Improving Landmark Localization with Semi-Supervised Learning

Sina Honari[1*], Pavlo Molchanov[2], Stephen Tyree[2], Pascal Vincent[1,4], Christopher Pal[3], Jan Kautz[2]
[1]University of Montreal, [2]NVIDIA, [3]Ecole Polytechnique of Montreal, [4]CIFAR
[1]{honaris, vincentp}@iro.umontreal.ca,
[2]{pmolchanov, styree, jkautz}@nvidia.com, [3]christopher.pal@polymtl.ca

arXiv:1709.01591v1 [cs.CV] 5 Sep 2017

Abstract

We present two techniques to improve landmark localization from partially annotated datasets. Our primary goal is to leverage the common situation where precise landmark locations are only provided for a small data subset, but where class labels for classification tasks related to the landmarks are more abundantly available. We propose a new architecture for landmark localization, where training with class labels acts as an auxiliary signal to guide the landmark localization on unlabeled data. A key aspect of our approach is that errors can be backpropagated through a complete landmark localization model. We also propose and explore an unsupervised learning technique for landmark localization based on having a model predict equivariant landmarks with respect to transformations applied to the image. We show that this technique, used as additional regularization, improves landmark prediction considerably and can learn effective detectors even when only a small fraction of the dataset has labels for landmarks. We present results on two toy datasets and three real datasets, with hands and faces, respectively, showing the performance gain of our method on each.

1. Introduction

Landmark detection is a central step when solving many complex vision problems. Examples include emotion recognition [14], face identity verification [31, 30], hand tracking [13, 7], gesture recognition [6], and eye gaze tracking [43, 20]. Reliable landmark estimation can simplify solutions to these problems and is often part of the pipeline for sophisticated, robust vision tools. Neural networks have recently yielded state-of-the art results on numerous landmark estimation problems [33, 12, 36, 40]. However, neural networks generally need to be trained on a large set of labeled data to be robust to the variations in natural images. Landmark labeling is a tedious manual work where precision is important; as a result, few landmark datasets are large enough to train reliable deep neural networks. On the other hand it is much easier to label an image with a single class label rather than the entire set of precise landmarks, and datasets with labels related to—but distinct from—landmark detection are far more abundant.

In this paper, we use the indirect supervision of class labels to guide classifiers trained to localize landmarks. The class label can be considered a weak label that sends indirect signals about landmarks. For example, a photo of a hand gesture with the label "waving" likely indicates that the hand is posed with an open palm and spread fingers, signaling a set of reasonable locations for landmarks on the hand. We leverage class labels that are more abundant or more easily obtainable than landmark labels, putting our proposed method in the category of multi-task learning. A common approach [44, 46, 41, 8] to multi-task learning uses a traditional convolutional network, in which a final common fully-connected (FC) layer feeds into separate branches, each dedicated to the output for a different task. This approach learns shared low-level features across the set of tasks and acts as a regularizer, particularly when the individual tasks have few labeled samples.

There is a fundamental caveat to applying such an approach directly to simultaneous classification and landmark localization tasks, because the two have opposing requirements: classification output needs to be insensitive (invariant) to small deformations such as translations, whereas landmark localization needs to be equivariant to them, i.e. follow them precisely with high sensitivity. To build in invariance, traditional convolutional neural networks for classification problems relied on pooling layers to integrate signals across the input image. However, tasks such as landmark localization or image segmentation require both the global integration of information as well as an ability to retain local, pixel-level details for precise localization. The goal of producing precise landmark localization has thus led to the development of new layers and network architectures such as dilated convolutions [38], stacked what-where auto-encoders [47],

*Part of this work was done when author was at NVIDIA Research

recombinator-networks [12], fully-convolutional networks [18], and hyper-columns [10], each preserving pixel-level information. These models have however not been developed with multi-tasking in mind.

