Hierarchical Part-based Disentanglement of Pose and Appearance

by

Farnoosh Javadi Fishani

Bachelor of Software Engineering, Sharif University of Technology, 2018

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

December 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Hierarchical Part-based Disentanglement of Pose and Appearance

submitted by **Farnoosh Javadi Fishani** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

Examining Committee:

- James J. Little, Professor, Department of Computer Science, University of British Columbia
  Supervisor
- Helge Rhodin, Assistant Professor, Department of Computer Science, University of British Columbia
  Co-supervisor
- Kwang Moo Yi, Assistant Professor, Department of Computer Science, University of British Columbia
  Additional Examiner

Abstract

Landmarks and keypoints are an important intermediate representation for image understanding and reconstruction. Although many supervised approaches exist, many of them require labels of the target domain, which exist for humans, but only for sparse keypoints and not for the breadth of object and animal classes present in our rich world. We propose a self-supervised approach for discovering landmarks from unstructured image collections by disentangling pose and appearance of object parts. In particular, we propose a hierarchical structure that helps to find more meaningful keypoint locations. We demonstrate that our simplifications and hierarchical extensions of prior work are effective quantitatively and qualitatively in 2D keypoint estimation and image modification operations tasks. Our approach eases the discovery of objects and their parts in domains for which no labeled data exist and thereby eases downstream tasks, such as behaviour classification for neuroscience applications, and intuitive image editing.

Lay Summary

Learning image representations is a fundamental task in Computer Vision, affecting quality of many other tasks such as object detection, motion transfer, image to image translation and pose estimation. A problem with common entangled representations is their lack of interpretability as they are usually output of a black-boxed neural network. Therefore, learning to disentangle and represent the independent latent characteristics of objects is of a high importance to enforce interpretability to image representations. Pose, appearance and object parts are some of these latent factors that that we are interested to encode in this work. We propose a new method, called HPD (Hierarchical Part-based Disentanglement), for learning structured object parts alongside with disentangling their spatial and appearance factors. Training needs no annotations or prior knowledge on any of the factors or object classes, and can be applied to any image dataset without any limitations. We demonstrate that our model provides an interpretable latent space that can be used for selective image modification combining pose and appearance from different images to synthesize novel images. In addition, the learned part-based representations could be used for the task of landmark detection where no annotations are available. We qualitatively and quantitatively evaluate the effectiveness of our model on two sets of data.

Preface

The entire work presented here is original work done by the author, Farnoosh Javadi, under the supervision of James J. Little and Helge Rhodin.
The first author is completely responsible for implementation of the model and the experiments. Prof. Little and Prof. Rhodin provided feedback continuously during each step.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my sincere gratitude to people who have supported me and helped me during my Master's.

First and foremost, I owe my deepest gratitude to my supervisors Prof. James J. Little and Prof. Helge Rhodin who have guided me throughout my Master's journey. Dr. Little's positive attitude, support, guidance and encouragement have always motivated me throughout the way especially whenever I got stuck with any disappointing problem. Dr. Rhodin's passion for exploring new ideas, vital feedback, and invaluable insights have always been inspiring and made this research possible. I could have never imagined better mentors for my first research experience and I will always be grateful for a great privilege and honor to work with them both. I would also like to thank Prof. Kwang Moo Yi for taking the time to be the second reader for this thesis.

I like to extend my thanks to my co-workers Ling Mei and Tim Straubinger for their technical advice and sharing their experiences which improved this research. Thanks should also extend to all my colleagues and lab-mates for their constructive feedback on this project. It has been a pleasure to study and learn alongside with all of the talented and enthusiastic members of UBC's Computer Vision group.

I would also like to thank my parents for their unconditional passion and support throughout my whole life. I would have never been in this place without their countless sacrifices to provide me the opportunities I have had in my life.

Last but not least, I want to thank all my amazing friends who have made my life more pleasant and joyful. Special thanks to my dearest, Hassan, who have always been there for me, and Setareh and Saeid who not only were my friends but a family in Vancouver.

Chapter 1

Introduction

Building algorithms that make sense of an image by decomposing it into parts and their appearance [35] is not only an intellectual endeavor but has practical utility. The position of objects, animals, persons, and their parts is an essential building block for automated behaviour analysis in neuroscience, it yields performance indicators in sports and medicine, and serves as control points for image editing. For understanding images of humans, supervised approaches are trained on massive datasets curated with millions of pose annotations [4, 17]. However, these do not generalize well to new settings where annotations are scarce, e.g., to animals or persons with unusual apparel.

To make automated pattern recognition adaptable to new domains, we propose an algorithm for disentangling images of humans and animals into the spatial location of their parts (pose) and their size (shape) and color (appearance) by using a new hierarchical formulation. Our structured representations are designed to provide additional meaning to the discovered parts and leads to more reliable localization.

The recent machine learning and computer vision literature is rich of methods with disentangling images into different factors of variation [14, 38, 43, 45]. We follow the stream of methods that use an auto-encoder framework with a structured latent space to explicitly separate pose, shape, and appearance into factored latent encodings. Because the number of parts is limited and their embedding is low-dimensional with localized spatial support, the encoder and decoder learn jointly

to attend to the most important image parts that occur frequently in the training set. For human pictures, this results into separate parts being assigned to torso, limbs and head. While no one-to-one correspondence between anatomical and discovered body parts can be enforced without additional supervision, in a self-supervised approach, the learned representations can be used to track the position of a part in a video or to synthesize new appearances by re-combining parts and appearances from different pictures.

Existing approaches are difficult to train since multiple loss functions and custom neural network layers need to be combined to facilitate disentanglement [10, 35, 38]. Our contributions are two-fold. First, we simplify an existing self-supervision pipeline by replacing an equivariance loss, first introduced by Lenc et al. [31] and used by similar methods for disentangled representation learning [35, 59], with a randomized switch in the control flow, thereby eliminating one of the necessary loss terms, frees the model from further tuning of training objective. We also reduce the model complexity by showing that a simpler encoder suffices in almost all application scenarios. Second, and most importantly, we enforce a child-parent relationship to the parts at training time. This hierarchy gives the discovered pose a new semantic level and supports the training of more fine-grained parts without adding considerable number of parameters to the model.

We evaluate our approach on the task of unsupervised landmark detection, image reconstruction and image editing on two datasets. We demonstrate that our contributions lead to accurate part localization and use established fashion datasets to showcase the editing capabilities qualitatively.

Chapter 2

Related Work

Our approach builds upon ideas from general representation learning, from learned pose, landmark, and shape models, as well as from existing hierarchical representations, for which we review the most related ones in the following.

2.1 Disentangled Representation Learning

Disentangled representation learning has become a hot topic in Computer Vision, as it adds interpretability to the black-boxed representations we get from the common image encoding methods [27, 47, 50]. The disentangled representations give us a sense of what factors of images a neural network thinks as the most important ones, and how it separates them. More importantly disentangled representations enable synthesizing novel images or image modifications by fixing some factors and changing the rest. Mathieu et al. [38] proposed a conditional generative model trained with adversarial loss to disentangle the hidden factors within a set of labeled observation. Tran et al. [52] and Peng et al. [42] disentangle pose and identity of human faces in a supervised manner. On the other hand, there are some GAN-based methods [9, 48] and VAE-based methods [18, 19, 28, 29, 56] for learning disentangled representations in a completely unsupervised manner. All the mentioned methods enforce disentanglement directly into the design of models. On the other hand, Rombach et al. [43] factorizes learned representations of existing models. Similar to these papers, we also learn disentangled representations; in

contrast we do not use adversarial loss or VAE objective which are challenging to train. Instead, we propose a non-variational auto-encoder framework for learning disentangled representations, trained in an unsupervised manner, only using the reconstruction loss that leads to a stable training process in comparison with GAN-based methods.

2.2 Pose and Appearance Disentanglement

Pose (shape) and appearance are two of the most important factors of images as they define the object spatially and semantically, and recently a lot of research has been done on their disentanglement. To learn disentangled pose and appearance, the models generally condition generative models on pose, shape or keypoints information. Many of them commonly assume the availability of pose or keypoints which are extracted by a pretrained pose or keypoints detector [1, 6, 13, 14, 36, 37, 45]. Although, these models work well, the constraint of observable pose and keypoints is strong that makes these models just applicable when a pre-trained pose detector is available for the domain like human bodies and faces. However, our model does not need any prior knowledge about the pose, shape, and keypoints. It discovers some parts that reflect visual concepts of images in a self-supervised manner. Therefore, it can be applied on any arbitrary-complex domain. Some other methods need multiple frames from videos [10, 22, 23] or pairs of images different in just one factor and the same in rest [49], specifically pairs of images from a single object but with different poses [15]. Such datasets are hard to obtain. In contrast, our model only needs single images, and at the same time it can also be generalized for videos by getting video frames as the input.

2.3 Discovery of Object Parts and Landmarks

Object parts or landmarks are one of the most important and common intermediate representations in computer vision, as they break objects into a set of meaningful components that focus on key regions of an object. Many approaches learn landmarks and parts in discriminative tasks such as image classification [5, 26, 30, 46], and pose estimation [8, 57]. The parts that are learned in a discriminative task emerge based on their semantic relation to the object and are optimized to work

best for that task and do not necessarily encode any information about the appearance, shape or pose. In contrast to these methods, our model learns the parts in an image modeling task in a way that for each part we learn the disentangled pose and appearance. The closest approach to ours is Lorenz et al. [35], which similarly learns the disentangled shape and appearance of the parts in an auto-encoder framework trained with the equivariance, adversarial and reconstruction loss. However, firstly, our model is just trained with the reconstruction loss which is more robust and easier to optimize and more importantly it learns the parts in a hierarchical manner that we show leads to predicting more meaningful parts.

