branch to learn the landmark points on clothing articles, which via transfer learning, can boost classification accuracy. This mask branch begins with a gated convolution of the feature map from our Inception backbone. This mask is then upsampled to generate a landmark mask for each landmark point on the article. By focusing on the landmark localization task, our model learns to hone in on the most important features of each clothing article, which furthers our classification objective. Employing these novel techniques, RopaNetBaseline achieves a staggering top-three accuracy of 80.89% and top-five-accuracy of 94.6% while RopaNet achieves top-three accuracy of 48.08% and top-five-accuracy of 49.6%

1. Introduction

1.1. Problem

Given a 300x300 image, which is featurized with the Inception [9] network, we focus on the following tasks:

1. **Landmark Annotation**: predict the (x,y) locations of several key-points pertaining to the structure of the clothing item on the image. There are four to eight different landmarks on any clothing item. Note that due to deformation and occlusion, some landmarks may not be visible.

2. **Category Classification**: classify a clothing article into one of fifty different categories including *Dress*, *Blazer*, *Hoodie*, *Romper*, etc.

Category classification is our primary objective, which is faciliated by a multi-task learning approach due to our landmark prediction. Referencing the FashionNet approach [5], we will jointly predict landmarks locations on the image of clothing, which has proven beneficial for category classification.

1.2. Why RopaNet Matters?

In recent years, advances in online and mobile shopping have lessened the need for brick-and-mortar shops. Many people look for and purchase clothes from their mobile phones. Accordingly, there have been recent efforts aimed at applying artificial intelligence to the fashion domain. To name a few hot areas, clothing classification, attribute prediction, and clothing item retrieval are all actively being worked on. These tasks are inherently difficult due to the fact that clothing images can be taken in various lighting conditions, are subject to occlusion and deformation, can become discolored overtime, and have a variety of different textures and styles. These factors culminate in a challenging problem.

1.3. Results Overview

RopaNetBaseline achieves top-three accuracy of 80.89% and top-five-accuracy of 94.6% while RopaNet achieves top-three accuracy of 48.08% and top-five-accuracy of 49.6%. Our more complex RopaNet, which predicts landmarks, suffers from slow-training and is hampered by an ineffective landmark mask. Accordingly, our simpler baseline is able to train much faster and performs well.

2. Related Work

2.1. Object Detection Architectures

Previous efforts were aimed at the object detection problem where regions in the image had to be classified. R-CNN [3], tackled this by introducing region proposals with a selective search algorithm, finally classifying the region with an SVM. While this was functional, R-CNN was prohibitively slow. As a result, Fast R-CNN [2] was introduced by the same author, differing from its predecessor by generating the proposals from the feature map instead of the original image. It used RoI pooling in order to measure fixed regions that could later be fed through a fully-connected

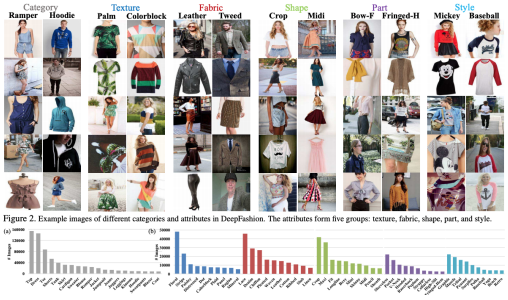


Figure 2. Example images of different categories and attributes in DeepFashion. The attributes form five groups: texture, fabric, shape, part, and style.

Figure 1: DeepFashion Dataset

layer for classification. Faster R-CNN [7] was later released and improved upon [2] by learning proposals instead of using a fixed, selective search algorithm. Then, Mask R-CNN [4] improved upon [7] by using ROIAlign to predict an un-quantized proposal region and predict a segmentation mask for the proposal region as well. Of the above proposed methods, this ties in the most with RopaNet, as we seek to classify what is seen in the clothing image with a convolutional neural network and fully connected layer framework while also predicting the mask for the landmark points. The YOLO paper [6] also introduced bounding boxes with confidence, which is akin to our landmark mask architecture. We use a softmax layer across the entire 300 x 300 matrix to determine our confidence in each pixel being a landmark point.