Current multi-task architectures that predict landmark locations [44, 46, 41, 8, 22] predict them at the final layer, i.e. *in parallel* with and at the same stage as classification. Many tasks ranging from emotion prediction, face identification, and gesture recognition to head-pose prediction can benefit from using landmarks as information *input* to a model. However, collecting labeled data for landmarks is almost always expensive compared to collecting data for a related classification task. *Therefore we propose a novel class of neural architectures which force classification predictions to flow through the intermediate step of landmark localization.*

The main aim of our model is to leverage auxiliary classification tasks and data, enhancing landmark localization by backpropagating classification errors through the landmark localization layers of the model. Specifically, we propose a sequential architecture in which the first part of the network predicts landmarks via pixel-level heatmaps, maintaining high-resolution feature maps by omitting pooling layers and strided convolutions. The second part of the network computes class labels using predicted landmark locations. To make the whole network differentiable, we use soft-argmax for extracting landmark locations from pixel-level predictions. Under this model, learning the landmark localizer is more directly influenced by the task of predicting class labels, allowing the classification task to enhance landmark localization learning.

Semi-supervised learning techniques [26, 23, 35] have been used in deep learning to improve classification accuracy with a limited amount of labeled training data. A recently proposed method [16] creates an implicit temporal ensemble of models by capturing models at different stages of training. The method penalizes variation in class predictions over time and across a variety of data augmentations applied to unlabeled training data, thus leveraging unlabeled data to learn invariant features and more robust models. Rather than penalizing variations in class predictions over time, here we also propose and explore an unsupervised learning technique for landmark localization where the model is asked to produce landmark localizations equivariant with respect to a set of transformations applied to the image. In other words, we transform an image during training and ask the model to produce landmarks that are similarly transformed. Importantly, this technique does not require the true landmark locations, and thus can be applied during semi-supervised training to leverage images with unlabeled landmarks.

In this paper we make the following contributions: 1) We propose a novel multi-tasking neural architecture, which a) predicts landmarks as a complete intermediate step before classification in order to use the class labels to improve landmark location learning, b) correspondingly maintains the resolution of feature maps for accurate landmark prediction, and c) uses soft-argmax for a fully-differentiable model in which end-to-end training can be performed with backpropogation, even from examples that do not provide labeled landmarks. 2) We also propose an unsupervised learning technique to learn features that are equivariant with respect to transformations applied to the input image. This technique can be applied on data that lack true landmark annotations. Combining contributions 1) and 2), we propose a robust landmark estimation technique which learns effective landmark predictors while requiring fewer labeled landmarks compared to current approaches.

2. Sequential Multi-Tasking

We refer to the new architecture that we propose here for leveraging class labels to guide the learning of landmark locations as *sequential multi-tasking*. This architecture first predicts the landmark locations and then uses the predicted landmarks as the input to the second part of the network that does classification (see Figure 1). In doing so, we create a bottle-neck in the network, forcing it to solve the classification task only through the landmarks. If the goal were to enhance classification, this architecture would have been harmful since such bottle-necks [11] would hurt the flow of information for classification. However, since our goal is landmark localization, this architecture enforces receiving signal from class labels through back-propagation to enhance landmark locations. This architecture can be applied whenever class labels can reliably be categorized when only considering the landmarks, without observing the image.

In order to make the whole pipeline trainable end-to-end, even on examples that do not provide any landmarks, we apply soft-argmax on the output of the last convolutional layer in the landmark prediction model. Specifically, let $M(I)$ the stack of K two-dimensional output maps produced by the last convolutional layer (with no non-linearity) for a given network input image I. The map associated to the kth landmark will be denoted $M_k(I)$. To obtain from it a single 2d location $L_k = (x, y)$ for the landmark we use the following soft-argmax operation:

$$L_k(I) = \text{soft-argmax}(\beta M_k(I)) \quad (1)$$

$$= \sum_{i,j} \text{softmax}(\beta M_k(I))_{i,j}(i,j) \quad (2)$$

where softmax denotes a spatial softmax of the map, i.e. $\text{softmax}(A)_{i,j} = \frac{\exp(A_{i,j})}{\sum_{i',j'}\exp(A_{i',j'})}$. β controls the softmax temperature, thus the peakiness of the resulting probability map, and (i, j) iterate over pixel coordinates. In short soft-argmax computes landmark coordinates $L_k = (x, y)$ as a

Figure 1. The architecture of the sequential multi-tasking model. **Top half:** basic sub-components of a sequential multi-tasking model. The first part of the network predicts the landmarks which is then fed into the second part of the network for classification. **Bottom half:** the specific architecture used in this paper. The landmark localization network is composed of several convolutional layers that preserve the image resolution without any pooling layers. A large kernel size is used to have a receptive field covering the entire image. The final layer generates as many maps as the number of landmarks. A soft-argmax is then applied on the output of the landmark prediction model. The classification network is composed of three fully connected (FC) layers whose final output is fed into a softmax. All FC layers, except the last one use dropout. All layers use rectified linear units (ReLU) non-linearity except the layer right before soft-argmax and the last FC layer right before softmax. These two layers have no non-linearities.

weighted average of all pixel coordinate pairs (i, j) where the weights are given by a softmax of landmark map M_k.