2.4 Hierarchical Representation Learning

Our hierarchical approach is closely related to the methods that learn structure of objects or the hierarchy of object parts [39]. Some of these methods need supervision to learn the hierarchy of parts. Grass model [32] needs prior knowledge in terms of segmentation of each shape into parts and StructureNet [40] needs semantic labels for each part. In contrast, our work is totally self-supervised without needing any sort of annotations. We assume a predefined hierarchy for the parts and enforce it to the model. Some of the unsupervised approaches for inferring the hierarchy and structure of parts take use of motion by looking at the videos and analyzing it [16, 21, 25, 55]. Unlike these approaches, we pre-define the hierarchy of parts as a binary tree structure and train it using only single images, without needing multiple frames and timestamps from videos.

Esmaeili et al. [12] proposes an unsupervised approach that leverages a two-level hierarchical objective to disentangle independent axes of variation of data by introducing a variational auto encoder framework. The disentanglement happens in a latent space without explicit spatial separation, which, however, is required for the intuitive editing and landmark discovery we target. Most relevant to ours, Paschalidou et al. [41] learns the primitive parts and their hierarchical structure from single images in the form of a binary tree. But unlike ours, they require 3D mesh models for supervision.

Chapter 3

Method

At a high level, our method learns to reconstruct an image from its pose and appearance, where the pose is extracted from an appearance transformed image and the appearance from a pose transformed image. This approach is similar to some recent methods [11, 28, 35], which we extend in the following with simplifications and a hierarchical extension.

As an intuition let us assume we have a triplet of images (x, x_1, x_2) where x and x_1 share the same appearance, and x and x_2 share the same pose. We also assume that each image can be generated by a decoding function D given its pose denoted by $\Phi^{pose}(x)$ and appearance $\Phi^{app}(x)$. We write,

$$x = D(\Phi^{app}(x), \Phi^{pose}(x)). \tag{3.1}$$

As x and x_1 share the same appearance, we have $\Phi^{app}(x) = \Phi^{app}(x_1)$, and similarly for pose we have $\Phi^{pose}(x) = \Phi^{pose}(x_2)$. Hence, we can rewrite Eq. 3.1 as:

$$x = D(\Phi^{app}(x_1), \Phi^{pose}(x_2)). \tag{3.2}$$

In other words, the image x can be reconstructed by the appearance of x_1 and pose of x_2. But the problem is that a dataset including triplets having mentioned constraints is hard to obtain, and it is not always available. In our method, we construct x_1 and x_2 by applying spatial transformation T_s and appearance transformation T_a on x, respectively.

3.1 Problem Formulation

Similar to Lorenz et al. [35], our goal is to learn a representation for an image x, denoted by $\Phi(x)$ which factorizes the image into its forming parts:

$$\Phi(x) = (\Phi_1(x), \Phi_2(x), \ldots, \Phi_k(x)), \tag{3.3}$$

where k denotes number of parts and $\Phi_i(x)$ is the representation of i^{th} part of image x. Furthermore, we want to disentangle the appearance and pose of each part. Hence, the part representation $\Phi_i(x)$ needs to be a combination of its appearance $\Phi_i^{app}(x)$ and $\Phi_i^{pose}(x)$, as in the following formula:

$$\Phi_i(x) = (\Phi_i^{app}(x), \Phi_i^{pose}(x)). \tag{3.4}$$

We know that the appearance representation should be invariant to change in appearance. Accordingly, if we apply spatial transformation T_s on x we should have $\Phi^{app}(x) = \Phi^{app}(T_s(x))$. And similarly the pose representation should be invariant to change in pose, in a way that: $\Phi^{pose}(x) = \Phi^{pose}(T_a(x))$, where T_a depicts an appearance transformation. In conclusion, by applying invariance constraints, we can rewrite the representation of each part (Eq. 3.4) as:

$$\Phi_i(x) = (\Phi_i^{app}(T_s(x)), \Phi_i^{pose}(T_a(x))), \tag{3.5}$$

which justifies using the appearance transformed images and spatially transformed images in our pipeline. Eq. 3.5 is similar to Eq. 3.2, but x_1 and x_2 are not given, instead we construct them by applying pose and appearance transformations on a single image x. Finally, the image representation $\Phi(x)$ would be an assembly of pose and appearance of its parts, respectively extracted from the appearance transformed image and the pose transformed image. Hence, Eq. 3.3 can be written as:

$$\Phi(x) = [(\Phi_i^{app}(T_s(x)), \Phi_i^{pose}(T_a(x)))]_{i=1}^{k}, \tag{3.6}$$

where $[\,]$ notation denotes the set of parts.

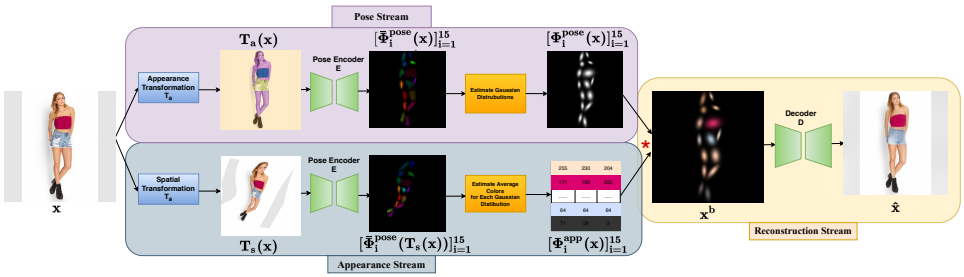

Figure 3.1: Pipeline of PD. The pose stream starts from the appearance transformed image $T_a(x)$, from which the encoder predicts a set of k activation maps $\bar{\Phi}_i^{pose}(x)$. Then, the Gaussian distributions $\Phi_i^{pose}(x)$ which represents parts shape are estimated as the way explained in Sect. 3.2.1. The appearance stream starts from giving the spatially transformed image $T_s(x)$ to the encoder for predicting part activation maps of that. This path continues by estimating the appearance vector $\Phi_i^{app}(x)$ for each part. In the reconstruction stream, first each part is multiplied by its color. Then, the blobby image x^b is created by Eq. 3.10. Finally, x^b is given to the decoder to reconstruct image x.

3.2 Baseline Model

In this section, we introduce the first version of our model termed as Part-based Disentanglement (PD) and explain our auto-encoder framework for encoding images into pose and appearance disentangled parts, where all the parts are in the same level. An overview of PD is shown in Fig. 3.1. For designing PD, we start by simplifying Lorenz et al. [35] and predict object parts in a flat, unstructured representation. We subsequently extend it by adding a hierarchical extension that arranges the predicted parts in a binary-tree hierarchical structure, in a way that at each level of hierarchy each parent part is broken into two finer, more detailed children parts. We name this version of our model HPD (Hierarchical Part-based Disentanglement).

3.2.1 Pose Stream

The goal of the pose stream is to predict a set of parts for a given image, in terms of a set of 2D Gaussian distributions which represents the shape and pose of the input image, denoted by $\Phi^{pose}(x) = [\Phi_i^{pose}(x)]_i$. In this stream, we start by applying

an appearance transformation T_a on the input image x to enforce pose representations invariance to change in appearance. The appearance transformation T_a simply changes the color of the image as a change in the appearance and we discuss its details in Sect. 3.4. The appearance transformed image is depicted by $T_a(x)$. A deeplabv3 [7] encoder $\mathbf{E}$ predicts a multi-channel activation map that encodes each part as a channel, shown by $\bar{\Phi}_i^{pose}$. Deeplabv3 performs well for image semantic segmentation task, which makes it suitable for our model. Because in this stream the pose encoder wants to assign each pixel to only one object part which is similar to image segmentation task.

Predicting activation maps from $T_a(x)$ and not directly x, prevents the pose stream from observing any information about the appearance of the original image, therefore it helps the disentanglement of pose. We assume that each object part has the spatial footprint of a multivariate normal distribution computed by Eq. 3.7.

$$\mathcal{N}(x;\mu,\mathbf{\Sigma}) = \exp\left(-\frac{1}{2}(x-\mu)\mathbf{\Sigma}^{-1}(x-\mu)^T\right), \tag{3.7}$$

where μ is the mean vector and $\mathbf{\Sigma}$ is the covariance matrix. Hence, we can estimate a Gaussian distribution $\Phi_i^{pose}(x)$ from each activation map. We compute Gaussian parameters μ and $\mathbf{\Sigma}$ as the weighted mean of the normalized activation map and the covariance matrix, respectively. μ can be interpreted as the center point of object part and $\mathbf{\Sigma}$ specifies the part's direction and size. Considering parts as Gaussians and not explicitly as a set of activation maps enforces a kind of regularization into the model that helps it to predict more localized and meaningful parts, specifically when activation maps are noisy and scattered. At the end of this stream, we end up having k 2D Gaussian distributions, denoted by $[\Phi_i^{pose}(x)]_{i=1}^k$, each representing one region of the image, and represents pose of the image, $\Phi^{pose}(x)$, as a whole.

3.2.2 Appearance Stream

The goal of this stream is to predict an appearance vector $\Phi_i^{app}(x)$ for each of the detected parts, $\Phi_i^{pose}(x)$ in Sect. 3.2.1. The set of appearance vectors, denoted by $[\Phi_i^{app}(x)]_{i=1}^k$ act as an appearance representation $\Phi^{app}(x)$ for the given image x, where k denotes the number of parts. In this stream, we start with applying

a spatial transformation T_s on image x which deforms object's pose, and makes appearance representation invariant to change of pose and shape. We explain the details of the T_s in Sect. 3.4. $T_s(x)$ denotes the spatially transformed image. Using the same encoder $\mathbf{E}$ as Sect. 3.2.1, firstly, we predict a set of activation maps for $T_s(x)$, depicted by $\bar{\Phi}_i^{pose}(T_s(x))$. Note that, the appearance vectors predicted from image x and image $T_s(x)$ should be the same, as the appearance encoding method needs to be invariant to object spatial deformations. This justifies predicting the appearance vectors from the spatially transformed image $T_s(x)$ and not the original image. Secondly, we estimate Gaussian distributions as the same way as described in Sect. 3.2.1, denoted by $\Phi_i^{pose}(T_s(x))$. Unlike Lorenz et al. [35] that predicts d-Dimensional appearance vectors by a separate appearance encoder, our PD uses the 3-channel RGB color of pixels as an appearance feature map. This technique simplifies the model and reduces the number of model's parameters significantly in comparison with the methods with more complex encoders, although makes the model limited to reconstructing simple appearances. To predict a 3-channel color for each part, we average pool colors of the spatially transformed image at all locations where part i has positive activation distribution, written as:

$$\Phi_i^{app}(x) = \frac{\sum_{u=1}^{width}\sum_{v=1}^{height} T_s(x).\Phi_i^{pose}(T_s(x))[u,v]}{\sum_{u=1}^{width}\sum_{v=1}^{height} \Phi_i^{pose}(T_s(x))[u,v]}, \tag{3.8}$$

where $[u,v]$ denotes the pixel location. Eq. 3.8 simply assigns the average color of each image region to the part representing that region. At the end of this stream, we would have k 3-channel RGB colors, individually denoted by $\Phi_i^{app}(x)$ each corresponds to one part $\Phi_i^{pose}(x)$, that represents the appearance of x as a whole: $\Phi^{app}(x) = [\Phi_i^{app}(x)]_{i=1}^k$.

