Hierarchical Category Detector for Clothing Recognition from Visual Data

Suren Kumar
Markable Inc.
suren@markable.ai

Rui Zheng
Markable Inc.
rui@markable.ai

Abstract

Clothing detection is an important step for retrieving similar clothing items, organizing fashion photos, artificial intelligence powered shopping assistants and automatic labeling of large catalogues. Training a deep learning based clothing detector requires pre-defined categories (dress, pants etc) and a high volume of annotated image data for each category. However, fashion evolves and new categories are constantly introduced in the marketplace. For example, consider the case of jeggings which is a combination of jeans and leggings. Detection of this new category will require adding annotated data specific to jegging class and subsequently relearning the weights for the deep network. In this paper, we propose a novel object detection method that can handle newer categories without the need of obtaining new labeled data and retraining the network. Our approach learns the visual similarities between various clothing categories and predicts a tree of categories. The resulting framework significantly improves the generalization capabilities of the detector to novel clothing products.

1. Introduction

Object detection in images and videos is an important computer vision research problem. It enables selection of the relevant region of interest for a specific category paving the way for a multitude of computer vision tasks including similar object search, object tracking, and collision avoidance for self-driving cars. Object detection performance is affected by multiple challenges including imaging noises (motion blur, lighting variations), scale, object occlusion, self-occlusion and appearance similarity with the background or other objects. Significant progress has been made on object detection by moving from the early feature based algorithms [1] to deep learning based end-to-end frameworks [9, 6, 10]. The focus has been to improve separation of objects belonging to a particular category from all the other objects and localization of the object in the image. However, this paradigm of going straight from images to object locations and their corresponding category ignores the correlation between multiple categories. The resulting methods have a high number of false positives because of classification errors between similar classes. Furthermore, addition of a novel object category requires re-training of the object detector.

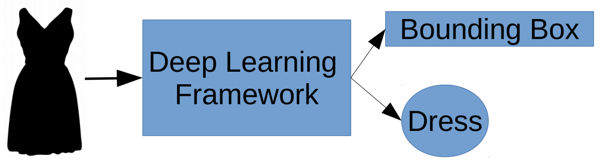
We propose an object detection framework that predicts a hierarchical tree as output instead of a single category. For example, for a *'t-shirt'* object, our detector predicts *['top innerwear'* $\mapsto$ *'t-shirt']*. The upper level category *'top innerwear'* includes *['blouses_shirts', 'tees', 'tanks_camis', 'tunics', 'sweater']*. The hierarchical tree is estimated by analyzing the errors of an object detector which does not model any correlation between the object cateogries. We propose three major research contributions;

1. We propose the first hierarchical detection framework for the clothing domain.
2. We propose a method to estimate the hierarchical/semantic tree by directly analyzing the detection errors.

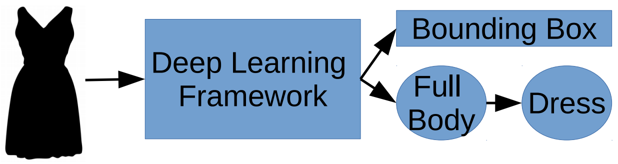
2. Related Work

Object detection computes bounding boxes and the corresponding categories for all the relevant objects using visual data. The category prediction often assumes that only one of the K total object categories is associated with each bounding boxes. The 1-of-K classification is often achieved by a 'Softmax' layer which encourages each object category to be as far away as possible from all the other object categories. However, this paradigm does not explicitly exploit the correlation information present in the object categories. For example, a 'jeans' is closer to 'pants' compared to 'coat'. We propose to exploit this correlation by first predicting 'lower body' and choosing one element from the 'lower body' category which is a set of 'jeans', 'pants', 'leggings' via hierarchical tree prediction.

There has been limited work on using the correlation between categories for improving detector performance. Redmon et. al [9] use a hierarchy based on WordNet [7] to combine multiple datasets. However, WordNet hierarchy and relationships are predefined by conceptual-semantic and lexi-



(a) Generic Detector



(b) Hierarchical Detector

Figure 1: Generic Object Detector and Hierarchical Detector. The hierarchical detector predicts a tree of categories as output compared to the generic detector that outputs a single category for each bounding box.

cal relationship. Wordnet hierarchy was also used by Deng et. al [2] to predict a visual categories at various level with a pre-specified accuracy. However, these predefined hierarchies do not take the visual similarity into account. For example, 'dress' and 'rompers' are not hyponyms of each other even though they are visually similar.

