Clothing Classification and Recognition

Asad Ashraf, Sai Sri Hari Saran Cherukupalli

May. 05, 2017

Boston University

Department of Electrical and Computer Engineering

Technical report No. ECE-2017-5

BOSTON UNIVERSITY

Clothing Classification and Recognition

Asad Ashraf, Sai Sri Hari Saran Cherukupalli

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

May. 05, 2017

Technical Report No. ECE-2017-5

This is a project for EC520 Digital Image Processing.

Summary

In recent years, there has been attempt to classify everything and anything by the big tech companies and research institutions to use in the real world for object detection. One big challenge is clothing item recognition since it's difficult to differentiate one item from another. In this project we attempted clothing recognition and detection, to find how far, we as students can go in successfully recognizing various clothing items. We fine-tuned pre-trained Convolutional Neural Network to classify clothing items and used a face detect the location of a person in the image. The location of the person helped us estimate regions of interests at the torso and legs, which we ran through classify to get back what the person was wearing. We had some successful results and also some failed results. The failed results were due to face detector and the difficulty in differentiating between different clothes.

Contents

List of Figures

1 Introduction

Recognition of clothing items in images is growing in demand, both in research and in applications being developed for it. There is a lot of potential with such technology since it can be applied in several fields. The fashion industry could use it to find out the latest trends while retailers could use it to find out what's in demand; and law enforcers could use it to identify people by their clothing. However, this isn't an easy task to do because of factors such as image quality and the ability to differentiate between different types of clothing items that affect the recognition success rate.

2 Literature Review

When it comes to classifying clothing for recognition, Brian Lao et al. (2015) [1] used a Convolutional Neural Network (CNN). In [1], a standard AlexNet CNN was used on an Apparel Classification with Style (ACS) Dataset which contains 89,484 images, each labeled with one of 15 hand-picked clothing categories, such as jacket, suit or shoe, for clothing type classification. It achieved an accuracy of 50.2% when fine-tuning all layers, which is much better than Support Vector Machines (35%), Transfer Forests (41.4%) and Random Forests (38.3%).

To detect the location of clothing items in an image, Brian Lao et al. (2015) [1] used a Region based Convolutional Neural Network (R-CNN), pre-trained model on CF dataset and modified with bounding box labels and extra training patches. They achieved a top validation accuracy of 93.4% during their phase-two training.

3 Problem Statement

There are several factors that affect clothing item recognition and detection. One of them is image quality. The quality of the image depends both on brightness and

resolution. Another factor to take into account for recognition is clothing itself. As Brian Lao et al. (2015) [1] states, “clothing can share similar characteristics (e.g. the bottoms of dresses vs. the bottoms of skirts), clothing can easily deform due to their material, certain types of clothing can be small, and clothing types can look very different depending on aspect ratio and angle.” When it comes to recognition, we have to these constraints into account.

As Brian Lao et al. (2015) [1] have shown in their implementation, a Convolutional Neural Network seems to be the best method to classify clothing items for recognition. Although not perfect, it does a better job of tackling some of these constraints that come with recognizing clothing than methods like Support Vector Machine. This is the basis of our project. We used a convolutional neural network for classifying clothing items. Brian Lao et al. (2015) [1] used 15 classes; to simplify our problem we only chose 5 classes. These classes are jackets, sweaters, shirts, pants and shorts; distinct enough to achieve better accuracy.

When it comes to clothing item detection, we used face detector to locate the person in the image and estimated the region where the clothes might be to get a region of interest. We ran this region of interest through the classifier to get results. Using R-CNN like Brian Lao et al. (2015) [1] used would not have been feasible for this project due to time constraints and lack of knowledge on the subject.