3.2.3 Reconstruction Stream

The goal of this stream is reconstructing image x, given the set of parts $[\Phi_i^{pose}(x)]_{i=1}^k$ and their appearances $[\Phi_i^{app}(x)]_{i=1}^k$. To this end, we combine the pose representation Φ^{pose} explained in Sect. 3.2.1, and the appearance representation Φ^{app} described in Sect. 3.2.2. As written in Eq. 3.9, we multiply each 2D Gaussian distri-

bution by its corresponding 3-channel RGB color to achieve colorful 2D Gaussian distributions referred as *blobs*, where each blob represents a visual component of image x, denoted by $\Phi_i(x)$.

$$\Phi_i(x) = \Phi_i^{app}(x) \cdot \Phi_i^{pose}(x), \tag{3.9}$$

where $\odot$ operation denotes element-wise multiplication. Note that, the appearance representations are 3-d and the pose representations are $|W| \times |H|$, where $|W|$ and $|H|$ are image width and height respectively. Therefore, at first we tile these representations to the size of $3 \times |W| \times |H|$ to be compatible for elementwise-multiplication.

For example, as shown in Fig. 3.2, for the human body, the colorful blobs would be located at arms, head, torso, etc. Afterwards, we create a single RGB image, depicted by x^b, from the set of k colorful blobs to enforce more regularization to the model. x^b is created by taking the maximum of all blobs as written in Eq. 3.10, and it is the sole input of the decoder *mD*. Giving the max of all Gaussian distributions, x^b, to the decoder instead of all of them implicitly activates only one Gaussian distribution for reconstructing each image region. Therefore, in practice, only one Gaussian distribution incorporate in reconstructing one image region, and the model is able to ignore some of the unnecessary information or noise of the other Gaussian distributions. In addition, taking the max of all 2D Gaussian distributions makes them not overlap, not having two different activation maps active for the same body region. Otherwise, just one of the identical or similar Gaussian distributions would be visible in x^b, incorporating to the reconstruction which is a waste for the model.

$$x^b = \max\Big[\Phi_i(x)[u,v]\Big], \tag{3.10}$$

where $i = 1 \dots k$ and $\Phi_i(x)[u,v]$ denotes the value of the 2D Gaussian distribution i ($\Phi_i(x)$) at location $[u,v]$.

A U-net [44] decoder $\mathbf{D}$ is used to reconstruct image x from x^b. The reconstructed image is denoted by: $\hat{x} = \mathbf{D}(x^b)$. By minimizing the l_2-norm of x and $\hat{x}$, the model learns to concentrate on object parts that are unambiguous and more

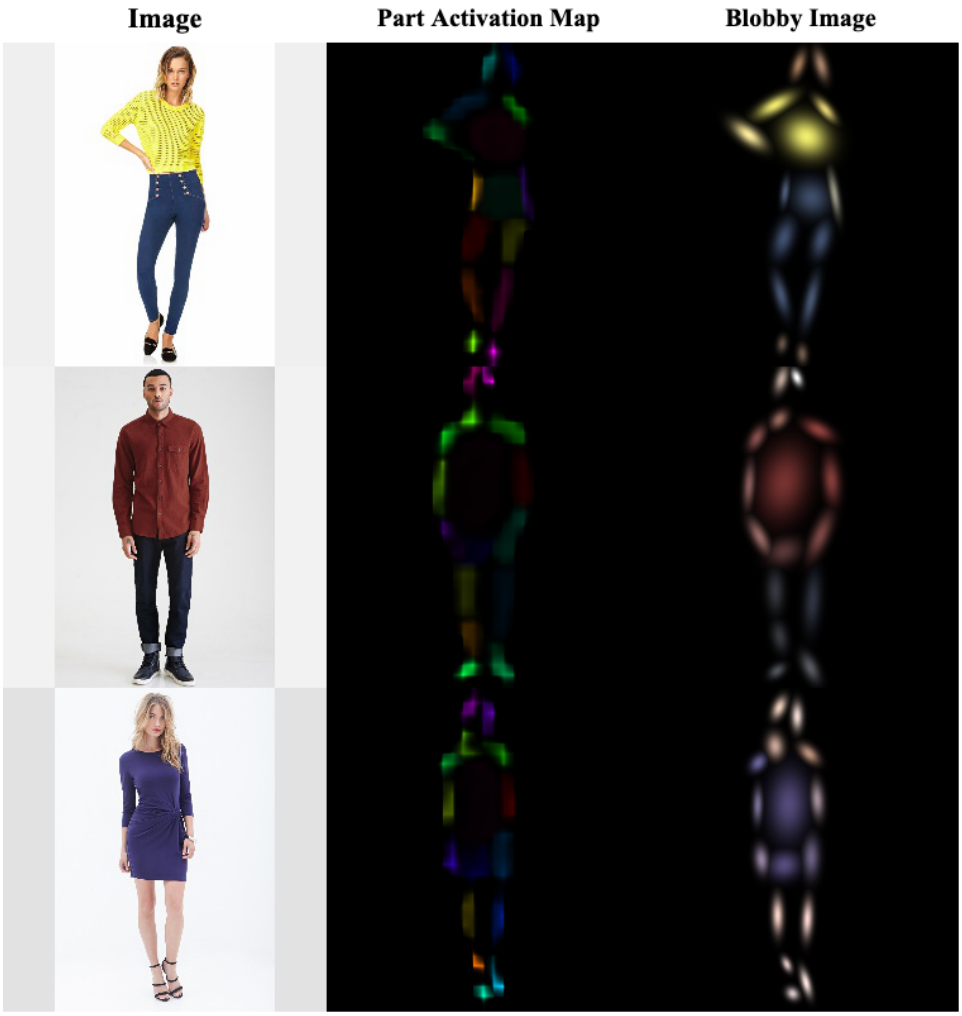

Figure 3.2: Blobs (colorful Gaussian distributions) The first column shows sample images, the second column shows the part activation maps predicted by the encoder **E**, and the last column shows the colorful Gaussian distributions which we refer as blobs. For full body human images the blobs are located at torso, feet, legs, shoulders, etc.

important for reconstruction. To achieve sharper images, it is a standard technique to train the model with a combination of reconstruction loss and adversarial loss [1, 35]. However, using adversarial loss needs defining a new adversarial task

and training a discriminator. As our task is not image generation, we preferred to use a combination of l_2-norm and perceptual loss that produces pleasing results, following Xu et al. [54] and Jakab et al. [22]. Instead of comparing raw pixel values directly, the perceptual loss, first proposed by Johnson et al. [24], compares features extracted from multiple layers of a deep network, which in our case is VGG-16. Our final objective function is:

$$l_{\text{rec}} = \|x - \hat{x}\|_2 + \beta l_{\text{perc}}(x, \hat{x}), \tag{3.11}$$

where β denotes the weight of the perceptual loss.

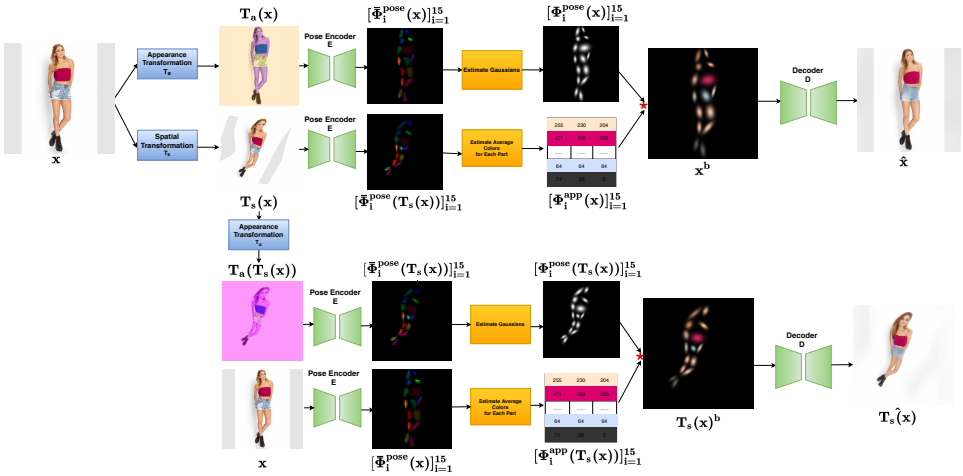

Figure 3.3: Swapping Technique. Using swapping technique, at each training step, we either follow the upper path or the lower path, which their only difference is that the role of image x and transformed image $T_s(x)$ is swapped in the lower path, but both of the paths follow the same process shown in Fig. 3.1.

