Machine Learning for Fashion Analysis

Practical application of fashion recognition using Deep Neural Networks

Fashion Analysis

"Half the money I spend on advertising is wasted; the trouble is I don't know which half." – John Wanamaker, 1922

Problem In fashion retail

→ *What are customers interested in?*

→ *How can I convince the customer to buy my products?*

Machine Leaning Solution:

- Let's build a fashion image analyzer, to analyze existing fashions and better reach the customer through targeted advertising

Challenges:

→ *What is important in the image?*

→ *What should the ML model recognize?*

→ *How can the ML model distinguish between a "Shirt" and a "Blouse"?*

Conclusion: A ML model needs to understand the key characteristics to analyze

1: “Breakfast at Tiffany’s”: http://youtube.com
2: “StreetStyle”: http://streetstyle.cs.cornell.edu/
3: “online fashion images”: http://de.zalando.ch

Fashion Analysis

Example usage:

Classic Retail *1

Customer approaches store

Online Retail

*2

Customer uploads

Bounding box location

DL

Category: ‘full’

ML Image Detector model

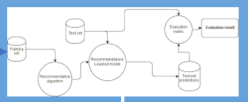
DL

ML Image Classifier model

Class: ‘dress’

ML models detect the current clothing items from image

“My Boutique”



Recommender Engine

*3

Match current pattern, recommend possible interesting items for customer from store

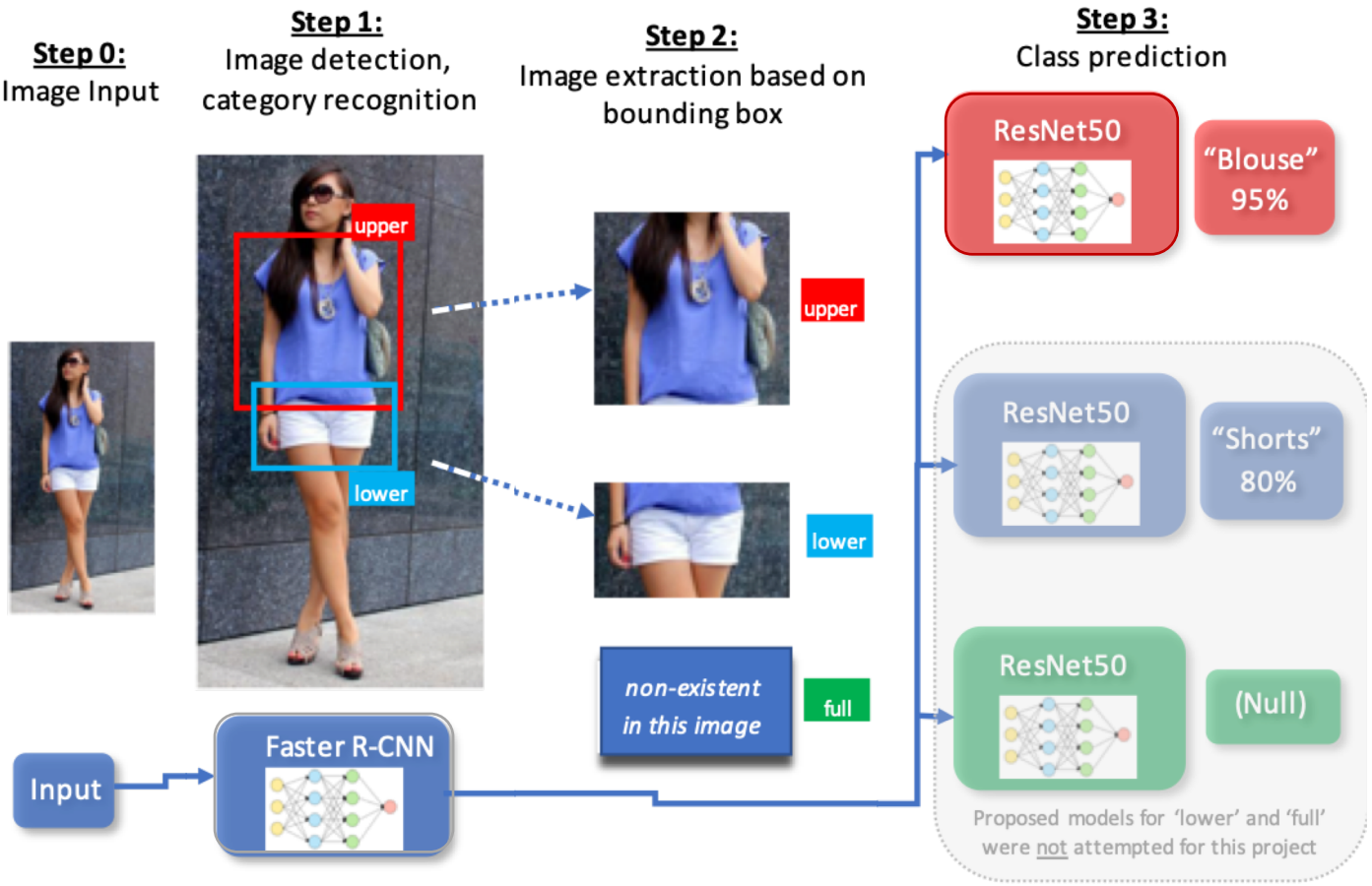
Use Case: Evaluation of image for fashion recommendation

