CATEGORY CLASSIFICATION AND LANDMARK LOCALIZATION FOR A FASHION DATASET

INF 592

Tutor: Professor Alexandre Alahi, Doctor Taylor Mordan
author:Pegah Khayatan

l'X
ÉCOLE
POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Acknowledgements

I would like to express my gratitude to my supervisor, professor Alexandre Alahi, who guided me throughout this project.
I would like to thank Doctor Taylor Mordan for providing me with guidance and numerous insightful suggestions.
My sincere thanks also goes to Carol Ortega for her great help with administrative procedures, George Adaimi for sharing his experience and code, and VITA Lab members for their support.

Abstract

In this internship, a plugin of a landmark localization pipeline, proposed by previous VITA Lab member Sven Kreiss, was created for deepfashion dataset and different parameters were tuned to improve the results.

It has been tried to give proper explanations for failure cases, for example when the loss explosion happens with a particular setup. Also, results from different setups and preprocessing are compared.

In the second part of the internship, the focus was to propose a network structure to jointly localize clothing landmarks and predict their categories. The method has been inspired by the work of VITA Lab member Taylor Mordan, to predict pedestrian attributes.

Contents

1 Introduction

1.1 Laboratory and internship

The principal goals of this internship were to acquire experience and knowledge in the field of computer vision and machine learning.

This objective has been satisfied by working on a project at the intersection of these two domains. The project was supervised by professor Alexandre Alahi and Doctor Taylor Mordan from Vita Lab. Vita Lab is one of Epfl laboratories doing research related to computer vision and artificial intelligence. More specifically, the laboratory pushes the limits of Artificial Intelligence (AI) in the context of transportation, mobility, and built environments.

1.2 Internship process

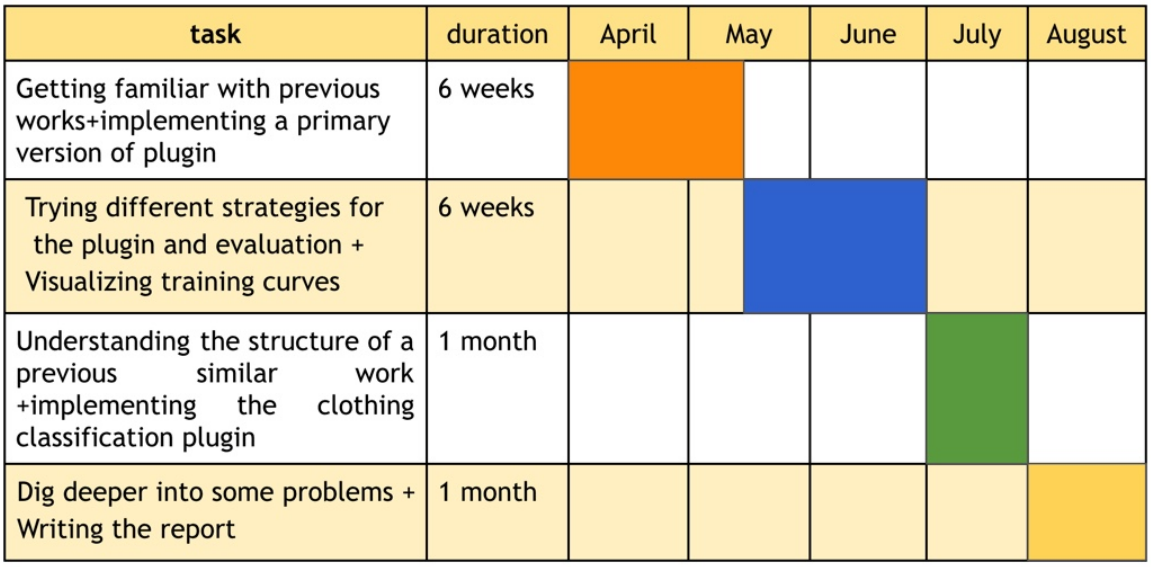
The first six weeks were dedicated to getting familiar with using HPCs (high performance computer cluster) of EPFL, studying the state-of-the-art methods and different datasets for fashion landmark detection, implementing the method introduced in one of the works, getting familiar with using the chosen dataset, and implementing a primary plugin for detecting landmark based on deepfashion dataset.

In the next six weeks, I tried different strategies for implementing the plugin, drawing training curves, and tuning parameters.

At the beginning of the fourth month, I started studying the code of Mordan et al. [2021] that is similar to the clothing classification part of my project. I also had to revise openpifpaf's code for certain parts of my implementation. This task along with its debugging took me about 4 weeks.

During the first two weeks of the fifth month, I was trying to dig deeper into some problems, such as loss explosion, faced during training the landmark localization plugin. During this period, I also cleaned my codes and finished some experiments and evaluations.

The last two weeks were dedicated to writing the report and evaluating some lately trained models.



task	duration	April	May	June	July	August
Getting familiar with previous works+implementing a primary version of plugin	6 weeks	🟧	🟧			
Trying different strategies for the plugin and evaluation + Visualizing training curves	6 weeks		🟦	🟦		
Understanding the structure of a previous similar work +implementing the clothing classification plugin	1 month				🟩	
Dig deeper into some problems + Writing the report	1 month					🟨

2 Retrospective of fashion task and related datasets

2.1 Fashion-related tasks

The large part of fashion in the industry motivates us to tackle related problems and the modern tools, deep neural network architectures, and large-scale fashion databases, enable us to do so. clothing recognition and segmentation Kalantidis et al. [2013], Hidayati et al. [2017], recommendation Han et al. [2017], Ma et al. [2017], and fashion landmark localization Liu and Lu [2018], Ge et al. [2019], Wang et al. [2018a] are the main tasks in this area.
In this internship, two problems in visual fashion analysis have been studied: fashion landmark localization and clothing category classification.
In the first part of this project, a landmark localization method is introduced with a focus on parameter tuning. Fashion landmarks are functional keypoints of clothing items, such as a collar, hemline, sleeves, and waistline. They can be considered an effective way to visually understand clothes. Landmark detection for clothes is subject to a number of inherent challenges compared to landmark detection for other usual items like humans, cars, and animals. These challenges include subtle appearance differences, variations in human poses, different shooting angles, apparel deformations, and self-occlusion.
In the second part, we propose a method to perform both landmark localization and clothing classification jointly. clothing classification has a variety of applications important to e-commerce, online advertising, internet search, and the visual surveillance industry.

The rest of this report is organized in the following manner: 1. Overview of previous visual fashion analysis methods 2. A quick review of important fashion datasets 3. Introduction to the base pipeline re-used for fashion landmark localization 4. Parameter tuning and experiments for landmark localization model 5. Introduction to the base work for category classification 6. Implementation details and primarily results 7. Conclusion and future work.

2.2 Overview of visual fashion analysis methods

Two general approaches exist for localizing the landmarks, coordinate-based and heatmap-based.
The first type regresses the coordinates of landmarks directly; it means the output is a set of estimated positions of the landmarks.
On the other hand, the second type regresses a confidence map over the 2D image; it means the more confident regions are where the landmark predictions should be placed. Different heat maps are used for different types of landmarks.

Most of the recent methods are heatmap-based as the latter captures more spatial and contextual information about the key points and yields more accurate results. It however faces a number of shortcomings; A 2-dimensional Gaussian distribution centering at the ground-truth joint position is used to convert discrete value ground-truth landmarks to ground-truth heatmaps. This transformation leads to smoother training but also introduces a quantization error problem when downscaling the continuous heat map.
The problem is even more pronounced for low-resolution datasets. Different strategies can be employed to overcome the drawbacks, such as increasing the resolution by including deconvolution layers; they however introduce costly computations to the network.
The landmark localization technique in this project is heatmap-based. In the following, we review works in both categories.

2.2.1 Coordinate-based methods

Liu et al. [2016a] proposed deepfashionNet, a convolutional network used on the deep fashion dataset from the same paper, which jointly predicts landmarks and attributes for clothing items and also models the clothes pairs.
Their network predicts landmarks and their visibility in the first stage. This information is then employed to gate the features, leading to local features. These features are invariant to deformations and occlusions as a result of being local.
The global and pooled features are then concatenated and passed through another layer to output categories and attributes and to model clothes pairs.
This model uses VGG16 Simonyan and Zisserman [2014] as backbone.

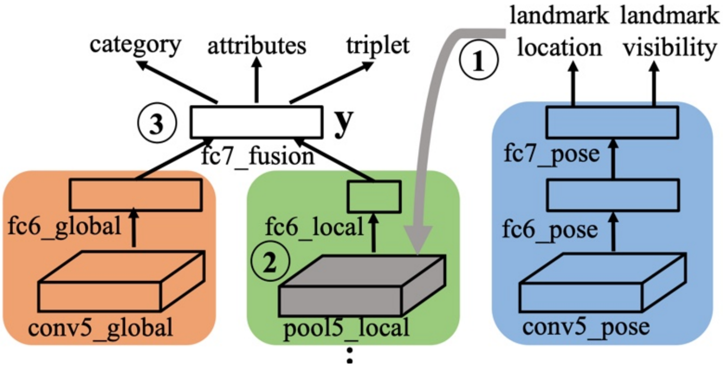


Figure 1: Pipeline of FashionNet, which consists of glob features branch (in orange), local features branch (in green) and pose branch (in blue). Shared layers are not shown here.

Liu et al. [2016b] introduces deep fashion alignment network (DFA). DFA consists of three stages where predictions are subsequently refined. The first stage predicts rough landmark positions and pseudo-labels (pseudo-labels are achieved by clustering landmark positions, indicating different clothing categories). The second and third stages refine these predictions by estimating offsets for landmarks and label clusters. The difference between the second and third steps is that the last one has two different branches for easy and hard images (based on the predicted offset in the second stage).

Yan et al. [2017] introduces Unconstrained Landmark Database (ULD), as a more challenging dataset for fashion landmark detection. They also propose a network Deep LAndmark Network (DLAN) based on DFA. However, DFA takes clothes bounding box as input while DLAN takes raw fashion images as input without any bounding box annotations. Thi work also addresses two main difficulties: 1. Robustness to scale variations 2. Robustness to global deviation from the center and local geometric deformations.
To address the first challenge, information from different scales is taken into account; a scale tower is constructed for each different scale. Each scale tower captures the convolutional responses for that scale by using exponentially expanded receptive fields (selective dilated convolution in Figure 2). The final convolutional response is obtained by selecting the element-wise maximum over the output of all scale towers. This process helps to adapt fine-grained single-scale filters to more flexible inputs and hence landmark localization becomes more robust to the change of scale.
It is desirable to remove background clutters from input images and transform target regions into canonical form because fashion inputs sometimes deviate from the center and experience local geometric deformations.

To regulate global and local variations in clothing items, two strategies are employed:

1. Spatial transformer Jaderberg et al. [2015] to learn to roughly align feature maps
2. Hierarchical recurrent spatial transformer module to predict global and local refinements to predicted landmarks; for each landmark j, its transformation is decomposed into a base global part (common for all landmarks) and a set of local refinement transformations:

$$\Theta_j(M) = \Theta_{global}.\Theta_{local} = \Theta(1).\prod_{i=2}^{M} \Theta_j(i-1 \to i)$$

the product on the right side of the equation indicates recurrent local refinement of the landmark j. M parameter is set to 3 in this work; meaning a local transformation is predicted and recurrently refined in 3 steps.