3. Methods

The proposed hierarchical prediction framework can be integrated with any existing object detector as the only change introduced is within the category prediction. Figure 1 shows the changes between the generic object detector and our detector. The generic detector can be any differentiable (e.g: any deep learning based detector) mapping $f(I) \mapsto bb, c$ that takes an input image I and produces a list of bounding boxes bb and a corresponding category c for each of the bounding box. The hierarchical detector learns a new differentiable mapping $f_h(I) \mapsto bb, \mathcal{F}(c)$ that produces a path/flow from the root category to the leaf category $\mathcal{F}(c)$ for each bounding box.

There are two steps involved in going from a generic detector to the hierarchical detector. First, we train a generic detector and estimate the category hierarchy tree as discussed in Section 3.2. Finally, using the category hierarchy, we retrain the deep learning framework with a loss function designed to predict the hierarchical category as detailed in Section 3.1.

3.1. Tree Prediction

We use the directed graph underlying the tree for predicting a tree/path from root node to leaf node for categories. Let T represent the entire tree consisting of all the categories as nodes and the hierarchical relationship as directed edges from parent node to children nodes. We use n, $s(n)$, $p(n)$, $\mathcal{F}(n)$ to denote the node, sibling set of a node, parent of a node, and path from the root node to a leaf node respectively. Consider a dummy directed graph as shown in Figure 2. All the nodes belonging to 'Level 0' are denoted as root nodes since they do not have any parents. Sibling set $s(n)$ denotes all the nodes that are on the same level and have a common parent. For example, $s(1) = \{1, 2, 3\}$ and $s(6) = \{4, 5, 6\}$. The path from the root to a leaf node includes all the nodes that lie on the way from a 'Level 0' node to a leaf node. For example, $\mathcal{F}(9) = 1, 6, 9$ and $\mathcal{F}(2) = 2$.

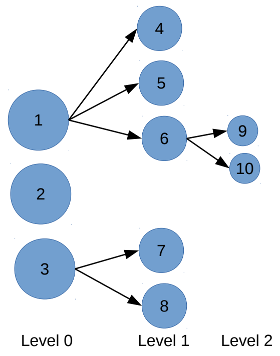


Figure 2: Instance of a category tree representing nodes at various levels

The probability of any node (or the category probability for a bounding box) given an image I is represented by $P(n|I)$. Using the underlying graph, this probability can also be expressed by a series of conditional probability over the path from the root node to the leaf node.

$$P(n|I) = P(l_0|I)P(l_1|l_0)....P(n|l_{q-1}) \quad (1)$$

where q is the total number of nodes along the path and all the nodes in the conditional probability computation belong to the path from the root to the leaf node, $\mathcal{F}(n) = (l_0, l_1, ..., l_{q-1}, n)$. We use the commonly used 'Softmax' layer to estimate the probability of each node [10]. Similar to the strategy of mixing datasets by Redmon et. al [9], we represent all the nodes in a single vector and have the last fully-connected (FC) layer predict scores for all of the nodes. Next, we use the underlying structure of the category tree to obtain probabilities for nodes at each level. For example, for a zero-th level node, one can write the probability as

$$P(l_0|I) = \frac{\exp c_0}{\sum_{c_i \in s(l_0)} \exp c_i} \quad (2)$$

where the 'Softmax' is only computed with respect to the sibling nodes. This encourages competition (1-of-K classification) only amongst siblings. Intuitively, our category estimator will first try to separate between major categories such as 'upper body', 'lower body', 'footwear', and subsequently estimate finer category for each of those categories, and so-on.