*1

1: "StreetStyle": http://streetstyle.cs.cornell.edu/

Fashion Analysis Models

Solution: Separate detection and classification tasks into separate models



By reducing complexity into separate tasks, a better result can be achieved
A multi-model approach was selected for analysis:

- Image Detector: "Faster R-CNN" based model
- Image Classifier: "ResNet50" based models

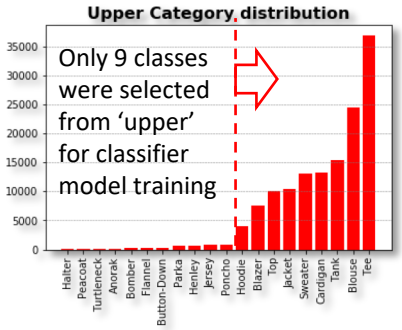
1: "DeepFashion": http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html
2: "StreetStyle": http://streetstyle.cs.cornell.edu/

Fashion Datasets

DeepFashion:
training dataset of fashion items

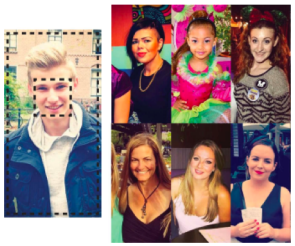
StreetStyle:
test dataset for fashion "in the wild"

DeepFashion[1]