2.2. Related Fashion Architectures

DeepFashion [5], proposes an architecture that uses a landmark branch to predict annotation locations, a local branch, which uses the predicted landmark locations to hone in on specific local features of the image, and a global branch to focus more on global features of the clothing image. RopaNet is inspired from this structure; yet, our feature extractor backbone is InceptionV3 [9] instead of VGG [8] and our landmark branch is far more complex, as it recreates a mask of the landmark locations in a 300 x 300 space while [5] merely predicts the locations of the landmarks. DeepFashion2 [1], however, proposes the Match R-CNN network, which is an extension of the previously discussed Mask R-CNN network [4]. Here, it must use the Mask R-CNN framework due to the fact that its dataset can contain multiple clothing items, whereas our dataset only includes one clothing item per image.

3. Data

We are using the DeepFashion dataset [5], which contains over 800,000 different 300x300 images that are labeled with attributes, landmarks, categories, and pairings between clothing images taken in different circumstances (i.e in a store catalog versus from a cell phone camera). As seen in figure 1, there are 1000 different attributes including style and texture, 50 different categories including Dress and Hoodie, and between four to eight landmark annotations for each image.

Data Exploration:
As seen in figure 1, the *top*s and *dress*es are grossly overrepresented in the dataset, which has significant effects on the ability of our model to train effectively without bias.

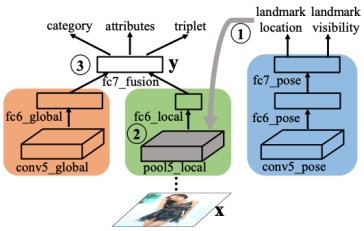
Preprocessing:
The data was separated into a folder of images and text files for specific tasks. Therefore, there is one file for landmark annotations where the left column is the path to the image and the right column has the four to eight landmarks as (x,y) pairs. There are similar files for category and attributes. When loading into a tensorDataSet, we noticed missing images in certain files. This was due to the mislabeling of certain image directories and paths: in one file there would be ”gold/white-tee.jpg” while another file would have ”gold/White-Tee.jpg”. To standardize across all files, we renamed all image paths to be their lowercase versions and then we made new files where all image paths in the left column were also made lowercase while the labeled info(category, attributes, landmarks, etc) in the right column was left untouched.

Additionally, Inception takes 299x299 images as input while our images in the dataset are 300 x 300. Thus, we resized all of them and normalized them in the range of 0 and 1. Note that the landmark positions also change slightly with this rescaling, so we are also working on that transformation since the original landmarks were written for a 300x300 image, not a 299x299 image.

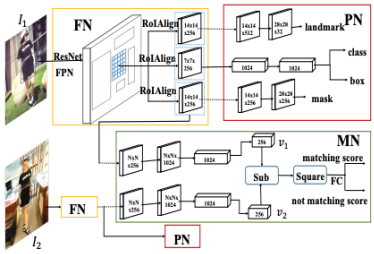
4. Methods

4.1. FashionNet

: As shown in figure 2, FashionNet [5], feeds an image through VGG-16 [8] and then proceeds to complete three branches: the *global* branch in orange, *local* branch in green, and *landmark* branch in blue. First, in the *landmark* branch, it annotates landmarks with an additional convolutional layers and two fully connected layers. It then pools this landmark information with the last convolutional layer output from VGG-16 to create a series of spatially *local* representations of the clothing in the image. On the left in orange, the last convolutional layer of VGG-16 is convolved once more before being fed to a fully connected layer. This branch focuses on spatially *global* features in the clothing image. The *global* and *local* outputs are then concatenated and fed to another fully connected layer to generate the categories and attributes. Note that in training, more weight in the loss function is given to landmark localization due to



(a) FashionNet



(b) Match R-CNN

Figure 2: Fashion Architectures

downstream dependencies on the quality of landmark annotations.