3.2.4 Swapping Technique

We expect the pose encoder mE to track the object shape consistently, predicting a consistent set of parts even under object deformations. That means the predicted parts from the spatially transformed image need to be the transformed version of

parts extracted from the original image, which can be written as:

$$\Phi_i^{pose}(T_s(x)) = T_s(\Phi_i^{pose}(x)). \tag{3.12}$$

This equivariance constraint was first proposed by Lenc et al. [31], and forces the encoder **E** to capture consistent object parts when the shape changes. Many similar methods [35, 51, 59] enforce this constraint by adding an additional term to the training objective named as equivariance loss. Adding the equivariance term to the loss function needs further tuning of the training objective. Instead of explicitly training the model with the equivariance loss, we propose a simple trick for training the model named as swapping technique, which implicitly enforces the equivariance constraint to the model.

In this version of training, PD randomly swaps the role of x and $T_s(x)$ in each training step as shown in Fig. 3.3. As a result, half of the times in the training process we extract the parts from the appearance transformed image and predict their colors from the spatially transformed image to reconstruct the original image. Instead, half of the times we extract the parts from the appearance transformed version of the spatially transformed image and extract the part colors from the original image to reconstruct the spatially transformed image. The intuition behind this trick is that x can be interpreted as the deformed $T_s(x)$. As a result, at each training step either the original image x or the spatially transformed image $T_s(x)$ is reconstructed. This technique enables us to train only with the reconstruction loss, which leads to more robust training than the proposed objective function of Lorenz et al. [35]. An overview of the training process using the swapping technique is shown in Fig. 3.3.

3.3 Hierarchical Model (HPD)

In this section, we discuss our hierarchical extension to PD, that predicts pose and appearance disentangled parts for an object in a hierarchical manner. We start with a fixed number of parts then in next levels of hierarchy, we break each part into two children parts in a binary tree structure. As a result, if we start from k_0 parts, and have h levels of hierarchy, at the the end we end up with $(2^h - 1)k_0$ structured parts. We chose binary tree because of its simplicity, however an unbalanced non-binary tree makes more sense as some parent parts are complex and need to be broken into

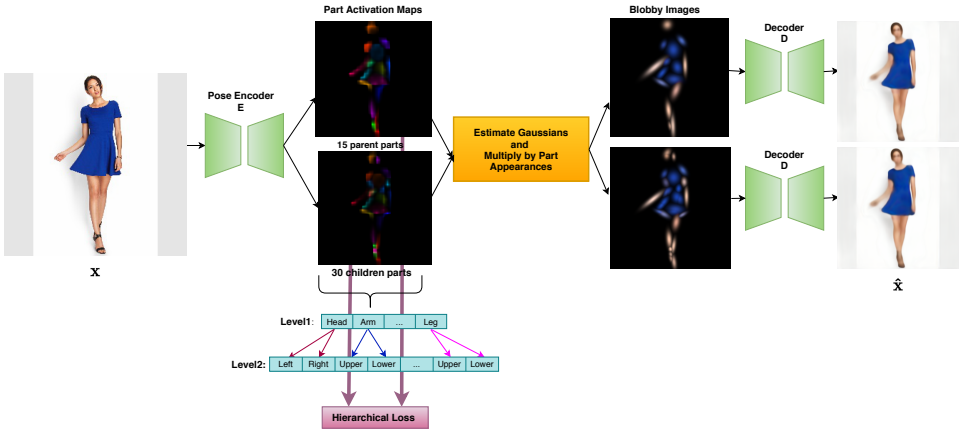

Figure 3.4: Pipeline of HPD. In this case, the pipeline starts with the pose encoder E predicting 45 activation parts. The first 15 of them (upper image) are parent activation parts and 30 are the children parts (lower image). The hierarchical loss would be computed by this activation maps following Eq. 3.13. Then, the Gaussian distributions and the appearance vectors are estimated to create the blobby images for each level of the hierarchy. The top blobby image corresponds to the first level and the bottom one corresponds to the second level of the hierarchy. Each of the blobby images is given to a decoder independently for reconstructing the image.

more than two children parts, and in contrast some parts are too simple and do not even need to be broken in further levels. But it is harder to enforce that structure to the network. In Sect. 4.3, we demonstrate that the structured object parts extracted by HPD are finer and more accurate which leads to better image reconstruction. An example of the HPD model architecture consisting of two levels of hierarchy, having 15 parts at the first level and 30 in the second level is shown in Fig. 3.4. The children parts are learned dependent on the location of their parents, as we force their center points to be close to the center point of their parents. We explain the details of the model in the following sections.

3.3.1 Pose and Appearance Stream

The goal of the pose stream is to learn structured pose and appearance disentangled object parts, that represent image pose as a whole $\Phi^{pose}(x)$. As in Sect. 3.2.1, we

start with applying T_a on x. Then, HPD's encoder $\mathbf{E}$ predicts $(2^h-1)k_0$ activation maps from $T_a(x)$ instead of k activation maps we had in PD. h denotes the total level of hierarchy and k_0 denotes the number of starting parts. We could have used a separate encoder for predicting activation maps of each hierarchy level, but to keep the model light-weight, we used one same encoder $\mathbf{E}$ for predicting all the activation maps, which achieved the same performance. Then, we estimate a 2D Gaussian distribution from each activation map, denoted by $\Phi_i^{pose}(x)$. But the important point is that we wish to enforce a binary-tree structure to the parts, in a way that each parent part has exactly two children parts. It is obvious that children of each parent part should be close to each other and to their parent, however not too close to present overlap. We enforce this constraint by minimizing the euclidean distance of center points of children parts and the parent's center point. We denote this term *hierarchical loss*. The center point of part i is the μ parameter of the i^{th} 2D Gaussian distribution. Thus, the hierarchical loss can be written as follows:

$$l_{\text{hrc}} = \sum_{r=1}^{h}\sum_{s=1}^{k_0}(\mu_{(2^{r-1}-1)k_0+s} - \mu_{(2^r-1)k_0+2s-1})^2 + (\mu_{(2^{r-1}-1)k_0+s} - \mu_{(2^r-1)k_0+2s})^2, \tag{3.13}$$

where r, s, and k_0 denotes the level of the hierarchy, part number in the hierarchy, and the initial number of parts, respectively. The index of s^{th} part in the hierarchy level r is $(2^{r-1}-1)k_0+s$, and parts number $(2^r-1)k_0+2s-1$ and number $(2^r-1)k_0+2s$ correspond to its children. At the end of this stream, we end up with $(2^h-1)k_0$ structured 2D Gaussian distributions, denoted by $[\Phi_i^{pose}(x)]_{i=1}^{k}$, each focusing on one image region, representing the pose of x as a whole $\Phi^{pose}(x)$. Note that the encoder $\mathbf{E}$ does not have a hierarchical design but it outputs activation parts that have a hierarchical structure.

The appearance stream works in the same way as explained in Sect. 3.2.2, but it predicts $(2^h-1)k_0$ 3D appearance vectors $\Phi_i^{app}(x)$ for each $\Phi_i^{pose}(x)$. Again, the set of the appearance vectors $[\Phi_i^{app}(x)]_{i=1}^{k}$ are considered as appearance representation $\Phi^{app}(x)$ of the given image x.

3.3.2 Reconstruction Stream

For reconstructing the original image, we combine pose and appearance representation of parts using Eq. 3.9. But, instead of having only one blobby x^b image in PD, we create one image for each level of hierarchy in HPD. Then, we feed each image to the decoder for reconstruction. In this way, all parts are incorporated into the reconstruction of the image exactly once, and there is no bias to parts on any level of the hierarchy. The r^{th} blobby image is created from the parts present in r^{th} level of hierarchy by the following formula:

$$x_r^b = \max \Phi_i^{app}(x) \cdot \Phi_i^{pose}(x), \tag{3.14}$$

where $r = 1 \dots h$, and $i = (2^{r-1} - 1)k_0 + 1 \dots (2^r - 1)k_0 + 1$. We feed each x_r^b to the U-net decoder $\mathbf{D}$ for reconstructing the image $\hat{x}$. Hence, we have $\hat{x} = \mathbf{D}(x_r^b)$. As a result, we end up with h reconstructed images that need to be incorporated in the loss function equally. The reconstruction loss can be rewritten as:

$$l_{\text{rec}} = \frac{1}{h} \sum_{r=1}^{h} \|x - \hat{x_r}\|_2 + l_{\text{perc}}(x, \hat{x}_r). \tag{3.15}$$

The final objective function is a combination of the reconstruction loss 3.15 and the hierarchical loss 3.13 written as:

$$l_{\text{final}} = l_{\text{rec}} + \alpha_0 l_{\text{hrc}}, \tag{3.16}$$

where α_0 denotes the weight of the hierarchical loss.

3.4 Transformations

The appearance transformation T_a and the spatial transformation T_s are important parts of our model. For T_a, firstly we shift the h channel of HSV space by a random value. After this step, some colors like black and white might still stay the same. Therefore, we also mix the 3-channel RGB image with a random base color. For T_s, we use thin plate splines [2] and rotations. Each TPS can be defined by its control points. We have a set of 6 predefined control points, and to create a T_s for

each image in the dataset, we randomly assign a weight to each set and linearly combine all 6 to get the final TPS. In order to not have a bias to left or right poses, 50% of the times, we flip all control points of final TPS along x axis, which leads to mirrored TPS. Note that flipping the control points does not mirror the transformed image, but only mirrors the TPS. As the final step, we rotate the result of final TPS on image, by an arbitrary angle between $(-60, 60)$. The combination of all these steps, would lead to T_s that can deform the shape and pose of a given image.

Chapter 4

Experiments

In this section we evaluate our model in various tasks including pose and appearance transfer, image reconstruction and landmark detection. In the first section 4.1 we show the qualitative results of PD model for the task of unsupervised part detection on the DeepFashion dataset. In Sect. 4.2 we use the learned disentangled representations of PD to locally or globally transfer the pose and appearance of objects and generate novel images. In Sect. 4.3 we compare the results of PD and HPD model on the part detection and image reconstruction task. Finally, We report the qualitative and quantitative results on the CelebA dataset in Sect. 4.4.