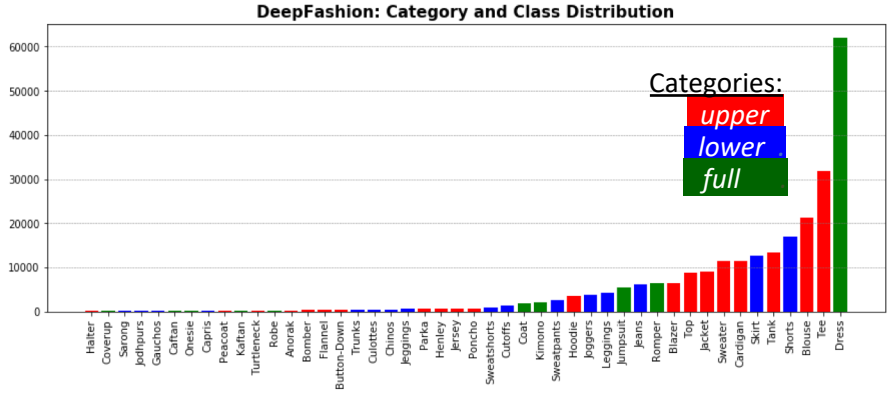


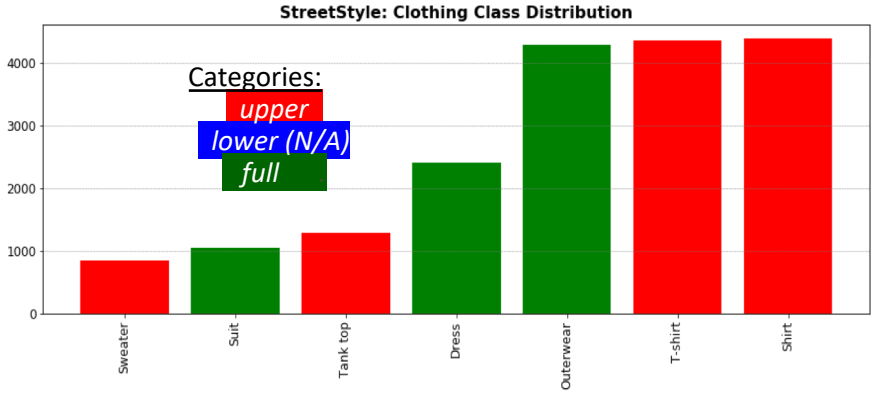
289k fashion images,
46 classes with bounding box annotations
Mostly curated fashion images

StreetStyle[2]



27k face & torso images
5 classes with bounding box annotation
Fashion images "in-the-wild"



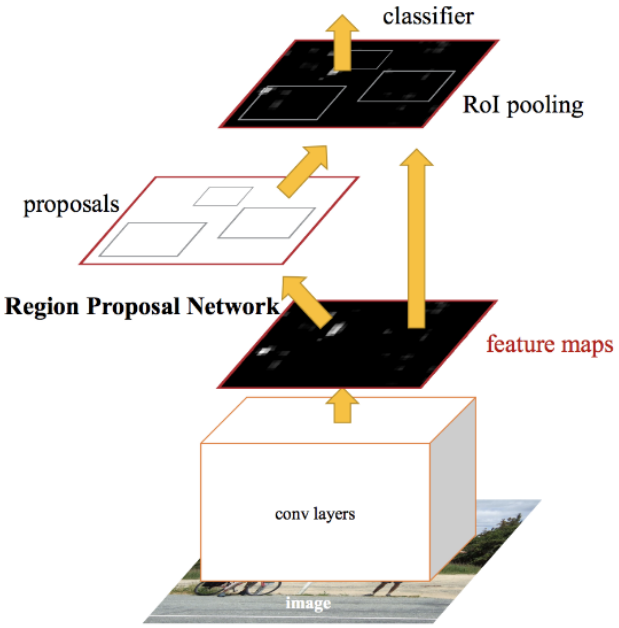


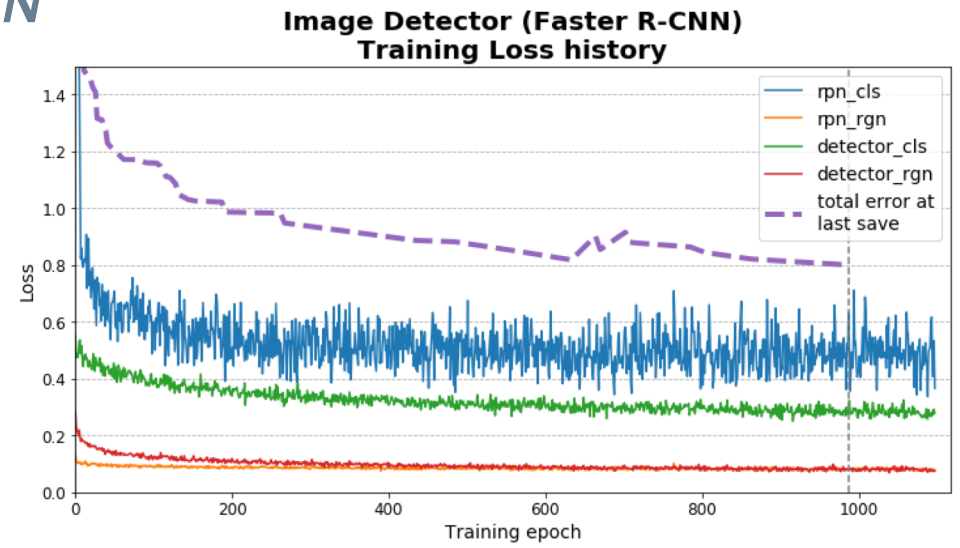
Training the models used only curated images
Testing models sets used both curated and "in-the-wild" type images