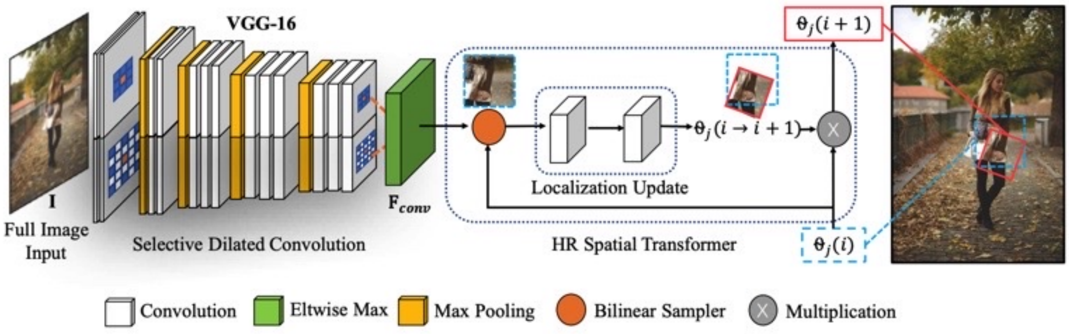


Figure 2: Pipeline of Deep LAndmark Network (DLAN) for unconstrained fashion landmark detection (clothing bounding boxes are not provided in both training and testing). DLAN contains two main modules, including a Selective Dilated Convolution for handling scale discrepancies, and a Hierarchical Recurrent Spatial Transformer for handling background clutters.

2.2.2 Heat map-based methods

Wang et al. [2018a] proposes a network for landmark localization, attribute detection, and category classification that is leveraged with high-level human knowledge and also uses two attention mechanisms. We shortly explore these two contributions:

1. *Fashion Grammar:* General clothing structure suggests a common set of connections, like a skeleton, between different landmarks (Kinematics grammar); for example a connection between the right collar and the right sleeve. Besides connectivity relations, there also exists a symmetrical association between landmarks (Symmetry grammar). This grammar can be considered as a graph where nodes are the confidence heatmaps for different landmarks and edges are connections defined in the grammar. A message passing system using Bidirectional Convolutional Recurrent Neural Network (BCRNN) is employed to iteratively update heatmaps and to improve localization of not correctly identified landmarks based on other heatmaps.

2. *Fashion Landmark-Aware Attention:* predicted landmark positions can be used to help features be more concentrated on key regions of clothes. A weight map can be constructed from the predicted heatmaps $\{S_i\}_{i=1}^K$, by a cross-channel average-pooling operation:

$$A^L = \frac{1}{K} \sum_{i=1}^{K} S_i$$

This attention is supervised and captures structural clues to help category classification and attribute prediction.

Clothing Category-Driven Attention: landmark-aware attention only captures supervised hints and may be insufficient to discover all the informative locations. Unsupervised and goal-driven attention is employed to cover up more information. The attention features are first pooled down to a very low resolution and then up-sampled and thus the attention has a large receptive field.

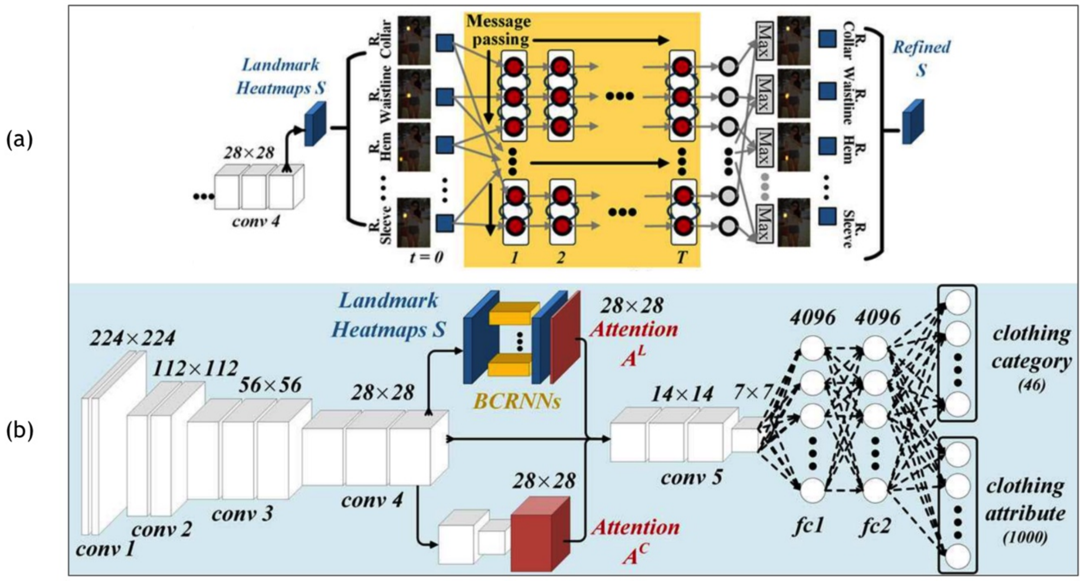


Figure 3: (a) Message passing over fashion grammars, where the blue rectangles indicate heatmaps of landmarks, and the red circles indicate BCRNN nodes. Within each BCRNN, message passing is performed over fashion grammars in two directions. With stacked of BCRNNs, the messages are iteratively updated and landmark estimations are refined. (b) The whole network architecture. A set of BCRNNs (yellow cubes) are established for capturing kinematics and symmetry grammars(blue cubes). Fashion landmark-aware attention A^L and clothing category-driven attention A^C (red cubes) are incorporated to enhance clothing features.

Liu and Lu [2018] proposes a model that shares the same layers with VGG-16 until conv3-4; two branches are then added, one for landmark localization, from which an attention branch is extracted that passes the predicted landmarks through a u-net structure Ronneberger et al. [2015], and one for category classification and attribute prediction, that uses the extracted spatial attention to refine features from conv3-4 layer. These refined features are then passed through conv5-1,2,3 layers, before getting divided into two fully connected branches for category classification and attribute prediction Figure 4.

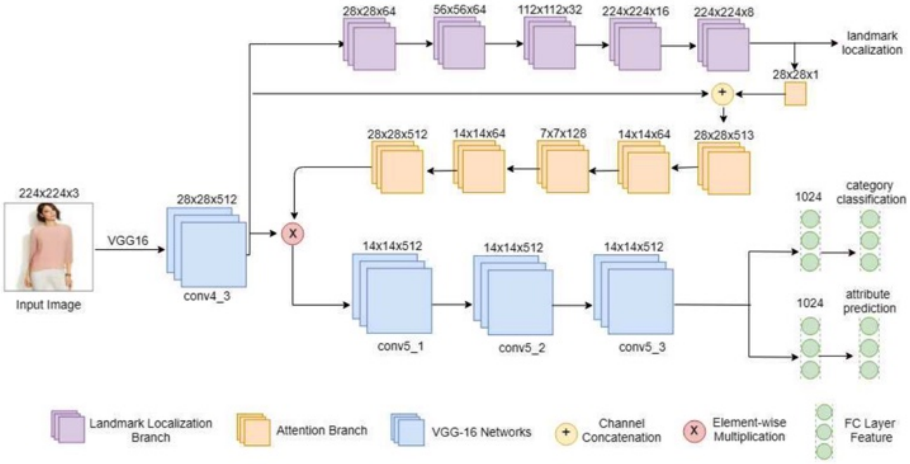


Figure 4

Li et al. [2019] proposes a network structure that uses as the main novelty a variation of non-local methods called spatial-aware non-local block (SANL). Non-local signifies a method that determines each pixel in a filtered image based on all the pixels in the input image Wang et al. [2018b]. In the original non-local block, the receptive field is the whole feature map. To find a way to reduce the number of locations in the receptive field (that should be trained in an end-to-end manner) means reducing the computational cost and likely would help to learn in a more effective way. Introducing spatial attention can filter out locations in the receptive field of a non-local block.
The spatial attention map used in a SANL block is obtained from the gradients of a feature map corresponding to the predicted category by pretrained ResNet-18 (using Gradient-weighted Class Activation Mapping or GCAM Selvaraju et al. [2017]). Using low-level feature maps for creating activation maps contains spatial information from low-level features and the same applies to high-level feature maps that lead to more semantic information.
The structure of the presented network is composed of three main parts: AttentionNet, CoraseNt and FineNet Figure 5.

AttentionNet is a branch that outputs the predicted category and also provides activation maps using Grad-CAM blocks; it consists of stacked convolutions and Grad-CAM blocks in between. Grad-CAM uses the gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept.
CoarseNet and FineNet both have landmark heat maps as targets, but the heatmap in CoarseNet is created with a larger sigma factor in the gaussian filter from the discrete ground truth landmark locations. The reason to have two networks with different targets is to avoid imbalance between positive and negative locations in a heatmap, by having a larger sigma, but also not to compromise the prediction accuracy.
CoarseNet has a feature pyramid architecture, where SANL blocks are added as attention between convolutional layers in the bottom-up path. Multi-scale predictions are made from the top-down ar-

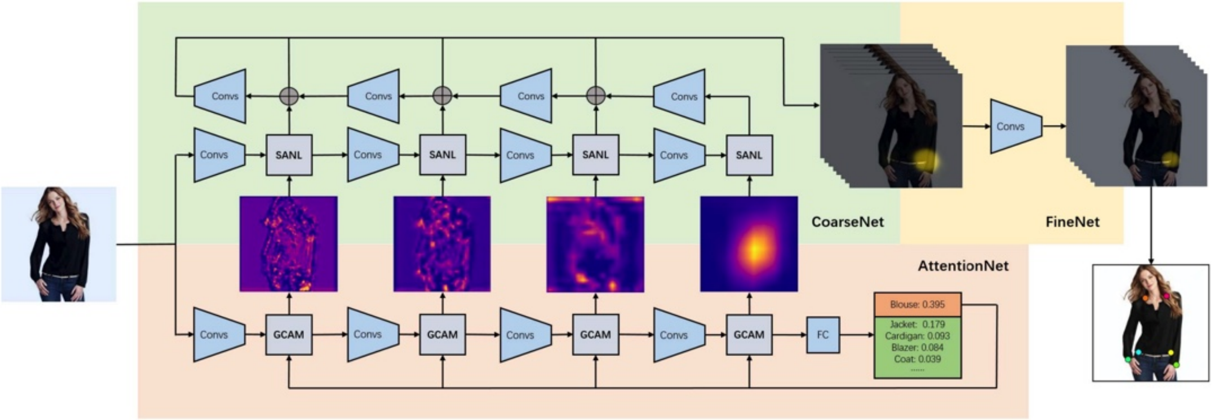


Figure 5

chitecture at all scales. The output of the CoarseNet is then passed to the convolutional FineNet to produce more detailed predictions.

In the same spirit as leveraging high-level human knowledge to further improve landmark localization, Yu et al. [2019] proposes a Layout-Graph Reasoning Module (LGR). This module is designed to enhance features and detect structure-consistent landmarks based on relationships among landmarks and their different categories.
For the sake of simplicity, we explain the structure of this module for the case of deepfashion dataset. In this dataset, eight types of landmark exist: left and right collars, sleeves, waistlines and hems. And clothing items are categorized into three subsets: upper, lower and full body.
LGR first maps convolutional features to graph node representations. These correspond to different types of landmarks, so 8 graph nodes are created in this step. These nodes are then clustered into clothes-part nodes (collar, sleeve, waistline and hem) and in the next level to body-part nodes (upper and lower). All the nodes are then clustered into one root node. The clustering operation uses graph convolution from Welling and Kipf [2016]. The root node is then passed through a series of graph deconvolution layers and is converted to body-part nodes, clothes-part nodes and leaf landmark nodes. Skip connections are employed between corresponding clustering and graph deconvolution layers to provide more consistency. The nodes in the last layer, leaf landmark nodes, are then mapped to convolutional features.