4.2. Match R-CNN

: Match R-CNN [1], which is based on Mask R-CNN [4], is for a different dataset called DeepFashion2 [1], where a single image contains multiple clothing items. This architecture seeks to perform the tasks of FashionNet at a bounding box-level to identify all clothing items and then use segmentation to classify each pixel in a clothing item bounding box. Due to the difference in objective, Match R-CNN takes two images as inputs, I_1 and I_2 and uses three primary network components. First, we have the feature extraction network (FN), a perception network (PN), and a match network (MN). Both I_1 and I_2 are fed to the FN, which has a ResNet backbone to extract features from the images in addition to a region proposal mechanism that is used to build feature maps for the candidate regions. These are then passed to the PN, which specializes in landmark annotation, clothing detection, and mask prediction, with small networks for each of these tasks, as shown above. Landmark annotation relies upon 8 convolutional and 2 pooling layers, while the clothing detection network uses 2 fully connected layers, where the first is for classification and the second is for bounding box regression. Finally, the mask prediction (segmentation) relies upon 4 convolutional layers, one pooling layer, and a final convolutional layer. Finally, the MN, which determines if I_1 and I_2 had the same clothing item, takes the FN represenations and uses 2 convolutional layers and a fully connected layer to generate v_1 and v_2. The matching score is determined by the distance between these vectors.

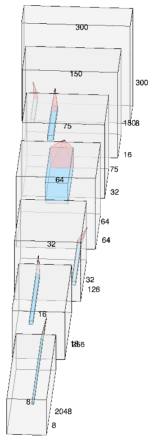


Figure 3: Landmark Branch Architecture

4.3. RopaNet-Baseline

: However, while we will implement multi-task learning approach above in [5] and [1], we have successfully implemented our own baseline method completely from scratch: we feed our image through InceptionV3 instead of VGG-16, take the last inception module output, which we then global average pool and feed into a fully connected layer to derive the image categories. We use cross-entropy loss to determine the model's confidence in the correct category. We also are using the Adam Optimizer with a learning rate of $5e^{-3}$. Since our baseline does not have the additional branches for landmark annotation and local and global featurization from FashionNet [5], we are convinced this is a strong lower-bound baseline that should improve with the additional multi-task learning. Note that we also differentiated from FashionNet in our choice our our underlying pretrained model, as we opted for InceptionV3 [9] instead. This was motivated by VGG's [8] difficulties in deployment on GPUs and memory constraints due to the large width of its convolutional layers. In future work, we hope to port this model for mobile phone use with TFLite, so Inception proved to be the better choice here.

4.4. RopaNet

: Like FashionNet [5], we use three branches: the landmark, local, and global branch.

4.4.1 Landmark Branch

Our landmark branch (figure 3) takes in the feature map from InceptionV3 [9], which is (8 x 8 x 2048). We use a 1 x 1 convolution with 2048 filters and a sigmoid activation

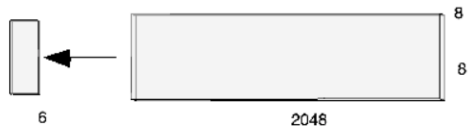


Figure 4: Local Branch Architecture

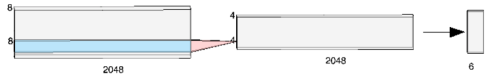


Figure 5: Global Branch Architecture

in order to generate a mask that can be applied to the feature map later in the local branch. After generating our 8 x 8 x 2048 mask with values between 0 and 1 due to the sigmoid activation function, we apply a series of transpose convolutions to effectively upsample to a 8 x 8 mask into a 300 x 300 x 8. Each of the 8 channels represents one of the landmarks. Following the Mask R-CNN paper [4], it's better to predict the landmarks separately via 8 different channels, which is what we did. In order to train this mask representation that is later upsampled to a 300 x 300 x 8 tensor, we apply a new loss function where the true landmark point (x,y) in the original image should be a 1 and all other pixels in the 300 x 300 tensor should be zero. This is faciliated by flattening each channel into a 300^2-vector, that is softmaxed, assigning values to every pixel location, where the grand-sum of all pixels' values is 1. Thus, our loss function first flattens the tensor and calculates the cross-entropy-loss. The gradient from this loss helps the model generate a mask that can be upsampled to localize the landmark annotations. This mask is important for the local branch.

4.4.2 Local Branch

The local branch's (figure 4) purpose is to focus on locally important features that will aid in classification downstream. Accordingly, it uses the gated mask from the landmark branch layer. Taking the gated mask from the landmark branch layer, we calculate the point-wise product with the feature map to generate a locally aware feature map, where important landmark points are emphasized. We then pass this locally aware feature map through a global average pool layer and then to a fully connected layer with 6 outputs.