To adapt a generic detector to hierarchical detector, we use cross-entropy between the predicted distribution in Equation 1 and the ground-truth annotation.

$$\mathcal{L}(I) = -\sum_x q(x|I) \log P(x|I) \tag{3}$$

where x are the individual elements of the vector representing all the categories, $P(I)$ and $q(I)$ denote the category probability and annotation vector for image I respectively. Both of these vectors are of dimension $|T|$, which is also the total number of nodes in the category tree T. The generic detector only has a single active element (a single category) in the annotation vector but our detector will have multiple activations to account all the labels from root node to the leaf-node. Intuitively, the loss function enforces the deep neural network to learn a representation for the tree path by encoding hierarchical information.

We also need to modify the backward propagation step to learn parameters of the deep neural network in order to predict hierarchical categories. The usage of sibling level 'Softmax' and the underlying graph structures induces a multiplier factor for each category. Consider the graph in Figure 2, and assume that an input image has category 9. The presence of category 9 also indicates the presence of categories along the path from leaf to root $(6, 1)$. The loss represented in Equation 3 will have three different active labels $(1, 6, 9)$. The loss for this image can be written as

$$\begin{aligned}\mathcal{L}(I) &= -(\log P(1|I) + \log P(6|I) + \log P(9|I)) \\ &= -(\log P(1|I) + \log P(6|1)P(1|I) + \\ &\log P(9|6)P(6|1)P(1|I)) \\ &= -(3\log P(1|I) + 2\log P(6|1) + \log P(9|6))\end{aligned} \tag{4}$$

Notably, Equation 4 demonstrates that when calculating gradients during backpropagation, we simply need to multiply the each node's 'Softmax' gradient by a multiplier factor. Due to competition amongst siblings, sibling nodes share same multiplier factor. The above example can be generalized and we present the algorithm to estimate the multiplier factor for each node in Algorithm 1. Given the category tree T and ground truth annotation $q(x|I)$ for an image I, we can find the corresponding leaf node l_q and the path $\mathcal{F}(l_q)$ to the root node l_0, and subsequently assign the level-distance from l_q as multiplier factor for all the nodes on path $\mathcal{F}(l_q)$ and their sibling nodes. The multiplier factor is zero for all the other nodes.

3.2. Hierarchy Estimation

To estimate the category tree T, one needs to estimate the visual similarity between various categories. Redmon

```
Data: $q(x|I)$,$T$
Result: Multiplier factor $m(n)$ for all nodes
Initialize $m(n) = 0 \; \forall \; n \in T$;
Find leaf node $l_q$ from $q(x|I)$;
// Traverse over all nodes in path from leaf to root
for $l_i = l_q$ to $l_0$ do
    $m(n) = (q - i + 1) \; \forall \; n \in s(l_i)$;
end
```

Algorithm 1: Multiplier factor estimation for each node

et. al [9] use a pre-defined hierarchy based on WordNet [7]. To the best of our knowledge, there is no prior work that organizes the visually similar categories for an object detector. Prior work has focussed on using attribute-level annotations to generate an annotation tag hierarchy instead of category-level information. However, such an approach requires large amounts of additional human effort to annotate each category with information such as, viewpoint, object part location, rotation, object specific attributes. Ouyang et al. [8] generate an attribute-based (viewpoint, rotation, part location etc.) hierarchical clustering for each object category to improve detection. In contrast, we only use category level information and generate a single hierearchical tree for all the object categories.

We estimate a category hierarchy by first evaluating the errors of a generic detector trained without any consideration of distance between categories and subsequently analyze the cross-errors generated due to visual-similarity between various categories. In the current work, we train a Faster-RCNN based detector [10] and evaluate detector errors inspired by the work from Hoiem et al. [4]. Specifically, we look for the false positives generated by the generic detector (Faster-RCNN detector in the current case) and compute all the errors that result from visually similar categories. These errors are computed by measuring all the false positives with bounding boxes that have an intersection-over-union (IOU) ratio between 0.1 to 0.5 with another object category. Intuitively, visually similar classes such as 'shoes' and 'boots' will be frequently misclassified with each-other resulting in higher cross-category false positive errors.