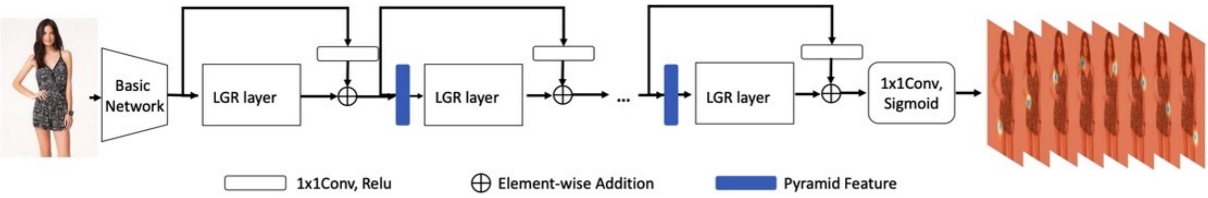


Figure 6

The network structure consists of a basic convolutional network for feature extraction and then a series of stacked LGR modules. Between LGR modules, intermediate features are enhanced across multi-scales by pyramid module Yang et al. [2017] and data bias is decreased by residual addition (Figure 6).

Ziegler et al. [2020] has its focus on adjusting previous networks to be more robust on less structured datasets when trained on large-scale fashion datasets. They hence propose data augmentation techniques for training and also model adjustments to make it more generalizable. We quickly describe each of these two:

1. *Data augmentation techniques:* Rotating images with a small angle can increase robustness in detection tasks[38]. As the first method of data augmentation, the image and its landmarks are rotated by a random angle in the range [0,2π]. However, rotation is not the only deformation that manifests in less structured datasets; clothes can also be loose. This deformation, elastic warping, can be modeled in an efficient way; the transformed image (image obtained after adding the deformation) is a version of the original image where certain pixels are displaced. So, the transformation can be considered as a mapping of pixels between the original image I and the transformed image $\tilde{I}$:

   $$\tilde{I}(\tilde{x}, \tilde{y}) = I(x(\tilde{x}, \tilde{y}), y(\tilde{x}, \tilde{y}))$$

   The mapping can be modeled using smooth displacement fields, $\Delta\bar{x}$ and $\Delta\bar{y}$:

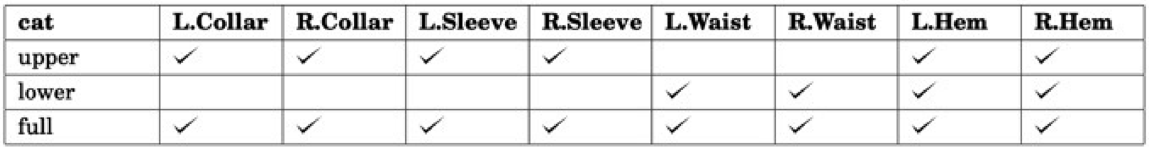
   $$\tilde{I}(\tilde{x}, \tilde{y}) = I(\tilde{x} + \Delta\bar{x}(\tilde{x}, \tilde{y}), \tilde{y} + \Delta\bar{y}(\tilde{x}, \tilde{y}))$$

   To construct smooth displacement fields, first, a number of pixels in the transformed image are chosen to be displaced. In the second step, the displacement field value (the intensity of displacement) at those points is randomly selected from a uniform distribution $U(\alpha, \alpha)$. Displacement fields are then convolved with a gaussian distribution to be smoothed (for example, $\Delta\bar{x}(\tilde{x}, \tilde{y}) = \Delta x(\tilde{x}, \tilde{y}) * G(\tilde{x}, \tilde{y})$ where $G(\tilde{x}, \tilde{y})$ is a gaussian filter).

2. *Network architecture adjustments:* first four convolutional layers of the vgg16 basenet are replaced with Averaged Oriented Response Convolutions (A-ORConvs) Wang et al. [2018c]. A-ORConvs was first introduced to take into account the scene orientation of the remote sensing images and encode the orientation-related information in the feature map. A Squeeze-ORAlign (S-ORAlign) layer is then used after the fourth convolutional layer to find the main response channel. The rest of the network is divided into two branches: landmark localization and category and attribute classification. The localization branch has the same structure as in Liu and Lu [2018]. Spatial attention is extracted from the landmark localization branch, and two other attention modules, category, and channel, that take conv4 output features as input. Category attention is modeled using a U-Net structure Ronneberger et al. [2015] and learns in an unsupervised manner, regions of an image that are important for classification. The channel attention is based on the Squeeze-and-Excitation block Hu et al. [2018]. These attentions are then combined with conv4 output features. The resulting features are passed through conv5-1,2,3 layers, before getting divided into two fully connected branches for category classification and attribute prediction.

3 Important Fashion Datasets

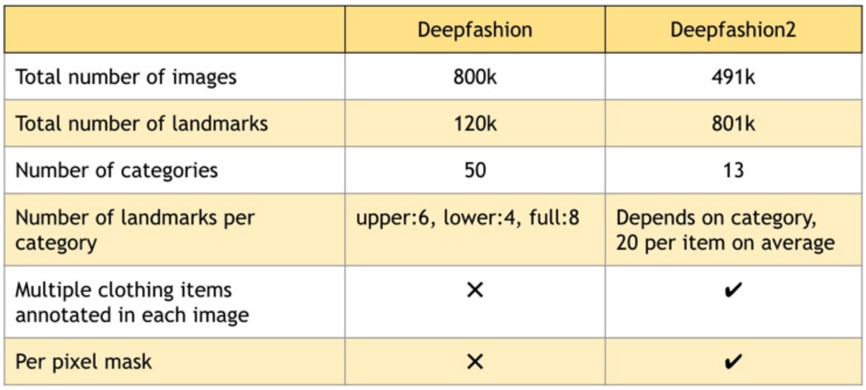
There are three main datasets for landmark detection for clothing items. Deepfashion Liu et al. [2016a] contains more than 800,000 images in total. The dataset provides benchmarks for different tasks, two of which are of interest for landmark localization: landmark localization subset and category classification and attribute prediction subset. These two subsets are richly annotated with massive attributes, clothing landmarks, and correspondence of images taken under different scenarios including store, street snapshot, and consumer. Three main types of clothing items are provided, upper, lower, and full body; each type of clothing has a fixed number of landmarks. Each landmark has one of the three visibility states, ”0” for visible, ”1” for invisible/occluded, and ”2” for truncated/cut-off.
Images in the category classification and attribute prediction subset are also labeled with 50 categories, and 1, 000 descriptive attributes.



cat	L.Collar	R.Collar	L.Sleeve	R.Sleeve	L.Waist	R.Waist	L.Hem	R.Hem
upper	✓	✓	✓	✓			✓	✓
lower					✓	✓	✓	✓
full	✓	✓	✓	✓	✓	✓	✓	✓

Figure 7: Keypoints in different categories of clothing

Deepfashion2 dataset Ge et al. [2019] is introduced to address some issues of deepfashion dataset, such as single clothing-item per image, sparse landmarks (4∼8 only), and no per-pixel masks. Compared to deepfashion, this dataset has less variety in categories (13 instead of 50), but each category is provided with a different set of landmarks, which is denser compared to deepfashion (about 20 for each category instead of at most 8 for full body category of deepfashion). The following table summarises the quantitative differences between these two datasets.



	Deepfashion	Deepfashion2
Total number of images	800k	491k
Total number of landmarks	120k	801k
Number of categories	50	13
Number of landmarks per category	upper:6, lower:4, full:8	Depends on category, 20 per item on average
Multiple clothing items annotated in each image	✕	✔
Per pixel mask	✕	✔

Figure 8: Deepfashion vs Deepfashion2

FashionAi Zou et al. [2019] is a fashion dataset with a hierarchical structure from the perspective of design, meaning The design regions (called attribute dimension, e.g. sleeve length, sleeve cuff, collar design, etc.) and their belonging designs (called attribute values, e.g. cap sleeves) are summarized in a hieratical structure. A total of 24 keypoint types are defined, along with 68 attribute dimensions, 245 attribute values, and 6 categories. The hierarchical structure reduces the trained attributes and also improves the comprehensiveness of the dataset at the same time.

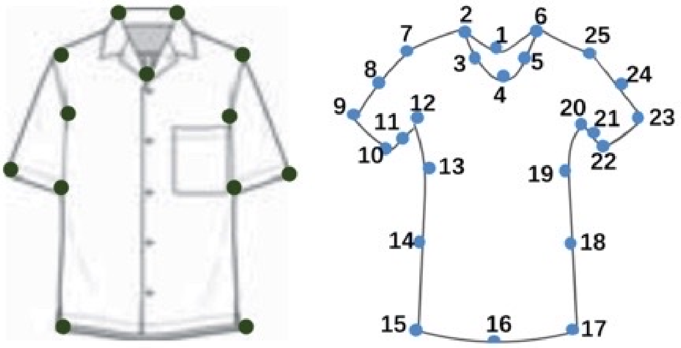


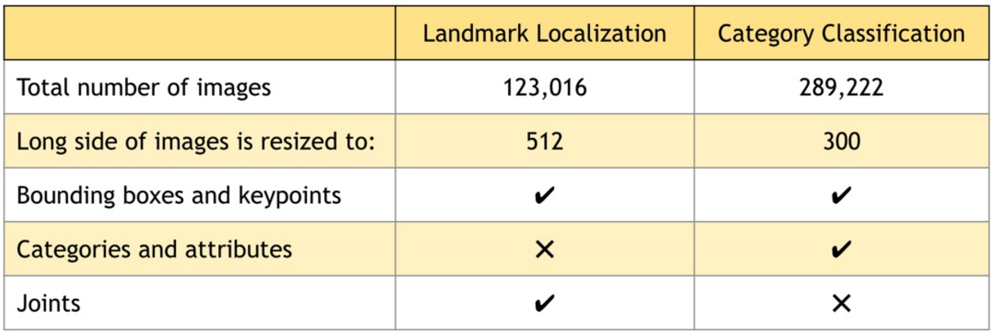
Figure 9: Landmarks for short sleeve shirt category in FashionAi vs deepfashion2

In this project, we mainly have worked with deepfashion dataset for the following reasons:

(a) The great number of works using the same dataset would facilitate the comparison.

(b) As the number of landmarks is less than deepfashion2, it would probably be an easier dataset to train on.

deepfashion dataset itself, have 5 different benchmarks, including landmark localization and category classification. Figure 10 summarizes main differences between these two parts of the dataset that have been used in this project.



	Landmark Localization	Category Classification
Total number of images	123,016	289,222
Long side of images is resized to:	512	300
Bounding boxes and keypoints	✔	✔
Categories and attributes	✕	✔
Joints	✔	✕

Figure 10: Comparison between data provided for landmark localization and category classification. In both subsets, the aspect ratios of original images are kept unchanged.

