Detecting Fashion Apparels and their Landmarks

Himani Saini

A Thesis
In the Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Masters (Computer Science) at
Concordia University
Montréal, Québec, Canada

September 2019

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Himani Saini
Entitled: Detecting Fashion Apparels and their Landmarks

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Nikolaos Tsantalis

_____ Examiner
Dr. Gregory Butler

_____ Examiner
Dr. Sudhir P. Mudur

_____ Supervisor
Dr. Jia Yuan Yu

Approved by _____
Dr. Dhrubajyoti Goswami
Chair of Department or Graduate Program Director

_____
Dr. Amir Asif,
Dean, Gina Cody School of Engineering and Computer Science

Date _____
25 September, 2019

Abstract

Detecting Fashion Apparels and their Landmarks

Himani Saini

Fashion landmarks are the functional key-points on the apparels that can be used for a more discriminative visual analysis of the apparel images. Such a framework can facilitate apparel alignment in displaying apparel images on the websites or help build a system to ensure dress code in a particular environment. However, challenges such as background clutter, human poses, scales and apparel variation can render such a task difficult. We present a conceptually simple, flexible, and general framework for apparels' landmark detection that can be simultaneously used for apparel classification and localization. In addition to the position of the landmarks in the apparels, we also classify the landmarks as visible or occluded in the same framework. The fashion landmark detection task is similar to joint localization and detection problems like human pose estimation, hence our approach extends stacked hourglass architecture, originally proposed to solve human pose estimation. We perform all these tasks in parallel using multi-task learning. Our proposed convolutional neural network is end-to-end differentiable and simple to train, since all these tasks are performed on the same architecture without any additional parameters to learn. Over the past few years, many modifications have been proposed to improve this architecture. We also compare the performances of some of these different variations of stacked hourglass architectures. These architectures leverage both global and local features captured by the deep convolutional neural networks to better localize the apparel in the image as well as the landmarks in those apparels. We test and analyze our results on DeepFashion dataset. We also weigh the trade-offs of the detecting the landmarks in a category-aware environment, i.e., pre-classified apparels and category-agnostic environment, i.e., unclassified apparels.

Acknowledgements

Throughout the writing of this thesis I have received a great deal of support and assistance. I would first like to thank my supervisor, Dr. Jia Yuan, whose expertise was invaluable in the formulating of the research topic and methodology in particular. I would like to acknowledge my colleagues from my internship at Dataperformers for their wonderful collaboration. You supported me greatly and were always willing to help me. I want to thank you for your cooperation and for all of the opportunities I was given to conduct my research and further my thesis . I would also like to thank my parents for their counsel and support. Finally, there are my friends, who were a great support in discussing the problems and findings, as well as providing a happy atmosphere outside of my research and proof-reading my work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fashion is essential to our society. It is an interplay of expressing oneself as well as presenting oneself appropriate to the formal and informal occasions. In the past few years, visual analysis of fashion items have received a lot of interest from the research community because of the gamut of applications ranging from apparel classification [1, 2, 3], retrieval [3, 4, 5, 6, 7], recommendation [8, 9, 10, 11], attribute prediction [1, 12, 13, 14] to style discovery and extraction [15, 16, 17, 18], outfit generation [19], to name a few. There has also been an attempt to directly measure the abstract social concepts such as popularity [20], adequacy [21] or fashionability [7, 18, 22]. Since fashion is a very subjective domain, defining an objective metrics is a very tricky task, hence much of this work focuses on learning the image similarity [23]. Solutions to such challenging problems can be applied to build a virtual apparel system for e-commerce or can be a key component in studying the social-behavioral aspects of fashion. Hence research in this field brings a lot of value to the fashion e-commerce industry, which is estimated to be $ 712.9 billion by 2022 [24]. The advent of deep convolutional neural networks and the availability of powerful computational resources have enabled researchers to draw meaningful information from the fashion images. This research is further advanced by the availability of the large-scale fashion datasets like Fashion MNIST [25], DeepFashion [12], FashionAI [26], DeepFashion2 [27], PaperDoll [28], etc. to tackle different tasks.

Fashion landmark detection is another such task that can bring enormous value to this domain. These fashion landmarks are the key-points located at the functional regions of the apparels e.g., neckline, cuff, etc. and much like human key-points in the problem of human pose estimation [29], they can represent the configuration of an apparel by itself or on a human body. These functional regions can better distinguish design, pattern or texture and category of the apparels. They can facilitate tasks like tracking, localized

auto-editing of apparel images, or by providing better insights into fashion trends, style analysis, etc. in order to build more customized recommendation systems. One of the key applications of such landmarks is to create a consistently-aligned inventory to boost the appeal of the apparel images on the websites. Detecting landmarks can help making sure that all the apparels displayed on the website are uniform in size and their corners, contours and angles are carefully aligned with respect to each other as well as to the alignment guidelines set for the apparel photography work-flow of the website. Another key application of these landmarks can be to develop a system to ensure the dress codes as per the occasion and environment. The relationship of these detected landmarks among each other can give meaningful insights on the type of apparel a person is wearing which can be checked against the established dress-code to flag the anomalies. Hence the detecting landmarks can bring effectiveness in both social and commercial fashion domains.

However, localization of these landmarks is a challenging problem because of the diversity of apparels, background clutter, occlusion, scale variation, frequent style changes, etc. In addition to these variations, the images also differ in their environment e.g., the images from an e-commerce platform are aesthetically and technically superior [30] to the images collected from a social media platform. Moreover, these images also exhibit human body and pose variation since people across different platforms take pictures in different poses and photos are taken from different camera viewpoints and focal lengths. On one hand, local details on apparel images like collar and cuff are more important to localize these landmarks, while on the other hand, global attributes of the apparel itself are more important in order to solve the spatial relationships among those landmarks, i.e., the left and right collar or the left and right sleeve.

In this work, we solve this problem using stacked hourglass architecture [31], which is a dominant approach forming the back-bone of many leading approaches for MPII dataset [32], a benchmark for human pose estimation. The key aspect of the stacked hourglass architecture is that it extracts and consolidates the features across all scales, leveraging both local and global information present in the images to learn the spatial relationships among the different locations of the image. It does so using repeated bottom-up and top-down processing, that constitutes its hourglass shape. Furthermore, this architecture is extended by stacking multiple hourglasses next to each other, which enables cascaded inferences along with intermediate supervision. Over years, many modifications have been made to this architecture. Two such variations are pyramid residual networks [33] and cascaded pyramid fusion [34]. These architectures were built to employ specific properties of convolutional neural networks like mutlti-scale features maps, etc. in order to further improve the predictions.

Deep convolutional neural networks have significantly improved the image classification and detection accuracy owing to the powerful baselines architectures like Fast/Faster R-CNN [35, 36], Mask R-CNN [37], single shot multibox detector (SSD) [38], you only look once (YOLO) [39], etc. All these methods are conceptually intuitive and robust with fast inference and training time and work by detecting the center, height and width of the bounding box. However, all these approaches use anchor boxes of different sizes and aspect ratios as the object detection candidates. But as noted in [40], using anchor boxes is not very computationally efficient because in order to detect a tight bounding box localizing the object in the image, a number of anchor boxes having sufficient overlap with the ground truth box are detected and compared. This large set of anchor boxes have a huge imbalance between positive and negative anchor boxes, which slows down the training [40]. Also, a lot of hyper-parameters and heuristic design choices, like size, aspect ratio are associated with these anchor boxes. In our work, instead of the center, we localize the top-left and bottom-right corners of the bounding boxes for each category of the apparel, in a similar way to landmarks on the apparel, without any need of anchor boxes. This works, because whereas the center of the object depends on the four corners of the object, the two corners can be estimated using the features from two diagonally opposite spatial locations of the objects. Mask R-CNN [37] architecture uses multi-task learning in order to simultaneously perform object detection, classification, segmentation and pose estimation in one single network. Our approach is motivated by Mask R-CNN in that sense, but unlike Mask R-CNN, which introduces branches to the base network and adds more parameters specific to learning specific tasks, our approach uses the same features to perform all these tasks in parallel.

The contributions of this work are as follow:

- We present a unified framework for fashion landmark detection, apparel classification and localization in the apparel images.
- We review the modularized structure of the stacked hourglass architecture with its variations and compare these architectures to propose the most suitable one to perform the described fashion analysis.
- Since the landmarks are sparse, i.e., different categories of apparels have different number of landmarks, we also explore the trade-off for the learning on pre-classified apparels, i.e., category-aware learning as well as on unclassified apparels, i.e., category-agnostic learning of these landmarks. e.g., the type of landmarks present in a T-shirt will be different from those present in the pants.

Thesis Outline

The remainder of this work is organized in the following manner. Chapter 2 enlists the related work in the fashion domain as well as in the related domains that try to solve similar joint localization and detection problems. We review some recent developments fashion landmark estimation and in other key-point detection problems. We also discuss recent developments in object detection. In Chapter 3, we formulate the problem of joint landmark detection and classification of apparels in apparel images. In Chapter 4, we explain the modular structure of the stacked hourglass architecture and its variations in a step by step manner. In Chapter 5, we do an exploratory analysis of DeepFashion dataset. We review the quantitative and qualitative results of the proposed methodology in Chapter 6 and 7 respectively. Finally, in Chapter 8, we conclude the results and provide future scope of this work.

Chapter 2

Related Work

In this chapter, we will review the previous work in the fashion domain. Fashion analysis in images has long been a research topic among the computer vision community and over the years, it has come a long way from preliminary analysis to providing rich information to identify trends and behaviours.

2.1 Fashion and Computer Vision

Traditional attempts to create powerful representation out of fashion images [8, 14] mostly relied on handcrafted features like SIFT [41], HOG [42] or color histograms. But the performance of these methods was limited by representation power of these extracted features. In order to learn spatial information, the probabilistic methods such as graph models [43] were used. Early apparel recognition attempts were based on a region-based interpretation of the apparel image modelled in a graph structure [44]. [43] on the other hand tackled this problem using a context-sensitive grammatical representations of apparels using artists' sketches. On the other hand, [45] combined the body contours with a low dimensional apparel model to represent apparels as a deformation from the underlying body contour and learns these deformations from the images using principal component analysis. Yet another work, [14] classified the apparels as semantic attributes of a person by extracting low-level features in a pose-adaptive manner and combining complementary features to learn attribute classifiers. Owing to the availability of the large-scale datasets [12, 25, 26, 27, 28] and powerful computational resources, there has been a shift in the research community towards deep learning based models [1, 6, 12]. These models outperform the prior approaches by a huge margin demonstrating the strong

representation power of neural network. However, the algorithms performing generic tasks like object detection or classification, [9, 35, 37, 46] do not perform as well in the fashion domain due to the huge variations in the apparel images as well as the platforms from which these images are collected.

In Deep Learning, neural networks provide a way to approximate and generalize on a given dataset. In traditional Computer Vision, image representations were hand engineered by the computer vision experts and domain experts. e.g., if certain patterns are to be detected in the cancer cells, the images of cancer cells need to be evaluated by the medical experts, who can then share that knowledge with the computer vision experts, so that the specific filters can be created in order to capture those patterns automatically. This limits the system of detection to the human understanding. Whereas, in convolutional neural networks, such filters are computed over the time with the data as input and the prediction as output. The predicted output is compared with the actual output and error is back-propagated through the network to update the computed filters. As observed in many experiments, the filters in the initial layers learn the local features like edges and contours, whereas the filters in later layers learn global shapes of the objects. We will discuss more about the inner workings of the convolutional neural network in chapter 4.

2.2 Joint Localization and Landmark Detection

Previous methods have studied joint localization and detection in the context of facial landmark detection and human pose estimation. Both are complex problems with enormous value. While facial landmark detection can be helpful in building powerful facial detection systems, human pose estimation has its own advantages for tracking, motion capture etc. In this section, we will briefly discuss the work done in both these domains and how these problems are different from fashion landmark detection.

2.2.1 Facial Landmark Detection

Facial landmark detection is a similar key-point localization and detection task where, the goal is to detect some essential key-points in human faces, e.g., the tip of the nose, the corners of the eyebrows, eyes, mouth, etc. Detecting these landmarks is a prerequisite for a variety of computer vision applications [47]. Some of the methods tried to tackle this problem using simple regression techniques [48, 49, 50]. These techniques use convolutional neural networks to learn facial features and then pass these features

through regressors in an end-to-end fashion [51, 52]. Putting such system in a cascaded architecture progressively improves the prediction [53, 54]. Another category of facial landmark detection uses heat-map regression over the same features [55, 56, 57]. These facial landmark detection methods also focus on the facial shape, e.g., in [53], the extreme head pose is used in addition to the landmark locations or as in [58] wherein rich facial deformations are used to aid the training process. But our problem is different from facial landmark detection since in the faces, the deformations in facial landmarks come from the expressions and poses, while these expressions and poses are contained within a common global shape. However, in fashion apparels, the diversity in apparels makes it much more difficult to localize the landmarks.

2.2.2 Human Pose Estimation

There has been an extensive research in human pose estimation [29, 33, 59, 60], which is another similar task in terms of localizing key-points of deformable entities. The objective in human pose estimation is to represent the orientation of a person in a graphical form, by locating the co-ordinates of some specific joints, called key-points and joining the valid pairs to form the limbs and hence the skeleton. The methods proposed to solve the pose-estimation have progressed from simple part detectors and elaborate body models [61, 62], to tree-structured pictorial models [63, 64], to using deep learning via convolutional neural networks. Convolutional neural networks benefit from spatial generalization ability, hence, help locating the object of interest irrespective of it's location in different images. This property arises from spatially shared parameters of convolutional neural networks. For key-point localization, the algorithms use heat-map regression, instead of numerical-coordinate regression because direct numerical-coordinate regression over the fully connected layers is a highly non-linear problem and lacks the spatial generalization ability [65]. Hence for heat-map regression is used instead where the heat-maps encode pixel-wise spatial estimate of the location of the key-points.

The work in this context can be mostly categorized in two different approaches: A *top-down* approach, where first a person detector detects bounding boxes for the people in the image and then a pose estimator estimates the key-points present in the each bounding box; and a *bottom-up* approach, where key-point proposals are grouped together into person instances. Typically, top-down approach depends upon the strength of the person-detector used, while the bottom-up approach depends upon the strength of the associating or grouping algorithms. Top-down approaches are less efficient when people are in close proximity to each other because the bounding box created by the person detector might contain the body parts of the other people in the image, hence creating false associations.

DeepPose [66] was proposed using numerical coordinate regression, where the first stage predicts the numerical coordinates and the later stages refine the predictions by calculating offsets. [67] was the first to use heat-map regression in a bottom-up fashion, first detecting the body parts using deep convolutional neural networks and then using Markov random field to find the association among the parts. Similar approach was followed by convolutional pose machines (CPM) [56], by using heat-map regression instead of numerical coordinate regression, hence each stage operates on the heat-maps produced by the previous stage, increasingly refining the co-ordinates of the key-points locations. The success of both the approaches imply that cascade arrangements are effective to distinguish the human-pose at a global level. OpenPose [68], was another popular bottom-up approach that feeds the extracted features of an image to two parallel branches, first predicting the heat-maps and the second one predicting the part affinity-fields, which represent the degree of association between the parts. The further processing involves predicting the bipartite graphs for the key-point pairs and then pruning the weaker links part affinity-field values. This was also modelled as a cascaded network. While these are bottom-up approaches and very efficient in the case of multi-person detection, stacked hourglass [31] was proposed as a top-down approach that leverages the cascaded network for refining the network predictions by introducing intermediate supervision. This network architecture has a high expression-ability since each stage extracts and combines features across different scales hence producing rich high resolution features of the images. This approach has since been the back-bone of many leading approaches on MPII dataset [33, 34, 59, 60, 69] and is the basis of our proposed methodology.

2.2.3 Fashion Landmark Detection

Initial work on Fashion Landmark detection [3] takes bounding boxes of the apparel images at the training as well as testing time. But such bounding boxes are hard to annotate for the real-time data, hence the algorithm should be able to learn from the full input images. In addition to input images, it takes 8 labels, i.e. upper, lower or full-body apparel categories, normal, medium or large poses and medium zoom-in or large zoom-in variations to learn pseudo-labels representing the relationship of one sample image to another. These pseudo-labels are then used to learn and refine the landmark localization in a cascaded network and learn the apparel configuration.

The next work [70], takes full images as an input as well as zoom-in or zoom-out labels to inject high-level human knowledge into the network. It uses *Selective Dilated Convolution* to handle scale discrepancies and *Hierarchical-Recurrent Spatial Transformer* network, as an attention mechanism to ignore the background clutter. Selective Dilated

Convolution use dilated filters instead of the traditional convolutional neural network filters in order to increase the receptive field of the image, keeping the same number of parameters. This helped in evaluating images at different scales, using 4 different dilation values. Hierarchical-Recurrent Spatial Transformer networks on the other hand are the modification of spatial transformer networks, which are used to encode a mechanism to handle affine distortions in the image. It learns the geometric transforms on the input image to focus more on the object in the image, ignoring the background. But since fashion images provide a lot of variations in terms of poses and background clutter, a series of geometric transformations, progressively for all the landmarks in an hierarchical manner. In addition to that, so as to not learn the same transforms again, a recurrent state of the transforms is saved for each landmark. [3, 70] employ direct regression over the fully connected layers, which is a highly non-linear problem and lacks the generalization ability [65]. When the features obtained after convolutional layers are flattened, there is a huge loss in the spatial features of those feature maps. Instead, recent approaches for both human pose estimation [33, 59, 60] and fashion landmark detection [13] predict heat-maps or confidence maps of positional distribution of each landmark, given the input image.

The recent approach [13] encodes the knowledge over fashion apparels by developing a domain specific grammar to learn kinematic and symmetric relations between apparel landmarks. *Bidirectional Convolutional Recurrent Neural Network* is used for message passing over the learned grammar. It also uses two attention mechanisms: one that is landmark-aware and involving domain-knowledge, and another directly focusing on the category relevant image regions. This attention is learned in a supervised manner and encodes semantic and textual constraints. While all these approaches work with different efficiencies, they produce features specific to fashion landmarks only and cannot be used for other tasks.