Predicted landmark coordinates are then fed into the second part of the network for classification. We will denote $P(\mathbb{C} = c|\mathbb{I} = I)$ the probability ascribed by the model to the class c given input image I, as computed by the final classification softmax layer.

Using soft-argmax, as opposed to a simple argmax, the model is fully differentiable through its landmark locations and is trainable end-to-end. When the model is trained to maximize the log probability of the true class labels in the second part of the network, this also guides the landmark locations through its gradients and corresponding parameter updates in the first part of the network.

A more standard end-to-end differentiable model might use a couple of convolution and pooling layers followed by fully connected layers before generating real-valued (x, y) key-point locations directly. But such an architecture gets sub-optimal results as the pooling layers do not maintain the location of the max-pooled features and this deteriorates the results on models that need fine-grained pixel level accuracy. This is our motivation for not using any pooling layers in the first part of the network that performs landmark localization. Our entire model is trained end-to-end to minimize the following cost:

$$\begin{aligned} \mathrm{C} = \; & \tfrac{1}{N} \textstyle\sum_{(I,\tilde{L},c)\in\mathcal{D}} \big(-\log P(\mathbb{C} = c|\mathbb{I} = I) + \\ & \lambda \tfrac{1}{K} \textstyle\sum_{k=1}^{K} ||\tilde{L}_k - L_k(I)||_2^2\big) + \gamma ||\mathbb{W}||_2^2, \end{aligned} \tag{3}$$

where $\mathcal{D}$ is the training set containing N triples $(I, \tilde{L}, c)$ of input image, labeled landmarks (when available), and true class. K is the number of landmarks. $\tilde{L}_k$ and $L_k(I)$ respectively correspond to the true and predicted landmark location. $\mathbb{W}$ represents the parameters of the model. The first part of the cost is the negative log likelihood on the class labels and affects the entire network. The second part is the squared Euclidean distance between true and estimated landmark locations and is used only when landmark labels are provided. This cost only affects the first part of the network. The last part of the cost is the ℓ_2-norm on the model's parameters (weight-decay regularization).

3. Equivariant Landmark Transformation

To further make the model's prediction consistent with respect to different transformations that are applied to the image, we propose the following unsupervised learning technique. Consider an input image I and the corresponding landmarks $L(I)$ predicted by the network. Now consider a small affine coordinate transformation T. We will use $T \odot \ldots$ to denote the application of such a transformation in coordinate space, whether applied to deform an bitmap image or to transform actual coordinates. If we apply this transformation to produce deformed image $I' = T \odot I$ and compute the resulting landmark coordinates $L(I')$ predicted by the network, they should be very close to the result of applying the transformation on landmark coordinates $L(I)$ i.e. we expect to have $L(T \odot I) \approx T \odot L(I)$.

To encourage this consistent behavior and increase robustness of the model's landmark prediction we thus define the following transformation cost:

$$\mathrm{C}_T = \frac{\alpha}{NK} \sum_{I\in\mathcal{D}} \sum_{k=1}^{K} \| \underbrace{T \odot L_k(I)}_{\hat{L}_k} - \underbrace{L_k(T \odot I)}_{L'_k} \|_2^2 \tag{4}$$

where α is a hyper-parameter controlling the strength of this additional regularization. The architecture for this technique, that we call *equivariant landmark transformation (ELT))* technique is illustrated in Figure 2. Multiple instances of C_T can thus be added to the overall training cost,

Figure 2. The equivariant landmark transformation (ELT) technique. Image I is transformed by affine transformation T to generate image I'. The same transformation is applied to predicted landmarks L of image I to generate $\hat{L}$. The model is encouraged to minimize the squared euclidean distance between $\hat{L}$ and the predicted landmarks L' of the transformed image I'. Both networks share the same parameters. Note that this technique can be applied even on images that do not have ground truth landmarks.