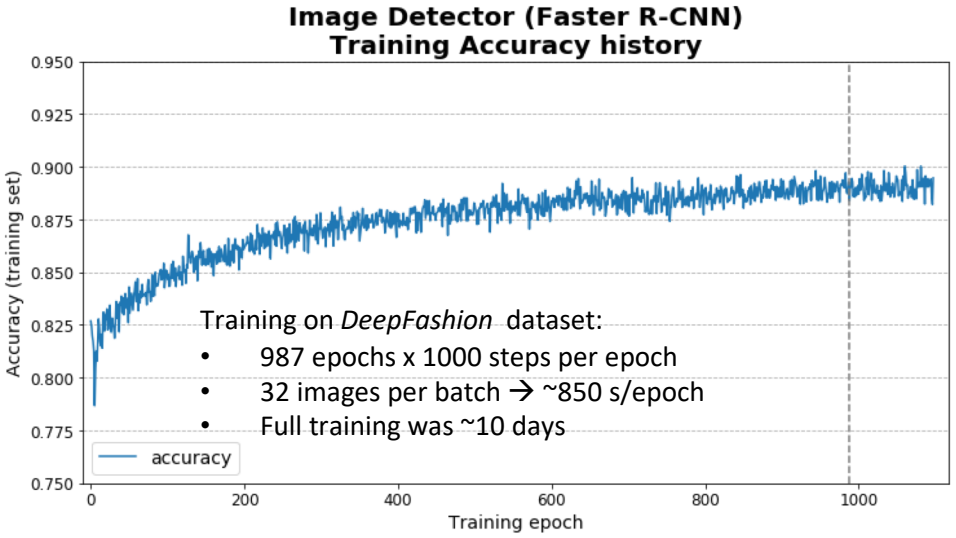
1: "Faster R-CNN": https://arxiv.org/abs/1512.03385

Image Detector: *Faster R-CNN*

Image Detector model:
"Faster R-CNN" architecture







Training on *DeepFashion* dataset:
- 987 epochs x 1000 steps per epoch
- 32 images per batch → ~850 s/epoch
- Full training was ~10 days

Categories:
- 'upper'
- 'lower'
- 'full'

Bounding box:

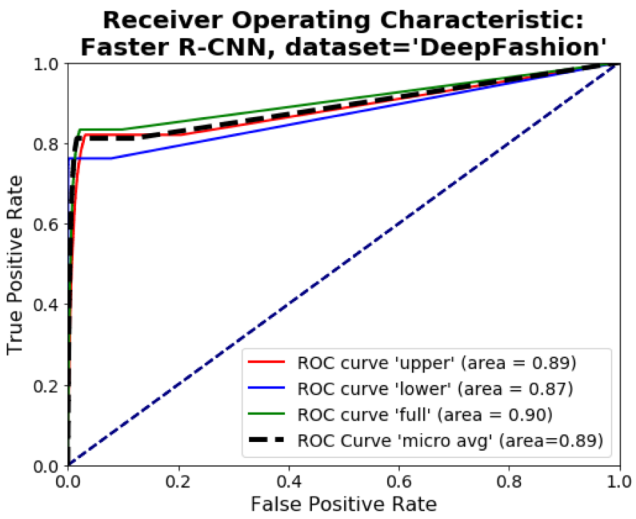
(x_1, y_1, x_2, y_2)