During our research, we realized that fashion landmark detection [12] is similar to human pose estimation in terms of the objective, yet is more challenging because of many reasons. The human key-points in the images are subject to rigid deformations, whereas the landmarks in fashion items are subject to non-rigid deformations like stretching and shrinking. Furthermore, apparels have a much larger scale variation than human poses. A same category of apparel can have it's landmarks at different places on human body. e.g., A dress can be a full-body dress, a medium-length dress or a short-length dress, making the localization of the dresses' landmarks more difficult. Also, local regions of fashion landmarks have larger appearance variations than those of human key-points. e.g., An upper-body woman apparel can be a T-shirt, a camisole, a racer back top or a tank top. Another challenge comes from the uneven number of landmarks in different apparels. In human pose estimation, the number of key-points are fixed throughout the dataset,

(fifteen in MPII [29]). Whereas, in fashion landmarks detection, different apparels have different number of landmarks. e.g., In DeepFashion [12], an upper-body apparel has six landmarks, a lower-body apparel has four landmarks and a full-body apparel has eight landmarks. Even if both upper and lower-body apparels are present in an image, the annotations are provided only for one of these categories. This makes it difficult for the network to learn the landmarks to predict in an image. In human pose estimation, the global cues captured by the later stages of convolutional neural network provide the spatial context for the localization of the key-points with respect to each other. Hence, the key-points that are difficult to localize and differentiate, e.g., shoulders and elbows, are learned with respect to the key-points that can be localized with greater confidence e.g., head or neck. But in case of fashion landmarks, the position of the sleeve can be anywhere on the arm with respect to the collars. So, properly learning the contextual information is more important for localization of fashion landmarks.

2.3 Object Detection and Classification

Convolutional neural networks based object detectors [35, 36, 37, 38, 39] have achieved advanced results on various benchmark datasets [71, 72]. The key component of these networks is the candidate boxes around all the objects in the image. These boxes are called the anchor boxes and for each candidate in the image, they are created in different sizes and aspect ratios, so as to produce a tighter bounding box localizing the object in the image. These prior anchor boxes are then used to perform both object classification, by predicting a class score and object localization, by proposing the bounding box coordinates. Most of the object detectors are divided into two categories: two-stage detectors and one stage detectors.

Two-stage detectors are called so because in the first stage, a sparse set of regions of interest are generated (ROIs) and in the second stage, a convolutional neural network is trained to classify them as a bounding box of the object instance. A post processing step like non-maximum supression [72] is used in order to select the most accurate bounding box from the proposed ones. This technique was first introduced by R-CNN [73], which uses traditional computer vision algorithm like selective search [74] in order to generate ROIs. These ROIs act as the input for the convolutional neural network. But since the input image could be of any resolution with only one object instance, this creates a lot of redundant computations, which are directly proportional to the number of ROIs generated. This was then improved by spatial pyramid pooling (SPP) [75] and Fast R-CNN [35] by first generating the feature maps from the image using some convolutional layers

and then generating ROIs from those feature maps. But since they are using two different algorithms to propose ROIs and process them, these networks cannot be trained end-to-end. This was taken care by Faster R-CNN [36] by using a region proposal network (RPN) to generate proposals from a set of pre-determined anchor boxes. Region-based fully convolutional networks (R-FCN) [76] improved it further by replacing the fully connected layer of Faster R-CNN with fully convolutional layer, hence preserving the spatial associations between the features from the last layer. This was further improved by Mask R-CNN [37] that introduced a mutli-branch network to perform object detection and instance segmentation in an end-to-end fashion. Mask R-CNN uses an ROIAlign layer in the RPN in order to remove the quantization effect while proposing ROIs. Later on, many modifications were proposed to improve the accuracy e.g., by generating anchor boxes at multiple scales preserving both local and contextual information [77], or by introducing cascades in order to refine the predictions [78].

One stage object detectors remove the ROI proposal step by densely placing the anchor boxes over an image and generating the final bounding box by scoring those anchor boxes and refining their coordinates through coordinate regression. This makes them computationally efficient and also allows the network to be trained end-to-end while maintaining competitive performance on the challenging benchmarks. Single shot detector (SSD) [38] densely places anchor boxes of different sizes and aspect ratios over the feature maps that are generated at multiple scales. This is because the deeper layers of the convolutional neural network have smaller receptive fields that can be used to detect small objects whereas, the shallower layers have larger receptive fields. These anchor boxes are then directly classified and refined. You only look once (YOLO) [39], another one-stage detectors, predicts bounding box coordinates directly from an image using coordinate regression, but it was later improved in YOLO9000 [79] by using anchor boxes. Deconvolutional single shot detector (DSSD) [80] modifies SSD by changing the network architecture to that of the hourglass network [31], combining local and global spatial features via skip connections. These one-stage object detectors were faster than two-stage detectors but were not as efficient, until RetinaNet [81]. RetinaNet uses focal loss, to dynamically adjust the weights of each anchor box.

However, there are drawbacks of using anchor boxes. Firstly, in order to train the detector to classify whether each anchor box sufficiently overlaps with a ground truth box, and a large number of anchor boxes is needed to ensure sufficient overlap with the ground truth boxes e.g., 20,000 in R-CNN, 40,000 in DSSD and 100,000 in RetinaNet. Hence only a small fraction of these anchor boxes overlap with the ground-truth boxes, creating a huge imbalance between positive and negative anchor boxes during training, causing the training to be inefficient and slow. Secondly, the use of anchor boxes introduces many

ad-hoc heuristics hyper-parameters and design choices in terms of the number of anchor boxes, their sizes and aspect ratios.

Recently many networks have been proposed for object detection *without using anchor boxes* e.g., DeNet [82] with directed sparse sampling, point linking network (PLN) [83] and CornerNet [40]. DeNet is a two-stage detector which generates ROIs by selecting features at manually determined locations relative to a region for classification and then determining how likely each location belongs to either the top-left, top-right, bottom-left or bottom-right corner of a bounding box, without identifying if they belong to the same object. It then uses another network to classify ROIs as the bounding box while rejecting poor ROIs. Whereas, PLN is a one-stage detector, which first predicts the locations of the four corners and the center of a bounding box. Then, similar to DeNet, for each corner location it predicts how likely each pixel location in the image is the center and for the center location, it predicts how likely each pixel location is the the top-left, top-right, bottom-left or bottom-right corner. It then combines the predictions from each corner and center pair to generate a bounding box. CornerNet is another one stage approach without anchor boxes that inspired the use of features from stacked hourglass architecture in our work. It uses a single convolutional neural network similar to an hourglass architecture to produce heat-maps of the top-left corner of all the instances of the same object, heat-maps of the bottom-right corner of all the instances of the same object and an embedding vector for each corner detection. The network is trained to produce similar embedding vectors for the corners belonging to the same object instance, which are then used to then pair the top-left and bottom-right for the a particular object instance in the image. This approach was inspired by [84], wherein similar embedding are used in a bottom-up human pose estimation to associate different key-points belonging to the same person together. Since a corner cannot be localized solely on the basis of the local evidence, it also uses corner pooling to explicitly encode some prior knowledge of corners' spatial definition and refine its predicted locations.

Since our network already produces rich feature maps combining the local and global features to compute the heat-maps for the fashion landmarks, we can leverage those feature maps to produce heat-maps for top-left and bottom-right corners, similar to CornerNet. But we do not use corner embedding and corner pooling. This is because DeepFashion dataset has only one object in an image so the top-left and bottom-right corners are predicted in pairs to begin with. And since these heat-maps will be predicted in parallel to the heat-maps for the fashion-landmarks, much richer prior information is explicitly encoded to spatially localize the corners and produce refined bounding boxes. As noted in [85], sharing representations between the related tasks enables the model to generalize better on the original task, since we are encoding more information from the training

signals of the related tasks.

In this chapter, we discussed the related work in other joint localization and detection tasks, such as facial landmark detection and human pose estimation as well in fashion landmark detection itself. We also discussed various object detection techniques extensively used over the past few year along with their merits and demerits. In the next chapter we will formulate our problem of fashion analysis to build a single framework for joint fashion landmark detection and apparel detection.

Chapter 3

Problem Formulation

Let X be the set of images, where i^{th} image $x^i \in X$ contains one fashion item to be annotated with the fashion attributes. Let Z be the set of all the pixel locations (u,v) in the coordinate space $\mathbb{R}^{m\times n\times C}$ of x^i, where $(m \times n)$ and C are the resolution and number of channels of x^i respectively and $u \in \{1,\ldots,m\}$ and $v \in \{1,\ldots,n\}$. Let Y be the label set representing the corresponding fashion attributes $y^i \in Y$. y^i consists of a vector of all fashion attributes, namely, a class label $\beta \in \{1,\ldots,\tau\}$ describing the type of the apparel using τ class labels; two opposite corner locations $\alpha_1, \alpha_2 \in Z$ (top-left and the bottom-right) localizing the apparel item in the image; and an ordered set of P fashion landmarks $(a^i_1,\ldots,a^i_P)$, such that, $y^i = (\beta^i, \alpha^i_1, \alpha^i_2, a^i_1, \ldots, a^i_P)$. In DeepFashion [12], $\beta \in \{1,2,3\}$ where the values 1, 2 and 3 represent the upper-body apparels (T-shirt, top, etc.), lower-body apparels (pants, shorts, etc.) and full-body apparels (dresses, jumpsuits, etc.) respectively. It has $P = 8$ landmarks, which are ordered as left collar, right collar, left sleeve, right sleeve, left waist, right waist, left hem, right hem. Each fashion landmark a^i_p (p^{th} fashion landmark for x^i) is a three dimensional vector $a^i_p = (u,v,\gamma)$, where $(u,v) \in Z$ indicates the location of the landmark and $\gamma \in \{0,1\}$ represents the visibility flag such that the values 1 and 0 indicate the visible and the occluded landmarks respectively. Figure 1 represent the landmarks on three different categories of the apparels in the DeepFashion.

Let $S_N = \{(x^1,y^1),\ldots,(x^N,y^N)\}$ be the training set containing N apparel images, where $(x^i,y^i) \in (X,Y)$. The objective is to learn a function $\varphi_{S_N} : X \to Y$, that predicts the fashion attributes $\hat{y^i}$ for given x^i. This function φ_{S_N} consists of a convolutional neural network ψ_{S_N} explained in Chapter 4, followed by a post processing function specific to different components of $\hat{y}^i = (\hat{\beta}^i, \hat{\alpha}^i_1, \hat{\alpha}^i_2, \hat{a}^i_1, \ldots, \hat{a}^i_P)$ as explained in the sections below.

Figure 1: Images for upper-body apparel, lower-body apparel and full-body apparel with annotated attributes. Blue dots represent the visible landmarks and the green dots represent the occluded landmarks on the apparels.

3.1 Landmark's Localization

For x^i, a sequence of ground-truth heat-maps is generated by creating P zero matrices $\{h_1^i,\ldots,h_P^i\}$ for P landmarks $(a_1^i,\ldots,a_P^i)$. Here $h_p^i \in \mathbb{R}^{m'\times n'}$, with $(m'\times n')$ as the selected output resolution. We represent this generated ground truth for the image x^i as $\mathbb{H}^i = \{h_1^i,\ldots,h_P^i\}$. For each landmark $a_p^i \in y^i$, its location (u,v) is mapped to the corresponding location (u',v') in h_p^i. Then, with (u',v') at the center, a discreet 2D gaussian [86] with standard deviation σ is calculated for (u',v') and its neighbouring locations up to width κ. Here $\kappa \in \mathbb{N}$. This 2D gaussian is given as a function $G : Z \rightarrow \mathbb{R}^{m'\times n'}$ such that,

$$
\begin{aligned}
G(u',v'|\sigma,\kappa) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{(q-\frac{\kappa}{2})^2+(r-\frac{\kappa}{2})^2}{2\sigma^2}}, \\
\forall q \in \{u'-\frac{\kappa}{2}, u'+\frac{\kappa}{2}\}, \\
r \in \{v'-\frac{\kappa}{2}, v'+\frac{\kappa}{2}\}.
\end{aligned}
\tag{3.1}
$$

This produces a surface whose contours are concentric circles with a gaussian distribution from the center point as shown in Figure 2 and Figure 3. In Figure 2, the top row corresponding to each apparel represent the gaussian heat-map for each of the landmarks. The circular back dot represents the approximate position of the landmark in its corresponding heat-map. These black dots, when zoomed in, look like the concentric circles in Figure 3. The pixel at location (u',v') receives the highest gaussian value and neighboring pixels receive smaller gaussian values as their distance from (u',v') increases, creating a soft

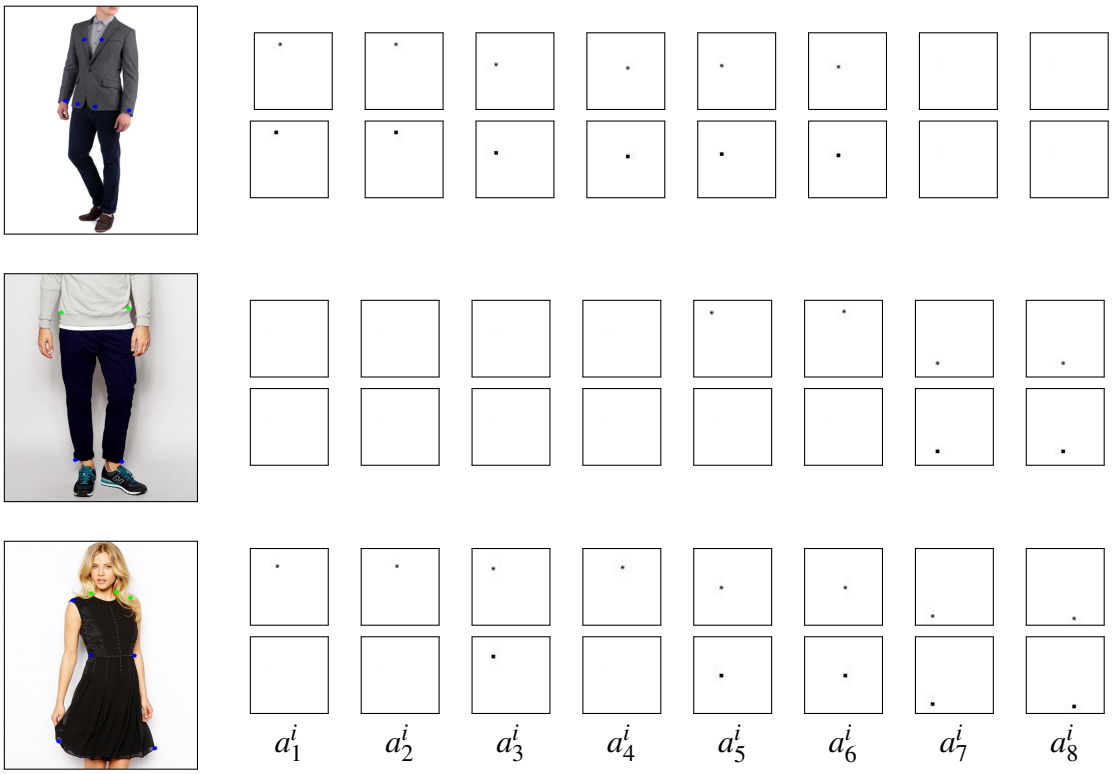


Figure 2: Upper, lower and the full-body apparels with landmarks. In the original images, the image shape is rectangular, owing to the standing poses, but these images are mapped to the resolution $m \times n$, where $m = n$. Then, using those images, ground-truth maps of resolution $m' \times n'$ are created. First row corresponding to each apparel represent the location heat-maps of $P = 8$ landmarks, ordered (from left) as left collar, right collar, left sleeve, right sleeve, left waist, right waist, left hem, right hem; and the second row represents the visibility-maps for those landmarks.

peak centered at (u', v'), decreasing symmetrically as the distance from (u', v') increases. It is most common to use $\sigma = 1/\kappa$ and $\kappa \in \{3, 5, 7\}$ [31, 33, 87].

Figure 3: Zoomed-in 2D gaussian on a heat-map.

A sequence of output heat-maps $\hat{\mathbb{H}}^i = \{\hat{h}_1^i, \ldots, \hat{h}_P^i\}$ is computed for x^i using ψ_{S_N}. These heat-maps are refined using the mean squared loss [88] between the set of output heat-maps $\hat{\mathbb{H}}$ and ground-truth heat-maps $\mathbb{H}$ for all the images in S_N, given as $\mathscr{L}_P^N$:

$\mathbb{R}^{m' \times n' \times P} \rightarrow \mathbb{R}$:

$$\mathscr{L}_P^N(\mathbb{H}, \hat{\mathbb{H}}) = \frac{1}{N}\frac{1}{P}\sum_{i=1}^{N}\sum_{p=1}^{P}\left\|h_p^i - \hat{h}_p^i\right\|^2, \tag{3.2}$$

where $\left\|h_p^i - \hat{h}_p^i\right\|^2$ is the l_2 loss [88] between the ground and the predicted p^{th} heat-map for x^i. This process is called heat-map regression. The estimated landmark locations are determined by computing the location of the first brightest pixel, i.e. the pixel with the maximum value in the corresponding heat-map. We represent the value of the matrix J at a given location (u,v) as $[J]_{u,v}$ For any given matrix J, if a function $\mathscr{M} : Z \rightarrow R$ defines the value of the matrix at the given the pixel location, such that,

$$\mathscr{M}\Big((u,v), J\Big) = [J]_{u,v}, \quad (u,v) \in Z, J \in \mathbb{R}^{m' \times n'}, \tag{3.3}$$

The pixel with the maximum value in the corresponding estimated heat-map is given as:

$$(\hat{u}_p^i, \hat{v}_p^i) = \underset{(u,v)\in Z}{\operatorname{argmax}}\ \mathscr{M}\Big((u,v), \hat{h}_p^i\Big). \tag{3.4}$$

This $(\hat{u}_p^i, \hat{v}_p^i)$ location obtained from $\hat{h}_p^i$ is the estimated location of the p^{th} landmark in $(\hat{a}_1^i, \ldots, \hat{a}_P^i)$ for x^i.