We compute the cross-category false positive matrix C (Size$(C) = J \times (J + 1)$), where J denotes the total number of categories in the dataset. The second dimension is higher than the first dimension to account for false positives that only intersect with background. The diagonal entries of the matrix C reflect the false positives resulting from poor localization and are ignored in the current analysis. Using the matrix C and a predefined threshold τ, we estimate the sets of categories that are similar to each other. This results in disparate groups of categories. All the sets with greater than 1 element are given new category names and all the elements for that set are assigned as children to the newly defined category. The above process readily generates a 2-level tree for categories, which is detailed in algorithm 2. We can iteratively apply this process on current root level categories to grow the category tree. For our experiments on the clothing dataset, we found that a 2-level category tree is enough to characterize the visual-similarity between various categories.

```
Data: $C$,$\tau$
Result: $T$
Initialize $T = \emptyset$;
for $i = 1$ to $J$ do
    for $j = 1$ to $J$ do
        if $C[i][j] \geq \tau$ then
            if $i \parallel j \in n; n \in T$ then
                // Add to the existing group;
                $n = n \cup \{i, j\}$
            else
                // Start a new group;
                $n = \{i, j\}$;
                $T = T \cup n$
            end
        end
    end
end
```

Algorithm 2: Generating visually similar groups from cross-category false positive error matrix

3.3. Novel Category Detection

Prior work has focussed on using attribute-level information apart from the category specific information to perform detection for novel object categories. Farhadi et al. [3] use attribute level information to detect objects from novel categories. For example, a new object category 'horse' is recognized as a combination of 'legs', 'mammal' and 'animal' categories. Attribute-based recognition requires one to learn attribute specific classifiers and also needs attribute level annotation for each of the object categories. In comparison, our method requires neither attribute annotations nor any attribute specific classifiers. For each new category, we assign an expected root-level category and subsequently estimate a bounding box with highest confidence score for that category.

We perform category specific non-maximal suppression to select unique bounding boxes for each leaf node category. For all the lower level categories, we also suppress the output by considering bounding boxes from all the children nodes. This helps us reduce spurious lower level category boxes.

Composite	Original Category
Footwear	Shoes, Boots, Sandals
Full Body	Dresses, Jumpsuits, Rompers/Overalls
Top Innerwear	Blouses/Shirts, Tees, Tanks/Camis, Tunics, Sweater
Top Outerwear	Coats/Jackets, Suitings/Blazers
Bags	Handbags, Clutches, Tote
Lower Body	Jeans, Pants, Leggings
Headgear	Cowboy Hat, Beanie/Knit Cap