4.1 Part Detection on DeepFashion

In this section we show the results of part and landmark detection on the DeepFashion [34] dataset, for which no pose annotations are available. In our work, we only used in-shop clothes images of DeepFashion, but only those that are full-body and from the front-view. All the images shown in this section are from the test set, which the model has not seen before. We randomly picked 10% of images as the test set.

Fig. 4.1 visualizes 15 out of 15 part activation maps of given images, spatially transformed images and appearance transformed images. Activation maps are learned in a self-supervised manner through an image reconstruction task. We

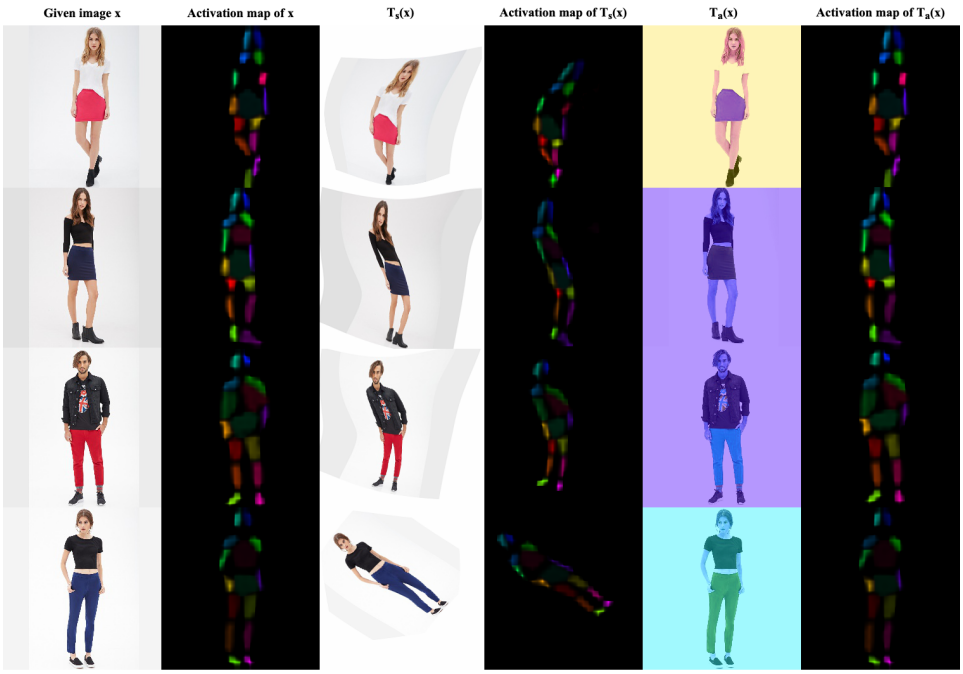

Figure 4.1: Part Detection. Visualization of 15 part activation maps for the given image x, spatially transformed image $T_s(x)$, and appearance transformed image $T_a(x)$. By comparing column 2 and 4 we can conclude that pose encoder **E** consistently track object parts even under image deformations, however the model is not explicitly trained with the equivariance loss. In addition, comparison of column 2 and 6 shows pose invariance to appearance changes.

can see from the resulting activation maps that important keypoints of humans like the wrists, torso, legs, and feet are correctly detected, even when there is a change in pose and appearance of the object. Although we do not use a term to enforce separation of part activation maps, we can see that they are automatically learned to not overlap, as it leads to lower reconstruction loss and better reconstructed images. In addition, it is seen that part activation maps of the original image x and the appearance transformed image $T_a(x)$ are almost identical which demonstrates that the pose encoder **E** is invariant to change in appearance. This means that the changes in the appearance of the person and background do not impact the detected pose. Furthermore, we can see that activation maps extracted from spatially trans-

formed images $T_s(x)$ are the spatially transformed versions of the activation maps detected from the original image. This means that, despite not explicitly using the equivariance loss proposed by Lorenz et al. [35] during the training, the model captures the equivariance consraint, by randomly switching the role of x and $T_s(x)$ for 50% of the iterations.

Fig. 4.2 shows our 15 learned 2D Gaussian distributions each acting as a part representation, and their corresponding predicted landmarks. The second column depicts all the merged part activation maps $[\bar{\Phi}_i^{pose}(x)]_{i=1}^k$, third column shows the merged Gaussian distributions $[\Phi_i^{pose}(x)]_{i=1}^k$ estimated from part activation maps, and the fourth column shows the blobby image x^b, generated by Eq. 3.10. We can see that x^b captures the overall shape of the object correctly, by combining part shapes and their appearances. We consider center points of part activation maps, which are μ parameters of Gaussian distributions as our predicted keypoints. Without any labels, our model can detect decent keypoints, especially for feet, wrist, torso, and leg regions.

We trained two versions of our model, one with l_2 as the reconstructions loss, and one with the combination of l_2 and perceptual loss (Eq. 3.11). Although, l_2 as a sole loss produces blurry images, we can see its combination with VGG-perceptual loss leads to sharper and crisper images, especially for challenging parts like faces and hair. Fig. 4.3 depicts this comparison by visualizing the reconstructed images. Again, all the images are from the test set.

4.1.1 Limitations

Although, our simplified PD model yields acceptable results for the majority of test set, there are some cases for which the model fails to predict accurate keypoints or reconstructions. We will explain these failure case scenarios in the following:

- **Complex clothing patterns:** As mentioned in Sect. 3.2.2, we do not have an appearance encoder. Instead, we take image average color over each Gaussian distribution as the appearance vector. This simplification leads to fewer

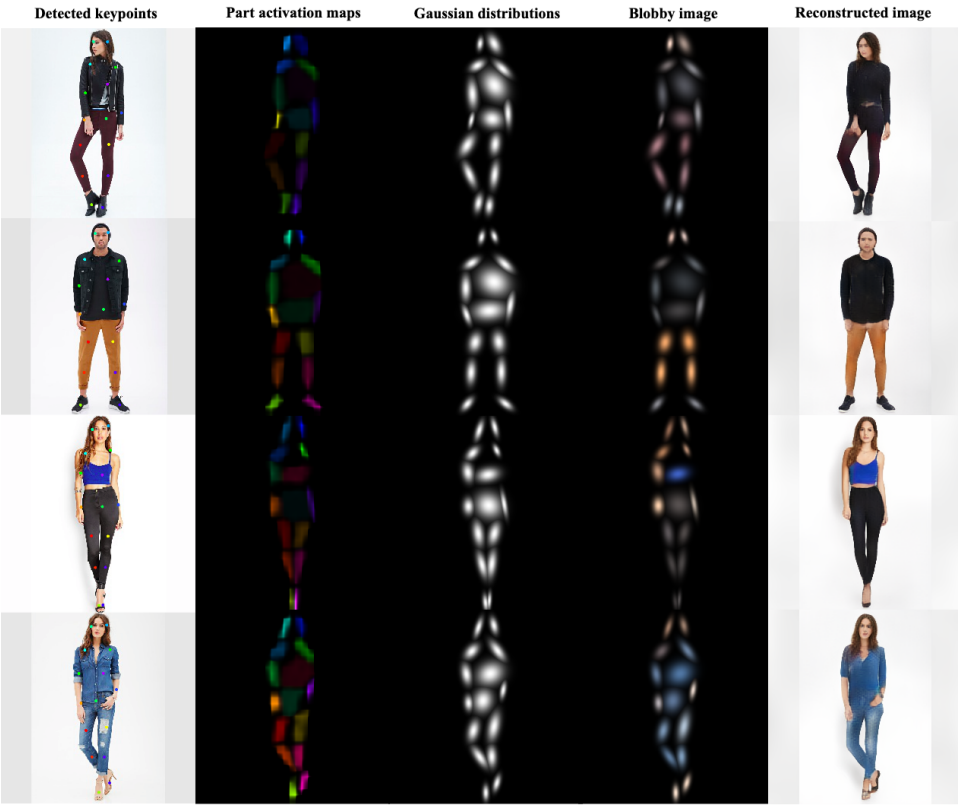

Figure 4.2: Part Detection. Visualization of detected unsupervised landmarks, part activation maps, Gaussian distributions, colorful Gaussian distributions, and reconstructed images respectively from left to right. The Gaussian distributions of column 2 are estimated from the part activation maps using Eq. 3.7, in a way that the μ of the Gaussian distributions are the center points of part activation maps, and $\mathbf{\Sigma}$ is the co-variance matrix. The blobby images shown in column 4 depicts multiplications of parts Gaussian distributions of column 3 by their corresponding color (Eq.3.10).

Figure 4.3: Image Reconstruction. Comparison of reconstructed images by using l_2 loss and perceptual loss.

Figure 4.4: Limitations of image reconstruction. The left set shows model inability to reconstruct complex patterns such as stripes and checkered patterns. The right set shows model bias to a specific skin color which has seen most in the training set. Note that, all samples are from the test set.

parameters and faster training, however makes the model unable to re-create complex clothing patterns like stripes, checkered, and polka dots. For these cases, as the model assigns one color to each part, reconstruction of clothing lacks detail, as shown in left side of Fig. 4.4. There is a trade-off between the complexity and size of the model and the amount of detail the model is able to capture. If we are interested in detailed reconstructions, a separate appearance encoder is needed to predict an n-D appearance vector per part.

- **Bias to light skin color:** Another problem is that unfortunately the dataset is not totally balanced, which causes the model to overfit to some particular colors for clothing, face and body as depicted in Fig. 4.4. Specifically, there are many images in the dataset with bare legs. Furthemore, the majority of the objects have light-color skin. Hence, sometimes in the reconstructed images we see a light-tone skin color although the given subject has a dark skin like the first row or the model mistakenly reconstruct bare legs instead of pants or boots like the other rows of Fig. 4.4.