Figure 3. Sample images from the Shapes dataset. Each 60×60 image contains one square and one triangle with randomly sampled location, size, and orientation.

each corresponding to a different transformation T[1]. Note that this can be applied even on images that do not provide labeled landmarks.

4. Experiments

To validate our proposed model, we begin with two toy datasets, Shapes and Blocks, in order to verify to what extent the class labels can be used to guide the landmark localization regardless of the complexity of the dataset. We show the results on these datasets in Sections 4.1 and 4.2. Later, we evaluate the performance of our model on three real datasets, Polish sign-language dataset [15] in Section 4.3, and two fiducial points datasets: Multi-PIE [9] in Section 4.4, and 300W [25] in Section 4.5. All the models are implemented in Theano [1].

4.1. Shapes Dataset

To begin, we use a simple toy dataset to demonstrate our method's ability to learn consistent landmarks without direct supervision. Images in our Shapes dataset (see Figure 4.1 for examples) consist of two white shapes positioned on a black background. The shapes, a triangle and a square, are each displayed with randomly sampled size, location, and orientation. The classification task is to identify which shape, 0 for triangle and 1 for square, is positioned closer to the upper-left corner of the image.

[1]In all our experiments we used the same transformations (scaling, translation, rotation) and level of transformation as we did when preprocessing the data.

Figure 4. Sample classification and landmark predictions from the Shapes dataset. The *top row* shows the two predicted landmarks, indicated by red and blue crosses, respectively. Without any explicit landmark supervision, the model has learned to associate the first landmark (red) with the triangle and the second landmark (blue) with the square. Meanwhile, the green shapes indicate the model has correctly classified the type of shape that is closer to the top-left corner of the image. The *second and third rows* show the first and second landmark feature maps, respectively, corresponding closely with the location of the respective shapes. (Best viewed in color with zoom.)

In keeping with the architecture in Figure 1, we trained a model with six convolutional layers with 7×7 kernels, followed by two convolutional layers with 1×1 kernels, then the soft-argmax (with $\beta = 1$) for landmark localization.[2] We maintain sixteen feature maps per layer, until the final layer where two landmark feature maps are predicted. Following the soft-argmax, predicted landmarks input to two fully connected (FC) layers of size 40 and 2, respectively. The model is trained with *only* the cross-entropy cost on the class label *without* labeled landmarks or the unsupervised transformation cost.

Figure 4 shows the predictions of the trained model on a few samples from the dataset. In the first row, the green shape corresponds to the shape predicted to be the nearest to the upper-left corner, which was learned with 99% accuracy. The red and blue crosses correspond to the first soft-argmax and second soft-argmax landmark localizations, respectively. We observe that the red cross is consistently placed adjacent to the triangle, while the blue cross is near the square. The input maps to each soft-argmax (shown in rows 2 and 3) also clearly indicate the location of a consistent shape. This experiment shows the sequential architecture proposed here properly guides the first part of the network to find meaningful landmarks on this dataset, based solely on the supervision of the related classification task.

4.2. Blocks Dataset

Our second toy dataset, Blocks, presents additional difficulty: each image depicts a figure composed of a sequence

[2]We also experimented with dilated convolutions, but found that landmarks were predicted with noticeably less precision.

Figure 5. The fifteen classes of the Blocks dataset. Each class is composed of five squares and one triangle. To create the each 60×60 image in the dataset, a random scale, translation, and rotation (up to 360 degrees) is applied to one of the base classes.

of five white squares with one white triangle at the head. See Fig. 5 for all fifteen classes of Block figures. Images in the dataset are created with random rotations of up to 360 degrees combined with random scale and translation. We split the dataset into train, validation, and test sets, each having 3200 images. For the sequential multi-tasking architecture, we use six convolution layers with kernels of size 9×9 followed by two convolutional layers with kernel size 1×1 before the soft-argmax output. All convolutional layers have eight feature maps, except for the final layer which produces five feature maps, one for each square. The classification section of the network has three fully-connected layers of sizes 256, 256, and 15, respectively, ending with softmax classification.