Table 1: New root-level categories and their children

4. Experiments

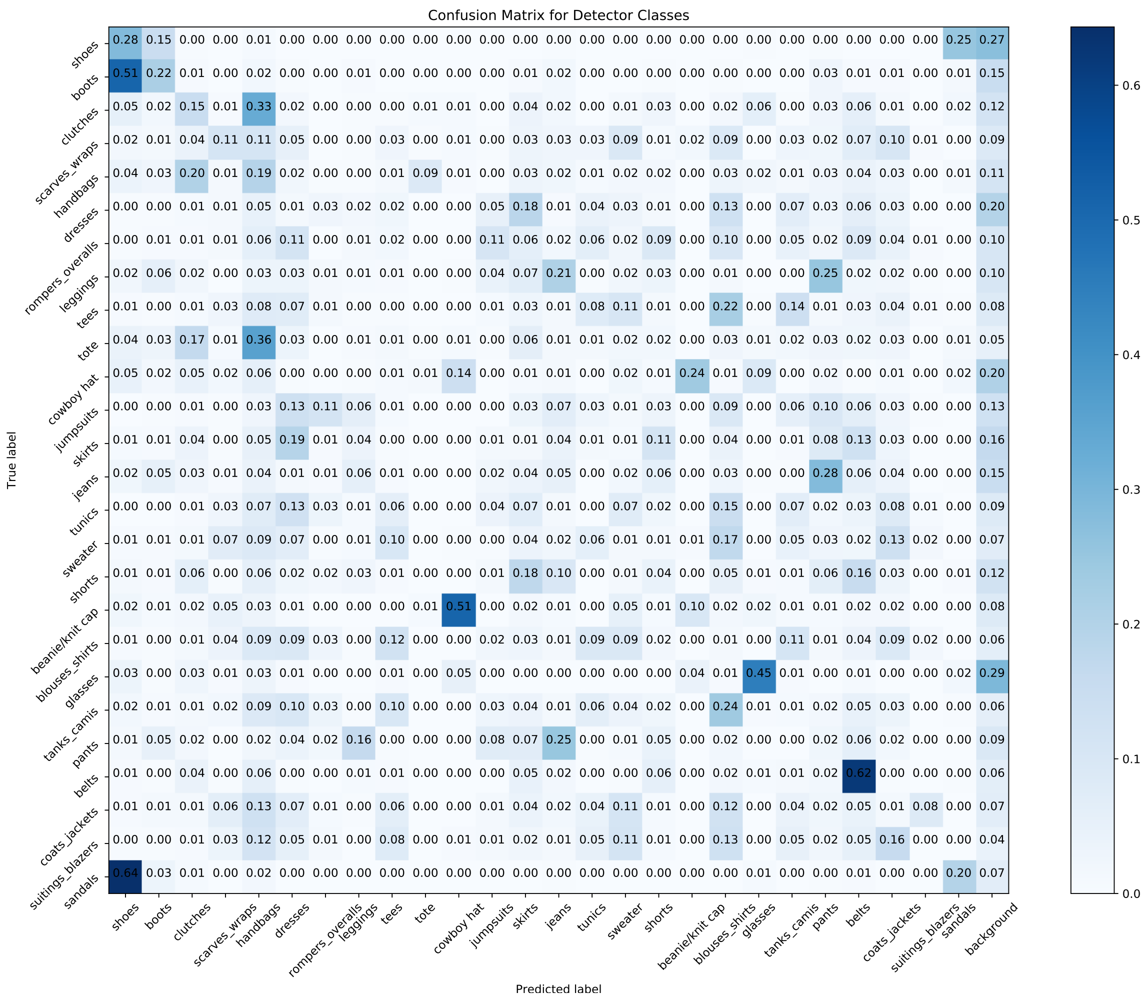
Due to the lack of a large standard dataset for clothing detection, we collect a large dataset of $97,321$ images from various fashion relevant websites, such as 'www.modcloth.com','www.renttherunway.com'. For all the images, we obtain human-annotation for all the fashion relevant items resulting in a total of $404,891$ bounding boxes across 43 different categories. We ignore all the categories that have less than 400 bounding boxes for training the object detector resulting in 26 valid categories. We split the dataset into training and testing set $80\% - 20\%$. Both detectors are trained and evaluated using the same training and testing set. We use the open-source deep learning framework CAFFE [5] for all of our experiments. During training, we use the same hyperparameters for both detectors and train them for $250,000$ iterations using RMSprop [11] optimizer.

4.1. Results

4.1.1 Category Tree Estimation

Following the standard practice for detector [10], we compute average precision for all different categories and summarize the results across categories using the mean-average precision. Average precision measure the area under the precision-recall curve for each object category. We use 0.5 pascal ratio as the threshold for true positive. We train the baseline generic detector on our dataset to compute the cross-error matrix C. Figure 3 shows the normalized error matrix.

As expected, from Figure 3, it is clear that the visually similar categories like 'shoes' and 'boots' are frequently misclassified with each other. We use Algorithm 2 to estimate the tree T based on detector error matrix C. Our algorithm finds 7 groups containing more than one element. Details of all the groups thus generated and their names are given in Table 1.