3.2 Landmark's Visibility

For x^i, a sequence of ground-truth visibility-maps $\mathbb{G}^i = \{g_1^i, \ldots, g_P^i\}$ is generated for P landmarks $(a_1^i, \ldots, a_P^i)$, where $g_p^i \in \mathbb{R}^{m' \times n'}$ is a zero matrix. These visibility maps encode each landmark's visibility information as a set of P binary maps. In each g_p^i, if the visibility flag γ in a_p^i is 1, the values within a certain width κ around the mapped ground truth location (u', v') are set to 1 and the values for the remaining background are set to 0, as shown in Figure 2 and Figure 4. In Figure 2, the bottom row corresponding to each apparel represent the visibility-map for each of the landmarks. The square back dot represents visibility state of the approximate position of the landmark in its corresponding heat-map. These black dots, when zoomed in, look like the square in Figure 2. This is to ensure that each of the approximate position in the corresponding heat-map of the landmark has the visibility state. Hence for the occluded landmarks g_p^i remains a zero-matrix.

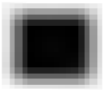


Figure 4: Zoomed-in part of a visibility-map.

A sequence of output visibility-maps $\hat{\mathbb{G}}^i = \{\hat{g}^i_1, \ldots, \hat{g}^i_P\}$ are computed for x^i using ψ_{S_N}. These computed visibility maps are refined using the pixel-wise binary cross entropy loss (BCE) [89] between the predicted and output visibility-map as $\Lambda : \mathbb{R}^{m' \times n'} \rightarrow \mathbb{R}$, such that,

$$\Lambda(g^i_p, \hat{g}^i_p) = -\sum_{q=1}^{m'} \sum_{r=1}^{n'} (\lambda^i_{p_{q,r}} \log(1 - \hat{\lambda}^i_{p_{q,r}})). \tag{3.5}$$

where $\lambda^i_{p_{q,r}}$ and $\hat{\lambda}^i_{p_{q,r}}$ respectively represent the probability of landmark's presence at the location (q,r) of the ground-truth visibility map g^i_p and predicted visibility map $\hat{g}^i_p$. Hence BCE loss over S_N is given as $\mathscr{V}^N_P : \mathbb{R}^{m' \times n' \times P} \rightarrow R$, such that,

$$\mathscr{V}^N_P(\mathbb{G}, \hat{\mathbb{G}}) = -\frac{1}{N}\frac{1}{P}\sum_{i=1}^{N}\sum_{p=1}^{P} \Lambda(g^i_p, \hat{g}^i_p), \tag{3.6}$$

The final predictions $\gamma = \{0,1\}$ of the visibility flag of the landmarks is determined by computing the first maximum pixel's value and then deciding the visibility flag as:

$$\hat{\gamma}^i_p = \begin{cases} 0, & \text{if } \max(\hat{g}^i_p) < 0.5 \\ 1, & \text{otherwise} \end{cases}, \quad p \in (1, \ldots, P). \tag{3.7}$$

This $\hat{\gamma}^i_p$ value obtained from $\hat{g}^i_p$ is the estimated visibility of the p^{th} landmark in $(\hat{a}^i_1, \ldots, \hat{a}^i_P)$ for x^i. Hence the output landmarks are $(\hat{a}^i_1, \ldots \hat{a}^i_P)$, where $\hat{a}^i_p = (\hat{u}^i_p, \hat{v}^i_p, \hat{\gamma}^i_p)$.

3.3 Bounding Box Localization

The locations of the top-left and the bottom right corners α^i_1, α^i_2 localizing the apparel in the image x^i are estimated in a similar fashion as that of the landmarks, i.e., via heat-map regression. The heat-maps are created using equation (3.1) as shown in Figure 5. For x^i, a sequence of ground-truth heat-maps $\mathbb{O}^i = \{o^i_1, o^i_2\}$ is generated for α^i_1, α^i_2 and the output heat-maps $\hat{\mathbb{O}}^i = \{\hat{o}^i_1, \hat{o}^i_2\}$ are computed using ψ_{S_N}. These output heat-maps are

also refined using l_2 loss as $\mathscr{B}_P^N : \mathbb{R}^{m' \times n' \times P} \rightarrow R$, such that,

$$\mathscr{B}_2^N(\mathbb{O}, \hat{\mathbb{O}}) = \frac{1}{N}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{2}\left\| o_j^i - \hat{o}_j^i \right\|^2, \tag{3.8}$$

Similar to landmark locations, the output locations of the corners $\hat{\alpha_1^i}, \hat{\alpha_2^i}$ are determined by computing the location of the first brightest pixel. Hence,

$$\hat{\alpha}_j^i = \underset{(u,v)\in Z}{\operatorname{argmax}}\ \mathscr{M}\Big((u,v), \hat{o}_p^i\Big). \tag{3.9}$$

3.4 Apparel Classification

Corresponding to the each of the bounding box locations α_1^i, α_2^i of image x^i, a sequence of binary class-map $\mathbb{T}^i = \{t_{11}^i, t_{12}^i, t_{21}^i, t_{22}^i, \ldots, t_{\tau 1}^i, t_{\tau 2}^i\}$ of each class is created indicating the class that corner location belongs to. These class-maps are created in the similar manner as of the visibility maps, i.e., in each class-map, values within a certain radius κ around the mapped ground truth location are set to 1 and the values for the remaining background are set to 0. Hence the class-maps to which the corner points do not belong are the zero-matrices. The class-maps for different apparels in DeepFashion dataset are shown in Figure 5. These visibility-maps are created so to ensure that each of the approximate corner location in the corresponding heat-map of the corner belongs to the same class. The sequence of output class-maps $\hat{\mathbb{T}}^i = \{\hat{t}_{11}^i, \hat{t}_{12}^i, \hat{t}_{21}^i, \hat{t}_{22}^i, \ldots, \hat{t}_{\tau 1}^i, \hat{t}_{\tau 2}^i\}$ are computed using ψ_{S_N} and are refined using BCE loss described in equation (3.3) as $\mathscr{C}_P^N : \mathbb{R}^{m' \times n' \times P} \rightarrow R$, such that,

$$\mathscr{C}_\tau^N(\mathbb{T}, \hat{\mathbb{T}}) = -\frac{1}{N}\frac{1}{\tau \times 2}\sum_{i=1}^{N}\sum_{j=1}^{\tau}\sum_{j'=1}^{2}\Lambda(\, t_{jj'}^i \,,\, \hat{t}_{jj'}^i \,), \tag{3.10}$$

The prediction of the class of each corner α_1^i and α_2^i localizing the apparel in the image is determined by getting the maximum pixel's value and then assigning the binary classification flag $\zeta_{jj'}^i$ as follow :

$$\zeta_{jj'}^i = \begin{cases} 0, & \text{if } \max(\hat{t}_{jj}^i) < 0.5 \\ 1, & \text{otherwise} \end{cases}, \quad j = 1 \ldots \tau \text{ and } j' = 1, 2. \tag{3.11}$$

Here, $\zeta_{jj'}^i = 1$ means that the corner j' belongs to class j . The final class label $\hat{\beta}^i$ is assigned to the image in the apparel, only if both the corners belong to that particular class.

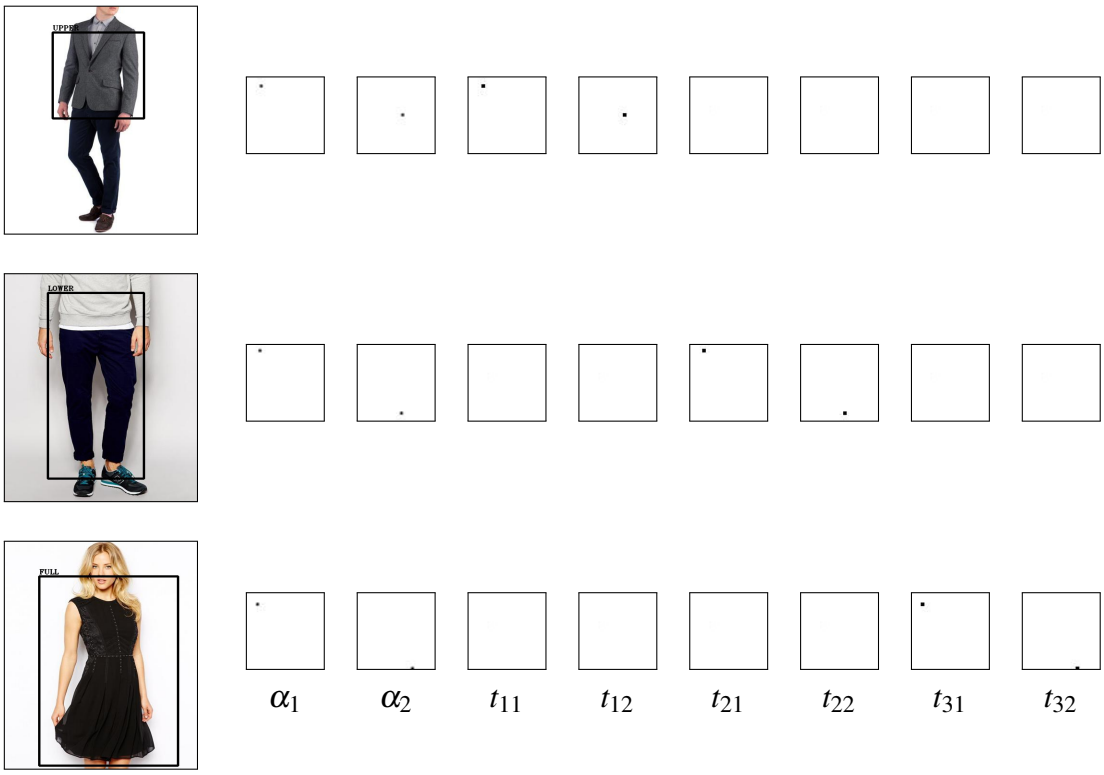


Figure 5: Upper, lower and the full-body apparels with bounding box heat-maps and class-maps. First two columns corresponding to each apparel represent heat-maps for top-left and bottom-right bounding box corners' location and the next four columns represents the class-maps for the apparels. Class maps t_{11}, t_{12} represent the presence or absence of upper-body apparel, t_{21}, t_{22} the lower-body apparel and t_{31}, t_{32} the full-body apparel.

3.5 Final Objective Function

The total loss is given by $L^N : \mathbb{R} \rightarrow \mathbb{R}$, as:

$$L^N = \nu_1 \mathscr{L}_P^N(\mathbb{H}, \hat{\mathbb{H}}) + \nu_2 \mathscr{V}_P^N(\mathbb{G}, \hat{\mathbb{G}}) + \nu_3 \mathscr{B}_2^N(\mathbb{O}, \hat{\mathbb{O}}) + \nu_4 \mathscr{C}_\tau^N(\mathbb{T}, \hat{\mathbb{T}}), \tag{3.12}$$

where ν_1, ν_2, ν_3 and ν_4 are the weights to adjust the impact of a particular loss while learning.

In this chapter, we formulated our problem of fashion analysis to build a single framework for joint fashion landmark detection and apparel detection. We defined different loss functions in order to tackle each of the tasks as well as the final objective, combining all the losses. In the next chapter, we will propose the convolutional neural network to solve this final objective.

Chapter 4

Methodology

We compute the set of heat-maps and visibility-maps using stacked hourglass network, which is a deep convolutional neural network architecture designed to solve the joint localization and detection problems. We define this network architecture as a function $\psi_{S_N} : X \to \mathbb{R}^{m' \times n' \times P}$. In this section, we first review the basic components of ψ_{S_N} as a simple convolutional neural network and then modify ψ_{S_N} as stacked hourglass network as well its adopted variations.

4.1 Building Block : Convolutional Neural Network

Convolutional neural networks [90] form the building block of the stacked hourglass architecture. These are special form of neural networks designed specifically for images. Convolutional Neural Networks consists of *convolutional layers* and *pooling layers*.

A typical deep convolutional neural network consists of the set of weights to be learned Θ and an activation function $\theta : \mathbb{R} \to \mathbb{R}$. In convolutional neural networks, these parameters Θ are characterized by a set of weight kernels or matrices W and biases vector B. This Θ is initialized using different initialization schemes [91]. The network is organized in L layers, hence the W and B can be decomposed into a union of non-empty disjoint subsets. Hence $\Theta = \cup_{l=0}^{L} \{W^{(l)}, B^{(l)}\}$. Fixing this triplet (L, Θ, θ) defines the architecture of the convolutional neural network. The first layer, i.e. $l = 0$ takes the image x^i in $(x^i, y^i) \in S_N$ as the input, and given $D^{(0)}$ number of weight matrices $W^{(0)} = \{w_1^{(0)}, \ldots, w_{D^{(0)}}^{(0)}\}$ and bias vectors $B^{(0)} = (b_1^{(0)}, \ldots, b_{D^{(0)}}^{(0)})$, it computes $D^{(0)}$ feature maps $\mathbb{F}^{(0)} = \{f_1^{(0)}, \ldots, f_{D^{(0)}}^{(0)}\}$ using a convolution and activation function.

Given any weight kernel $w_d^{(0)} \in W^{(0)}$, such that $w_d^{(0)} \in \mathbb{R}^{k\times k\times C}$, a convolution function defined as $z : \mathbb{R}^{k\times k} \to \mathbb{R}$ is applied to each element of the input image, such that at each location, the product between each element of the weight matrix and the input element it overlaps is computed and the results are summed up to obtain the output in the current location. Applying this function to the input image produces an intermediate feature map, represented as $f_d^{'(0)}$. Hence, the convolution function, determining the value of each location in a feature map $f_d^{'(0)} \in \mathbb{F}^{(0)}$ after receiving input image x^i is given by:

$$[f_d^{'(0)}]_{u,v} = z(x^i, w_d^{(0)}) = \sum_{q=0}^{k-1}\sum_{r=0}^{k-1}\sum_{c=1}^{C}[x^i]_{u+q,v+r,c} \cdot [w_d^{(0)}]_{q,r,c} + b_d^{(0)}, \quad (u,v) \in \mathrm{Z}. \tag{4.1}$$

The element wise addition is performed across different channels. For the purposes of simplicity we take the input image as gray-scale i.e single channel $C = 1$. Hence, equation (4.1) will be transformed to:

$$[f_d^{'(0)}]_{u,v} = z(x^i, w_d^{(0)}) = \sum_{q=0}^{k-1}\sum_{r=0}^{k-1}[x^i]_{u+q,v+r} \cdot [w_d^{(0)}]_{q,r} + b_d^{(0)}, \quad (u,v) \in \mathrm{Z}. \tag{4.2}$$

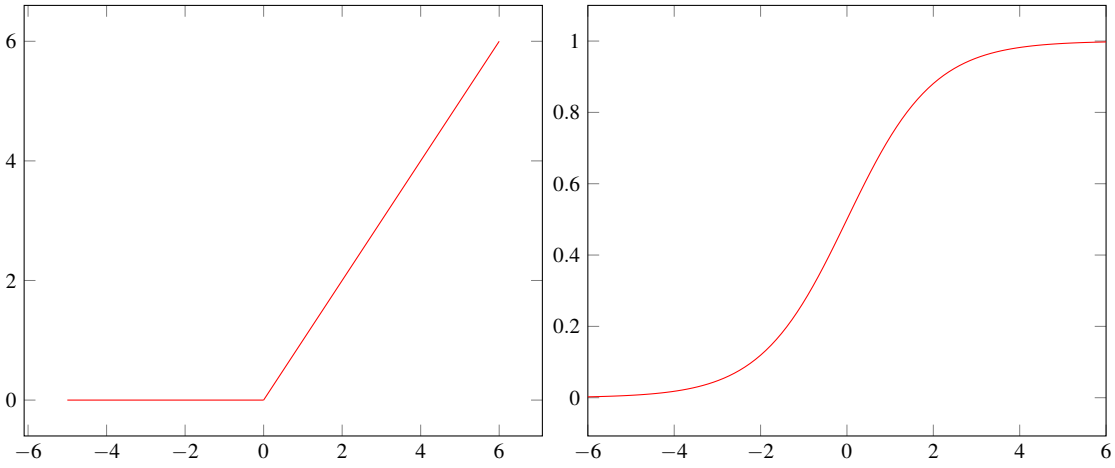
An activation function $\theta : \mathbb{R} \to \mathbb{R}$ like sigmoid, hyperbolic tangent (tanh), rectified linear activation unit (ReLU) [91, 92] is applied to $f_d^{'(0)}$ to provide the final output feature map $f_d^{(0)}$ as :

$$f_d^{(0)} = \theta[f_d^{'(0)}]_{u,v}, \quad (u,v) \in \mathrm{Z}. \tag{4.3}$$

The choice of activation function depends upon the problem at hand. In our problem, we use $\theta = ReLu$ for all the hidden layers. For the purpose of learning, the activation function should be closely linear, so as not to saturate, yet non-linear so as to capture the complexities of the dataset. Figure 6 (a) shows the non-linearity curve for the ReLU activation function. With it's values ranging from $[0, \infty)$, ReLU is linear for all the positive value and non-linear for all the negative values. This helps ReLU provide more sensitivity to the its inputs and avoids saturation, unlike other activation functions that limit the activations to a small range, e.g., sigmoid function (Figure 6 (b)) limiting the range to $(0, 1)$.

Here $f_d^{(0)} \in \mathbb{R}^{\bar{m}\times\bar{n}}$, where $(\bar{m}, \bar{n}, D^{(0)})$ are determined by the input shape of $\mathbb{F}^{(0)}$. For the first layers, since the input is the image x^i, $(\bar{m}, \bar{n})$ depends upon (m, n), kernel size (k), zero-padding (zp) and kernel-stride (st) as:

$$(\bar{m} = \bar{n}) = \frac{m + 2 * zp - k}{st} + 1. \tag{4.4}$$



(a) ReLu activation function (b) Sigmoid activation function

Figure 6: Activation functions.