Initially we trained the model with only cross-entropy on the class labels and evaluated the quality of the resulting landmark assignments. Ideally, the model would consistently assign each landmark to a particular block in the sequence from head (the triangle) to tail (the final square). However, in this more complex setting, the model did not predict landmarks consistently across examples. The addition of the transformation cost induces the model learns relatively consistent landmarks between examples *from the same class*, but this consistency does not extend between different classes. Unlike the Shapes dataset—where there was a consistent, if indirect, mapping between landmarks and the classification task and where there were only a single triangle and square–the correspondence among the classification task, and landmark identities is more tenuous in the Blocks dataset. Hence, we introduce a labeled set of true landmark locations and evaluate the landmark localization accuracy by having a different percentage of train-set being labelled with true landmarks.

Table 1 compares the results using the sequential multi-tasking model in the following scenarios: 1) using only the landmarks (Seq-MT (L)), which is equivalent to training only the first part of the network, 2) using landmarks and class labels (Seq-MT (L+C)), which trains the whole network on class labels and the first part of the network on landmarks, 3) using landmarks and the the transformation cost (Seq-MT (L+T)), and 4) using three costs together (Seq-MT (L+T+C)). We compare the performance of these

Figure 6. Our implementation of the multi-tasking architecture that is commonly used in the literature [44, 46, 41, 8] (Comm-MT). The model takes an image and applies 5 half conv layers (half conv layers maintain the 2D resolution of the feature maps). The conv layer indicated by $\times 4$ is repeated 4 times without weight sharing. The model then applies a 2×2 pooling layer reducing the size of the 2D maps by half on each dimension. Another half conv layer is applied, followed by a 2×2 pooling layer. The output of the pooling layer is flattened and given to 2 FC layers with dropout and relu non-linearities. The last FC layer is then connected to two branches, one for the classification task and another for the landmark localization. All conv layers and all FC layers, except that last FC layer, use relu non-linearity.

models while varying the percentage of images labeled with true landmark locations (L).[3]

When using the transformation cost in training (scenarios 3 and 4), we only apply it to images that do not provide true landmarks to simulate semi-supervised learning [4].

As shown in Table 1 [5], the *Seq-MT (L+C)* improves upon *Seq-MT (L)*, indicating that class labels can be used to guide the landmark locations. By adding the transformation cost, we can improve the results considerably. With *Seq-MT (L+T)* better performance is obtained compared to *Seq-MT (L+C)* indicating the unsupervised learning technique can substantially enhance performance. However, the best results are obtained with both class labels and the transformation and landmark costs. See Fig. 7 for prediction samples when only 5% of the data are labeled with landmarks.

Since our model can be considered to be a multi-tasking network, we contrast it with other multi-tasking architectures that are commonly used in the literature. We compare with two architectures: 1) The "common" multi-tasking architecture [44, 46, 41, 8] where sub-networks for each task share a common set of initial layers ending in a common fully-connected layer. See an illustration of this architecture

[3] We ensure that the set of examples with labeled landmarks is class-balanced.

[4] This is done to avoid unfair advantage of our model compared to other models on examples that provide landmarks. However, this approach can be applied to any image, both with and without labeled landmarks.

[5] The results are averaged over five seeds. We have tuned L2 separately for these models. λ and α coefficient are set to 1 for models that use them. We used $\beta = 0.01$ for models with soft-argmax and 25% dropout in FC layers in our experiments.

in Fig. 6.[6] We train two variants of this model, one with only landmarks (*Comm-MT (L)*) and another with landmarks and class labels (*Comm-MT (L+C)*) to see whether the class labels improve landmark localization. 2) Heatmap multi-tasking, where to avoid pooling layers we follow the recent trend of maintaining the resolution of feature maps [33, 10, 18, 12] and features detected for landmark localization do not pass through a FC layer. With Heat-map multitasking architecture we maintain the resolution of the feature maps for landmark localization and use a softmax on top of it. See Fig. 8 for an illustration of this architecture. The heatmaps right before the softmax layer are taken as input to the classification model. We refer to this model as *Heatmap-MT*. Note that this model doesn't have a bottleneck such as *Seq-MT (L+C)*.

As shown in Table 1 [7], the common multi-tasking approach is performing much worse than our sequential architecture for landmark estimation. The drawback of this architecture is the usage of pooling layers which leads to sub-optimal results for landmark estimation. The model trained with extra class information performs better than the model trained only on landmarks. Heatmap-MT performs better than the Comm-MT, indicating better results can be obtained without pooling layers. However, Heatmap-MT performs worse than Seq-MT, which is due to using softmax instead of softargmax. The softmax cannot be more accurate than the number of discrete elements in the softmax, however, the soft-argmax can get any real number. Moreover, in Heatmap-MT the class label is mostly helping in low percentage of labeled data, but in Seq-MT it is helping for all percentage of labeled data. We believe this is due to creating a bottle-neck of landmarks before class label prediction, which causes the class labels to impact on landmarks more directly through back-propagation.