Image detector model was trained for 3 categories with bounding box information

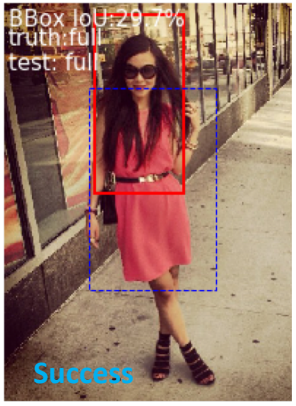
Image Detector: *Faster R-CNN*

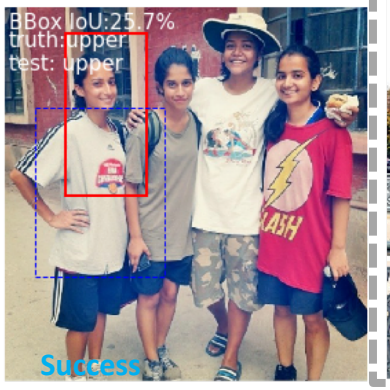
DeepFashion:

Overall Results of test datasets



StreetStyle:





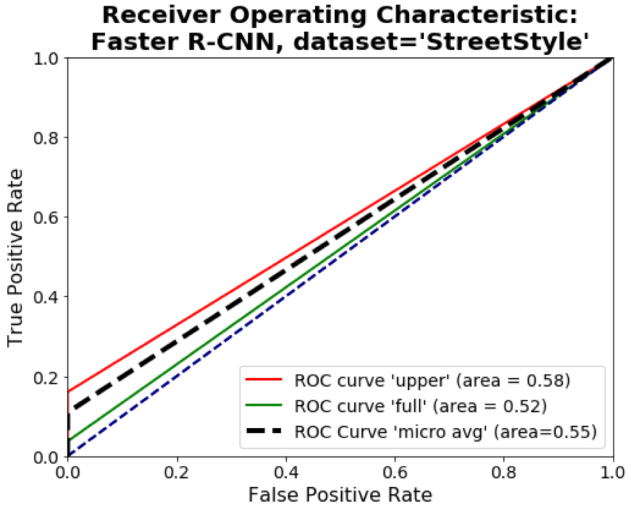
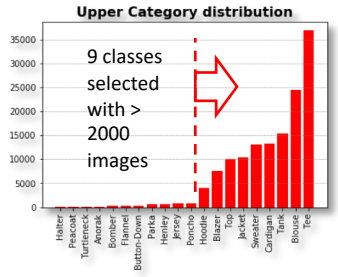
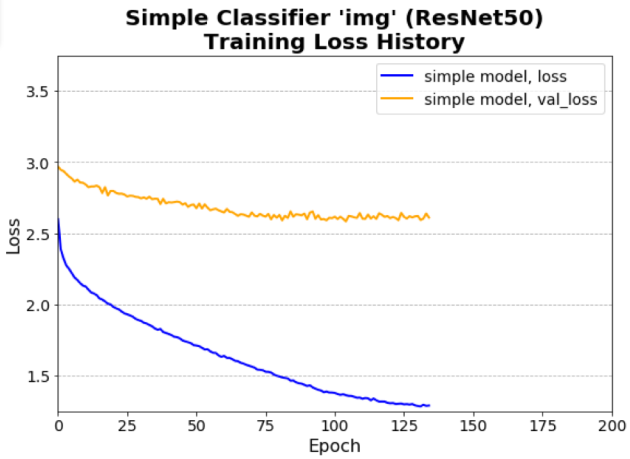


Image Detector: *ResNet50*

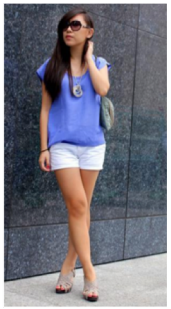
Image Classifier models:

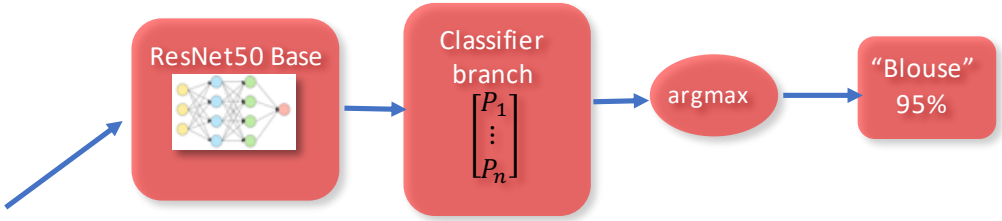
"ResNet50" architecture with and without bounding box entrainment