- **White clothing** Another problem that impacts the accuracy of predited keypoints negatively is that the model mistakes the white clothing for the background. In our DeepFashion dataset, all the images have a simple whitish background. As a result, even without assigning any Gaussian distribution to the background, model can reconstruct the white background. As a result, for reconstruction, the decoder simply remembers to set white to any empty region of the blobby image. And sometimes it does the same for clothing. It does not assign any Gaussian distribution to white shirts or pants, but already knows to assign white color to empty regions of blobby image. This might lead to good reconstruction, but the extra Gaussian distributions are assigned to meaningless locations, leading to inaccurate and inconsistent keypoints. Fig. 4.5 visualizes this problem. For the first two rows, we had expected to have a Gaussian distribution assigned to the torso but instead it is assigned to somewhere close to the right arm which leads to poor keypoint prediction for torso. And similarly for the last two examples, we have inaccurate keypoints for the hips and torso.

Figure 4.5: Limitations of part detection. Visualization of detected keypoints, part activation maps and colored Gaussian distributions for samples wearing white clothing. The image shows model's failure on detecting meaningful keypoints for these cases as it mixes up white clothing with the background.

4.2 Pose and Appearance Transfer

Disentangled representations allow image modification and synthesizing new images by fixing some of the learned factors and changing the rest. In our case, the disentangled factors are parts, appearance and pose. Therefore, in this section we explain our set of experiments for evaluating the learned pose and appearance disentangled representations in the task of local or global Pose and Appearance transfer. We reported our results on the test set of DeepFashion dataset.

For synthesizing new images, we can get an object appearance and pose from different source images, then combine the new factors and generate a novel image which has the appearance of one image and the pose of a different one. In other words, the model can transfer the appearance or pose from one image to another, as these two factors are disentangled. Fig. 4.6 shows synthesized images where the target appearance of all parts are extracted from the left column and target shape of all the part are extracted from the top row. As seen in the image, the model is able to synthesize an unseen person in variety of unseen poses or clothing. Note that, while the model is not trained on pairs of images but single images, it can combine factors of image pairs.

Furthermore, we learn the appearance and pose factors per part which enables us to do local changes to specific body parts and control each part independent from the others. In our approach, not necessarily the whole pose or appearance can be transferred, but local changes are also possible, which enforces more control over the image synthesis process. Given an image, we can fix the appearance and shape of all body parts except for, for example, the head. Then, extracting the appearance or pose of the head from different images, we can synthesize novel images which only differ in face appearance or the head shape, but are the same in all other parts. In Fig. 4.7, the whole pose and appearance representations are extracted from the top row, except for the pants. For the pants parts, appearance representations are extracted from the images on left. As a result, in each column, we see one person in the same pose but wearing different pants. In Fig. 4.8, the opposite experiment is shown. For all the parts, appearance and pose come from source images (the top

Figure 4.6: Pose and Appearance Transfer. The image visualizes novel images which their pose come from the top row and their appearance come from the left column. Each column depicts one person in various clothing and different hair colors.

Figure 4.7: Local Appearance Transfer. Visualization of selective image editing, where pants appearance of objects in the top row are extracted from the left column. Each column shows a person having the same pose and appearance except for the pants.

Figure 4.8: Local Pose Transfer. Visualization of selective image editing, where the pose of legs and feet of the object in the top row is changed according to the left row. The pose representations for legs and feet are extracted from the left row. Each column shows a person with same appearance but in different lower body poses.

row) expert for legs. We extract pose representations of legs and feet from the left images. As a result, in each column we have on person with the same clothings who poses only the legs differently.

Note that, as no annotation is used during training, the learned parts mappings to body regions are not available beforehand. For example, before training we do not know that which activation maps correspond to legs. They can be learned in any order. Hence, after the training process, all activation maps need to be visualized individually so that we can infer the activation maps and body regions correspondences.

4.2.1 Limitations

Although, some decent results are shown in the previous section for pose and appearance transfer task, the model does not yield good results for transferring pose or appearance from every random source image to any other image. The appearance source image and the pose source image pairs require to have some conditions, otherwise synthesized image do not have a high quality. Fig. 4.9 depicts some pose and appearance transfer failure cases. Firstly, the body parts for both appearance source image and the pose source image should correspond to each other. As the model detects the most important regions of an image in a totally self-supervised way, we cannot enforce the part labels to it. Sometimes, one object region can be easily be reconstructed by just assigning one Gaussian distribution to it, however the reconstruction of the same region for other images might need assigning more Gaussian distributions to capture more details. One example is a bald head versus a head with long hair. For the bald head just one single part is enough although for reconstructing the long hair 2 or 3 Gaussian distributions might be needed. Hence, the part appearances of these two images do not match and cannot be transferred. In the top rows of Fig. 4.9 we have shown this problem. In the first row, the shirt part of appearance source image corresponds to the skirt part of the pose source image which leads to mistakenly transferring the shirt color to the skirt. And in the next two rows, the feet of the pose source is occluded that leads to mistakenly

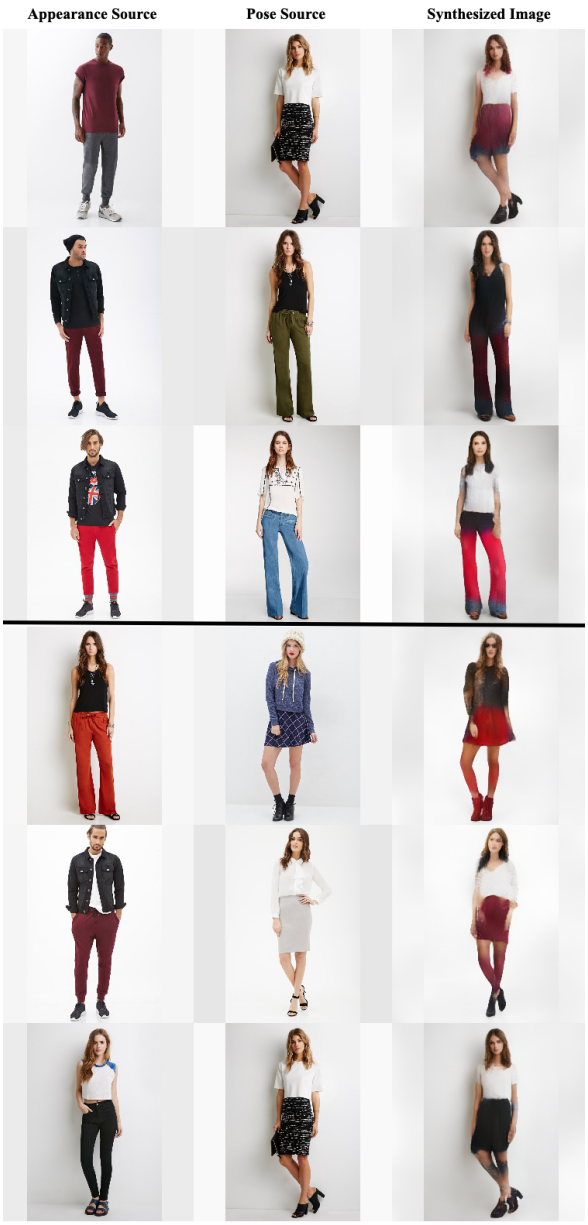

Figure 4.9: Limitations of pose and appearance transfer. Novel images shown in column 3 are generated by combining the appearance and pose of two objects in columns 1 and 2. But the model does not achieve good results for any arbitrarily pairs. In the first set, the model fails due to parts incorrespondences between two source images. And in the second set, transferring the appearance of pants to skirts generate is not successful.

transferring the shoes appearance of the appearance source to the pants of the pose source and and not the shoes. Secondly, the clothing types of each pair should be similar to get decent pose and appearance transfer results. For example, the appearance of a person wearing pant does not transfer well to a person wearing a short skirt as shown in the bottom image set of Fig. 4.9. Because in the dataset, the objects wear skirts usually with bare legs. As a result, the model is biased and has trouble changing the skin color to pants color when the appearance source is wearing pants and the pose source is wearing short skirt.

4.3 Comparison of PD and HPD on DeepFashion

PD predicts object parts in a flat way that leads to unstructured representations, however HPD detects object parts in a hierarchical way that enforces a tree structure to the part-based representations. Our experiments showed that HPD provides more controlled and localized parts than PD which leads to better image reconstruction capturing more details. HPD is trained with an additional loss term, hierarchical loss computed by Eq. 3.13 that forces the children parts to be close to their parents which prevents them from distributing all over the image. We find HPD works better particularly for cases that the number of predicted landmarks is large (more than 20). In these cases, a network have less control over the parts to assign them to meaningful regions as there are so many of them. As a result, PD ends up predicting parts, some of which are very alike and overlap with each other or some others are assigned to the background regions instead of the object parts, not being able to capture the details of the objects as expected. But for HPD the results are better and the parts are finer, as it enforces more control over the parts, not letting them spread over the image.

Fig. 4.10 visualizes the set of parts $\Phi_i(x)$ predicted by PD and HPD. For this experiment, PD is trained to predict 30 Gaussian distributions in a flat way, however HPD is trained with two levels of hierarchy, having 15 parent Gaussian distributions in the first level and 30 children Guassian distributions in the next and final level. The parents and children distributions are trained simultaneously. All the

Figure 4.10: Comparison of PD and HPD on the task of part detection. The image visualises colored Gaussian distributions predicted by PD model, and HPD. Comparing column 2 and 4 shows the hierarchical extension leads to more detailed and meaningful parts. In addition, HPD model works better especially on the challenging samples who wear white clothes.

images are from the test set of DeepFashion dataset, which are never seen in the training process. Firstly, by comparing resulting parts of the first level and the second level of hierarchy, we can see that some important object regions that could not be captured in the first level of hierarchy like the feet or upper arms are captured in the second level. Generally, there are finer and smaller Gaussian distributions in the second level as each parent distribution is broken into two in the next level. In addition, although there is no explicit term in the loss function to force the parts not to be identical, almost all of the Gaussian distributions are separated and capture different object keypoints in the HPD model. Secondly, when we increase the number of final parts from 15 in Fig. 4.2 to 30 in Fig. 4.10, we expect to have finer parts capturing more details of the object. But, we can see in the second column that PD does not necessarily leads to finer and more detailed parts. For example, when the number of detected landmarks are set as 30 PD still assigns 3 Gaussian distributions to the left leg similar to Fig. 4.2, which had 15 landmarks. But, for this case of 30 landmarks HPD assigns 6 Gaussian distributions to the left leg, being able to capture more details of that. In addition, about 9 of the predicted parts by PD are almost identical that makes just at most 21 of them visible in the blobby image (second column of Fig. 4.10). Predicting identical parts can be considered as a waste of model complexity and parameters, since repetitive parts do not add more information for reconstructing the original image. However, for HPD, 29 out of 30 resulting Gaussian distributions are well separated and clearly visible in the blobby image (second column of Fig. 4.10). Generally, by comparing the second column and the last column we can infer that the hierarchical extension improves part localization and their controlability and it is less prone to predicting irrelevantly distributed parts.