These feature maps $\mathbb{F}^{(0)}$ serve as the input for the subsequent layer. We represent the feature maps of any layer l layers as $\mathbb{F}^{(l)}$. We represent the joint convolution and activation function mapping the feature maps of one layer to its subsequent layer as $\mathscr{F}_l : \mathbb{F}^{(l-1)} \rightarrow \mathbb{F}^{(l)}$ the network builds in a feed-foreword fashion as:

$$\mathbb{F}^l = \mathscr{F}_l(\ \mathbb{F}^{(l-1)}\ \mid\ W^{(l)}, B^{(l)}). \tag{4.5}$$

Such multiple non-linear layers can asymptotically approximate any complicated function. The network also consist of pooling layers that use pool kernels $w' \in W$ to bring down the resolution of the feature maps by a factor of k, which is the size of w'. This helps encoding a degree of invariance in the convolutional output with respect to translation and elastic distortions in the feature maps. The pooling operation $\mathscr{P}_l : \mathbb{F}^{(l-1)} \rightarrow \mathbb{F}^{(l)}$, outputs the maximum value of the input at a given position, that falls within the kernel. Hence each pixel in the resulting feature maps $f_d^{(l)} \in \mathbb{F}^{(l)}$ when the input is the image x^i can be given as :

$$[f_d^{(l)}]_{u,v} = \max([f_d^{(l-1)}]_{u+q,v+r}),\ \ \forall q,r \in \{1,\ldots,k\},\quad (u,v) \in \mathrm{Z}. \tag{4.6}$$

The network consists of combination of these convolution and pooing layers. The feature maps in $\mathbb{F}^{(l)}$ are used as the input for the following convolution or pooling layers and the process is repeated till the input image x^i has been down-sampled to a very low resolution representing its key features. These key features are then used to compute the output $\mathbb{F}^{(O)}$, which can be a classification score, regression score, segmentation map, or like in our case, set of visibility maps, heat maps and class maps. We represent $\odot$ as a composite

function taking input from the sub-function. Hence,

$$\mathbb{F}^{(O)} = \mathscr{F}_L \odot \mathscr{F}_{L-1} \odot \mathscr{P}_{L-2} \odot \ldots \odot \mathscr{F}_3 \odot \mathscr{P}_2 \odot \mathscr{F}_1 \odot x^i. \tag{4.7}$$

is an example of a convolutional neural network architecture. The initial layers of such networks capture the local information of the input image x^i and the later layers capture the global information of x^i encoding the semantic relationships of different components of the image.

4.2 Stacked Hourglass Network and its variations

A stacked hourglass architecture consists of a set of Δ hourglass modules $\{\phi_1, \ldots, \phi_\Delta\}$ stacked together, where each module $\phi_\delta : \phi_{\delta-1} \rightarrow \{\hat{\mathbb{H}}^i_\delta, \hat{\mathbb{G}}^i_\delta, \hat{\mathbb{O}}^i_\delta, \hat{\mathbb{C}}^i_\delta\}$ is a function that captures and consolidates both global and local features in a single feed-forward network built using convolutional neural network as explained in Section 4.1 and produces a set of heat-maps, visibility maps and class maps. The hourglass shape comes from the bottom-up processing by convolutional and pooling layers and the top-down processing by up-sampling the down-pooled feature layers. The local and the global information captured by the layers is consolidated using skip layer connections, hence combining information at each scale. This is essential because a person's orientation in the image, the arrangement of different landmarks in the apparels and landmarks' relationship with each other as well as to the overall location of the apparel in the image encode the semantic information necessary for the different components of our analysis. These hourglass modules further comprise of a set of M residual units, i.e., $\phi_\delta = \{\omega_1, \ldots, \omega_M\}$ formed using convolutional neural networks, where for each residual unit in bottom-up processing, there is a corresponding unit in the top down processing, giving it an hourglass shape. We can perform intermediate supervision by stacking such hourglasses together end-to-end. The stacked structure refines the final predictions by allowing repeated bottom-up and top-down processing across different scales. The stacked hourglass architecture constitutes of the following functional blocks.

Convolutional neural networks explained (CNN) in Section 4.1 form the building block of the residual units $\{\omega_1, \ldots, \omega_M\}$ [2, 46]. A deep convolutional neural network built using equation (4.5) suffers from a degradation problem, i.e., the accuracy decreases with the depth and then saturates after a point. Residual units [2, 46] solve this problem because instead of asymptotically approximating a complicated function, it is easier to approximate the same residual function. Hence the added layers are constructed as identity

mappings. Instead of (4.5), we use the following to build our network in a feed-forward fashion:

$$\mathbb{F}^{(l)} = \mathscr{F}_l(\ \mathbb{F}^{(l-1)}\ \mid\ W^{(l)}, B^{(l)}) + \mathbb{F}^{(l-1)}, \tag{4.8}$$

We build each ω_μ using equation (4.8), but we add three more convolutional layers before adding the identity mappings back to the output feature maps. Hence, $\omega_\mu : \mathbb{F}^{(l-1)} \rightarrow \mathbb{F}^{(l)}$ consists of three convoutional layers as:

$$\omega_\mu = \mathbb{F}^{(0)} + \mathscr{F}_3 \odot \mathscr{F}_2 \odot \mathscr{F}_1(\ \mathbb{F}^{(0)}\ \mid\ W^{(0)}, B^{(0)}). \tag{4.9}$$

Here, $\mathscr{F}_3$ is implemented using 1×1 convolutions [46] to map the depth of the output of layer $\mathscr{F}_2$ to the depth of input $\mathbb{F}^{(0)}$ in order to perform element-wise addition between the processed feature maps and the input feature maps. This is also called a bottle-neck layer and the element-wise addition is called a skip-layer connection. For each residual unit, we use $D = 256$ for both input and the output layers. Figure 7 depicts a the residual unit.

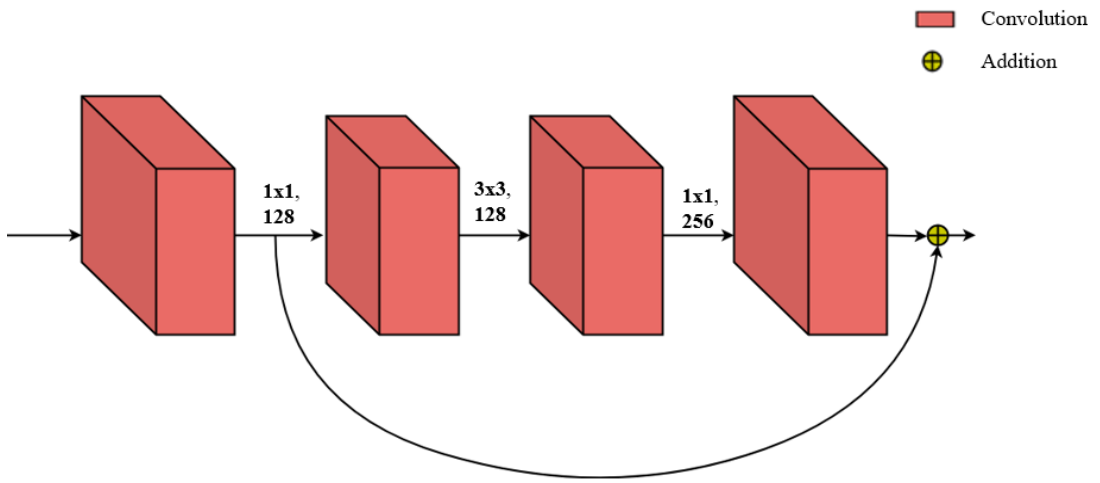


Figure 7: Residual-unit used throughout the stacked hourglass architecture. The first layer has 256 feature maps which are mapped down to 128 using a bottle-neck layer. After convolution using a 3×3 filter, these features maps are mapped back to 256 and added to the input feature maps to produce the final feature maps of this unit.

First, each input image x^i is processed through two residual units (ω_1^o, ω_2^o) to create feature maps of the selected output resolution (m', n'), i.e the same resolution as the ground-truth and predicted maps. These feature maps will serve as input to the first layer of the hourglass module. We represent this initial convolutions as $\mathscr{I}$.

$$\mathscr{I}(x^i) = \omega_2^o \odot \omega_1^o \odot x^i, \tag{4.10}$$

An hourglass module ϕ_δ consists of bottom-up and top-down processing using residual units. During bottom-up processing, residual units equation (4.9) and max pooling layers equation (4.6) are used to process features down to a very low resolution. At each max pooling step, the network branches off and applies one more residual units at the original pre-pooled resolution. The network reaches the lowest resolution at 4×4, where weight kernels are applied to learn the spatial features across the entire space of the image. At the lowest resolution, the network does the top-down processing using nearest neighbor up-sampling [93]. This up-sampling is followed by element-wise addition of two sets of features across the respective scales. The first hourglass produces the set of intermediate maps as below:

$$\{\hat{\mathbb{H}}_1^i, \hat{\mathbb{O}}_1^i, \hat{\mathbb{G}}_1^i, \hat{\mathbb{C}}_1^i\} = \phi_1(\mathscr{I}) = \mathscr{F}_0 \odot \Bigg(\omega_M \odot \omega_1\Bigg(+ \cdots + \Big(\omega_{\frac{M}{2}+2} \odot \omega_{\frac{M}{2}-2} + \big(\omega_{\frac{M}{2}+1} \odot \omega_{\frac{M}{2}-1} + (\omega_{\frac{M}{2}} \ldots \odot \mathscr{P}_2 \odot \omega_2 \odot \mathscr{P}_1 \odot \omega_1(\mathscr{I}))\big)\Big) \ldots \Bigg)\Bigg), \tag{4.11}$$

Out of these maps, heat-maps $\{\hat{\mathbb{H}}_1^i, \hat{\mathbb{O}}_1^i\}$ are used as it is, while the visibility-maps $\hat{\mathbb{G}}_1^i$ and class-maps $\hat{\mathbb{C}}_1^i$ are passed through a sigmoid activation in order to restrict their values between 0 and 1. We do so, because these maps essentially contain the probability scores. The sigmoid activation is shown in Figure 6 (b). Figure 8 represents the stacked hourglass architecture in a pictorial form.

The output of the hourglass module is passed through 1×1 convolutions in $\mathscr{F}_0$. For $\mathscr{F}_0$, D is set as the total number of output maps to be predicted, which in our case is P landmarks, P visibility flags of the landmarks, 2 bounding box corners and τ class-maps for each of the bounding box corners. Hence $D = (2P + 2 + \tau * 2)$. These final feature maps for hourglass ϕ_1 are the set of ordered maps for the image x^i as $(\hat{\mathbb{H}}_1^i)$ heat-maps for the landmarks' location, $(\hat{\mathbb{G}}_1^i)$ for the landmarks' visibility, $(\hat{\mathbb{O}}_1^i)$ for the bounding box corners and $(\hat{\mathbb{C}}_1^i)$ for the class of the bounding box corners.

Each hourglass after the first one takes the intermediate maps produced by the previous hourglass as input. The intermediate maps $\{\hat{\mathbb{H}}_\delta^i, \hat{\mathbb{G}}_\delta^i, \hat{\mathbb{O}}_\delta^i, \hat{\mathbb{C}}_\delta^i\}$ are further reevaluated and refined by multi-stage training by mapping their $D = (2P + 2 + \tau * 2)$ channels to 256 using 1×1 convolutions. These, when element-wise added to the intermediate features from the previous hourglass $\phi_{\delta-1}$, serve as an input to the next hourglass, which generates another set of predictions. This repeated bottom-up, top-down inference preserves spatial locations of features, which is important to bring different features together to form a coherent understanding of the scene.

$$\{\hat{\mathbb{H}}^i, \hat{\mathbb{G}}^i, \hat{\mathbb{O}}^i, \hat{\mathbb{C}}^i\} = \psi_{S_N} = \mathscr{F}_\Delta \odot \phi_\Delta \odot \ldots \mathscr{F}_2 + \mathscr{F}_3 \odot \phi_3 \odot \mathscr{F}_1 + \mathscr{F}_2 \odot \phi_2 \odot \mathscr{F}_1 \odot \phi_1(\mathscr{I}). \tag{4.12}$$

Another key feature of the stacked hourglass architecture is the multi-stage architecture. In the original network design, $\delta = 8$ hourglass modules are stacked end to end, without sharing the weights, and the loss in equation (3.12) is applied to each intermediate prediction $\{\hat{\mathbb{H}}^i_\delta, \hat{\mathbb{G}}^i_\delta, \hat{\mathbb{O}}^i_\delta, \hat{\mathbb{C}}^i_\delta\}$ using the same ground truth maps $\{\mathbb{H}^i, \mathbb{G}^i, \mathbb{O}^i, \mathbb{C}^i\}$.

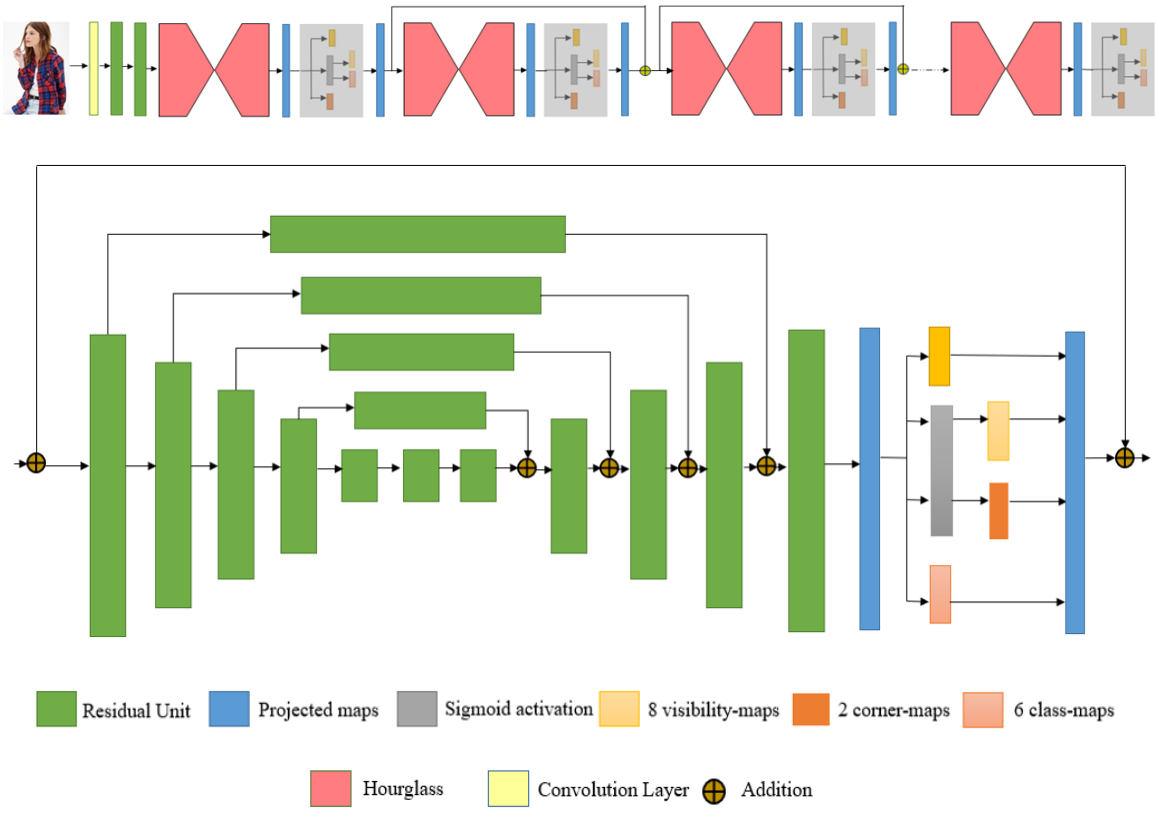


Figure 8: Stacked hourglass architecture. The top figure represents the overall stacked hourglass and the middle figure represents the internal structure of the hourglass as per equation (4.11).

All the maps are repeatedly refined by minimizing final loss in equation (3.12) using an optimization algorithm like RMSProp or Adam [94]. We use Adam in our experiments because Adam uses adapted learning rate as well as momentum, hence updating the weight matrices with more appropriate gradients.

Pyramid Residual Networks

Pyramid residual networks modify the stacked hourglass architecture by replacing the residual units $\{\omega_1, \ldots, \omega_M\}$with pyramid residual modules (PRM) $\{\omega'_1, \ldots, \omega'_M\}$ that learn features at different resolutions in each layer, hence forming a feature pyramid. These feature pyramids are similar to Laplacian pyramids [95] obtained for the images by down-

sampling the image at different sub-sampling ratios. Such functionality is inherent to convolutional neural networks since every convolutional and pooling layer reduces the size of the input. Such a modification is motivated by the fact that while stacked hourglass architecture captures and consolidates global and local information, it still doesn't account for many scale changes due to inter-personal body shape variations, view-point variations, apparel-variation and foreshortening. This modified architecture tackles such variations using PRM.

In each ω'_{μ} The input is first processed through a simple convolutional layer as in equation (4.5) and then the network branches off to create feature pyramids. A PRM can be formulated as:

$$\mathbb{F}^{l} = \mathscr{R}\left(\mathbb{F}^{(l-1)} \mid W^{(l)},\, B^{(l)}\right) + \mathbb{F}^{(l-1)} \tag{4.13}$$

where $\mathscr{R}\left(\mathbb{F}^{(l-1)} \mid W^{(l)},\, B^{(l)}\right)$ can be decomposed as :

$$\begin{aligned}\mathscr{R}\left(\mathbb{F}^{(l-1)} \mid W^{(l)},\, B^{(l)}\right) = \mathscr{F}\left(\sum_{j=1}^{K} \mathscr{F}\left(\mathbb{F}^{(l-1)} \mid w_j^{(l)},\, b_j^{(l)}\right) \mid w_{j+1}^{(l)},\, b_{j+1}^{(l)}\right) \\ + \mathscr{F}\left(\mathbb{F}^{(l-1)} \mid w_0^{(l)},\, b_0^{(l)}\right)\end{aligned} \tag{4.14}$$

where, K denotes the number of pyramid levels. Here $W^{(l)} = \{w_0^{(l)}, w_1^{(l)}, \ldots, w_j^{(l)}, w_{j+1}^{(l)}\}$, $B^{(l)} = \{b_0^{(l)}, b_1^{(l)}, \ldots, b_j^{(l)}, b_{j+1}^{(l)}\}$ are the set of parameters to be learned. Outputs of each transformation $\left(\mathbb{F}^{(l-1)} \mid w_j^{(l)},\, b_j^{(l)}\right)$ are up-sampled to original scale of $\mathbb{F}^{(l-1)}$, summed together, and further convolved to give the final output of the PRM unit.