4 Implementation

For our project, we took two approaches. First approach was to classify the clothes only without detection and the second was to classify and detect the clothing item. For classification only, we used the Inception Model that was trained on ImageNet data set and fine-tuned the model using a method called transfer learning. Convolutional Neural Networks set weights and biases, through which it can classify, objects. Transfer learning takes an already pre-trained model with initialized weights and biases and replaces the final layer of the model with a layer based on a given dataset. Through back propagation, it

tweaks the weights and biases to minimize the loss function. Since the weights will not start from zero, rather they will start from a value close to the expected weights, it is a fast and accurate approach to classifying objects. For this approach, we used only 3 classes – Jacket, shirt and sweater and on the ACS dataset with 1000 images from each class.

For our second approach, we fine-tuned a pre-trained Alexnet on 5 classes- Jackets, shirts, sweaters, pants and shorts. There were 5000 images in each class, 80% was used for training and the rest for testing. To detect a person in an image we used the Viola M Jones face detector function in Matlab. This helped us locate the upper body of the person in the image.

Figure 1: Image of person with boundary box around face.

As seen on Figure 1, a boundary box was drawn around the upper body. We use this boundary box to estimate two regions of interest – Torso and legs.

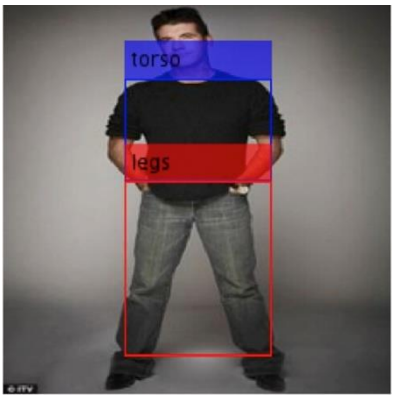

Figure 2: image with torso and leg labeled.

Moving and scaling the original boundary box around the face estimated the region of interests shown in Figure 2. We ran the region of interests to get the label of the clothing item.

Since we used a face detector, we had to take some assumptions for the program to run. The assumptions are that there should only be one person in the image, the person should be looking straight at the camera and be standing upright.

5 Experimental Results:

Below are our results on training accuracy and some tests that we ran on our pictures. Using inception model in tensorflow we got an accuracy of 97.5 with 1000 iterations.

```
2017-05-02 21:18:59.190669: Step 900: Train accuracy = 95.0%
2017-05-02 21:18:59.190724: Step 900: Cross entropy = 0.159681
2017-05-02 21:18:59.263980: Step 900: Validation accuracy = 95.0% (N=100)
2017-05-02 21:19:00.005765: Step 910: Train accuracy = 96.0%
2017-05-02 21:19:00.005873: Step 910: Cross entropy = 0.089757
2017-05-02 21:19:00.079153: Step 910: Validation accuracy = 98.0% (N=100)
2017-05-02 21:19:00.816369: Step 920: Train accuracy = 97.0%
2017-05-02 21:19:00.816425: Step 920: Cross entropy = 0.081593
2017-05-02 21:19:00.889612: Step 920: Validation accuracy = 98.0% (N=100)
2017-05-02 21:19:01.634963: Step 930: Train accuracy = 99.0%
2017-05-02 21:19:01.635020: Step 930: Cross entropy = 0.073641
2017-05-02 21:19:01.708324: Step 930: Validation accuracy = 99.0% (N=100)
2017-05-02 21:19:02.445205: Step 940: Train accuracy = 99.0%
2017-05-02 21:19:02.445259: Step 940: Cross entropy = 0.055930
2017-05-02 21:19:02.518066: Step 940: Validation accuracy = 97.0% (N=100)
2017-05-02 21:19:03.253162: Step 950: Train accuracy = 99.0%
2017-05-02 21:19:03.253215: Step 950: Cross entropy = 0.074049
2017-05-02 21:19:03.326031: Step 950: Validation accuracy = 98.0% (N=100)
2017-05-02 21:19:04.062321: Step 960: Train accuracy = 100.0%
2017-05-02 21:19:04.062374: Step 960: Cross entropy = 0.042008
2017-05-02 21:19:04.135018: Step 960: Validation accuracy = 95.0% (N=100)
2017-05-02 21:19:04.872827: Step 970: Train accuracy = 100.0%
2017-05-02 21:19:04.872883: Step 970: Cross entropy = 0.038199
2017-05-02 21:19:04.946149: Step 970: Validation accuracy = 97.0% (N=100)
2017-05-02 21:19:05.681384: Step 980: Train accuracy = 97.0%
2017-05-02 21:19:05.681437: Step 980: Cross entropy = 0.073152
2017-05-02 21:19:05.754088: Step 980: Validation accuracy = 98.0% (N=100)
2017-05-02 21:19:06.488275: Step 990: Train accuracy = 99.0%
2017-05-02 21:19:06.488360: Step 990: Cross entropy = 0.051503
2017-05-02 21:19:06.561134: Step 990: Validation accuracy = 96.0% (N=100)
2017-05-02 21:19:07.223820: Step 999: Train accuracy = 97.0%
2017-05-02 21:19:07.223873: Step 999: Cross entropy = 0.092597
2017-05-02 21:19:07.295903: Step 999: Validation accuracy = 97.0% (N=100)
Final test accuracy = 97.5% (N=353)
Converted 2 variables to const ops.
```