Table 4.1 shows that hierarchical extension can also lead to better quantitative results in terms of the pixel-wise reconstruction error. Note that, for all evaluation experiments, both part appearance representations and pose representations are extracted from the original image x not the spatially transformed image $T_s(x)$ or appearance transformed image $T_a(x)$. Then part appearance and pose representations are combined and given to the decoder to reconstruct the original image. We reported pixel-wise error of the PD model with 30 keypoints and the HPD model

Method	Encoder's Parameters	Pixel-wise Error
PD	**39,641,182**	0.2921
HPD	39,645,037	**0.2202**

Table 4.1: Reconstruction error of PD and HP model in image generation task on the DeepFashion dataset. Numbers show HPD achieves better results having only a few parameters more than PD.

with two levels of hierarchy, 15 keypoints in the first level and 30 in the second level in Table 4.1. Pixel-wise error is computed by taking the average sum of the squared difference of pixel values over the image, written as follows:

$$\text{Pixel-wise Error } = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|H||W|}\left(\sum_{u=1}^{|H|}\sum_{v=1}^{|W|}(x_{u,v}-\hat{x}_{u,v})^2\right), \tag{4.1}$$

where N is number of total images in the test set, $|H|$ and $|W|$ are the images height and width respectively, x is the data sample and $\hat{x}$ is the reconstructed image. It can be concluded from the results that parts predicted by HPD improve image reconstruction as they capture more details of the object. In addition, the number of each model parameters is reported in Table 4.1 as a factor of model complexity. We can infer from the results that HPD improves the reconstruction task without adding a huge number of parameters and complexity to the model, as its encoder has only $3,855$ more than the flat PD model, and the decoder is same for both models.

4.4 Landmark Detection on CelebA

In this set of experiments we evaluate our model quantitatively and qualitatively through the task of unsupervised landmark detection, and compare our results with Lorenz et al. [35] as the baseline. We also provide an ablation study to assess the importance of each individual module of our model. For this set of experiments, we used the CelebA[33] dataset. It includes $200K$ images of celebrity faces. Following [22, 35, 51], we divided the dataset into three folds: CelebA without the MAFL subset, the MAFL training set, and the MAFL test set. Firstly, we train the

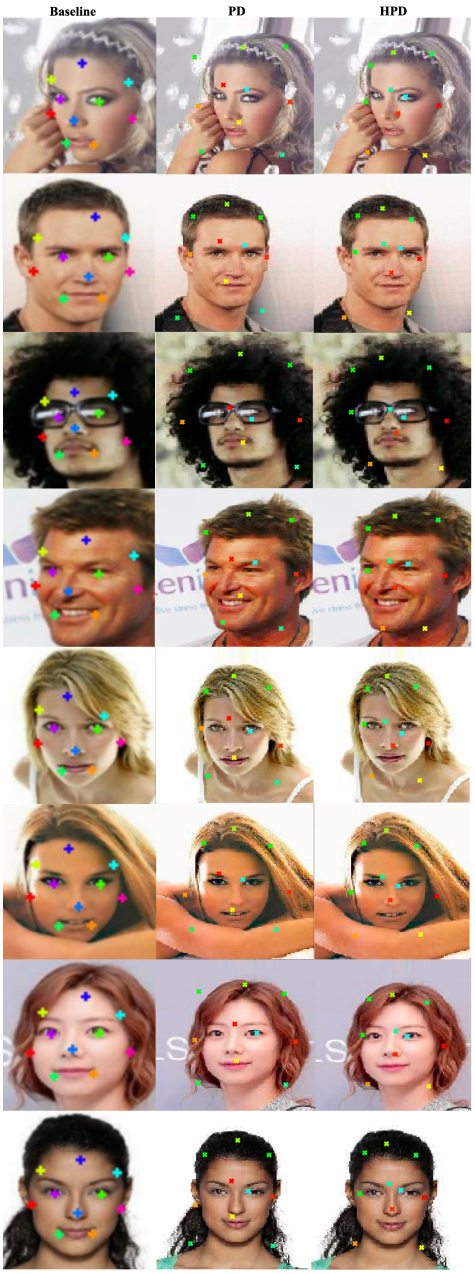

Figure 4.11: Unsupervised landmark detection. Visualization of 10 unsupervised landmarks predicted by the baseline, PD and HPD models from left to right. Note that, the images of the first column are more zoomed as we copied them directly from their paper, but all models are trained with 128×128 images.

variations of our model (PD and HPD) on the CelebA dataset excluding the MAFL subset. Then we freeze the network weights to predict the unsupervised keypoints. In the next step, as the mapping of our unsuperviesd keypoints to the ground-truth ones is not available, we train a linear regressor to learn this mapping. One linear layer without a bias term is trained on the training set of the MAFL dataset to map our unsupervised keypoints to the 5 ground-truth keypoints. Lastly, for qualitative and quantitative results, we test the model on the test set of the MAFL subset.

Fig. 4.11 visualizes our 10 unsupervised keypoints predicted by the two variations of our model, PD and HPD, and compares them with keypoints predicted by the baseline[35]. We can see that our keypoints track the face shape properly, however it is evident that the baseline does better on this as its keypoints are more compact and focused to the face. The semantic meaning behind each keypoint is not obvious, but we can see some of the keypoints are assigned to meaningful parts like the nose, right eye, and top of the head. HPD parts are more localized and less distributed over the image in comparison with PD which leads to tracking the face more closely.

Fig. 4.12 shows landmark detection results predicted by the best version of our model HPD for faces in different angles. We trained the model with a structure defined in Sect. 3.3 with two levels of hierarchy, predicting 5 parts in the first layer and 10 in the second and last layer. The 10 cross markers shown in Fig. 4.12 represent the unsupervised detected the keypoints of the last layer of HPD. The regressed keypoints are the output of the linear layer which maps the 10 unsupervised keypoints to 5. We can see that predicted keypoints correspond to the ground-truth ones specifically for the eyes and the mouth keypoints, even when the face is not looking straight.

4.4.1 Ablation Study

To assess the impact of each module, we tested 4 different models each differing in one module and compared the results with the baseline in terms of the landmark

Figure 4.12: Landmark detection. We show 10 unsupervised keypoints alongside with their mappings to 5 keypoints and the ground truth keypoints for sample faces from different angles. The cross markers show unsupervised keypoints. The ground truth keypoints and regressed keypoints are shown by hollow circles and solid circles respectively.

Method	Model's Parameters	Landmark Detection Error
Baseline[35]	74,171,543	7.54 (3.24)
Baseline + Swapping 4.4.1	74,171,543	7.12
Baseline + RGB colors 4.4.1	56,903,565	6.25
PD 4.4.1	56,903,565	5.87
HPD 4.4.1	56,904,850	**5.79**

Table 4.2: Ablation Study. We report landmark detection error of 5 versions of our model each differing in one module on MAFL test set. Results demonstrate that HPD model which combines all our proposed techniques yields the best results. Note that, the reported error of 7.54 for the baseline is achieved by our own implementation of that model, however it achieved error of 3.24 according to their paper.

detection error. Following [3, 35, 51], the landmark detection error is measured by Eq. 4.2 as the average landmark distance to ground-truth, normalized as percentages with respect to inter-ocular distance.

$$\text{Landmark Detection Error} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{5}\sum_{j=1}^{5}\frac{\|x_{ij}-\hat{x}_{ij}\|_2}{\text{inter-ocular distance}(x_i)}, \tag{4.2}$$

where inter-ocular distance is the distance between the two eyes, which is measured as the euclidean distance of ground truth keypoints of the left and right eye, and $\hat{x_{ij}}$ depicts the j_{th} predicted keypoints of the i_th sample. We report the number of model parameters and the landmark detection error in Table 4.2.

The four variations of models in Table 4.2 are described in below:

- **Baseline + Swapping** : In this model, we used [35] as the core but instead of explicitly using the equivariance loss in the training, we used the swapping technique. In the swapping technique, we swap the original image with the spatially transformed image for 50 percent of the times. As a result, half of the times the spatially transformed image is reconstructed instead of the original image which makes the pose encoder predict part activation maps that consistently track the object part they represent, even when the object is deformed. This simple technique frees us from further tuning the loss term

and improves the results.

- **Baseline + RGB colors**: We wanted to assess the impact of using a simplified appearance encoder which instead of predicting a n dimensional appearance vector for each part, it assigns a 3-channel RGB color to them. As a result, in this model we replaced the appearance encoder of Lorenz et al [35] with our simplified appearance encoder. This change freed up $17,267,978$ parameters which reduces the memory usage without harming the results.

- **PD** This model is described in Sect. 3.2. In this version, we omitted the appearance encoder and the equivariance loss of the baseline; instead we used the RGB colors and the swapping technique. The combination of these changes leads to better landmark detection results.

- **HPD** This model is described in Sect. 3.3. In this version, we kept all our changes to the baseline plus predicting the parts in a hierarchical manner. For the HPD model, we can define an arbitrary number of hierarchies but in this particular experiments, we used two levels of hierarchy with 5 keypoints in the first level and 10 in the last. It is evident from our qualitative and quantitative results that the hierarchical extension improves the results qualitatively and quantitatively in terms of the reconstruction error and the landmark detection error. Furthermore, the hierarchical extension only adds about $1,285$ parameters which is negligible in comparison with over 56 million of total parameters.