In order to do the sub-sampling, instead of max-pooling in stacked hourglass, fractional pooling [96] is used to build PRMs. This is because max-pooling sub-samples the image too fast (by ratio k, the kernel width), making it difficult to create smooth pyramids. Furthermore, the disjoint nature of the pooling regions may limit the generalization ability of the produced feature maps. In fractional pooling, the original resolution is gradually reduced by the sub-sampling ratio of j^{th} pyramid as :

$$s_j = 2^{-T\frac{j}{K}}, \;\; j = 0, \ldots, K,\; T \geq 1 \tag{4.15}$$

where $s_j \in [2^{-T}, 1]$, denotes the relative resolution compared to the input features.

Furthermore, two residual units in the initial convolutions block $\mathscr{I}$ of the stacked hourglass architecture are also replaced by two PRM units and the rest of the network

builds and computes the prediction maps in the similar fashion to the basic stacked hourglass architecture as in equation (4.12). Figure 9 represents a PRM unit.

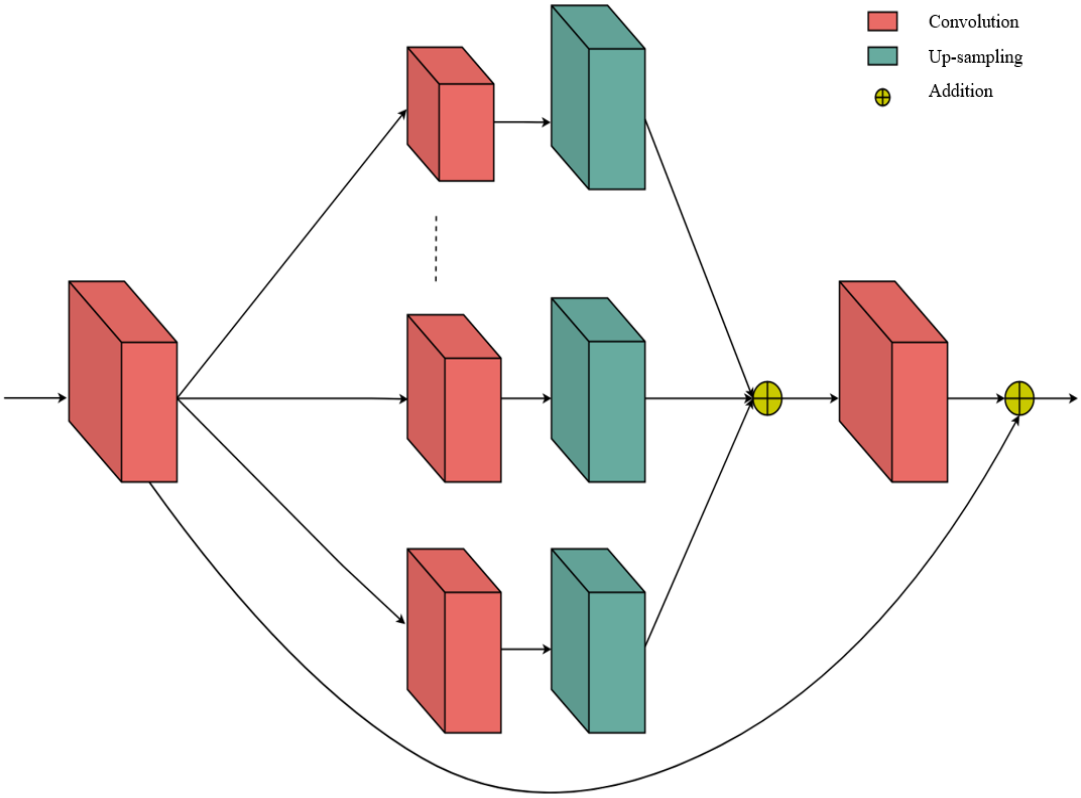


Figure 9: PRM-unit used throughout the stacked hourglass architecture. Input feature maps are pooled at different rations and then up-sampled to the input resolution, where they are added together and convolved again to produce the final feature map.

Cascade Prediction Fusion

Cascade prediction fusion (CPF) network leverages the multi-stage architecture of stacked hourglass network by gradually integrating different semantic information from the initial stacks to the later stacks. The prediction maps computed at the previous stack are used to as a prior to guide the prediction of the present stack. The prediction maps $\{\hat{\mathbb{H}}^i_{\delta-1}, \hat{\mathbb{G}}^i_{\delta-1}, \hat{\mathbb{O}}^i_{\delta-1}, \hat{\mathbb{C}}^i_{\delta-1}\}$ from stack $\phi_{\delta-1}$ undergoes a 1×1 convolution to increase channels to 256 and are then fused with the original intermediate feature maps for stack ϕ_δ using element-wise addition. $\{\hat{\mathbb{H}}^i_\delta, \hat{\mathbb{G}}^i_\delta, \hat{\mathbb{O}}^i_\delta, \hat{\mathbb{C}}^i_\delta\}$ in ϕ_δ are computed using this fused feature map as input. Figure 10 illustrate the resulting architecture.

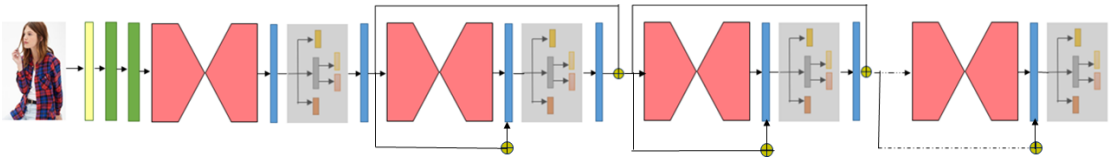


Figure 10: Cascade prediction fusion (CPF) modification of stacked hourglass architecture

4.3 Evaluation Metrics

The different tasks in the fashion analysis presented in this work have different evaluation metrics. These evaluation metrics are based on the evaluation metrics used in previous works [12, 72]. None of the previous work evaluated the visibility state of the landmarks, so, we define our own metrics to evaluate that.

Landmark's Location

We use normalized error (NE) and percentage of detected landmarks (PDL) metrics to evaluate and compare our results. These metrics are computed between final predicted and ground-truth location of the landmarks.

NE

The landmark predictions are computed using the argmax function over the predicted heat-maps respectively using equation (3.4). Normalized errros (NE), defined as $NE : \mathbb{R}^{2\times P} \rightarrow \mathbb{R}$ is the l_2 distance $\left\|a_p^i - \hat{a_p^i}\right\|^2$ between the predicted and the ground truth landmarks in the normalized coordinate space (i.e. divided by height/width of the image). Note that this l_2 distance is calculated only between the (u,v) location of the landmarks. Smaller values of NE mean better results. Also, this normalized error is normalized in the dimensions of the heat-maps. Hence they give an approximate distance between the predicted and ground-truth heat-maps with respect to the location of their actual landmarks.

$$NE_P^N(a,\hat{a}) = \frac{1}{N}\frac{1}{P}\sum_{i=1}^{N}\sum_{p=1}^{P}\frac{\left\|a_p^i - \hat{a_p^i}\right\|^2}{\bar{m}}. \tag{4.16}$$

PDL

$PDL : \mathbb{R}^{2\times P} \rightarrow \mathbb{R}$ is calculated as percentage of detected visible ($\gamma = 1$) landmarks with normalized error under the threshold η. In our experiments, we take $\eta \in \{0.5, 0.7\}$.

$$PDL_P^N(a,\hat{a}) = \frac{\text{count}\left(\frac{\left\|a_p^i - \hat{a_p^i}\right\|^2}{\bar{m}} < \eta\right)}{N * P} \times 100, \qquad (4.17)$$
$$i \in \{1,\ldots,N\},\ \ p \in \{1,\ldots,P\}.$$

Landmark's Visibility

For the landmarks' visibility, we calculate the percentage of visibility states classified correctly. This metrics is calculated only on the γ values of the landmarks calculated using equation (3.7). We call this metrics $PDLv : \mathbb{R}^P \rightarrow \mathbb{R}$, given as:

$$PDLv\,{}_P^N(a,\hat{a}) = \frac{\text{count}\left(a_p^i(\gamma) = \hat{a_p^i}(\gamma)\right)}{N * P} \times 100. \qquad (4.18)$$

Apparel Detection

The apparel's bounding box localization and the apparel classification tasks are jointly evaluated using average precision metrics, popularly used in COCO object detection [71] tasks or PASCAL VOC object [72] detection tasks. Average precision (AP) metrics involves measuring intersection over union (IOU) of the predicted and the ground truth bounding box using Jaccard distance [97]. The ground truth corner locations α_1^i, α_2^i and predicted corner locations $\hat{\alpha_1^i}, \hat{\alpha_2^i}$ calculated for image x^i using equation (3.9) are used to compute the ground truth bounding box Υ^i and predicted bounding box $\hat{\Upsilon}^i$. This IOU value is used to determine whether a detection is valid (true positive) or not (false positive). IOU is calculated as the overlapping area between the predicted bounding box $\hat{\Upsilon}^i$ and the ground truth bounding box Υ^i over their area of union.

$$IOU = \frac{\text{area } (\hat{\Upsilon}^i \cup \Upsilon^i)}{\text{area } (\hat{\Upsilon}^i \cap \Upsilon^i)}, \tag{4.19}$$

The boxes having the correct classification label and an (IOU) value greater than a threshold ι are counted as true positives (TP), and the ones having the value smaller than ι are counted as false positives (FP). Whereas, the bounding boxes with the incorrect classification labels but IOU value greater than ι are considered as false negatives (FN). Since all the images are annotated with the ground truth bounding box, there are no true negatives. The values of TP, FP and FN are used to calculate precision and recall. Precision ρ identifies the relevant objects by calculating percentage of correct positive predictions given as:

$$\rho = \frac{TP}{TP+FP}, \tag{4.20}$$

whereas recall is the measure of all the relevant cases given as the percentage of true positives detected among all relevant ground truths. Recall is given as:

$$r = \frac{TP}{TP+FN}, \tag{4.21}$$

These values are used to compute a precision/recall curve to evaluate any object detector [98]. The value ι is changed by plotting the curve for each object class. A good object detector would have high precision as the recall increases, which indicates its ability to identify only the relevant objects. Whereas, a poor object detector needs to increase the number of detected objects to maintain a high recall value. That's why, a precision/recall graphs starts with high precision values, decreasing as recall increases. AP summarizes the shape of precision/recall curve, calculated as the mean precision at a set of 11 equally spaced recall levels $r \in [0, 0.1, \ldots, 1]$, taking maximum precision whose recall value is greater than r, as in PASCAL VOC challenge [98]. If $\hat{\rho}$ represents the interpolated precision, AP is given as:

$$AP = \frac{1}{11} \sum_{r \in [0, 0.1, \ldots, 1]} \hat{\rho}(r), \tag{4.22}$$

with,

$$\hat{\rho}(r) = \max(\rho(\tilde{r})), \;\; \tilde{r} \geq r. \tag{4.23}$$

and $\rho(\tilde{r})$ is the measured precision at recall $\tilde{r}$.

In this chapter, we described the convolutional neural networks and how they are used to build the stacked hourglass architecture and it's two other variations in order to solve this final objective of joint fashion landmark detection and apparel detection. We also defined the metrics to evaluate each of these tasks in our final objective of joint fashion landmark detection and apparel detection individually. These metrics can be used to compare the methods used in previous works as well. In the next chapter, we will do an exploratory analysis of the DeepFashion dataset.

Chapter 5

DeepFashion Dataset

DeepFashion[12] is a comprehensively annotated apparels dataset that contains apparel categories, attributes, landmarks, as well as cross-pose/cross-domain correspondences of apparel pairs. The landmark annotation itself is comprehensive since in addition to the landmarks, it also contains the high-level apparel categories, the pose-variations and the bounding box coordinates localizing the apparel in the images. These landmarks are apparel-centric, hence can be very different from the joints of human body in the problem of human pose estimation. Figure 11 shows some of the sample images in the DeepFashion dataset.

DeepFashion dataset has about 123,000 images, richly annotated to support multiple tasks like landmark detection and apparel classification and apparel detection for fashion analysis. These annotations contain eight landmarks, namely left collar, right collar, left sleeve, right sleeve, left waist, right waist, left hem and right hem; three high-level apparel categories, i.e., upper (e.g., shirts), lower (e.g., pants) and full-body (e.g., dresses) apparels depending upon the body-parts they are designed for; five viewpoint variations, i.e., normal, medium or large and medium zoom-in or large zoom-in poses; and two bounding box coordinates, i.e. top-left and bottom-right corners' coordinates. Also, since the landmarks on the apparels are frequently occluded in images, these annotations also contain the visibility status of the landmarks. Originally the visibility flag $\gamma \in \{0, 1, 2, 3\}$, indicating the occluded, visible, truncated and non-existent landmarks respectively. Truncated landmarks means that the landmark exists in the apparel but has been cropped in the image and non-existent landmark means that the landmark does not exist for this particular apparel, e.g., the left and right collar or the left and right sleeve do not exist in a lower-body apparel like pants. Table 1 represents the existing and non-existing landmarks in each category. But for the training purposes, we consider all the visibility states other

than visible as occluded, hence $\gamma \in \{0, 1\}$, where values 1 and 0 indicate the visible and the occluded landmarks respectively.

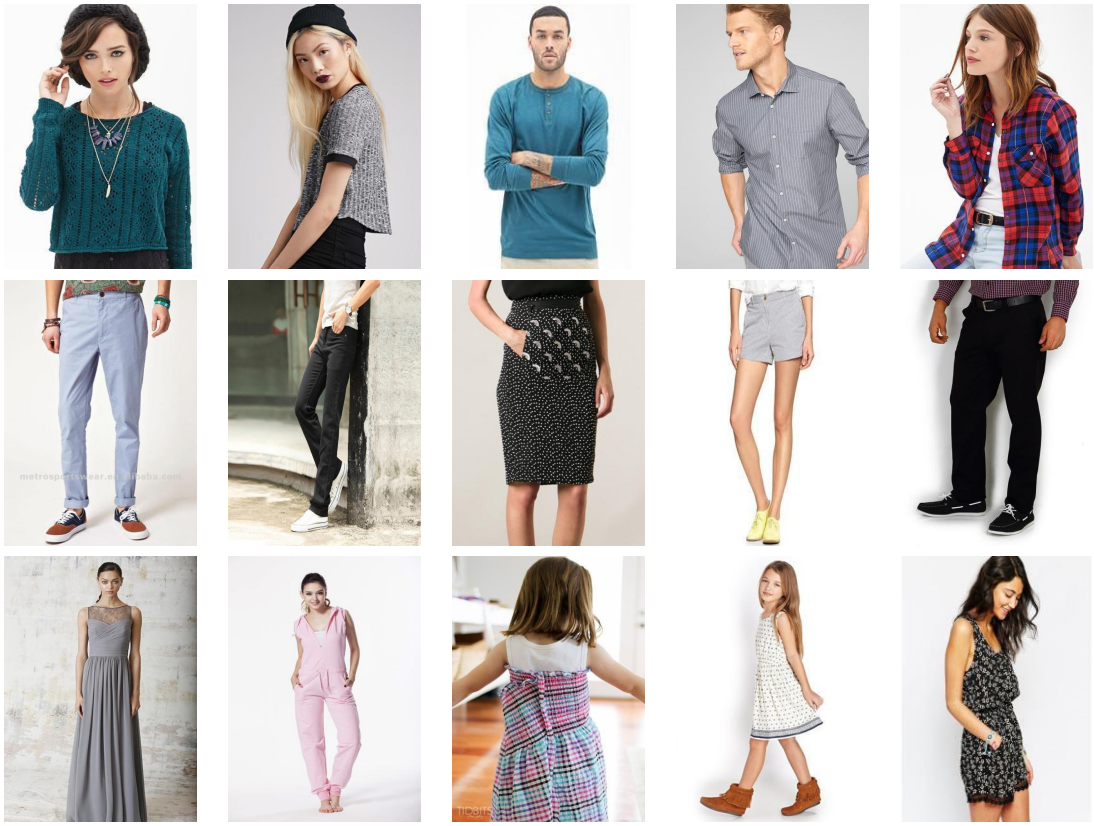


Figure 11: Sample images from DeepFashion dataset. First row : upper-body apparel images, second row : lower-body apparel images, third row : full-body apparel images.

Table 1: Presence of landmark in each apparel category, i.e., left collar, right collar, left sleeve, right sleeve, left waist, right waist, left hem, right hem respectively. Here, *P* represents the number of landmarks in each apparel category.

Category	**P**	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**
upper	6	√	√	√	√	√	√	×	×
lower	4	×	×	×	×	√	√	√	√
full	8	√	√	√	√	√	√	√	√
overall	8	√	√	√	√	√	√	√	√

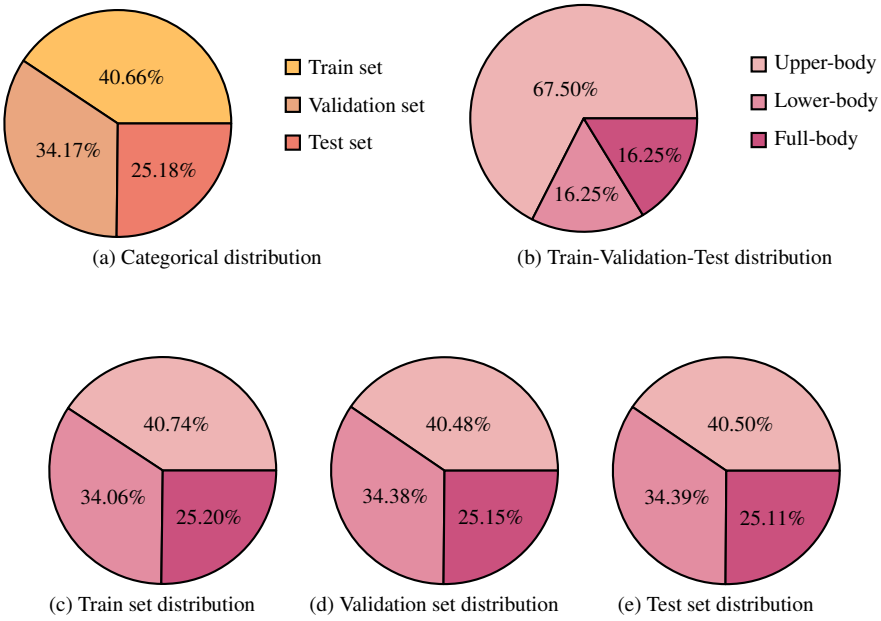
This dataset is divided into train, test and validation set. We train our architectures on the train dataset, modify the training process on the validation set and then evaluate the results on the test dataset. Table 2 represents the statistics of images in the DeepFashion dataset. Figure 12 (a,b) show the detailed statistic of images in train, validation and test sets as well as their distribution in their respective categories. Figure 12 (c,d,e) show the

statistics of landmarks distribution for all apparel categories as well as in the test, train and validation divisions. These statistic are important to analyze the category-aware and category agnostic learning for the landmarks detection. As we can see, the distribution of the images in different apparel categories is consistent in the test, train and validation dataset. Figure 13 shows the distribution of the visibility state's in each of the subset. This distribution is also consistent through the train, test and validation set. However, full-body apparels in the train set have relatively fewer occluded landmarks than the ones in full-body validation and train set. Similar inconsistency is also there in upper-body apparel datasets as upper-body train set has more occluded landmarks than the upper-body apparels test and validation dataset. This might result in a very small bias while evaluating the accuracy of the visibility states of the landmarks.