	shoes	boots	clutches	scarves_wraps	handbags	dresses	rompers_overalls	leggings	tees	tote	cowboy hat	jumpsuits	skirts	jeans	tunics	sweater	shorts	beanie/knit cap	blouses_shirts	glasses	tanks_camis	pants	belts	coats_jackets	suitings_blazers	sandals	background
shoes	0.28	0.15	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.27
boots	0.51	0.22	0.01	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.01	0.00	0.01	0.15
clutches	0.05	0.02	0.15	0.01	0.33	0.02	0.00	0.00	0.00	0.01	0.01	0.00	0.04	0.02	0.00	0.01	0.03	0.00	0.02	0.06	0.00	0.03	0.06	0.01	0.00	0.02	0.12
scarves_wraps	0.02	0.01	0.04	0.11	0.11	0.05	0.00	0.00	0.03	0.00	0.01	0.00	0.03	0.03	0.03	0.09	0.01	0.02	0.09	0.00	0.03	0.02	0.07	0.10	0.01	0.00	0.09
handbags	0.04	0.03	0.20	0.01	0.19	0.02	0.00	0.00	0.01	0.09	0.01	0.00	0.03	0.02	0.01	0.02	0.02	0.00	0.03	0.02	0.01	0.03	0.04	0.03	0.00	0.01	0.11
dresses	0.00	0.00	0.01	0.01	0.05	0.01	0.03	0.02	0.02	0.00	0.00	0.05	0.18	0.01	0.04	0.03	0.01	0.00	0.13	0.00	0.07	0.03	0.06	0.03	0.00	0.00	0.20
rompers_overalls	0.00	0.01	0.01	0.01	0.06	0.11	0.00	0.01	0.02	0.00	0.00	0.11	0.06	0.02	0.06	0.02	0.09	0.00	0.10	0.00	0.05	0.02	0.09	0.04	0.01	0.00	0.10
leggings	0.02	0.06	0.02	0.00	0.03	0.03	0.01	0.01	0.01	0.00	0.00	0.04	0.07	0.21	0.00	0.02	0.03	0.00	0.01	0.00	0.00	0.25	0.02	0.02	0.00	0.00	0.10
tees	0.01	0.00	0.01	0.03	0.08	0.07	0.01	0.00	0.00	0.00	0.00	0.01	0.03	0.01	0.08	0.11	0.01	0.00	0.22	0.00	0.14	0.01	0.03	0.04	0.01	0.00	0.08
tote	0.04	0.03	0.17	0.01	0.36	0.03	0.00	0.01	0.01	0.00	0.01	0.00	0.06	0.01	0.01	0.02	0.02	0.00	0.03	0.01	0.02	0.03	0.02	0.03	0.00	0.01	0.05
cowboy hat	0.05	0.02	0.05	0.02	0.06	0.00	0.00	0.00	0.01	0.01	0.14	0.00	0.01	0.01	0.00	0.02	0.01	0.24	0.01	0.09	0.00	0.02	0.00	0.01	0.00	0.02	0.20
jumpsuits	0.00	0.00	0.01	0.00	0.03	0.13	0.11	0.06	0.01	0.00	0.00	0.00	0.03	0.07	0.03	0.01	0.03	0.00	0.09	0.00	0.06	0.10	0.06	0.03	0.00	0.00	0.13
skirts	0.01	0.01	0.04	0.00	0.05	0.19	0.01	0.04	0.00	0.00	0.00	0.01	0.01	0.04	0.01	0.01	0.11	0.00	0.04	0.00	0.01	0.08	0.13	0.03	0.00	0.00	0.16
jeans	0.02	0.05	0.03	0.01	0.04	0.01	0.01	0.06	0.01	0.00	0.00	0.02	0.04	0.05	0.00	0.02	0.06	0.00	0.03	0.00	0.00	0.28	0.06	0.04	0.00	0.00	0.15
tunics	0.00	0.00	0.01	0.03	0.07	0.13	0.03	0.01	0.06	0.00	0.00	0.04	0.07	0.01	0.00	0.07	0.02	0.00	0.15	0.00	0.07	0.02	0.03	0.08	0.01	0.00	0.09
sweater	0.01	0.01	0.01	0.07	0.09	0.07	0.00	0.01	0.10	0.00	0.00	0.00	0.04	0.02	0.06	0.01	0.01	0.01	0.17	0.00	0.05	0.03	0.02	0.13	0.02	0.00	0.07
shorts	0.01	0.01	0.06	0.00	0.06	0.02	0.02	0.03	0.01	0.00	0.00	0.01	0.18	0.10	0.00	0.01	0.04	0.00	0.05	0.01	0.01	0.06	0.16	0.03	0.00	0.01	0.12
beanie/knit cap	0.02	0.01	0.02	0.05	0.03	0.01	0.00	0.00	0.00	0.01	0.51	0.00	0.02	0.01	0.00	0.05	0.01	0.10	0.02	0.02	0.01	0.01	0.02	0.02	0.00	0.00	0.08
blouses_shirts	0.01	0.00	0.01	0.04	0.09	0.09	0.03	0.00	0.12	0.00	0.00	0.02	0.03	0.01	0.09	0.09	0.02	0.00	0.01	0.00	0.11	0.01	0.04	0.09	0.02	0.00	0.06
glasses	0.03	0.00	0.03	0.01	0.03	0.01	0.00	0.00	0.01	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.01	0.45	0.01	0.00	0.01	0.00	0.00	0.02	0.29
tanks_camis	0.02	0.01	0.01	0.02	0.09	0.10	0.03	0.00	0.10	0.00	0.00	0.03	0.04	0.01	0.06	0.04	0.02	0.00	0.24	0.01	0.01	0.02	0.05	0.03	0.00	0.00	0.06
pants	0.01	0.05	0.02	0.00	0.02	0.04	0.02	0.16	0.00	0.00	0.00	0.08	0.07	0.25	0.00	0.01	0.05	0.00	0.02	0.00	0.00	0.02	0.06	0.02	0.00	0.00	0.09
belts	0.01	0.00	0.04	0.00	0.06	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.05	0.02	0.00	0.00	0.06	0.00	0.02	0.01	0.01	0.02	0.62	0.00	0.00	0.00	0.06
coats_jackets	0.01	0.01	0.01	0.06	0.13	0.07	0.01	0.00	0.06	0.00	0.00	0.01	0.04	0.02	0.04	0.11	0.01	0.00	0.12	0.00	0.04	0.02	0.05	0.01	0.08	0.00	0.07
suitings_blazers	0.00	0.00	0.01	0.03	0.12	0.05	0.01	0.00	0.08	0.00	0.01	0.01	0.02	0.01	0.05	0.11	0.01	0.00	0.13	0.00	0.05	0.02	0.05	0.16	0.00	0.00	0.04
sandals	0.64	0.03	0.01	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.20	0.07