Among all the tested variations, the HPD model yields the minimum error of 5.79 with a small gap with PD. Note that, as the original code for the baseline model [35] was in Tensorflow and the evaluation code was not available, we reimplemented the model on our own. We used our own setup for training and testing including the transformations which might be the reason for the gap between their reported number in the paper [35] (3.24) and ours (7.54).

4.5 Implementations Details

For implementation, we used the Pytorch framework. We found all the model variations are sensitive to the batch size. For batch sizes less than 8, the Gaussian

distributions were not well localized to body regions, instead there were horizontally distributed over the object. But with batch sizes equal greater than 16 they started to capture meaningful parts. For all the models we used 32 as the batch size with the accumulating gradient technique and 0.001 as the learning rate with Adam optimizer. For all experiments with the hierarchical extension, the weight of the hierarchical loss is set as 0.1. Our pose encoder **E** has the same architecture of Chen et al. [7], and decoder **D** has U-net [44] architecture with four downsampling Convolution layers, four upsampling Convolution layers, and four ResNet connections. In the learning phase, each pass over the whole training set took about half an hour for the DeepFashion dataset on a single core NVIDIA Titan Xp GPU.

Chapter 5

Conclusion

In this thesis, we present HPD, an auto-encoder framework for detecting parts that form an object and disentangling their pose and appearance in an unsupervised manner. Our model can produce high quality disentangled representations which can be used in various tasks such as unsupervised landmark detection, novel image synthesis, and local or global pose and appearance transfer. We suggest a swapping technique that enables the training using only reconstruction loss as the objective function instead of its combination with the equivariance loss used in similar methods [35, 51, 59]. This simple technique frees the model from further tuning of the training objective and leads to robust training. In addition, we propose a much simpler encoder that, instead of predicting n dimensional appearance representations like [35], predicts a single RGB color for each part as the appearance representation. This substitution saves millions of model parameters, making it a more lightweight network in comparison with Lorenz et al., yet achieving comparable results. Furthermore, we propose a new method for detecting object parts in a hierarchical manner which enforces a binary-tree structure to the detected parts. We show in our experiments that the hierarchical extension can lead to more meaningful keypoints and better quantitative results in terms of landmark detection error and image reconstruction error. Although we could not beat the quantitavie results reported by Loren et al. [35] paper, we believe that our simplified architecture and proposed ideas have potential in a future work.

5.1 Limitations and Future Work

The proposed method in this thesis does not come without limitations. At last, we list some of the model's limitations and our proposed solutions that can be tested to improve the results in the future.

- The visual quality of synthesised images by a neural network is not always ideal, and our method is not an exception. To improve the quality of images we can train the model with an additional adversarial training objective as proposed by Isola et al [20]. For this purpose a discriminator need to be added on top of the model to classify the generated images by the auto-encoder as real or fake.

- For the set of experiments on CelebA, we trained models for 30 epochs and reported the results in Table 4.2, which outperforms the baseline. But as the baseline model has more parameters in comparison with ours, it might outperform us after longer training. To keep the comparison fair, the model should be trained longer.

- We showed our model performs well on the DeepFashion and CelebA datasets, but its performance on other datasets is an open question. It would be interesting to assess the performance of the model for objects other than humans like cats [58] and birds [53] in the future.

- We discussed that in our proposed HPD model children parts locations are dependent on their parents, as we force μ of the children Gaussian distributions to be close to the μ of their parents Gaussian distributions. But currently there is no constraint on the $\mathbf{\Sigma}$. In the future, the hierarchical loss can be revised in a way that the standard deviation of the children Gaussian distributions be dependent to their parents as well as the mean.

- We believe the new idea of enforcing a hierarchical structure to the learned parts has a great potential for future work. Although the proposed hierarchical method is defined generally for any arbitrarily levels of hierarchy, we just used up to 3 levels of hierarchy for our experiments. The impact of hierarchy levels on the quality of keypoints and image reconstruction is an

open area. In addition, in our experiments the number of predicted keypoints were limited to under 30, however we believe the strength of the hierarchical extension emerges on cases that need to predict a large number of keypoints. Further experiments can be done in the future to assess this assumption.

Bibliography

[1] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8340–8348, 2018. → pages 4, 12

[2] F. L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989. doi:10.1109/34.24792. → pages 17

[3] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *2013 IEEE International Conference on Computer Vision*, pages 1513–1520, 2013. doi:10.1109/ICCV.2013.191. → pages 40

[4] Z. Cao, T. Simon, S. Wei, Y. Sheikh, et al. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. → pages 1

[5] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 321–328, 2013. → pages 4

[6] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5933–5942, 2019. → pages 4

[7] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. → pages 9, 42

[8] X. Chen and A. L. Yuille. Parsing occluded people by flexible compositions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3945–3954, 2015. → pages 4

[9] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. → pages 3

[10] E. L. Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pages 4414–4423, 2017. → pages 2, 4

[11] A. Dundar, K. J. Shih, A. Garg, R. Pottorf, A. Tao, and B. Catanzaro. Unsupervised disentanglement of pose, appearance and background from images and videos. *arXiv*, pages arXiv–2001, 2020. → pages 6

[12] B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy, and J.-W. Meent. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2525–2534. PMLR, 2019. → pages 5

[13] P. Esser, J. Haux, T. Milbich, and B. Ommer. Towards learning a realistic rendering of human behavior. In *ECCV Workshops*, 2018. → pages 4

[14] P. Esser, E. Sutter, and B. Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. → pages 1, 4

[15] P. Esser, J. Haux, and B. Ommer. Unsupervised robust disentangling of latent characteristics for image synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2699–2709, 2019. → pages 4

[16] S. J. Gershman, J. B. Tenenbaum, and F. Jäkel. Discovering hierarchical motion structure. *Vision research*, 126:232–241, 2016. → pages 5

[17] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. doi:10.1109/CVPR.2018.00762. → pages 1

[18] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. → pages 3

[19] Q. Hu, A. Szabó, T. Portenier, P. Favaro, and M. Zwicker. Disentangling factors of variation by mixing them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3399–3407, 2018. → pages 3

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. → pages 44

[21] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 740–747, 2014. → pages 5

[22] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in neural information processing systems*, pages 4016–4027, 2018. → pages 4, 13, 36

[23] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8787–8797, 2020. → pages 4

[24] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. → pages 13

[25] P. P. Juan-Manuel Perez-Rua, Tomas Crivelli and P. Bouthemy. Discovering motion hierarchies via tree-structured coding of trajectories. In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 106.1–106.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi:10.5244/C.30.106. URL https://dx.doi.org/10.5244/C.30.106. → pages 5

[26] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5546–5555, 2015. → pages 4

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90, 2017. → pages 3

[28] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015. → pages 3, 6

[29] A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018. → pages 3

[30] M. Lam, B. Mahasseni, and S. Todorovic. Fine-grained recognition as hsnet search for informative image parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2520–2529, 2017. → pages 4

[31] K. Lenc and A. Vedaldi. Learning covariant feature detectors. In *European conference on computer vision*, pages 100–117. Springer, 2016. → pages 2, 14

[32] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. → pages 5

[33] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. → pages 36

[34] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016. → pages 19

[35] D. Lorenz, L. Bereska, T. Milbich, and B. Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2019. → pages 1, 2, 5, 6, 7, 8, 10, 12, 14, 21, 36, 38, 40, 41, 43

[36] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. In *Advances in neural information processing systems*, pages 406–416, 2017. → pages 4

[37] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2018. → pages 4

[38] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in neural information processing systems*, pages 5040–5048, 2016. → pages 1, 2, 3

[39] N. J. Mitra, M. Wand, H. Zhang, D. Cohen-Or, V. Kim, and Q.-X. Huang. Structure-aware shape processing. In *ACM SIGGRAPH 2014 Courses*, pages 1–21. 2014. → pages 5

[40] K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. Mitra, and L. Guibas. Structurenet: hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics*, 38(6), 2019. → pages 5

[41] D. Paschalidou, L. V. Gool, and A. Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1060–1070, 2020. → pages 5

[42] X. Peng, X. Yu, K. Sohn, D. N. Metaxas, and M. Chandraker. Reconstruction-based disentanglement for pose-invariant face recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1623–1632, 2017. → pages 3

[43] R. Rombach, P. Esser, and B. Ommer. Making sense of cnns: Interpreting deep representations & their invariances with inns. In *Proceedings of the European Conference on Computer Vision*, 2020. → pages 1, 3

[44] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. → pages 11, 42

[45] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018. → pages 1, 4

[46] R. Sicre, Y. Avrithis, E. Kijak, and F. Jurie. Unsupervised part learning for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6271–6279, 2017. → pages 4

[47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. → pages 3

[48] K. K. Singh, U. Ojha, and Y. J. Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*, 2019. → pages 3

[49] A. Szabó, Q. Hu, T. Portenier, M. Zwicker, and P. Favaro. Challenges in disentangling independent factors of variation. *arXiv preprint arXiv:1711.02245*, 2017. → pages 4

[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. → pages 3

[51] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. → pages 14, 36, 40, 43

[52] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017. → pages 3

[53] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. → pages 44

[54] Y. Xu, C. Yang, Z. Liu, B. Dai, and B. Zhou. Unsupervised landmark learning from unpaired data, 2020. → pages 13

[55] Z. Xu, Z. Liu, C. Sun, K. Murphy, W. T. Freeman, J. B. Tenenbaum, and J. Wu. Unsupervised discovery of parts, structure, and dynamics. In *International Conference on Learning Representations*, 2018. → pages 5

[56] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016. → pages 3

[57] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3073–3082, 2016. → pages 4

[58] W. Zhang, J. Sun, and X. Tang. Cat head detection-how to effectively exploit shape and texture features. In *European Conference on Computer Vision*, pages 802–816. Springer, 2008. → pages 44

[59] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee. Unsupervised discovery of object landmarks as structural representations. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2694–2703, 2018. → pages 2, 14, 43