DeepFashion mainly contains well-posed shop images, with mostly white background. The images contain different profile variations of the apparels but they have been mostly taken at the same camera angle hence very few images with the possibility of foreshortening. Because of the same reasons, there are very few variations in the pose orientation of people wearing the apparels. The Figure 14 depicts the histogram of average horizontal orientations of the pose vectors of the images in the complete dataset as well as in the respective apparel categories. Here an average horizontal orientation vector is calculated as the average of the clockwise angle between the mirroring pairs of annotations in each image and the X-axis. e.g., right and left collar, right and left sleeve, right and left waist, etc. And the average vertical orientation vector is calculated as the average of the clockwise angle between the vertical pairs of annotations in each image and the X-axis. e.g., right collar and right waist, left collar and left waist, right waist and right hem, and left waits and left hem. The histograms show the distribution of the poses is consistent throughout the dataset, except for the full-body apparels. Full body apparels contain more variations in the poses. Furthermore, as discussed earlier, DeepFashion also has labels for view-point variations, i.e., normal, medium or large and medium zoom-in or large zoom-in poses. Figure 15 represents the statics for viewpoint variation of the images. This is necessary to evaluate our results both quantitatively and qualitatively, since the landmarks in a certain view-point might be more difficult to localize than the others.

Table 2: Data Distribution in DeepFashion.

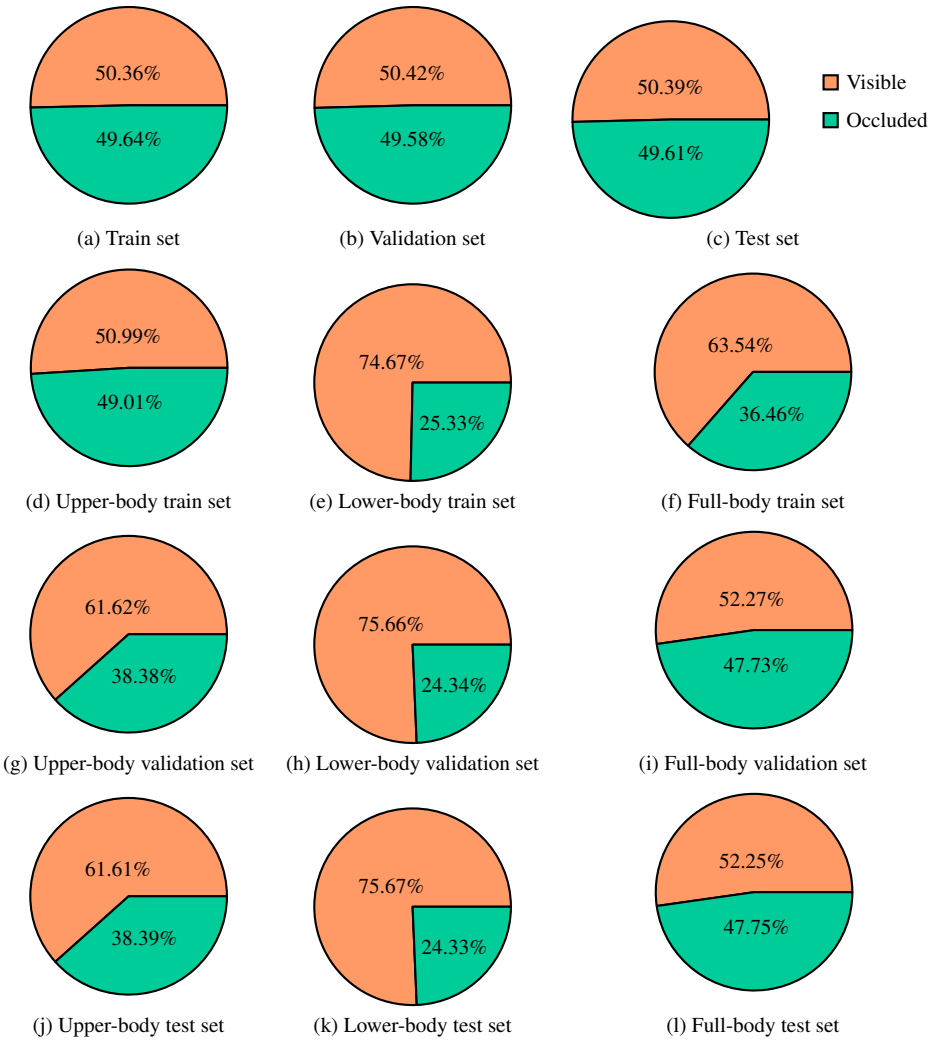
Category	**Total**	**Train**	**Validation**	**Test**
upper	42031	28283	6873	6875
lower	30972	20925	5027	5020
full	50013	33825	8092	8096
overall	123016	83033	19992	19991



(a) Categorical distribution (b) Train-Validation-Test distribution

(c) Train set distribution (d) Validation set distribution (e) Test set distribution

Figure 12: DeepFashion statistics.



(a) Train set (b) Validation set (c) Test set

(d) Upper-body train set (e) Lower-body train set (f) Full-body train set

(g) Upper-body validation set (h) Lower-body validation set (i) Full-body validation set

(j) Upper-body test set (k) Lower-body test set (l) Full-body test set

Figure 13: Landmark's visibility and occlusion statistics in DeepFashion dataset.

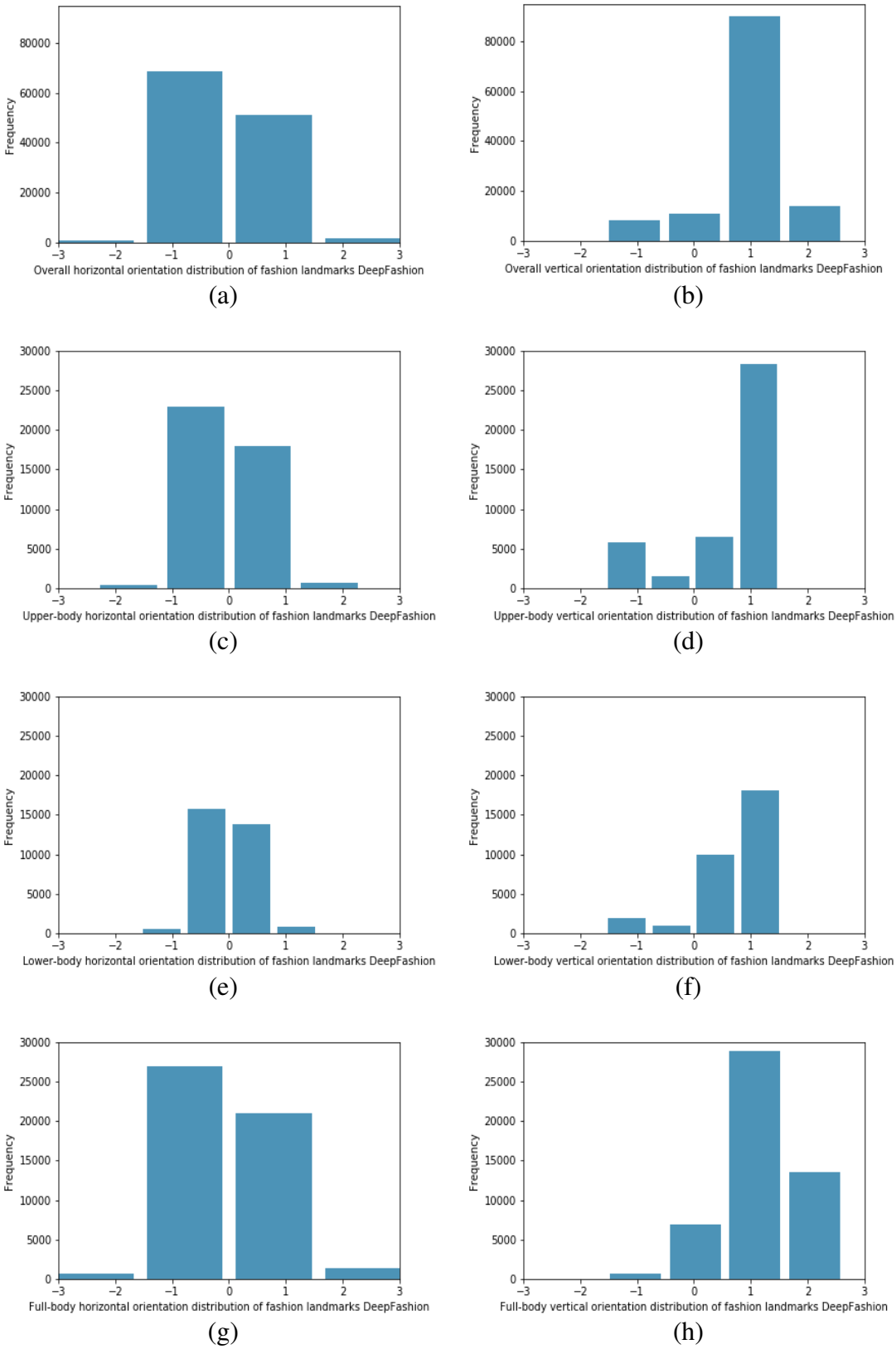
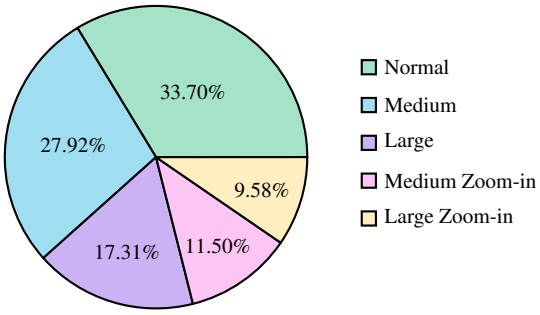
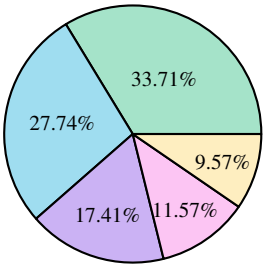


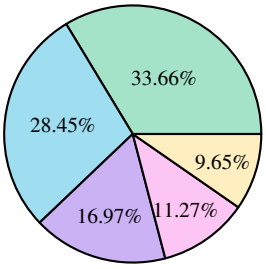
Figure 14: Distribution of horizontal and vertical orientations of the fashion landmarks. The orientation values in the x-axis is in radians.



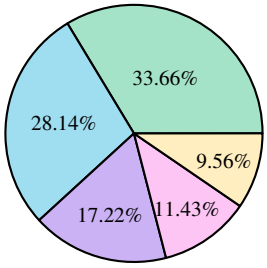
(a) DeepFashion pose-variation distribution.



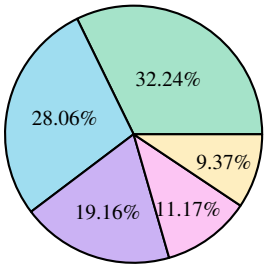
(b) Train



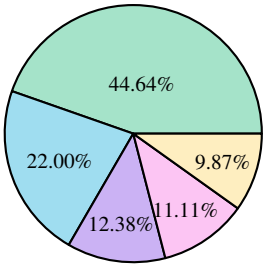
(c) Validation



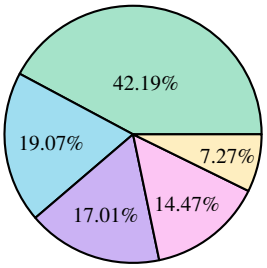
(d) Test



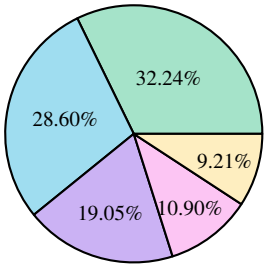
(e) Upper-Body Train



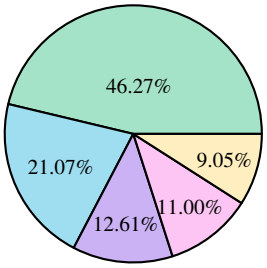
(f) Lower-Body Train



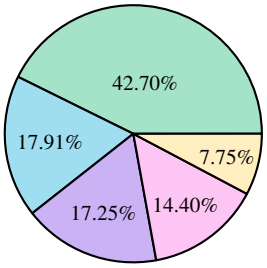
(g) Full-Body Train



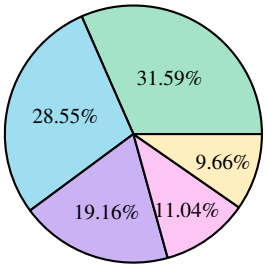
(h) Upper-Body Validation



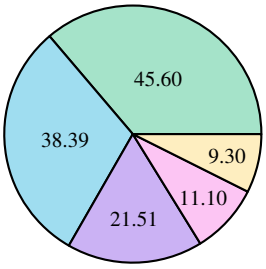
(i) Lower-Body Validation



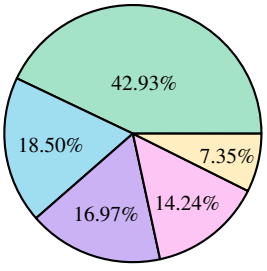
(j) Full-Body Validation



(k) Upper-Body Test



(l) Lower-Body Test



(m) Full-Body Test

Figure 15: View-point variation statistics in DeepFashion dataset.

Chapter 6

Implementation and Analysis

There are a lot of hyper-parameters that determine the performance of the convolutional neural network architectures in deep learning, e.g., the number of layers, the number of weight matrices, the input resolution of the image, the optimizer to be used for learning, the learning rate etc. In the stacked hourglass architecture used in this work, there are many additional hyper-parameters to be determined like the number of stacks to be used, the number of residual modules used in both the top-down and bottom-up processing etc. Many of these choices are based on plain heuristics, but we conducted various experiments in order to check which configuration of the architecture work the best to solve this scenario and what is the trade-off between the faster and the better learning. In this chapter, we will discuss various implementation settings and analyse the qualitative and quantitative results on the proposed setting.

6.1 Implementation

The input images are re-sized to 256×256 resolution, hence $m = n = 256$. The average resolution of the images in DeepFashion dataset is about 340×400. For the sake of simplicity, we take square images as input. That's why we chose $m = n = 256$. Reducing the resolution will result in the loss of a lot of features and the resulting feature maps will be less richer in comparison. The initial block of each architecture $\mathscr{I}$ brings the resolution down to 64×64, hence $m' = n' = 64$. This is the resolution of the input and the output of each stack, as well as of the ground truth and predicted maps, i.e., $\{\hat{\mathbb{H}}^i, \hat{\mathbb{G}}^i, \hat{\mathbb{O}}^i, \hat{\mathbb{C}}^i\}$ and $\{\mathbb{H}^i, \mathbb{G}^i, \mathbb{O}^i, \mathbb{C}^i\}$. At this point, the feature maps are rich enough from the initial block of convolution as they contain features from the bigger resolution. We chose 64×64

as the output resolution of the feature maps so as this gives a good balance between the richness of the consequent feature maps and the number of further convolutions we can apply using the residual modules. Each hourglass has $M = 15$ residual modules, out of which, there are 5 residual modules in the bottom-up processing, 5 residual modules in the top-down processing and 5 residual modules that are applied on the residual modules of the bottom-up processing before adding the via skip-connections to the residual modules from the top-down processing. After the top-down processing each stacked hourglass ϕ_δ reaches down to the feature maps of resolution 4×4 in $\omega_{\frac{M}{2}}$. Shallower architectures provided some loss in accuracy, while deeper architectures only improve the accuracy by maximum 0.5%, while increasing the number of parameters to be learnt.

All the architectures are implemented using Torch and optimized with Adam optimizer. The learning rate starts at 0.0007 and decreases by 10 after the validation accuracy pleatues. We also experimented with 0.0002 and 0.0005 learning rate, but it did not create any difference in the training accuracy. This is because, no matter what learning rate we start with, Adam optimizer adjusts it based on the moving mean of the gradients. The parameters are randomly initialized. We train each architecture for 100 epochs with batch size 10. The accuracy saturates after about 70-80 epochs. Table 3 lists the number of parameters to be learned in each architecture. The number of additional parameters to be learned in pyramid residual networks are 19.63% more than the ones in stacked hourglass architecture, whereas the additional parameters are only 0.08% more in cascade prediction fusion variation. In the further analysis, we will represent these architectures with the abbreviations mentioned in Table 3.

Table 3: Number of parameters in each stacked hourglass variation

Network architecture	**Parameters**
Stacked hourglass (SHG)	24,608,038
Stacked hourglass: Pyramid residual network variation (SHGP)	29,440,492
Stacked hourglass: Cascade prediction fusion variation (SHGC)	24,627,030

The initial block $\mathscr{I}$ uses 7×7 filters to bring down the resolution of the input image to the required resolution but in the hourglasses, we use only 3×3 weight matrices, i.e, $w_d^{(l)} \in \mathbb{R}^{3\times3}$. This is because, as noticed in [9], layers with two 3×3 weight matrices, cover the same spatial area as one 5×5 matrix but the later has more number of parameters to be learned than the former. Hence it is beneficial to use smaller weight matrices while training. The initial stacked hourglass architecture used $\Delta = 8$ hourglasses for training, but as demonstrated in [99], using more that 8 hourglasses causes saturation. So, in our experiments, we only used 2 hourglasses in all the architectures. Testing is performed

with PDL thresholds $\eta = \{0.5, 0.7\}$. These values are empirically determined from the validation set. Figure 16 shows the improvement of PDL and PDLv accuracy over the training period. We observe that multi-task training provides better results in the first few epochs in comparison to the individual landmark detection training. In multi-task training, the PDL score saturates after 30 epochs, hence the similar or better results can be achieved with fewer epochs in comparison to individual landmark detection training. This though might be counter-intuitive, since in multi-task training, the architecture has the more information to learn, but it works because it has been noticed in many experiments involving the multi-task training that the adjunct tasks help to train better on the subsequent task as well. Furthermore, the joint architecture, did not learn anything new about the landmarks, as we will discuss later in the chapter, but it learnt the landmarks faster than it's individual counterpart. For pyramid residual modules, we used $K = 5$ pyramids in a residual module. The stacked hourglass architecture with pyramid residual models performs the best for all the categories in category-aware training as well as in the category-agnostic training, but it takes almost one more day to train than the other architectures, owing to the increased number of parameters to be learned.