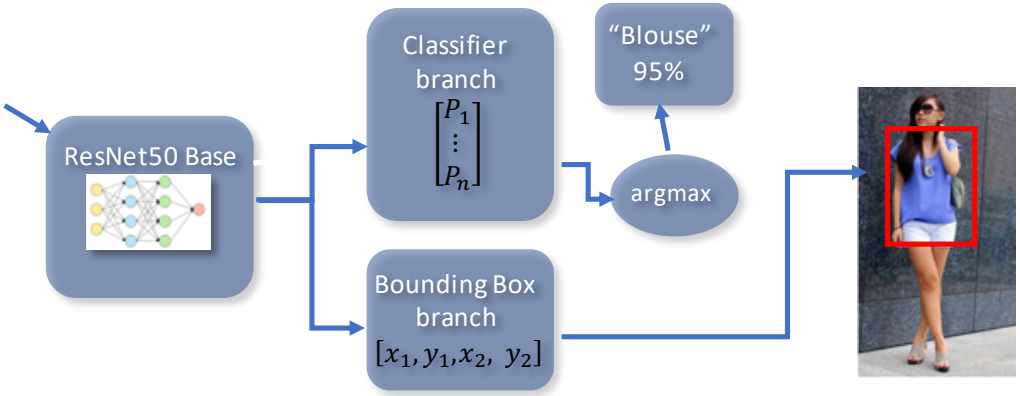




Simple classifier 'img'







Feature classifier 'img+bbox'

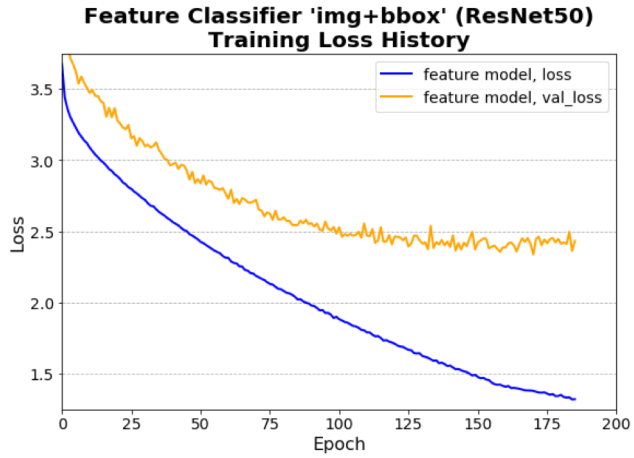
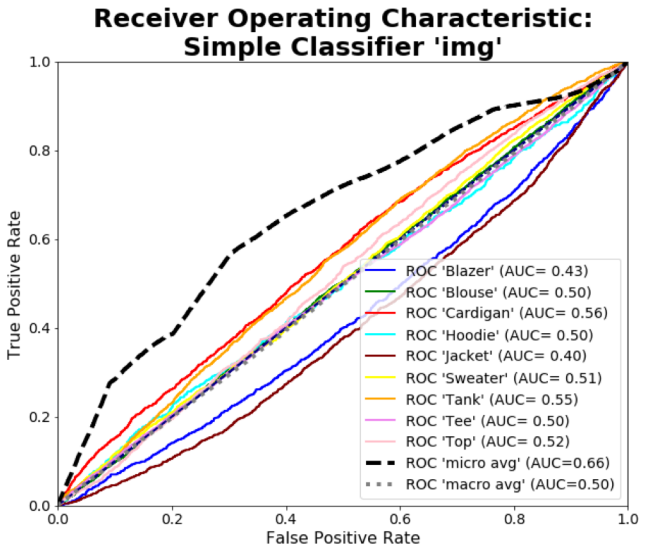


Image Classifier 'img' and 'img+bbox' models were both attempted to determine the influence of feature inclusion in model training

Image Classifier: *ResNet50*

Simple Classifier 'img'

Overall Results of test datasets



Feature Classifier 'img+bbox'