4.4.3 Global Branch

The global branch's (figure 5) purpose is to focus on globally important indicators that will aid in classification downstream. To do this, it takes the unmodified InceptionV3 [9] feature map and applies a 5x5 convolution to the feature map. This output is passed through a global average pool layer and finally to a fully connected layer with 6 outputs.

Model	Top-3 Accuracy	Top-5 Accuracy
FashionNet	**82.58**	**90.17**
RopaNet-Baseline-100	25	50
RopaNet-Baseline-1000	61.39	75
RopaNet-Baseline-5000	80.89	94.6
RopaNet-10000	48.08	49.6

Table 1: Model Accuracy: RopaNet-Baseline-N denotes the size of train and test sets

4.5. Final Steps & Optimization Paradigm

At this point, we concatenate the 6-vectors from the global and local branches and apply a fully connected layer with 50 outputs to predict our final category. This output is softmaxed and the loss for the classification class is cross-entropy. Now, the gradients computed from classification loss and the landmark loss are propogated backwards to the model. Note that since landmark localization is a particularly difficult task due to the granularity of predicting one pixel out of 300^2, we train our landmark branch twice as often as our global and local branches.

4.6. Metrics

Our baseline metrics include accuracy, top-3-accuracy, and top-5-accuracy, precision, and recall, which are appropriate for this classification task. We also calculate our landmark loss, as discussed above.

5. Experiments

We focused on various runs of our RopaNet baseline and three-branch RopaNet model, as we implemented both models from scratch.

5.1. Experimental Results

5.1.1 Evaluation Results

RopaNet vs Baseline

Although our baseline is relatively simple, we see in the table that our baseline outperforms our complex RopaNet model with landmark annotation. There are several reasons why this is the case. As seen in figures 8 and 7, our baseline trains much faster and converges to a lower training loss. This is expected, however, as the size of RopaNet's training set is twice the size of our baseline. Additionally, the baseline has much fewer parameters to train than the complex RopaNet, which has three branches with multiple convolutional and fully-connected layers. Nevertheless, the

Model	Precision	Recall
RopaNet-10000	.93	.04

Table 2: RopaNet Precision & Recall

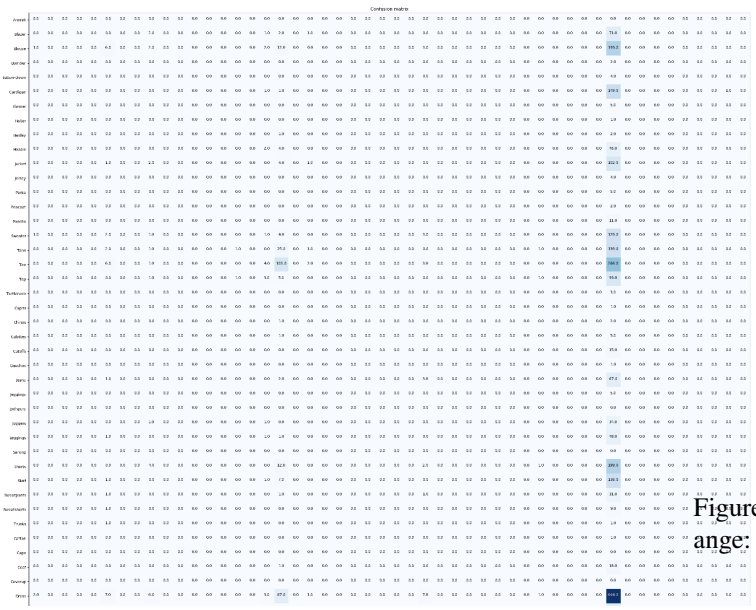


Figure 6: Confusion Matrix

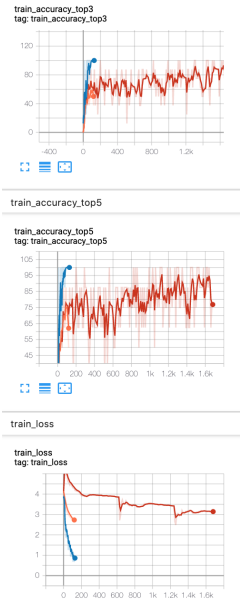


Figure 7: Training Set Graphs: Blue: Ropa-Base-100; Orange: Ropa-Base-1000; Red: Ropa-Base-5000

difference in performance suggests something is amuck in the RopaNet model. Given additional time, I would perform a visualization of the local branch layers to determine if our mask from the landmark layer is actually hurting performance by gating out too much information.