Clothing items in deepfashion are divided into three main categories of upper, lower, and full body. The set of landmark defined for each category is different 3; for example, upper body clothes have right collar keypoint, whereas for lower body clothes this keypoint is not defined. Different landmarks are: right collar, left collar, right sleeve, left sleeve, right waistline, left waistline, right hem, and left hem.

cat	L.Col	R.Col	L.Sle	R.Sle	L.Wai	R.Wai	L.Hem	R.Hem
upper	✓	✓	✓	✓			✓	✓
lower					✓	✓	✓	✓
full	✓	✓	✓	✓	✓	✓	✓	✓

4 Landmark Localization

In this project, to perform a landmark localization task, an already established pipeline, pifpaf Kreiss et al. [2019, 2021], is used and the main focus is on tuning the hyperparameters. In the following, we describe the main parts of this pipeline, before explaining its plugin for deepfashion exclusively.

4.1 pifpaf

4.1.1 Introduction

Part intensity fields and part association fields (pifpaf) is a pose detection pipeline, specially designed to address challenges that arise in autonomous navigation settings; low-resolution images along with a wide viewing angle and the existence of various scales of humans are the main difficulties in this application, compared to usual pose detection tasks.
On top of addressing the mentioned challenges, pifpaf undertakes a bottom-up approach for detecting poses (meaning it first starts by detecting joints and then tries to associate them; top-down techniques first use a person detector and then output joints within the detected bounding boxes). This approach is helpful when the detection scene is occluded and bounding boxes clash.
Remark: landmark, keypoint, and joint is used interchangeably in the following sections.

4.1.2 Method

Pifpaf consists of a base neural network architecture with two head networks, pif and paf.
PIF detects and localizes joints (or more broadly, body parts). It has a composite structure; at every output location (i, j), a PIF predicts a confidence c, a vector (x, y), pointing to the closest body part of a certain type, with spread b and a scale σ (annotated as $p^{ij} = \{p^{ij}_c, p^{ij}_x, p^{ij}_y, p^{ij}_b, p^{ij}_\sigma\}$). The predicted confidence map, p_c, is coarse but can be refined when fused with other components of PIF. To create this high-resolution confidence map the following equation is used:

$$f(x, y) = \sum_{ij} p^{ij}_c \mathcal{N}(x, y|p^{ij}_x, p^{ij}_y, p^{ij}_\sigma)$$

where $f(x, y)$ is the refined confidence map and $\mathcal{N}(x, y|p^{ij}_x, p^{ij}_y, p^{ij}_\sigma)$ is an unnormalized gaussian, with width p_σ over the regressed targets from the Part Intensity Field (the points that are pointed at from each position in the output; for example, (p^{ij}_x, p^{ij}_y) is the coordinate of the position that the vector map of PIF points to from the position (i, j).
The above equation also points out the grid-free nature of the localization.

PAF head constructs poses by connecting joints together. At each output position for a specific landmark connection (from the defined skeleton on the dataset), it predicts two vectors pointing to the two joints/landmarks that this association is connecting, confidence of the connection, and two spatial precisions for the regressions (annotated as $a^{ij} = \{a_c^{ij}, a_{x1}^{ij}, a_{y1}^{ij}, a_{b1}^{ij}, a_{x2}^{ij}, a_{y2}^{ij}, a_{b2}^{ij}\}$).

To give an illustrative example, in the COCO dataset, there are 19 connections for the person class; one of these connections is the right-knee-to-right-ankle association; the PAF predicts all the mentioned fields for this association at every output location.

Two steps are taken to determine PAF fields for an association, such as right-knee-to-right-ankle, at a particular feature map location:

(a) the closest joint of either of the two ends of the association is determined for each output location.

(b) the ground truth pose (or skeleton) is used to find the vector component representing this association.

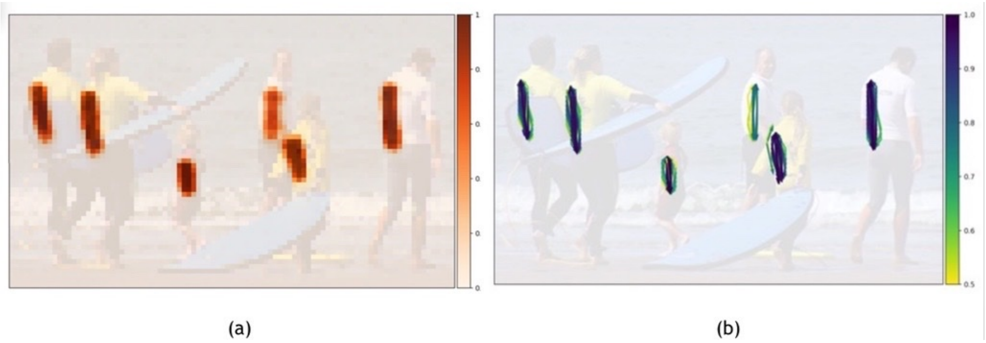


Figure 11: (a) Confidence field of PAF for the left shoulder to left hip association in COCO dataset for human pose detection. Every location of the feature map is the origin of two vectors that point to the shoulder and hip. The shown map is the confidence of associations at their origin. (b) Vector field of PAF for feature map locations where a_c is more than a certain threshold.

PIFPAF Decoder converts the output of PIF and PAF to sets of landmark coordinates. The decoding process is fast, greedy, and similar to the method used in Papandreou et al. [2018]:

(a) A new pose is seeded from the highest values in the high-resolution confidence map of PIF.

(b) Given a starting joint such as x, a set of scores is calculated over a PAF association with one extremity at x.

$$s(\boldsymbol{a}, \overrightarrow{x}) = a_c \; exp(-\frac{\|\overrightarrow{x} - \overrightarrow{a_1}\|}{b_1}) \; f_2(a_{x2}, a_{y2})$$

In the above equation, $\boldsymbol{a}$ represents an association composite field over one output pixel, a_c is the PAF confidence value at that location, and $f_2(a_{x2}, a_{y2})$ is the high-resolution part intensity field at the second vector's target location.
The second extremity with the highest score association is considered the second joint's new position. To confirm the position of the proposed joint, the same process is done starting from the second joint. Once a connection to a new joint has been made, this decision is final and the joints at the extremities of this connection have fixed positions.

(c) The same operation is repeated (starting from the last positioned landmark) until a full pose is completed.

4.1.3 Loss Functions

The smooth L1 loss is used for the confidence component of PIF, L1 loss for the scale component, and Laplace losses for all vectorial components.
As mentioned earlier, pifpaf has a way to deal with scale diversity of a human pose; scale dependence is taken into account in the regression loss SmoothL1 Girshick [2015] or Laplace loss Kendall and Gal [2017].
SmoothL1 loss produces softer gradients around the origin and it is formulated as:

$$L1_{smooth} = \begin{cases} 0.5(x-\mu)^2/r & \text{if } |x-\mu| < r \\ |x-\mu| - 0.5r & \text{otherwise} \end{cases}$$

to take the scale component into consideration via the r parameter; For a person instance bounding box area of A and keypoint size of σ, r is set proportionally to $\sqrt{A}\sigma$.

Laplace loss is also a L1-type loss that is attenuated via the predicted spread b in PAF fields:

$$L = |x-\mu|/b + log(2b)$$

4.2 PifPaf Plugin for Deepfashion Landmark Localization

4.2.1 Training

For experiments, the Deepfashion dataset is used. In order to use PifPaf, Deepfashion dataset should first be converted to COCO dataset style. This task includes mapping visibilities and writing over the keypoints and bounding boxes.
In Deepfashion visibility convention, ”0” represents visible, ”1” invisible/occluded, and ”2” truncated/cut-off. In COCO visibility convention ”0” represents not labeled (in which case $x = y = 0$), ”1” labeled but not visible, and ”2” labeled and visible. the mapping from Deepfashion to COCO visibility is hence $\{0 \rightarrow 2, 1 \rightarrow 1, 2 \rightarrow 0\}$.
After converting Deepfashion dataset to COCO style dataset, plugin parameters should be set: 1. skeleton 2. preprocessing functions.

1. For the primary experiments, two skeleton types have been tested, one where collars are not connected to waistlines and the other where they are connected (Figure 12). Models trained with skeleton B obtained more promising results in primary evaluations; for this reason, succeeding trainings were conducted using B skeleton.

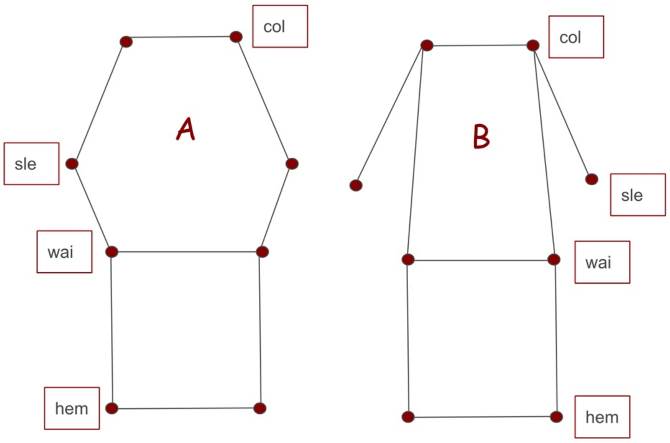


Figure 12: two skeleton types for deepfashion dataset

2. A number of data augmentation techniques have been tried out to train different models, but they all have these main components: 1.color jittering with 40% variation in brightness and saturation 2.random horizontal flipping 3.random cropping and padding, to have images of the same size in the dataloader.

Figure 15 summarizes different pre-processings, on top of mentioned common ones used in all cases. Most of the trainings have been conducted on category classification and attribute prediction subset of deepfashion, as it was also the practice of previous works. However, a few models have been trained and evaluated on the landmark localization subset for the sake of comparison; these latter are indicated by dashed lines in Figure 15.
A few examples of annotated images after processing are shown in Figures 13 and 14.