Figure 3: Cross classification matrix with false positives errors between various categories

4.1.2 Accuracy Computation

Table 2 shows the mAP comparison between the generic and the proposed hierarchical detector. Since the generic detector does not generate any of the newly generate groups, we generate AP results for the new categories by averaging the performance across their children. This is reasonable since the detection of 'Dress' or 'Jumpsuits' also indicates the presence of 'Full Body' clothing category. Our results show that hierarchical detector significantly improves the mAP over the generic detector on new categories, and it also improves the mAP of generic classes by 0.8%. Notably, the improvement in the performance of hierarchical detector is because of the ability to capture visual information at a higher level.

Hierarchical detector can reduce missing detections from the generic detector for ambiguous examples. For example, it is hard to clearly identify the type of 'top innerwear' occluded by a 'coat' or 'jacket'. But the hierarchical detector can still detect that the clothing item hidden underneath is an instance of 'top innerwear' because of the hierarchical information representation. Figure 4 shows some examples of ambiguous instances that are identified by the hierarchical detector. Hierarchical detector also suppresses sibling output in constrast to generic detector as shown in Figure 5.

4.1.3 Novel Category Detection

The hierarchical nature of our detection output allows us to represent information at various scales. For example, 'Top Innerwear' category captures the commonalities between all the children categories. We use this aspect of our framework to perform detection on novel categories that the generic detector has not been directly trained on. Each novel category is assign as a root-level category and we compute the maximum confidence detection for both the children and root-level category. We collect a small test-set where the generic detector fails because these are novel