Precision, Recall, and Confusion Matrix

Additionally, our high precision and local recall metrics seen in our table demonstrate that RopaNet is not performing well in distinguishing between categories for classification. This is further evidenced by our confusion matrix (see figure **??**) where the model is overly biased towards predicting *dress*. As mentioned earlier in the *data* section, the data distribution is dominated with *dress* pictures, so our model has become biased towards predicting *dress* due to its overwhelming prevalence in the training set. In future work, we must better balance the dataset to mitigate this issue.

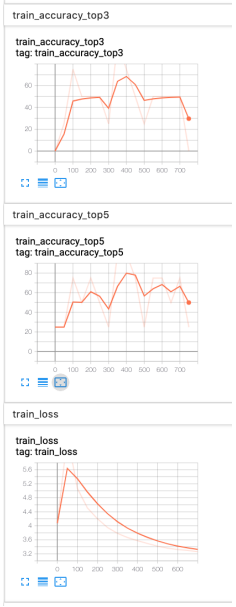


Figure 8: RopaNet Train Set Graphs

5.1.2 Metric Plots

Issues with Landmark Branch

When viewing figures 7, 8, and 9, we see that our landmark loss, while decreasing, decreases extremely slowly and appears to converge earlier than intended. Accordingly, the RopaNet train loss also cannot decrease at the rate and to the level that we see with our baseline model. Again, an ablation study of the landmark layers, especially the mask, is necessary to diagnose its issue in training. Nevertheless, picking one pixel out of 300^2 pixels as a landmark point is not a simple task, so slow training is not too unusual in this

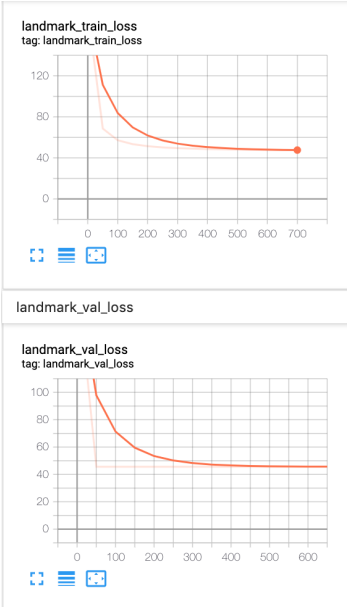


Figure 9: RopaNet Landmark Loss

Figure 10: Error Analysis: Model predicted correctly

Figure 11: Error Analysis: Model predicted incorrectly

case.

5.1.3 Error Analysis

We've randomly selected two examples in the test set that our RopaNet model evaluated: figure 10 is an example where the model predicted correctly whereas figure 11 is where the model predicted incorrectly. while there are similarites between sweatpants and a dress, I attribute this misclassification to over-representation of the dress class in the original dataset.

5.2. Architectural & Hyperparameter Search

For our local layer, since we wanted to highlight important local features in the image, we tried using a MaxPool instead of and AveragePool to hone in on striking features in the image. This suprisingly perfomed badly, which is why we swithced to using AveragePool.

When training RopaNet, we experimented with training our landmark layer three times for every one time the rest of the model trained. This did not improve performance and increased training time, so we switched back to training it two times for every one time for the rest of the model.

6. Conclusion

As a major takeaway, we found that our simpler baseline performed much better than our complex model for a host of reasons: ease of training, less parameters, etc.

Additionally, we found the inherent issue in training with an unbalanced dataset. Downstream predictions will become biased towards the dominant classes.

In future work, we hope to follow the Inception strategy [9] and build a deeper network to see if we can boost performance. Our current RopaNet has extremely shallow local and global branches.

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. *CoRR*, abs/1901.07973, 2019.

[2] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[3] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[4] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[5] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: powering robust clothes recognition and retrieval with rich annotations. *CVPR*, pages 1096–1104, 2016.

[6] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[7] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.