In addition to that, we also perform data augmentation on the training set while training. We apply data augmentation as a Markov process, in which augmentation is performed via a sequence of random transformations. A weight matrix learned on augmented data learns the averaged version of transformed features and a data dependent variance regularization term [100]. Hence it improved generalization by inducing invariance and reducing model complexity. In our experiments we apply a sequence of transformations such as scaling, rotation, flipping, and adding color noise, on the images from the training set, with parameters drawn randomly from hand-tuned ranges. We empirically choose the interval $[-45, 45]$ degrees for rotation, $[0.75, 1.25]$ for scaling. This regularization helps to avoid the over-fitting in the training process.

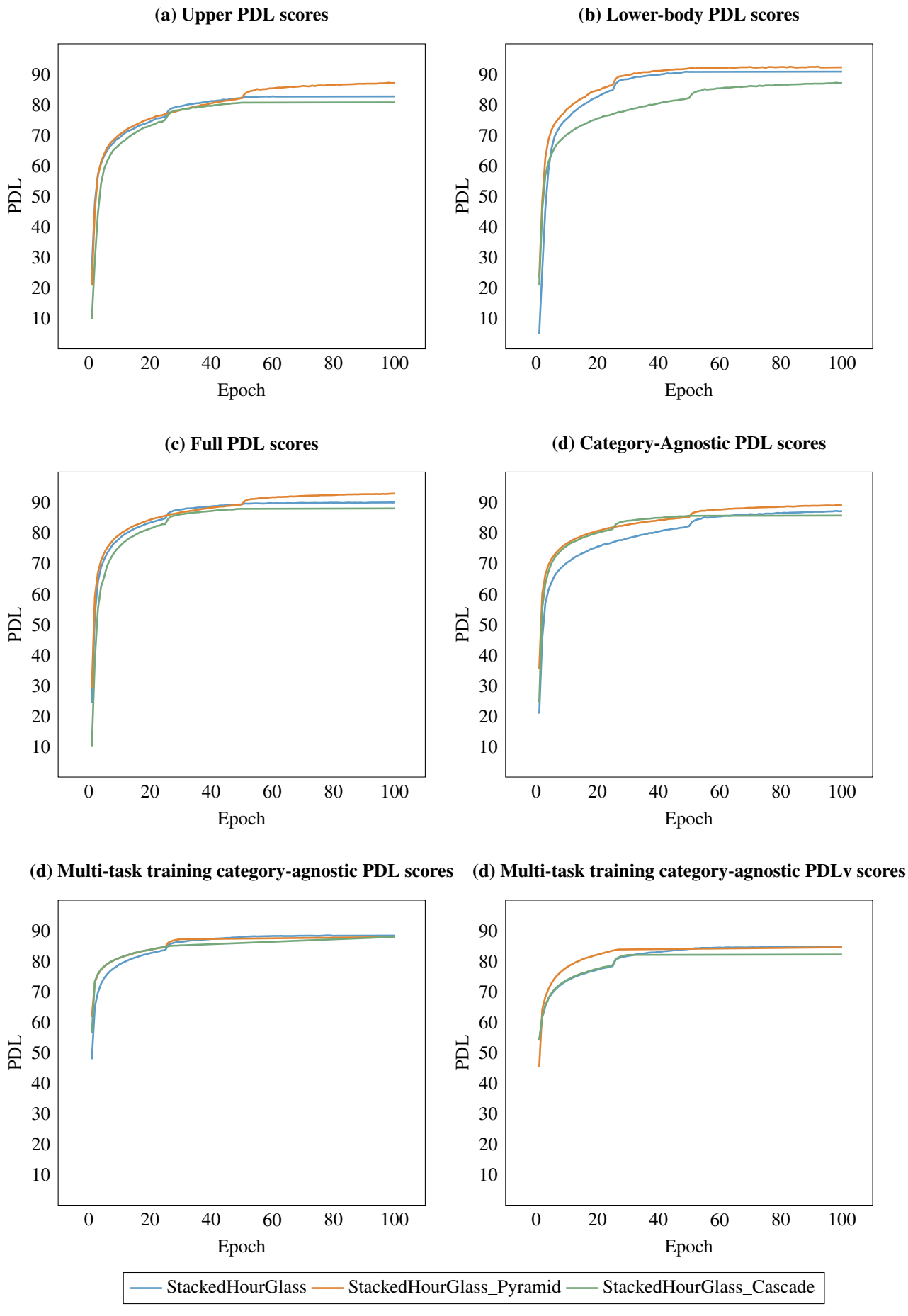


Figure 16: PDL accuracy during the training process.

6.2 Analysis and Discussion

We implemented our architectures using the hyper-parameters discussed in the previous section. In this section, we will analyze the results of all the experiments performed on these architectures for all the tasks, both quantitatively and qualitatively.

6.2.1 Landmark's Localization

For landmark localization, since different categories of apparels have different number of landmarks, in the initial experiments, we explore the trade-off for the learning on pre-classified apparels, i.e., category-aware learning as well as on unclassified apparels, i.e., category-agnostic learning of these landmarks. In these experiments, we only predict the fashion landmarks, i.e., these predictions are without the multi-task learning. Table 4 and Table 5 list the PDL results on the category-aware experiments on the test and validation set of DeepFashion. Similarly Table 6 and Table 7 list the results on the category-agnostic experiments on test and validation dataset of DeepFashion respectively.

The results indicate that category-aware learning for fashion landmarks provides only a small advantage over the category-agnostic learning. But it only increases the number of parameters to learn on the whole dataset, since the chosen architecture would have to be trained three times, each time for a particular category. So, even if the landmarks are sparse, a simple stacked hourglass architecture provides a good generalize-ability to detect the fashion landmarks on all types of apparel images.

Table 4: Category-aware PDL comparison on DeepFashion validation dataset at $\eta = 0.5$.

Method	**Category**	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	upper	81.6	80.8	75.5	76.3	71.0	70.5	×	×	75.9
SHGp	upper	81.8	81.0	75.7	76.5	71.3	70.7	×	×	76.4
SHGc	upper	80.7	80.8	74.1	75.5	70.1	69.6	×	×	75.0
SHG	lower	×	×	×	×	87.3	87.3	85.8	86.2	86.7
SHGp	lower	×	×	×	×	87.6	87.7	86.2	86.7	87.1
SHGc	lower	×	×	×	×	86.4	86.0	85.0	85.4	85.7
SHG	full	93.5	94.0	79.0	78.9	85.9	85.7	84.0	84.4	85.7
SHGp	full	93.4	94.1	79.1	79.0	85.9	85.8	84.2	84.3	85.8
SHGc	full	92.9	93.4	76.4	77.2	85.0	84.8	83.3	82.7	84.5

Table 5: Category-aware PDL comparison on DeepFashion test dataset at $\eta = 0.5$.

Method	**Category**	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	upper	80.2	79.5	75.3	76.1	70.7	69.9	×	×	75.2
SHGp	upper	80.9	79.8	75.6	76.5	70.9	70.3	×	×	76.0
SHGc	upper	80.1	79.5	73.6	75.6	70.7	69.4	×	×	74.8
SHG	lower	×	×	×	×	87.4	87.3	85.6	86.5	86.7
SHGp	lower	×	×	×	×	87.7	87.7	86	86.9	87.1
SHGc	lower	×	×	×	×	85.9	85.8	84.5	85.4	85.4
SHG	full	93.5	94.0	79.2	78.9	85.7	85.6	84.1	84.2	85.6
SHGp	full	93.6	94.1	79.3	80	85.8	85.7	84.2	84.3	85.7
SHGc	full	93.1	93.5	76.6	76.9	84.9	84.8	83.0	82.5	84.4

Table 6: Category-agnostic PDL comparison on DeepFashion validation dataset without multi-task learning at $\eta = 0.5$.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	87.3	87.1	77.5	78.1	77.9	77.1	84.5	84.9	81.8
SHGp	88.0	87.5	78.1	78.8	78.5	77.5	84.9	85.2	82.3
SHGc	86.9	86.7	76.4	77.2	76.7	76.2	83.7	83.8	81.1

Table 7: Category-agnostic PDL comparison on DeepFashion test dataset without multi-task learning at $\eta = 0.5$.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	87.3	87.2	77.4	77.9	77.7	77.2	84.5	84.7	81.7
SHGp	88.1	87.5	78.1	78.3	78.5	77.6	84.9	85.2	82.3
SHGc	86.6	86.7	76.4	77.3	76.6	76.3	83.5	83.7	81.1

Table 8: Category-agnostic PDL comparison on DeepFashion validation dataset without multi-task learning at $\eta = 0.7$.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	94.2	94.0	86.1	86.1	86.4	87.7	87.5	91.5	89.2
SHGp	94.7	94.5	86.7	86.6	86.9	87.9	88.3	92.1	89.7
SHGc	91.6	91.4	82.6	83.5	83.6	82.8	87.8	88.0	86.4

Table 9: Category-agnostic PDL comparison on DeepFashion test dataset without multi-task learning at $\eta = 0.7$.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	92.2	92.0	84.2	84.5	84.6	84.8	87.8	89.0	87.3
SHGp	92.4	92.6	84.5	84.9	85.0	85.4	88.0	89.6	87.7
SHGc	89.6	89.5	80.8	82.1	82.3	81.1	87.5	85.5	84.8

We also observe that among all the types of apparels, the fashion landmarks in the upper-body apparels are the most difficult to localize. This might be because of the variability in the designs of the upper-body apparels. e.g., T-shirt, tank-tops, camisoles, shirts, etc. have a huge variation in the location of the landmarks. Also, among all the landmarks in the fashion items, the left and the right collar are the most easy to localize, whereas, the left and right sleeve and the left and right waist are the most difficult to localize in both the category-aware and category-agnostic learning. But looking at the results of the category-aware learning, we can conclude that the difficulty in localizing the left and right waist majorly comes from the upper body apparels. We can also conclude that among all three architectures, simple stacked hourglass architecture and it's variation with PRM provide competitive results. Whereas, replacing residual units with PRMs increases the number of parameters but doesn't have a correspondingly significant effect on the results. However, CPF variation of the stacked hourglass architecture doesn't provide better results over the simple hourglass architecture, whereas, it did perform better for human pose estimation [34]. This might be because human pose estimation consists of fewer deformations than the apparels and the features computed in the previous hourglass might be more erroneous in apparels than in the human poses. So, while we expect those features to strengthen our learning ability, they contribute more towards the error.

But since in the category-agnostic learning, all the eight landmarks are predicted, irrespective of the category of the apparel, there are also some predictions on the random locations of the image. This is because of the argmax function used in equation (3.4). The heat-maps related to the landmarks that are not present in the apparel contain very-very low confidence values but the argmax function gives the location containing largest confidence value among those low ones. In the images that contain only one apparel, these locations do not correspond to anything, hence are random, but in the images that contain more than one apparel, they correspond to the landmarks of the other apparel. e.g., If an image contains a person wearing pants and shirt, but is only annotated for the shirt, the last two landmarks will correspond to the left and right ankles of the pants. This does not impact the PDL scores since PDL is calculated only using the landmarks for which annotations are present and visible. Clearly, a simple landmarks based architecture

cannot handle such randomness. But when coupled with a detection task, the predicted class can help to ignore the landmarks that do not belong to the predicted class. This way, the predictions are more refined and accurate.

Table 10 and Table 11 list the results of the landmark detection when the architectures are trained on multiple-tasks in parallel. The results indicate that the multi-task learning does not have any impact on the PDL score of the landmark detection when compared as a stand-alone task. We also plot the PDL vs normalized error for different landmarks as shown in Figure 17. Here, we average the predictions of the symmetric landmarks like left and right collars, sleeves, waist and hems respectively. Since these landmarks do not have an absolute definite position on the apparels, we also compute the PDL score with $\eta = 0.7$. These are listed in Tables 8, 9, 12, 13. Having an absolute position means having only one pixel location that represents the landmark. e.g., In human pose estimation, location of wrist or elbows have a very low scope for error, since these occupy small areas in the image, but in case of fashion landmarks, the functional regions are of larger area, as in, a collar or waist has a comparatively larger area where a landmark can be predicted. Hence, we can consider the landmarks closer up-to $\eta = 0.7$ threshold as correct.

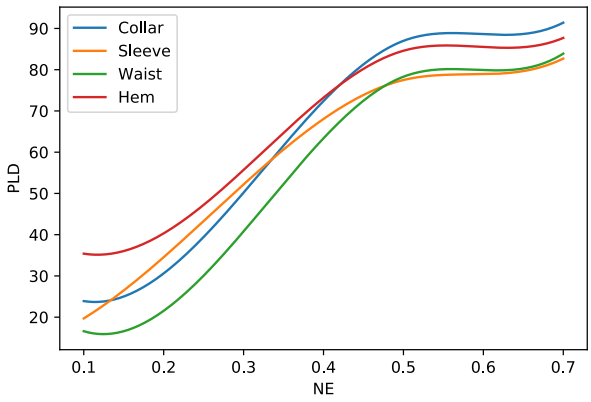


Figure 17: Normalized Error (NE) vs Percentage Landmark Detection (PDL) for on standard stacked hourglass network using multi-task learning

Table 10: Category-agnostic PDL comparison on DeepFashion validation dataset with multi-task learning at $\eta = 0.5$.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	Σ
SHG	86.9	87.1	77.2	77.8	78.4	78.2	85.1	84.1	81.9
SHGp	87.2	87.4	77.5	78.1	78.7	78.5	85.5	84.5	82.2
SHGc	86.5	86.7	76.1	76.9	77.2	77.2	84.3	83.0	81.1

Table 11: Category-agnostic PDL comparison on DeepFashion test dataset with multi-task learning at $\eta = 0.5$.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	86.9	87.1	77.2	77.8	78.4	78.2	85.1	84.1	81.8
SHGp	87.3	87.3	77.5	78.1	78.7	78.6	85.3	84.4	82.1
SHGc	86.2	86.6	76.2	77.5	77	76.9	84.1	83.1	81

Table 12: Category-agnostic PDL comparison on DeepFashion validation dataset with multi-task learning at $\eta = 0.7$.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	94.1	94.0	86.0	86.1	86.0	87.5	87.5	91.5	89.1
SHGp	94.8	94.4	86.8	86.5	86.9	87.9	88.2	92.2	89.7
SHGc	91.6	91.4	82.6	83.5	83.6	82.8	87.8	88.0	86.4

Table 13: Category-agnostic PDL comparison on DeepFashion test dataset with multi-task learning at $\eta = 0.7$.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	92.2	92.0	84.2	84.5	84.6	84.8	87.8	89.0	87.3
SHGp	92.4	92.6	84.5	84.9	85.0	85.4	88.0	89.6	87.7
SHGc	89.7	89.4	80.8	82.0	82.4	81.1	87.5	85.5	84.8

6.2.2 Landmark's Visibility

Table 14 and Table 15 list PDLv scores for all the landmarks, when trained in parallel to the landmark localization in a multi-task learning environment on validation and testing set respectively. In the front facing images for upper-body and full-body apparels, the occlusion around the collars usually come from the hair of the person wearing the apparels. In the upper-body and the lower-body apparels, the occlusion usually comes from the overlap between the upper-body and the lower-body apparels. e.g., A person wearing pants and shirt might have the shirt tucked in, which would cause occlusion if the image is annotated with the upper-body apparel or an un-tucked shirt, which would cause occlusion if the image is annotated with the lower-body apparels. As the results indicate, the visibility flag of waist is more difficult to capture than any other landmark. In the lower-body apparels, the occlusion comes from the kind of shoes the person is wearing. Apart from that, the occlusion comes from the pose of the person in the image. Side poses often hide the landmarks on one side of the apparels with other body parts. The stacked

hourglass architecture is able to capture the visibility state of the landmarks with about 80% accuracy.

Table 14: PDLv comparison on DeepFashion validation dataset.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	83.2	82.1	83.1	81.8	72.3	71.7	84.8	84.9	80.5
SHGp	83.4	82.3	83.4	81.9	72.8	71.9	85.2	85.2	80.8
SHGc	82.2	80.9	81.9	80.5	70.9	70.5	83.8	83.9	79.3

Table 15: PDLv comparison on DeepFashion test dataset.