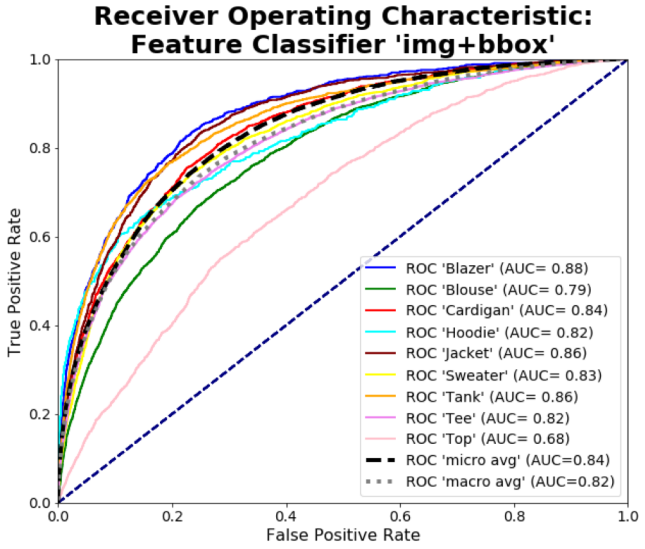
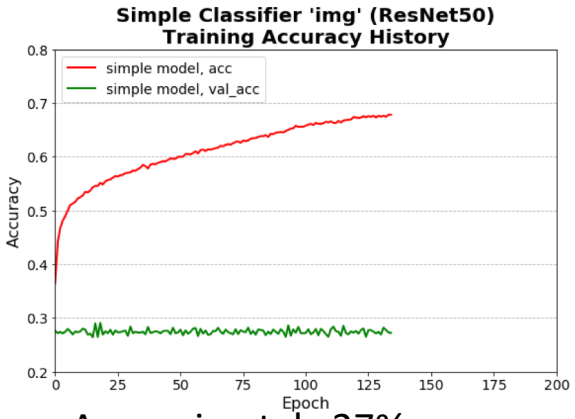


Image Detector: *ResNet50*

What went wrong with the Simple Classifier ('img') ?

Simple classifier

class	images of class in test set	Case 1 y_test_Img == y_true, y_test_ImgBB == y_true	Case 2 y_test_Img != y_true, y_test_ImgBB != y_true	Case 3 y_test_Img == y_true, y_test_ImgBB != y_true	Case 4 y_test_Img != y_true, y_test_ImgBB == y_true	*'img' model correct*	*'img+bbox' model correct*
Blazer	*1040*	0	491	0	549	*0*	*549 (53%)*
Blouse	*3389*	0	1941	0	1448	*0*	*1448 (43%)*
Cardigan	*1828*	0	766	0	1062	*0*	*1062 (58%)*
Hoodie	*541*	0	358	0	183	*0*	*183 (34%)*
Jacket	*1473*	0	1239	0	234	*0*	*234 (16%)*
Sweater	*1822*	0	1353	0	469	*0*	*469 (26%)*
Tank	*2097*	0	1284	0	813	*0*	*813 (39%)*
Tee	*5143*	3293	0	1850	0	*5143 (100%)*	*3293 (64%)*
Top	*1387*	0	1084	0	303	*0*	*303 (22%)*
Totals	*18720*	3293	8516	1850	5061	*5143 (27%)*	*8354 (45%)*



Approximately 27% accuracy which is equal to N_{tee} / N_{total}

…the Simple Classifier minimized its error by choosing the majority class for every prediction. The resulting model is useless for prediction…

By better feature engineering, more attributes can be linked with the image aspects and thus lead to better model predictions

Conclusions

What is important for fashion image analysis?

…Datasets:

- *A well-balanced class distribution is as important as number of samples*
- *A model trained on curated images will learn to predict only curated images…a better model will be trained on images as you wish to detect them ("in-the-wild")*
- *However, make sure there is something to learn from the images ("annotations")*

…Models:

- *In general, a model which addresses more features ('breadth') will outperform a complicated model ('depth') which does not address the features of the image.*
- *Be sure to check the end results, to make sure your model didn't take a 'lazy' approach (i.e. guess the majority class)*

Questions?

Original clip: “Breakfast at Tiffany’s”