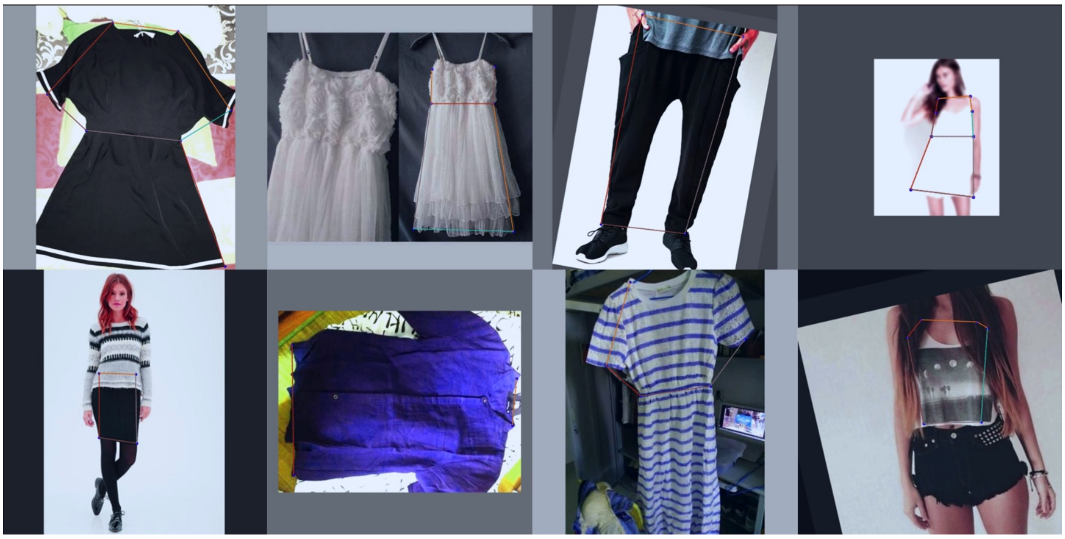


Figure 13: Examples of annotated processed raw images; some are rotated by 20°

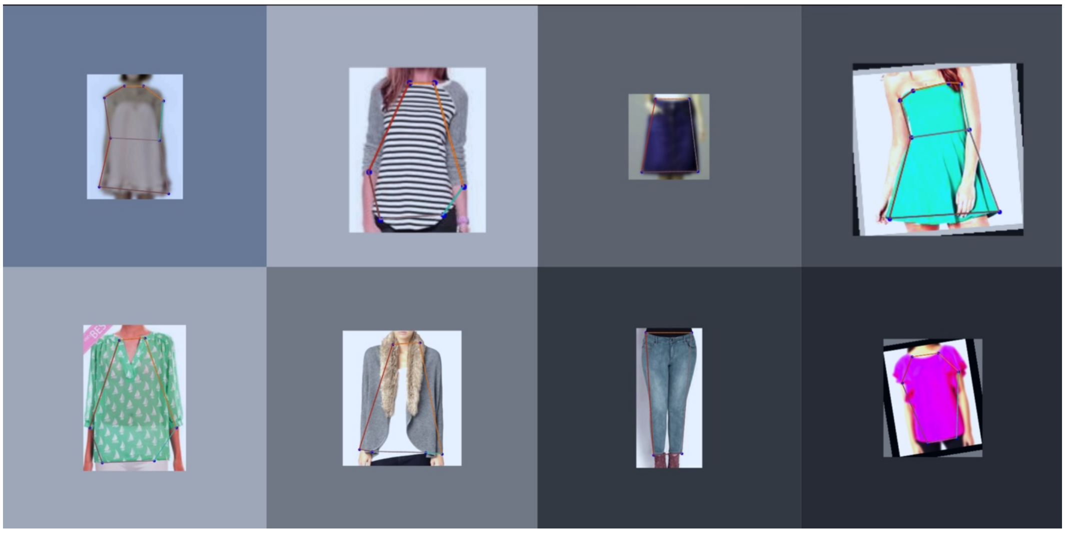


Figure 14: Examples of annotated processed images cropped by their bounding box; some are rotated by 20°

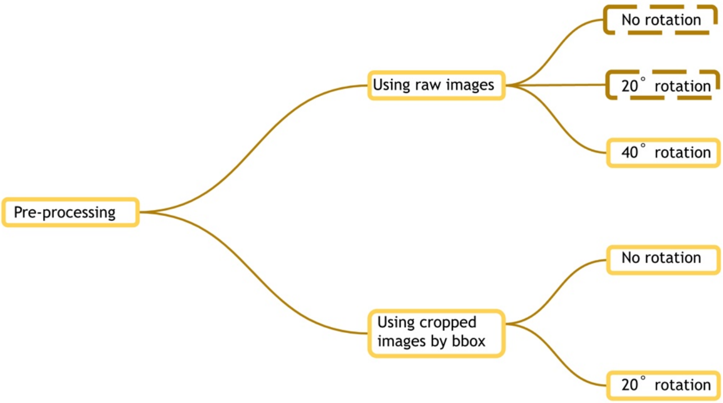


Figure 15: Pre-processings have been carried out over the category classification subset of the dataset. The dashed pre-processing has also been tried with the landmark localization dataset.

In the training, we consider two main categories: 1. Single model and 2. Combined model. The single model predicts all the landmarks for all of the three categories (upper, lower, full) together, using one 'single' model; and the keypoints that are not present in a certain category should be predicted as non-visible. For example, no collar keypoint is defined for lower body clothes; hence, the target will set the visibility of both collars to zero. The loss over invisible points will not be back-propagated.

On the other hand, a combined model uses separate predictors for different categories: a model is trained on each category; a different model for each of the upper, lower, and full body categories to localize their corresponding keypoints.

An exhaustive list of the different trainings, along with their backbone and pre-processing, can be found in the appendix 7.2.

Training mistake

In my first try to train and evaluate the combined model, I realized that the detection of the combined model is weaker than the single model; one example of such can be seen in Figure 16, where the left image represents the detection made by a single model, in the middle image no detection is made by the combined model and the right image is the ground truth annotation.

The reason was that the model had not been trained well, because of the low number of samples for each category separately. In the next trials, the annotation has been accumulated: for example, in training samples for the upper body model, also the full body images have been added, along with their first six keypoint annotations. This last change was useful as expected. The reported results for the combined strategy correspond to model that uses greater number of training samples.

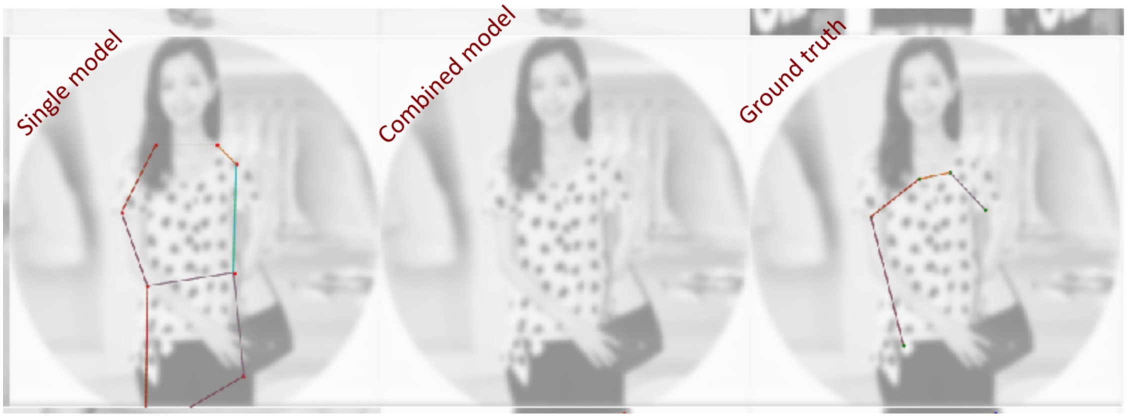


Figure 16

4.2.2 Loss explosion during training

For certain backbones and certain pre-processings, we experience loss explosion during trainins. The list of trainings, and their explosion state (if no loss explosion is experienced, their last training epoch is indicated) can be found in appendix 7.2.

A try to solve the loss explosion

To solve the problem, I tried changing the loss functions. The change was inspired by the two following remarks:

(a) it has been observed that for certain backbones, large learning rates, 10^{-4} for instance, makes the loss more stable than using a smaller learning rate, 10^{-5} for instance; for example, for certain backbones, loss explosion happens in an early epoch when a learning rate of $10^{-5}$is used, but the training continues till the desired epoch for a learning rate of 10^{-4}.
On top of that, the experiments demonstrated that training may resume with a lower learning rate when starting with a model trained with a higher learning rate for several epochs.

(b) as mentioned earlier in the Loss section, Laplace loss uses the predicted spread, b, as a way to take the diversity of scales into account:

$$L = |x - \mu|/b + log(2b)$$

During the initial phase of training, predictions are poor and the network better predicts large spreads (b) to reduce the first term at the cost of a logarithmic penalty from the second term. As the network learns to predict better, it starts to predict smaller values for b to reduce the logarithmic penalty while the prediction loss gets also smaller.

From the two above observations, I made the guess that the loss explosion is probably caused by large predictions for the spread; observing the loss terms was also confirming the guess The regression loss was also large in those early epochs.
Based on the above guess, I changed the regression loss from Laplace to a simple smooth L1 loss with a fixed radius. Unfortunately, the latter change did not stop those training from experiencing loss explosion.

To address/understand this issue, I studied the structure of backbones used in training, different resnets and shufflenets, along with the effect of different pre-processings. It has been tried to point out some observations that can possibly lead to understanding the reason behind the loss explosion:

(a) Large networks are more prone to loss explosion.

(b) Images that are rotated by even a small angle are generally more prone to loss explosion than the case where no rotation is applied as data pre-processing.

(c) The models trained with larger random rotations are trained better if not experiencing a loss explosion.

(d) When the images are cropped by their bounding box before pre-processing, and the same square edge is used for cropping/padding images, the model is more prone to explosion (the effective space taken by the image in the processed one is smaller)

Recent guess to explain loss explosion

The pre-processing shows that the background color (not a part of an image) can be different for different samples. This latter can a possible reason for loss explosion that is being further studied.

4.2.3 Evaluation Metrics

The main metric on deepfashion dataset is normalized error (NE). It is defined as the Euclidean distance between the position of the ground truth and predicted keypoints in the normalized coordinate space (i.e. divided by the width/height of the image).

$$\frac{\sum_{n=1}^{n_{vis}} \sqrt{(x_p - x_{gt})^2 + (y_p - y_{gt})^2}}{resolution}$$

Another metric has also been used in the paper that introduced deepfashion, Liu et al. [2016a] which is the percentage of detected landmarks (PDL) Toshev and Szegedy [2014]. However, this metric is surprisingly not reported in its succeeding works.
In this project, to quantify the detection performance separately from prediction performance, we introduce other metrics (and also consider PDL as a metric):

(a) *NE over confidently predicted keypoints.* Pifpaf plugin also includes a prediction score for each keypoint. By setting a threshold, we can filter confident predictions, take its intersection with ground truth visible keypoints, and then compute the normalized error over this intersection.

(b) *percentage of detected keypoints not present in the image.*

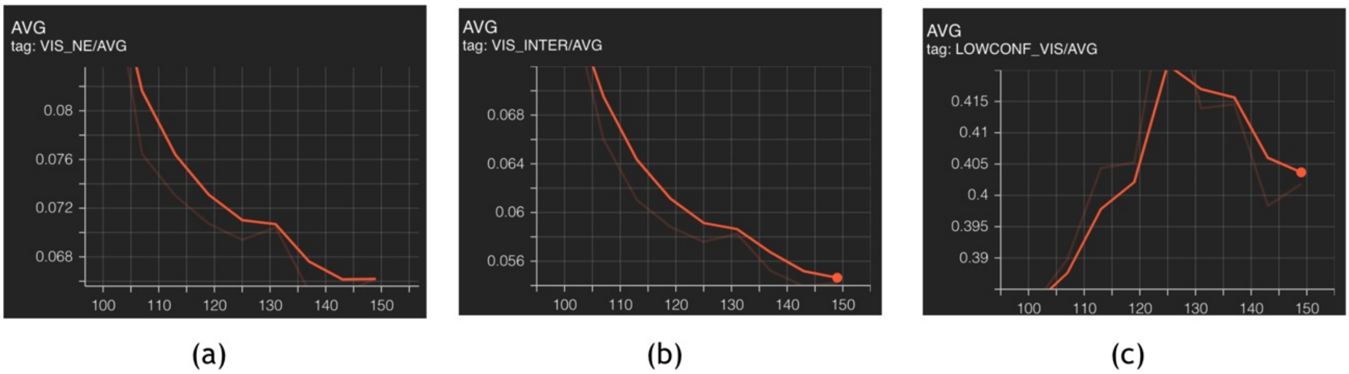
(c) *percentage of detected keypoints not visible in the image*

A keypoint can be present in the image but not visible; this happens for example when two pieces of clothing are occluding each other.