Method	**l.collar**	**r.collar**	**l.sleeve**	**r.sleeve**	**l.waist**	**r.waist**	**l.hem**	**r.hem**	$\sum$
SHG	81.5	80.1	81.9	80.7	70.6	69.8	85.9	86.0	79.6
SHGp	81.7	80.3	82.2	80.8	70.8	70.2	85.9	86.2	79.4
SHGc	80.4	78.9	80.8	79.4	69.6	69.6	84.8	85.0	78.5

For the final results, we check the predicted class of the landmarks and display the landmarks associated with that class. Figure 18 represents that ground truth and predicted analysis on many images from the validation and test datasets. In our experiments, we observed that the localization of fashion landmarks in the images as well as their visibility states go hand in hand. As discussed earlier, occlusion of the landmarks is mainly of two types. One, if the landmarks are in the image but are occluded by something as shown in Figure 18 (d, e, f). In this case, the ground truth location is available with the visibility flag set as occluded, which is indicated as green in the images. In the other case, the landmarks are present for the apparel but are truncated in the image, as shown in Figure 18 (r). In this case, the ground truth values for the location are unknown and are set to zero. In the former case, since we deem a landmark prediction as correct if it lies within a certain threshold within the ground truth location, it is possible to have some offset between the ground truth and the predicted location, as in the landmarks are not predicted with a pixel wise accuracy. On qualitative evaluations of these predictions, we noticed that the network was able to learn subtle patterns in occlusion and set the visibility flag accordingly. e.g., While predicting the right and left collars, most of the occlusion in the front facing and the side facing images come from the hair, and there is often a small offset between the collar locations covered by the hair and the collar locations not covered by the hair. Images in Figure 18 (d, g) compare the ground truth and predicted landmarks on some upper-body apparels and as it turns out, the visibility state is often predicted correctly with respect to the predicted location of landmarks. We observed similar patterns in the predictions of the landmarks where occlusion can be caused by some body parts. e.g., Images in Figure 18 (f, i) have ground truth occlusion on the landmarks on the waist because of the

back, arms and hands, but when the landmark locations are predicted with an offset, at a place that doesn't have such occlusion, the visibility state of the landmarks is predicted accurately. These would amount to some loss of quantitative accuracy in the visibility states of the landmarks, while these predictions are qualitatively sound. This indicates that the network is clearly able to generalize on the common patterns of occlusion and produce accurate results. In addition to that, at some places, the faulty ground truth locations of the landmarks were also corrected. e.g., Images in Figure 18 (j, k, l) have more accurate predicted landmarks than the ground truth landmarks.

On the other hand, some error in the predictions come from the presence of upper-body apparels and the lower-body apparels in the same image. In category-agnostic learning, before post processing the landmarks as per the predicted class, the network predicts all 8 landmarks for all types of apparels. But since in DeepFashion, only one apparel is annotated in an image, the images that contain both upper-body apparels and lower-body apparels have predictions for all the landmarks. As shown in Figure 18 (m, p), the image contains landmarks of the upper-body apparel, but the network treats it like a full-body apparel and contains correct landmark predictions for the hems as well. Similarly, Figure 18 (n, q) has a lower-body apparel classified as full and upper-body apparel. While calculating the accuracy of the predicted landmark locations, we only consider the visible landmarks, so this does not impact the PDL accuracy for the landmarks, but it does impact the accuracy of the visibility states of all the landmarks. This also shows the scalability of this network to a dataset that contains the annotations of the landmarks of all the apparels in the images.

6.2.3 Apparel Detection

Table 16 represents apparel detection AP score for all the landmarks, when trained in a multi-task environment on test set. We also compare the results with standard SSD and R-CNN. We trained these architecture using transfer learning with the weight values obtained after training on COCO dataset [90]. Our multi-task architecture trained from scratch does better than the other traditional object detection architectures. This is because of the rich features obtained via stacked hourglass architecture and more attention owing to to multi-task learning. We also compare the AP score of the multi-task learning with respect to different pose variations. As shown in Table 17, the AP scores are pretty good for normal, medium and large poses but the scores are very low for medium zoom-in and large zoom-in poses. Hence, the medium-zoom in and the large zoom-in poses are the most difficult to detect and classify correctly.

Table 16: The AP score for apparel detection on test set.

Method	**upper**	**lower**	**full**	**AP**
SHG	59.8	61.2	52.7	57.9
SHGp	61.0	63.2	53.2	58.9
SHGc	58.4	60.4	51.9	56.1
SSD	53.2	55.7	50.8	53.0
R-CNN	53.2	55.7	50.9	53.2

Table 17: The AP score for apparel detection on different variations in the test set.

Variation	**upper**	**lower**	**full**	**AP**
Normal	72.3	70.0	79.4	73.9
Medium	80.5	75.1	73.7	76.4
Large	86.2	69.2	70.5	75.4
Medium zoom-in	33.7	27.7	20.0	27.1
Large zoom-in	55.3	63.6	53.3	57.4

As described earlier, the DeepFashion dataset contains annotation of only one apparel in an image, hence the images that contain both upper-body and lower-body apparels are subject to faulty detection as in Figure 18 (m, n, o, p, q) because the network has generalized over all the categories and would provide stronger detection for the more prominent apparel category in the image. However, the parameters behind decision of this prominence are unknown, it could depend upon the size the apparels are occupying in the image or on the texture or the dominant features that it has generalized on. We also noticed that many upper-body apparels have been miss-classified as full-body apparels. This could be because these upper-body apparels, when paired with certain lower -body apparels, might look similar to full-body apparels as in Figure 18 (m). This, on one side indicates the ability of the network to learn generalized spatial features of the full-body apparels, but also indicates the difficulty to distinguish between the upper-body and full-body apparels. But in addition to these miss-classifications, we noticed that the predicted bounding boxes are more tightly bounded than the ground truth bounding boxes. Figure 18 (d, f, l, r) indicate some examples where the predicted bounding boxes more tightly localize the apparel than the ground-truth bounding boxes.

Figure 18: Sample images from test and validation dataset of DeepFashion. Left images in the image pair indicate the ground truth and the right images indicate the predicted apparel analysis.

In this chapter, we discussed the implementation details of the proposed architectures

and evaluated and compared them for fashion landmark detection, the visibility states of the landmarks as well as apparel detection in the DeepFashion dataset. We presented both qualitative and quantitative analysis of the results. The results indicate the applicability of the proposed methodology in fashion domain. In the next chapter, we will discuss the future work and the scalability of these approaches.

Chapter 7

Conclusion and Future Work

We explored three architectures to detect fashion landmarks on apparel images. We also evaluated the category-aware and category-agnostic training of different apparel categories. The results conclude that among all three approaches, stacked hourglass is the best fit to solve this problem. We also conclude that, when compared between category-aware and category-agnostic learning for the landmarks on the fashion apparels, category-aware learning only provides a small benefit but introduces thrice the number of parameters to be learnt. So, even when the landmarks are sparse, all the stacked hourglass architectures can learn to detect the landmarks with a competent accuracy. We also conclude that stacked hourglass architecture with pyramid residual modules gives the best accuracy but introduces 19% more parameters to learn. Among all the hyper-parameter choices that we had to make in this architecture, a stacked-hourglass architecture with 2 stacks and 15 residual modules, operating at the input image of size 256×256, and producing heat-maps of size 64×64 gives a good trade-off between the number of parameters to be learnt, the training time and the final accuracy on the test set. Of course these settings might change given a different dataset. This implementation setting gives an approximate position of the landmark that is offset by a certain value from the ground-truth landmark. Since these landmarks are on the functional region of clothes, certain magnitude of offsets still produce the landmarks on the approximate locations. This architecture is able to discern the subtle patterns in occlusion when it comes to the visibility states of the landmarks. Using this architecture, we can clearly predict the occlusion when caused by hair or other parts of the body. In some of the cases, even the occlusion caused by the clothes belonging to other parts of the body are also clearly distinguished.

Stacked hourglass architecture computes rich features for a variety of tasks. This architecture has already been used in solving complex tasks like facial landmark detection

and human pose estimation. But our successful experiments with apparel detection indicate that these features are rich enough to learn any complex task individually or in addition to the landmarks or key-points. The computed features can also be used for segmentation or as a generator network in GAN architectures. Both segmentation and generators in GANs contain an encoder-decoder structure, for which the output features are of the same resolution as the input features. This stacked hourglass architecture can be modified for segmentation by computing the final features maps of the input resolution for each category of the objects in segmentation task. In GAN networks, the final output of the generator is a synthetic image. Such modification is also easy to implement by just changing the depth of the output feature maps to 3, corresponding to the RGB channels. This indicates the versatility of this architecture to be applied for different tasks.

This fashion analysis using stacked hourglass architecture is limited by the dataset used in this work. DeepFashion has the annotation of only one apparel per image, whereas this is not a real-time scenario. DeepFashion2 [27] is a similar recently released dataset for the similar task. DeepFashion2 has about 800,000 images, richly annotated with 13 categories of apparels. It contains a bounding box as well as a segmentation mask for each apparel in the image. Whereas, DeepFashion has only eight landmarks, DeepFashion2 dataset has about 54 landmarks on whole, annotated for 13 categories of apparels. These landmarks provide a lot more discriminative analysis than mere eight landmarks. In addition to pose, it has variations with respect to scale and occlusion, making this dataset much more realistic to work with. As explained earlier, this stacked hourglass architecture can also be used to perform segmentation as well. The apparel segmentation can be done in parallel to apparel detection and apparel's landmark detection, by simply by adding another set of 13 feature maps to be learned, corresponding to the 13 categories of apparels. In addition to that, in order to detect multiple apparels in the image, this stacked hourglass architecture can also be used to compute the embedding of each corner as per [84] in order to group the corners that belong to the same apparel. The 54 landmarks can be learnt in a category-agnostic manner as we did for DeepFashion. Hence there is a lot of work that can be done in this domain, using this very architecture but a more advanced dataset.

Bibliography

[1] J. Huang, R. S. Feris, Q. Chen, and S. Yan, "Cross-domain image retrieval with a dual attribute-aware ranking network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1062–1070, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*, pp. 630–645, Springer, 2016.

[3] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang, "Fashion landmark detection in the wild," in *European Conference on Computer Vision*, pp. 229–245, Springer, 2016.

[4] J. Fu, J. Wang, Z. Li, M. Xu, and H. Lu, "Efficient clothing retrieval with semantic-preserving visual phrases," in *Asian Conference on Computer Vision*, pp. 420–431, Springer, 2012.

[5] Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel, "Visual search at pinterest," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1889–1898, ACM, 2015.

[6] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 27–35, 2015.

[7] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun, "Neuroaesthetics in fashion: Modeling the perception of fashionability," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 869–877, 2015.

[8] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg, "Hipster wars: Discovering elements of fashion styles," in *European Conference on Computer Vision*, pp. 472–488, Springer, 2014.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[10] C. Packer, J. McAuley, and A. Ramisa, "Visually-aware personalized recommendation using interpretable image representations," *arXiv:1806.09820*, 2018.

[11] P. Agarwal, S. Vempati, and S. Borar, "Personalizing similar product recommendations in fashion e-commerce," *arXiv:1806.11371*, 2018.

[12] S. Q. X. W. Ziwei Liu, Ping Luo and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[13] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu, "Attentive fashion grammar network for fashion landmark detection and clothing category classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4271–4280, 2018.

[14] H. Chen, A. Gallagher, and B. Girod, "Describing clothing by semantic attributes," in *European Conference on Computer Vision*, pp. 609–623, Springer, 2012.

[15] E. Simo-Serra and H. Ishikawa, "Fashion style in 128 floats: Joint ranking and classification using weak data for feature extraction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 298–307, 2016.

[16] V. Gabale and A. P. Subramanian, "How to extract fashion trends from social media? a robust object detector with support for unsupervised learning," *arXiv:1806.10787*, 2018.

[17] A. Dalmia, S. Joshi, R. Singh, and V. Raykar, "Styling with attention to details," *CoRR*, vol. abs/1807.01182, 2018.

[18] A. Jain, Y. Gupta, P. K. Singh, and A. Rajan, "Understanding fashionability: What drives sales of a style?," *CoRR*, vol. abs/1806.11424, 2018.

[19] T. Nakamura and R. Goto, "Outfit generation and style extraction via bidirectional lstm and autoencoder," *arXiv:1807.03133*, 2018.

[20] K. Yamaguchi, T. L. Berg, and L. E. Ortiz, "Chic or social: Visual popularity analysis in online fashion networks," in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 773–776, ACM, 2014.

[21] W. H. Lin, K. Chen, H. Chiang, and W. Hsu, "Netizen-style commenting on fashion photos: Dataset and diversity measures," *CoRR*, vol. abs/1801.10300, 2018.

[22] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 507–517, International World Wide Web Conferences Steering Committee, 2016.

[23] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, "Where to buy it: Matching street clothing photos in online shops," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3343–3351, 2015.

[24] "Fashion industry-statistics, trends & strategy, https://www.shopify.com/enterprise/ecommerce-fashion-industry."

[25] "Fashion mnist dataset : https://github.com/zalandoresearch/fashion-mnist."

[26] "Fashion ai dataset, http://fashionai.alibaba.com/datasets/."

[27] Y. Ge, R. Zhang, L. Wu, X. Wang, X. Tang, and P. Luo, "A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images,"

[28] "Paperdoll dataset: http://vision.is.tohoku.ac.jp/ kyamagu/research/paperdoll/."

[29] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2014.

[30] H. Talebi and P. Milanfar, "Nima: Neural image assessment," *IEEE transactions on image processing*, vol. 27, no. 8, pp. 3998–4011, 2018.

[31] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*, pp. 483–499, Springer, 2016.

[32] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3686–3693, 2014.

[33] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," in *The IEEE International Conference on Computer Vision*, vol. 2, 2017.

[34] H. Zhang, H. Ouyang, S. Liu, X. Qi, X. Shen, R. Yang, and J. Jia, "Human pose estimation with spatial contextual information," *arXiv:1901.01760*, 2019.

[35] X. Wang, A. Shrivastava, and A. Gupta, "A fast r-cnn: Hard positive generation via adversary for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2606–2615, 2017.

[36] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1440–1448, 2015.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2961–2969, 2017.

[38] 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*, pp. 21–37, Springer, 2016.

[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[40] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the european Conference on Computer Vision*, pp. 734–750, 2018.

[41] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, Nov. 2004.

[42] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition - Volume 1 - Volume 01*, CVPR '05, (Washington, DC, USA), pp. 886–893, IEEE Computer Society, 2005.

[43] H. Chen, Z. J. Xu, Z. Q. Liu, and S. C. Zhu, "Composite templates for cloth modeling and sketching," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 943–950, IEEE, 2006.

[44] A. Borras, F. Tous, J. Lladós, and M. Vanrell, "High-level clothes description based on colour-texture and structural features," in *Iberian conference on Pattern Recognition and Image Analysis*, pp. 108–116, Springer, 2003.

[45] P. Guan, O. Freifeld, and M. J. Black, "A 2d human body model dressed in eigen clothing," in *European Conference on Computer Vision*, pp. 285–298, Springer, 2010.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[47] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 397–403, 2013.

[48] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2387–2395, 2016.

[49] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 532–539, 2013.

[50] D. Ramanan and X. Zhu, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2879–2886, Citeseer, 2012.

[51] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3476–3483, 2013.

[52] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou, "A deep regression architecture with two-stage re-initialization for high performance facial landmark detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3317–3326, 2017.

[53] S. Zhu, C. Li, C.-C. Loy, and X. Tang, "Unconstrained face alignment via cascaded compositional learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3409–3417, 2016.

[54] P. Dollár, P. Welinder, and P. Perona, "Cascaded pose regression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1078–1085, IEEE, 2010.

[55] Y. Li, B. Sun, T. Wu, and Y. Wang, "Face detection with end-to-end integration of a convnet and a 3d model," in *European Conference on Computer Vision*, pp. 420–436, Springer, 2016.

[56] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724–4732, 2016.

[57] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1021–1030, 2017.

[58] Y. Wu and Q. Ji, "Robust facial landmark detection under significant head poses and occlusion," *CoRR*, vol. abs/1709.08127, 2017.

[59] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, "Adversarial posenet: A structure-aware convolutional network for human pose estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1212–1221, 2017.

[60] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1831–1840, 2017.

[61] D. Ramanan, "Learning to parse images of articulated objects," NIPS, 2006.

[62] X. Ren, A. C. Berg, and J. Malik, "Recovering human body configurations using pairwise constraints between parts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Volume 1*, vol. 1, pp. 824–831, IEEE, 2005.

[63] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3681, 2013.

[64] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Strong appearance and expressive spatial models for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3487–3494, 2013.

[65] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv:1312.4400*, 2013.

[66] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, 2014.

[67] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Advances in neural information processing systems*, pp. 1799–1807, 2014.

[68] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *arXiv:1812.08008*, 2018.

[69] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2334–2343, 2017.

[70] S. Yan, Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Unconstrained fashion landmark detection via hierarchical recurrent transformer networks," in *Proceedings of the 25th ACM International Conference on multimedia*, pp. 172–180, ACM, 2017.

[71] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

[72] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[73] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[74] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[75] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[76] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, pp. 379–387, 2016.

[77] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

[78] Z. Cai and N. Vasconcelos, "Cascade R-CNN: high quality object detection and instance segmentation," *CoRR*, vol. abs/1906.09756, 2019.

[79] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, 2017.

[80] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv:1701.06659*, 2017.

[81] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "Automatic ship detection based on retinanet using multi-resolution gaofen-3 imagery," *Remote Sensing*, vol. 11, no. 5, p. 531, 2019.

[82] L. Tychsen-Smith and L. Petersson, "Denet: Scalable real-time object detection with directed sparse sampling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 428–436, 2017.

[83] X. Wang, K. Chen, Z. Huang, C. Yao, and W. Liu, "Point linking network for object detection," *arXiv:1706.03646*, 2017.

[84] A. Newell and J. Deng, "Pixels to graphs by associative embedding," in *Advances in neural Information Processing Systems*, pp. 2171–2180, 2017.

[85] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017.

[86] Gussian Blur, "Gaussian blur — Wikipedia, the free encyclopedia."

[87] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *The european Conference on Computer Vision*, September 2018.

[88] "L2 loss:
https://letslearnai.com/2018/03/10/what-are-l1-and-l2-loss-functions.html."

[89] "Binary cross entropy loss:
https://gombru.github.io/2018/05/23/cross_entropy_loss/."

[90] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," pp. 319–345, 1999.

[91] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

[92] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1026–1034, 2015.

[93] R. Olivier and C. Hanqiang, "Nearest neighbor value interpolation," *arXiv:1211.1768*, 2012.

[94] P. Li, "Optimization algorithms for deep learning."

[95] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on communications*, vol. 31, no. 4, pp. 532–540, 1983.

[96] B. Graham, "Fractional max-pooling," *CoRR*, vol. abs/1412.6071, 2014.

[97] N. H. Sulaiman and D. Mohamad, "A jaccard-based similarity measure for soft sets," in *2012 IEEE Symposium on Humanities, Science and Engineering Research*, pp. 659–663, IEEE, 2012.

[98] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results." http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[99] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the european Conference on Computer Vision*, pp. 466–481, 2018.

[100] T. Dao, A. Gu, A. J. Ratner, V. Smith, C. D. Sa, and C. Ré, "A kernel theory of modern data augmentation," *CoRR*, vol. abs/1803.06084, 2018.