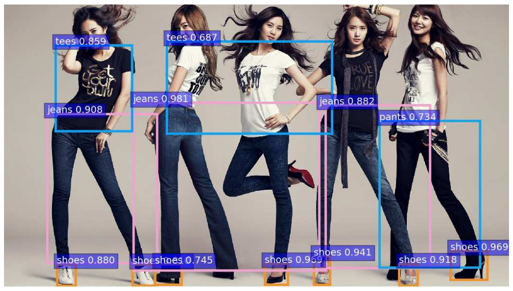
Category	Generic	Hierarchical
Shoes	0.8835	0.8880
Jeans	0.8898	0.8996
Boots	0.7765	0.7847
Tanks/Camis	0.4932	0.4922
Rompers/Overalls	0.3722	0.4239
Tunics	0.2281	0.2140
Scarves/Wraps	0.4140	0.3644
Coats/Jackets	0.8035	0.8143
Handbags	0.8066	0.8115
Sweater	0.6763	0.6776
Dresses	0.9712	0.9729
Pants	0.6038	0.6070
Clutches	0.6505	0.6574
Shorts	0.8447	0.8485
Leggings	0.1721	0.1933
Sandals	0.6320	0.6496
Tees	0.4836	0.4982
Beanie/Knit cap	0.7192	0.7745
Tote	0.2162	0.2393
Belts	0.2383	0.2296
Cowboy hat	0.9175	0.9239
Blouses/Shirts	0.6842	0.6993
Glasses	0.7696	0.7773
Suitings/Blazers	0.1865	0.1810
Skirts	0.7061	0.7052
Jumpsuits	0.6966	0.7190
mAP	0.6091	0.6172
Footwear	0.7640	0.8955
Headgear	0.8184	0.8883
Top Innerwear	0.5131	0.7702
Top Outerwear	0.4950	0.7379
Full Body	0.6800	0.9368
Lower Body	0.5552	0.9261
Bags	0.5578	0.7576
mAP	0.6127	0.6654

Table 2: Average Precision for Generic (Faster-RCNN) and its Hierarchical version for each category

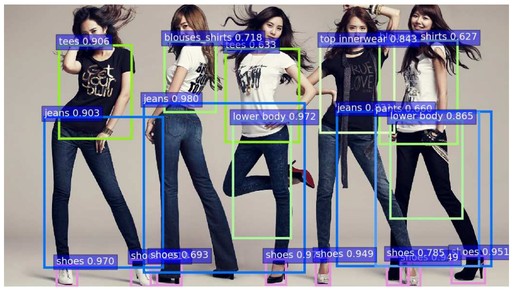
categories. The results of this set are demonstrated in Table 3. It shows the hierarchical detector can fix the failures of the generic detector by detection of root categories.

5. Conclusion

We proposed a novel framework for predicting hierarchical categories for a detector. The hierarchy between categories is only based on visual similarity. Our hierarchical detector demonstrates the ability to capture information at various scales and generalizes the detector to novel categories that our detector has not been directly trained on. The proposed detector improves detection performance over the state-of-the-art detector on a clothing dataset.

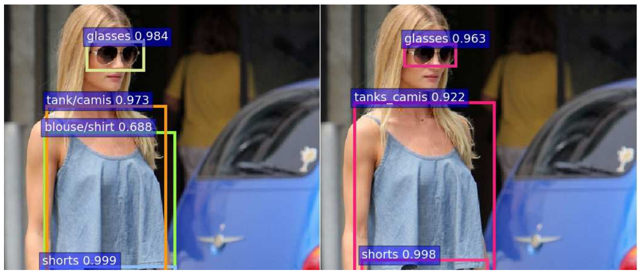


(a) Generic Detector



(b) Hierarchical Detector

Figure 4: Hierearchical detector can correct missing detections from generic detector.



(a) Generic Detector (b) Hierarchical Detector

Figure 5: Generic detector predicts two different bounding boxes for two sibling categories which is suppressed by the hierearchical detector.

Category	Root Category	Total Images	True Positive	False Positive
Polos	Top Innerwear	165	157	8
Hoodies	Top Innerwear	239	215	14
Briefcase	Bags	132	132	0

Table 3: Detection Performance on Novel Categories

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 1

[2] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3450–3457. IEEE, 2012. 2

[3] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2352–2359. IEEE, 2010. 4

[4] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. *Computer Vision–ECCV 2012*, pages 340–353, 2012. 3

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 4

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 1

[7] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990. 1, 3

[8] W. Ouyang, H. Li, X. Zeng, and X. Wang. Learning deep representation with large-scale attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1895–1903, 2015. 3

[9] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016. 1, 2, 3

[10] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2, 3, 4

[11] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012. 4