In Figure 17, we can see curves for a) average NE for all landmarks in different epochs b) average NE for confidently predicted landmarks in different epochs c) average NE for not confidently predicted landmarks in different epochs. Remark: not the best model is used in these curves, but one of the primary trainings.

4.2.4 Evaluation and inference

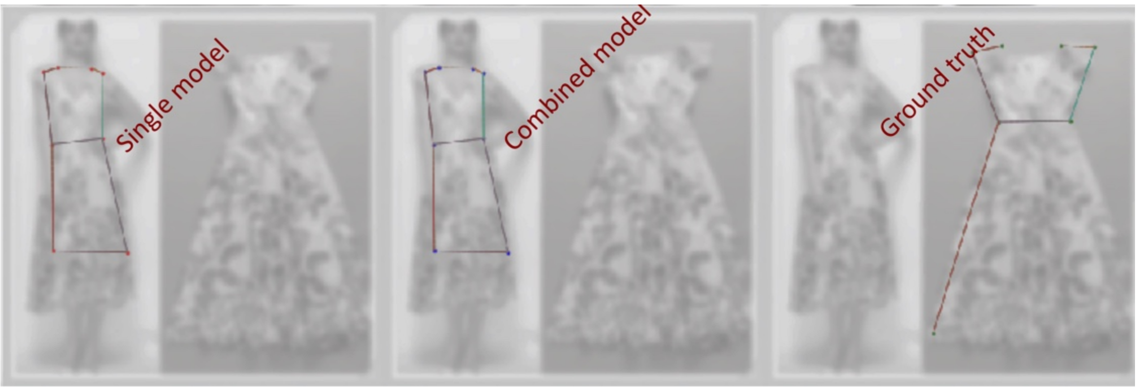
Single model case: Pifpaf plugin may output more than one prediction (instance) per image when several clothing items are present. However, in the dataset, just one clothing item is annotated. To find the prediction that corresponds to the annotated instance in the ground truth, we have a number of choices: to use the ground truth itself to find the closest pair (comparing the distance of predictions with annotated instance), to decide based on the prediction score and choose the prediction with the highest one, or to use the ground truth bounding box of the annotated instance to filter out not suitable predictions.
Using the ground truth itself to find the closest pair gives the best results compared to two



(a) (b) (c)

Figure 17

other inference choices, as is expected; it however is not fair for comparison purposes. For fair comparison and comparability with other works, the image is first cropped by a bounding box, and then the prediction is made over the cropped image. Cropping is also useful when more than one instance is in the image and the most confidently predicted instance is not annotated one. One example of such is given in the Figure ??, where the predicted instance by both single and combined models is different from the annotated instance.



Combined model case: As in the case of the single model, to find the prediction that corresponds to the annotated instance in the ground truth, we have a number of choices. Let's say we have M_1, M_2, M_3, three models respectively for localizing keypoints of upper, lower, and full body. One option is to pick the right model based on the ground truth category, and the second option is to use the model that gives the prediction with the highest score.
The first method resulted in better predictions (lower NE) and the reported results are also obtained in the same manner; it, however, takes advantage of ground truth information and may not be used in inference mode.

4.2.5 Results and Comparison with Other Works

Category Classification subset of dataset

Over the shufflenet16 backbone and test subset of the category classification dataset, six models are evaluated: two models trained on processed cropped images with 20° rotation and no rotation, two models trained on processed raw images with 40° rotation (one model pretrained on wholebody dataset), and two models trained on processed raw images with 20° rotation and no rotation.
The best result was obtained by a shufflenet16 model pretrained on wholebody dataset, and fine-tuned on processed raw images with random 40° rotations. The model trained with the same setup but from scratch performs on par with this latter.
Other works have mostly reported evaluation on category classification subset of deepfashion, and so here we make a comparison with other techniques on this subset.

Table 1: Average normalized error (over test subset of deepfashion-c) reported in different works

n	**paper**	**year**	**conference/journal**	**average NE**
1	FashionNetLiu et al. [2016a]	2016	IEEE computer vision and pattern recognition	0.0872
2	DFALiu et al. [2016b]	2016	European Computer Vision	0.0660
3	DLANYan et al. [2017]	2017	ACM Multimedia	0.0643
4	BCRNNWang et al. [2018a]	2018	IEEE computer vision and pattern recognition	0.0484
5	Liu and Lu [2018]	2018	ECCV	0.0474
6	Shajini and Ramanan [2021]	2021	The Visual Computer	0.0425
7	SFLMHe et al. [2021]	2021	Multimedia Modeling	0.0398
8	Lee et al. [2019]	2019	IEEE/CVF	0.0393
9	Chen et al. [2019]	2019	IEEE/CVF	0.0342
10	Xie and Chen [2022]	2022	ICGIP	0.0334
11	SANLLi et al. [2019]	2019	IEEE Multimedia and Expo (ICME)	0.0299
12	Chen et al. [2020]	2020	IOS Press	0.0297

n	paper	year	conference/journal	average NE
13	Wang et al. [2021]	2021	IEEE/ ICTAI	0.0189
14	pifpaf plugin	2021	IEEE Transactions on Intelligent Transportation Systems	0.0370

Table 2: Individual normalized error (over test subset of deepfashion-c) reported in different works.

n	L.Collar	R.Collar	L.Sleeve	R.Sleeve	L.Waist	R.Waist	L.Hem	R.Hem
1	0.0854	0.0902	0.0973	0.0935	0.0854	0.0845	0.0812	0.0823
2	0.0628	0.0637	0.0658	0.0621	0.0726	0.0702	0.0658	0.0663
3	0.0570	0.0611	0.0672	0.0647	0.0703	0.0694	0.0624	0.0627
4	0.0415	0.0404	0.0496	0.0449	0.0502	0.0523	0.0537	0.0551
5	0.0332	0.0346	0.0487	0.0519	0.0422	0.0429	0.0620	0.0639
6	0.0323	0.0334	0.0443	0.0472	0.0368	0.0370	0.0553	0.0558
7	0.0316	0.0330	0.0428	0.0442	0.0378	0.0369	0.0454	0.0466
8	0.0312	0.0324	0.0427	0.0434	0.0361	0.0373	0.0442	0.0475
9	0.0295	0.0297	0.0363	0.0361	0.0311	0.0313	0.0394	0.0402
10	0.0247	0.0267	0.0314	0.0389	0.0312	0.0336	0.0357	0.0380
11	0.0249	0.0256	0.0341	0.0348	0.0260	0.0259	0.0338	0.0346
12	0.0256	0.0251	0.0318	0.0324	0.0271	0.0286	0.0328	0.0341
13	0.0154	0.0155	0.0224	0.0224	0.0162	0.0159	0.0203	0.0207
14	0.0267	0.0271	0.0408	0.0417	0.0509	0.0519	0.0281	0.0286

Using a combined structure we obtain normalized errors reported in table 3. The backbone used for upper and lower body categories is a shufflenetv2k16 trained from scratch for 200 epochs, and for the full body category, the model is shufflenetv2k30 trained for 60 epochs. The results for each individual category are also reported in the table 4. The evaluation with artificial skeleton places landmarks on predefined positions based on the size of the image, instead of having all the predictions at a corner, when no prediction is made by the model.

Table 3: Normalized error using a combined model

n	method	AVG NE	L.Col	R.Col	L.Sle	R.Sle	L.Wai	R.Wai	L.Hem	R.Hem
1	without artificial skeleton	0.0571	0.0474	0.0497	0.0700	0.0674	0.0642	0.0644	0.0470	0.0467
2	with artificial skeleton	0.0630	0.0538	0.0569	0.0756	0.0770	0.0687	0.0693	0.0517	0.0512

Table 4: Normalized error for different categories using a combined model

n	category	L.Col	R.Col	L.Sle	R.Sle	L.Wai	R.Wai	L.Hem	R.Hem
1	upper body	0.0674	0.0726	0.0867	0.0844	0.0958	0.0967		
2	lower body					0.0537	0.0554	0.0596	0.0597
3	full body	0.0301	0.0297	0.0569	0.0543	0.0498	0.04918	0.0466	0.0455

Landmark Localization subset of dataset

Over the shufflenet16 backbone and test subset of the landmark localization dataset, two models are evaluated: both trained on processed raw images, one with 20° random rotations, and the other with no rotation.

The results are reported for a shufflenetv2k16 model trained from scratch with random 20° rotations (line 1) and a shufflenetv2k16 model pretrained on wholebody and fine-tuned for 100 epochs (line 2), are reported in table 5.

Table 5: normalized error using two models on deepfashion-l dataset

n	AVG NE	L.Col	R.Col	L.Sle	R.Sle	L.Wai	R.Wai	L.Hem	R.Hem
1	0.0573	0.0333	0.0335	0.0568	0.0601	0.0697	0.0733	0.0623	0.0692
2	0.0693	0.0380	0.0382	0.0639	0.0643	0.0921	0.0949	0.0782	0.0846

4.2.6 Other experiments

One interesting experiment is to evaluate the models on different scales. To simulate this variation, we can crop and pad images to different sizes. It is of course expected to have the best performance when the evaluation and training sizes are the same. To observe the change in performance with cropping/padding size variation, we plot the error/crop-size curve with one of the trained models (the experiment was done before obtaining the best model, and so the results are not optimal; the correlation however should remain the same across different models.).

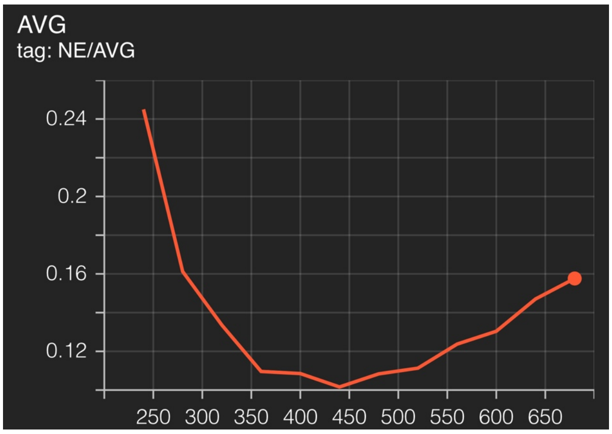


Figure 18: Normalized error (vertical axis) over cropping/padding size

The code for converting deepfashion to coco style, training (the best model that was obtained), and evaluation of this plugin is available at https://github.com/Pegah-source/pifpaf_deepfashion.git.

5 Clothing Classification

the implementation of this part is inspired by Mordan et al. [2021], which uses openpifpaf as a base network and then adds different heads (like pif and paf heads in the original pifpaf) to predict pedestrian attributes. In the following, we give a quick introduction to this work and then explain our similar method for the case of clothing classification.