Figure-3: accuracy results for inception model.

The above figure shows our results using inception model in tensorflow we got an accuracy of 97.5 with 1000 iterations.

```
>> ec520train
Training on single GPU.
Initializing image normalization.
```

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	4.13	0.7151	51.56%	0.0010
2	50	27.83	0.0052	100.00%	0.0010
4	100	51.90	0.0010	100.00%	0.0010
6	150	79.67	0.0005	100.00%	0.0010
8	200	106.93	0.0003	100.00%	0.0010
10	250	131.17	0.0002	100.00%	0.0010
12	300	155.65	0.0002	100.00%	0.0010
14	350	180.62	0.0002	100.00%	0.0010
16	400	204.02	0.0001	100.00%	0.0010
18	450	227.57	0.0001	100.00%	0.0010
20	500	251.22	0.0001	100.00%	0.0010

Figure-4: accuracy results of AlexNet

Using Alexnet we got a minibatch accuracy of 100% and 97.6% accuracy with 20 epochs.

Some test results on the pictures which we ran our programs on :

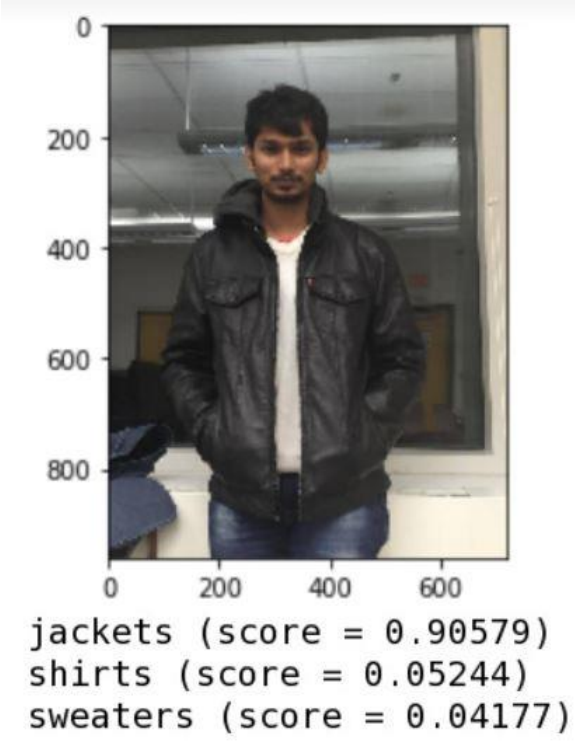

Figure-5: Accurately labeled image with jackets.

This shows that our network classifies person wearing jacket with a 90% accuracy which is correct.

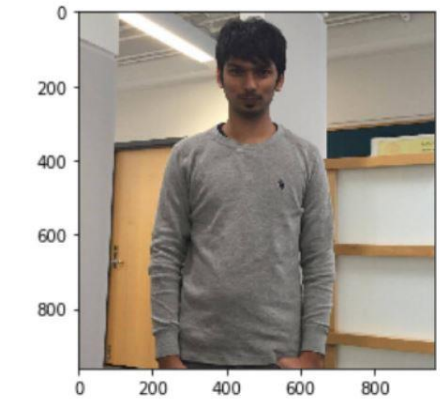

Figure-6 :Accuratly labeled image with sweaters.