5.1 Detecting 32 Pedestrian Attributes for Autonomous Vehicles

Mordan et al. [2021] Proposes Multi-Task Learning (MTL) model to jointly detect pedestrians and to predict 32 pedestrian attributes from a single image.
The model has four main components: 1.Base model, which is the same as the base network of openpifpaf 2.Fork normalization, which is a technique to effectively handle the loss in both the base network and the heads when the number of tasks increases 3.Head networks to predict different attributes in an image 4.Decoders to convert a set of field predictions made on the whole image to attribute predictions for different instances in an image.
Heads are first employed to predict attributes that would be used in *instance detection.* To this end, they predict two fields: 1. *confidence score field*: Confidence score indicates how likely cells (x,y) are to belong to pedestrian instances. 2 *a vector field to localize the centers* of the bounding boxes of pedestrian instances.
A threshold is used over the confidence field to filter out background pixels and give a coarse segmentation of instances in the image. However, due to the large stride of the network, this coarse segmentation is not enough to separate instances in more crowded scenes, but just to filter out certain cells and keep the others as possible positions of instances.
Retained cells along with the vector field to the center of instances are then used to obtain a number of estimated centers. The estimated centers are expected to be more representative of different instances than the segmentation map of instances in the last step. OPTICS algorithm Ankerst et al. [1999] is then used to cluster estimated centers into sets of different instances.

After the detection step, Given an attribute f, the prediction is made for a pedestrian instance by a voting system over the cells in the cluster of this instance on the field associated with the attribute f.

The above work presents a multi-tasking framework for pedestrian attribute prediction that envelopes different tasks into one single model. Multi-tasking frameworks may face disadvantages with respect to single-task models when the number of tasks increases. This work introduces fork normalization as a way to handle loss imbalance between the main network and heads in multi-tasking frameworks. Please find more details about the proposed solution in the appendix 7.

5.2 Clothing Classification Based on Openpifpaf

Inspired by the above structure for detecting pedestrian instances and their attributes, we propose a similar network for jointly learning keypoint localization and category classification of clothing items.

To this end we consider the following heads to be built on top of the base openpifpaf network: 1.Pif head 2.Paf head 3.Category classification head.
The first two heads are already implemented in openpifpaf and we tried to explain them shortly in the previous sections 4. In contrast, we add a third head to the network, with a simple architecture: a convolution layer with a kernel size of one is applied to the output of the base network to adjust the number of channels to the number of categories. Then the maximum value is picked up from the two-dimensional output corresponding to each category. At last, a softmax layer is added to put the values into the right range. The loss function for this head is cross-entropy loss.
We add fork normalization on top of the base network, before the heads.
The main challenge in this task was to maintain the versatility of the implementation with respect to openpifpaf. In other words, it was important for us to add components on top of openpifpaf and not to apply changes directly in its structure. One of the parts that had to remain unchanged was the predictor in openpifpaf: this part calls a decoder to convert the output of heads into final predictions. Hence, the decoder associated with PIF and PAF heads had to be integrated with the decoder of the category classification head. On top of that, some structures had to be re-implemented so that the three heads are treated in one piece.
Just one model (backbone and learning rate) has been trained with this structure: shufflenetv2k30 backbone, 10^{-4} learning rate, momentum of 0.95, weight-decay of 10^{-5}, and trained for 150 epochs. Results are reported in Table 6.

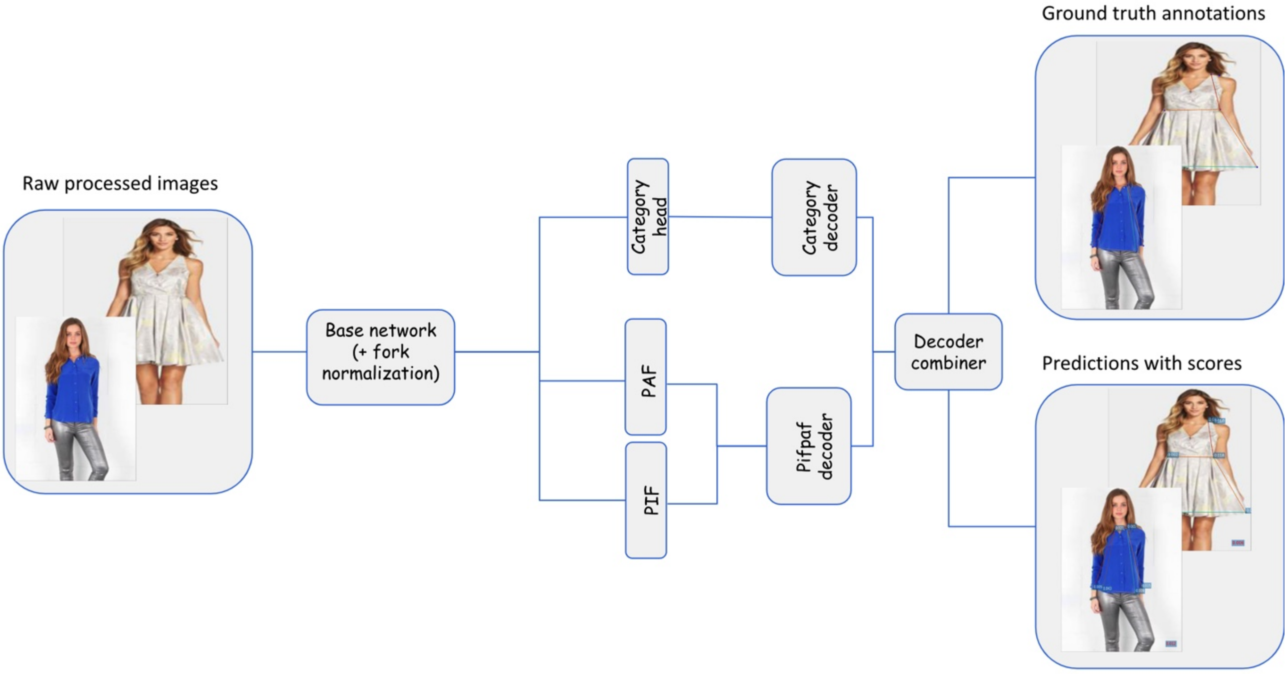


Figure 19

The code for training this plugin is available at https://github.com/Pegah-source/openpifpaf_clothing_classification.git.

Table 6: The results of clothing classification for different works

Method	Category	
	top3	top5
Chen et al. [2012]	43.73	66.26
Huang et al. [2015]	59.48	79.58
Liu et al. [2016a]	82.58	90.17
Lu et al. [2017]	86.72	92.51
Corbiere et al. [2017]	86.30	92.80
Wang et al. [2018a]	90.99	95.78
Singh et al. [2021]	91.05	95.35
Liu and Lu [2018]	91.16	96.12
Zhang et al. [2020]	91.99	96.44
pifpaf plugin	89.51	93.64

6 Future Work

There are a few directions for future works at the continuation of this project:

(a) Loss explosion can be studied in more depth to better understand its source.

(b) I tried in this project to also train a landmark localization plugin for deepfashion2 dataset. But the training was unsuccessful, probably due to the great number of keypoints defined in the dataset. However, it most probably should be possible to train a working plugin, by adapting different parameters. One direction for future work can be to try different ideas for this plugin to work; separating 13 different categories and training a model for each is a possibility (inspired by Lin [2020]).

(c) The openpifpaf plugin has the advantage of being able to detect keypoints for several instances in an image, unlike most of the other methods. A direction for future work can be to define a quantitative metric or structured way to prove this advantage.

7 Appendix

7.1 Loss Imbalance and Fork Normalization

In the usual case, the loss $\mathcal{L}$ of the backbone in a multi-tasking model is the sum of all the losses across different branches. In this case, for a feature $\mathfrak{z}$ in the backbone, the gradient $\mathcal{L}/\mathfrak{z}$ can be bounded by the norms of all task-specific gradients on the same feature:

$$\mathcal{L} = \sum_{t=1}^{T} \lambda_t \mathcal{L}_t$$

$$\|\frac{d\mathcal{L}}{d\mathfrak{z}}\| \leq \sum_{t=1}^{T} \|\lambda_t \frac{d\mathcal{L}_t}{d\mathfrak{z}}\| \leq T \max_{1 \leq t \leq T} \|\lambda_t \frac{d\mathcal{L}_t}{d\mathfrak{z}}\|$$

Assuming the weights λ_t are not normalized by the number of tasks, T, the bound increases with T. The bound does not indicate that the loss should increase by the number of tasks, but this increase has been observed in the experiments conducted in Mordan et al. [2021].
On the other hand, normalizing the weights λ_t by T would lead to decrease in the norm of a branch loss with respect to its task specific feature, with a factor of T.
In order to get gradient bounds, in the backbone and also branches, independent of the number of tasks T, Mordan et al. [2021] proposed a modification in the way task-specific gradient join in the backward pass, before propagating through the backbone. They consider a set of T parameters $\kappa = (\kappa_1, \kappa_2, ..., \kappa_T)$ to weigh the gradients from different branches in the merging.

$$\begin{cases} \frac{d\mathcal{L}}{d\mathfrak{z}} = \sum_{t=1}^{T} \kappa_t \lambda_t \frac{d\mathcal{L}_t}{d\mathfrak{z}} & \text{for } \mathfrak{z} \text{ in backbone} \\ \frac{d\mathcal{L}}{d\mathfrak{z}_t} = \lambda_t \frac{d\mathcal{L}_t}{d\mathfrak{z}_t} & \text{for } \mathfrak{z}_t \text{ in branch } t \end{cases}$$

They first experiment with the proposition of having $\sum_{t=1}^{T} \kappa_t = 1$, but this lather yields to actual norm decreasing in the backbone with T. To address this, the relation between the gradient norm and the number of tasks is modeled as linear in log-log space. This results in new gradient weighting parameters of the form $\kappa_t = 1/T^{\beta}$, where β is a hyper-parameter. From the measurement data $\beta = 0.5$ under this modeling.

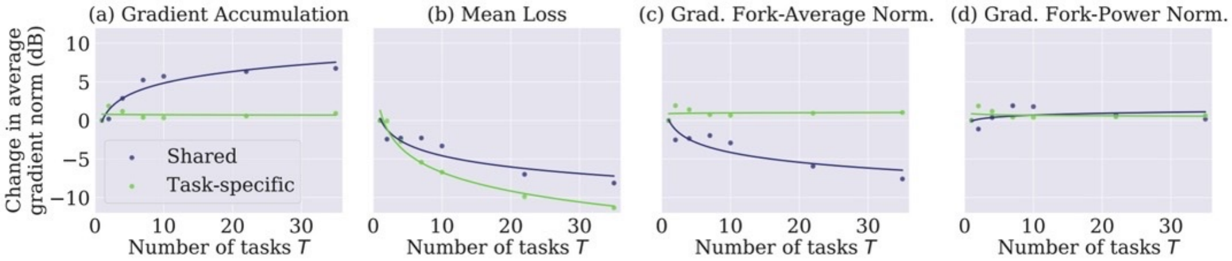
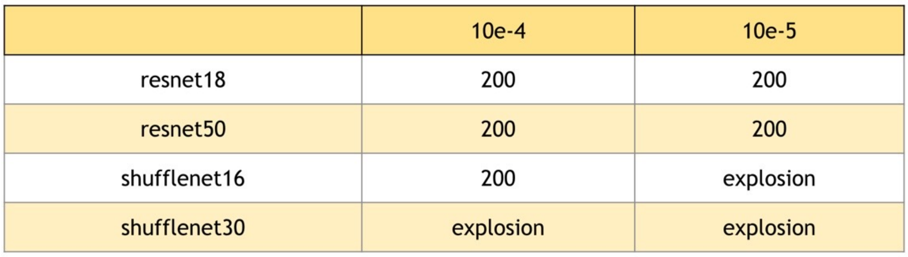


Figure 20: Relative change in average gradient norm in different setups

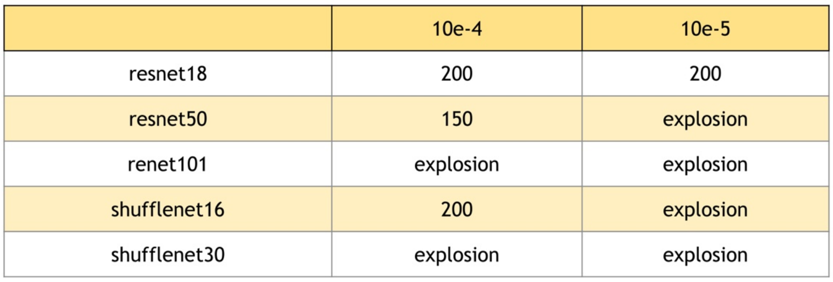
7.2 Exhaustive list of trainings

7.2.1 Landmark Localization subset of dataset:



	10e-4	10e-5
resnet18	200	200
resnet50	200	200
shufflenet16	200	explosion
shufflenet30	explosion	explosion

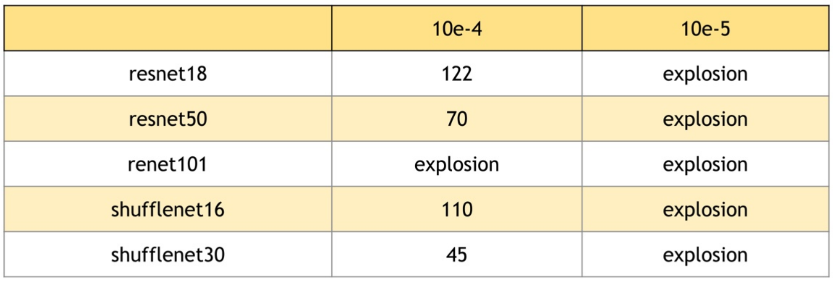
Figure 21: Pre-processing: No rotation and no bounding box cropping, trained on deepfashion-l



	10e-4	10e-5
resnet18	200	200
resnet50	150	explosion
renet101	explosion	explosion
shufflenet16	200	explosion
shufflenet30	explosion	explosion

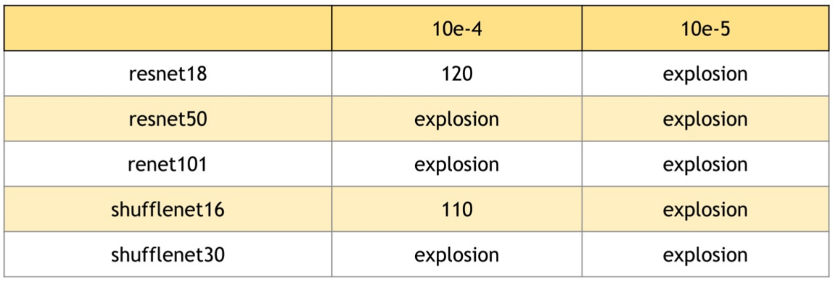
Figure 22: Pre-processing: random 20° rotation and no bounding box cropping, trained on deepfashion-l

7.2.2 Category Classification subset of dataset:



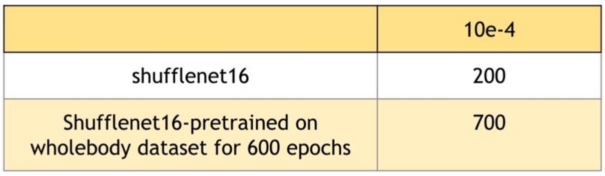
	10e-4	10e-5
resnet18	122	explosion
resnet50	70	explosion
renet101	explosion	explosion
shufflenet16	110	explosion
shufflenet30	45	explosion

Figure 23: Pre-processing: No rotation and no bounding box cropping, trained on deepfashion-c



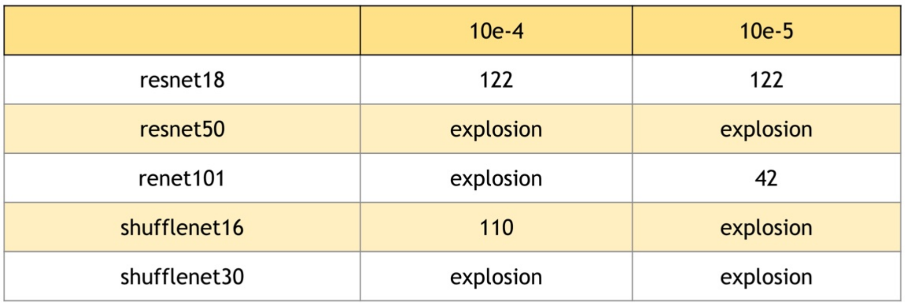
	10e-4	10e-5
resnet18	120	explosion
resnet50	explosion	explosion
renet101	explosion	explosion
shufflenet16	110	explosion
shufflenet30	explosion	explosion

Figure 24: Pre-processing: random 20° rotation and no bounding box cropping, trained on deepfashion-c



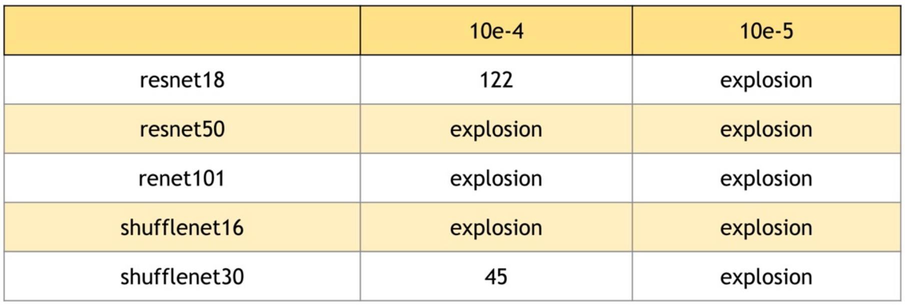
	10e-4
shufflenet16	200
Shufflenet16-pretrained on wholebody dataset for 600 epochs	700

Figure 25: Pre-processing: random 40° rotation and no bounding box cropping, trained on deepfashion-c



	10e-4	10e-5
resnet18	122	122
resnet50	explosion	explosion
renet101	explosion	42
shufflenet16	110	explosion
shufflenet30	explosion	explosion

Figure 26: Pre-processing: No rotation and bounding box cropping



	10e-4	10e-5
resnet18	122	explosion
resnet50	explosion	explosion
renet101	explosion	explosion
shufflenet16	explosion	explosion
shufflenet30	45	explosion

Figure 27: Pre-processing: random 20° rotation and bounding box cropping

References

Taylor Mordan, Matthieu Cord, Patrick Pérez, and Alexandre Alahi. Detecting 32 pedestrian attributes for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Yannis Kalantidis, Lyndon Kennedy, and Li-Jia Li. Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 105–112, 2013.

Shintami C Hidayati, Chuang-Wen You, Wen-Huang Cheng, and Kai-Lung Hua. Learning and recognition of clothing genres from full-body images. *IEEE transactions on cybernetics*, 48 (5):1647–1659, 2017.

Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1078–1086, 2017.

Yihui Ma, Jia Jia, Suping Zhou, Jingtian Fu, Yejun Liu, and Zijian Tong. Towards better understanding the clothing fashion styles: A multimodal deep learning approach. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Jingyuan Liu and Hong Lu. Deep fashion analysis with feature map upsampling and landmark-driven attention. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5337–5345, 2019.

Wenguan Wang, Yuanlu Xu, Jianbing Shen, and Song-Chun Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4271–4280, 2018a.

Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou 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, 2016a.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Ziwei Liu, Sijie Yan, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Fashion landmark detection in the wild. In *European Conference on Computer Vision*, pages 229–245. Springer, 2016b.

Sijie Yan, Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Unconstrained fashion landmark detection via hierarchical recurrent transformer networks. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 172–180, 2017.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

Olaf Ronneberger, Philipp Fischer, and Thomas 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.

Yixin Li, Shengqin Tang, Yun Ye, and Jinwen Ma. Spatial-aware non-local attention for fashion landmark detection. In *2019 IEEE international conference on multimedia and expo (ICME)*, pages 820–825. IEEE, 2019.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018b.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Weijiang Yu, Xiaodan Liang, Ke Gong, Chenhan Jiang, Nong Xiao, and Liang Lin. Layout-graph reasoning for fashion landmark detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2937–2945, 2019.

Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *proceedings of the IEEE international conference on computer vision*, pages 1281–1290, 2017.

Thomas Ziegler, Judith Butepage, Michael C Welle, Anastasiia Varava, Tonci Novkovic, and Danica Kragic. Fashion landmark detection and category classification for robotics. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 81–88. IEEE, 2020.

Jue Wang, Wenchao Liu, Long Ma, He Chen, and Liang Chen. Iorn: An effective remote sensing image scene classification framework. *IEEE Geoscience and Remote Sensing Letters*, 15(11): 1695–1699, 2018c.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

Xingxing Zou, Xiangheng Kong, Waikeung Wong, Congde Wang, Yuguang Liu, and Yang Cao. Fashionai: A hierarchical dataset for fashion understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11977–11986, 2019.

Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European conference on computer vision (ECCV)*, pages 269–286, 2018.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

Majuran Shajini and Amirthalingam Ramanan. An improved landmark-driven and spatial–channel attentive convolutional neural network for fashion clothes classification. *The Visual Computer*, 37(6):1517–1526, 2021.

Ruhan He, Yuyi Su, Tao Peng, Jia Chen, Zili Zhang, and Xinrong Hu. A structured feature learning model for clothing keypoints localization. In *International Conference on Multimedia Modeling*, pages 629–640. Springer, 2021.

Sumin Lee, Sungchan Oh, Chanho Jung, and Changick Kim. A global-local embedding module for fashion landmark detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

Ming Chen, Yingjie Qin, Lizhe Qi, and Yunquan Sun. Improving fashion landmark detection by dual attention feature enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

Huosheng Xie and Jiaqi Chen. Layout-aware bidirectional transfer network for fashion landmark detection. In *Thirteenth International Conference on Graphics and Image Processing (ICGIP 2021)*, volume 12083, pages 593–602. SPIE, 2022.

Ming Chen, Hang Ying, Yingjie Qin, Lizhe Qi, Zhongxue Gan, and Yunquan Sun. Adaptive graph reasoning network for fashion landmark detection. In *ECAI 2020*, pages 2672–2679. IOS Press, 2020.

Rui Wang, Jun Feng, and Qirong Bu. Fashion landmark detection via deep residual spatial attention network. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 745–752. IEEE, 2021.

Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

Huizhong Chen, Andrew Gallagher, and Bernd Girod. Describing clothing by semantic attributes. In *European conference on computer vision*, pages 609–623. Springer, 2012.

Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Proceedings of the IEEE international conference on computer vision*, pages 1062–1070, 2015.

Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5334–5343, 2017.

Charles Corbiere, Hedi Ben-Younes, Alexandre Ramé, and Charles Ollion. Leveraging weakly annotated data for fashion image retrieval and label prediction. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 2268–2274, 2017.

Maneet Singh, Shruti Nagpal, Mayank Vatsa, and Richa Singh. Enhancing fine-grained classification for low resolution images. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

Yuwei Zhang, Peng Zhang, Chun Yuan, and Zhi Wang. Texture and shape biased two-stream networks for clothing classification and attribute recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13538–13547, 2020.

Tzu-Heng Lin. Aggregation and finetuning for clothes landmark detection. *arXiv preprint arXiv:2005.00419*, 2020.