The above figure shows that our network classifies person wearning sweater with a 96% accuracy which is correct.

We used AlexNet in Matlab to classify images , used person detector to get the location of person in the picture and used boundary boxes to classify the location of image.

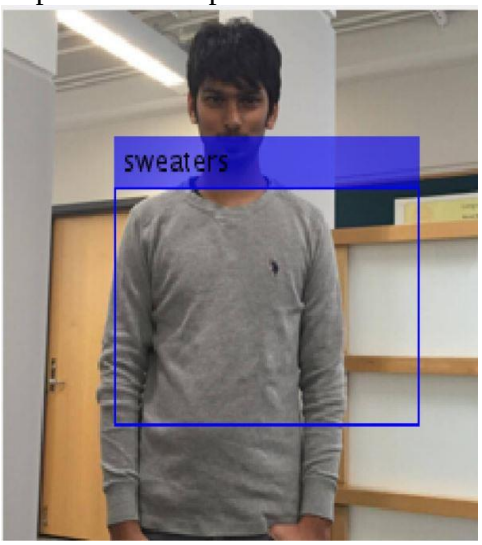

Figure-7:accuratly labeled image with bbox

The above figure shows that our classifier classified a person wearing a sweater which is correct.

Figure-8: accuratly labeled image with bbox

The above figure shows that our classifier classified a person wearing a shirt and a pant and drew a box around it which is accurate.

Figure 9 : accuratly labeled image

The above figure shows that our classifier classified a person wearing a shirt and shorts and our network classified it accuratly.

Failed Results:

Below are some of the results where the classifer failed to classify accuractly in the given box region.

Figure 10 : incorrectly labeled shorts as sweaters

The above figure shows that our classifier classified a person wearing a shirt accuratly but it failed to detect shorts and our network classified it as sweaters which is not correct.

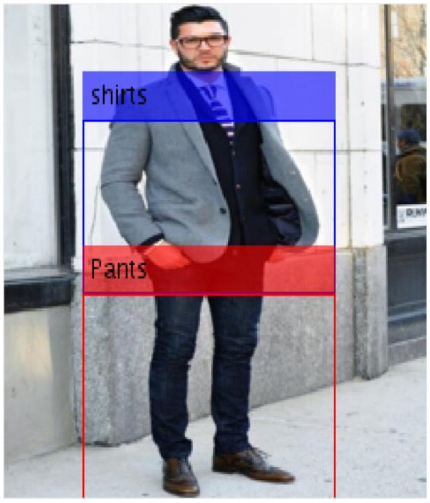

Figure 11 :incorrectly labeled jackets as shirts.

The above figure shows that our classifier classified a person wearing a pants accuratly but it failed to detect Jackets and our network classified it as Shirts which is not correct.

Figure-12: Results using R-CNN

We used R-CNN to classify images , but it was taking very long time for very short data so we switched to person detector.

4 Conclusions and possible improvements

Since we had several failed results, we could implement our classification and detection in a different way. First of all, we could avoid the Viola M Jones Face detector and use R-CNN to get better results. The face detector had several issues for which we had to make several assumptions for our program to run. Secondly, we could use another dataset. The dataset we had was fine for the classes we had to predict, but the dataset for cloaks had the Indian dress, Sari, in it. This isn't a cloak and could lead to false results. We only trained 5 classes for the project; possibly in the future we could do several more and make a better recognition program.

References

1] B. Lao, K. Jagadeesh. Convolutional Neural Networks for Fashion Classification and Object Detection, 2015. http://cs231n.stanford.edu/reports

[2] L.Bossard, M.Dantone, C.Leistner, C.Wengert, T.Quack, L.Gool. Apparel Classification with Style. http://people.ee.ethz.ch/~lbossard/projects/accv12/index.htm