4.3. Hands Dataset

Our first experiment on real data is landmark estimation on hands images captured with color sensors. Most common image datasets with landmarks on hands such as NYU [34] and ICVL [32] do not provide class labels. Also, most of the prior work in landmark estimation for hands is based on depth data [34, 32, 28, 29, 27]. We use the Polish hand dataset (HGR1) [15, 21], which provides 898 RGB images with 25 landmarks and 27 gestures from Polish sign

[6]We tried other variants of this architecture such as 1) a model that goes directly from the feature maps that have the same size as input image to the FC layer without any pooling layers and 2) a model that has more pooling layers and goes to a lower resolution before feeding the features to FC layers. Both models achieved worse results. Model 1 suffers from an over-abundance of parameters when going to FC layer. Model 2 suffers from pooling layers, and deeper models have slightly worse results on this task.

[7]The classification output layer in *Comm-MT* model uses sigmoid non-linearity. It worked better than abs(tanh) or having no non-linearity. The classification output layer in *Heatmap-MT* model has no-non-linearity.

Table 1. Error of different architectures on blocks dataset. The error is reported in pixel space. An error of 1 indicates 1 pixel distance to the target landmark location. The first 4 rows show the results of Seq-MT architecture, depicted in Fig. 1. The 5th and 6th rows show results with common multi-tasking architecture, depicted in 6. The last two rows show the results of the heatmap based multi-tasking architecture depicted in Fig. 8. Since the transformation cost is applied only to the data that do not provide landmarks, the results of these models for 100% labeled data would be similar to the ones that do not apply transformation cost.

Model	Percentage of Images with Labeled Landmarks					
	1%	5%	10%	20%	50%	100%
Seq-MT (L)	8.33	3.95	3.35	1.98	1.19	0.44
Seq-MT (L+C)	8.02	3.45	3.20	1.67	1.05	**0.38**
Seq-MT (L+T)	6.42	1.94	1.37	1.16	0.85	
Seq-MT (L+T+C)	**6.25**	**1.70**	**1.26**	**1.07**	**0.74**	
Comm-MT (L)	12.89	11.56	10.72	9.39	5.04	3.41
Comm-MT (L+C)	12.28	11.19	10.36	9.01	4.21	2.97
Heatmap-MT (L)	10.09	6.59	5.27	3.82	2.78	2.01
Heatmap-MT (L+C)	9.27	6.35	5.62	3.75	3.14	2.23

Figure 7. Sample landmark prediction on blocks dataset using sequential multi-tasking models when only 5% of data is labeled with landmarks. The rows shows the model predictions in this order from top to bottom: 1) Seq-MT (L), 2) Seq-MT (L+C), 3) Seq-MT (L+T), and 4) Seq-MT (L+T+C). The green cross shows the true landmark location and the red cross shows the model's prediction. Best viewed in color with zoom.

Figure 8. Multi-tasking architecture using heatmaps (Heatmap-MT). Landmarks are detected using conv layers without subsampling or pooling layers. A softmax layer is used for landmark prediction in the output layer. Landmark heatmaps right before softmax layer are fed to a series of pool and conv layers which is then passed to FC layers. The last FC layer is fed to softmax for classification. All layers use relu non-linearity except the layers right before the softmax. The first two FC layers use dropout.

Figure 9. Left: 25 landmarks of the HGR1 hands dataset. Right: sample classes from the HGR1 dataset.

Figure 10. Examples of our model predictions on the test set of the HGR1 dataset. GT represents ground-trust annotations, while numbers 100, 50, and 20 indicate which percentage of the training set with labeled landmarks used for training. Results are computed with Seq-MT (L+T+C) model (denoted *) and Seq-MT (L). Best viewed in color with zoom.

language captured with uncontrolled lightning and uncontrolled background from 12 subjects.

Figure 9 shows the correspondence of 25 landmarks with example gestures from different classes in the HGR1 dataset. Due to occlusion, different classes have different number of landmarks in range [17, 25]. To train the model properly, we mask out occluded landmarks during training and evaluation. The transformation cost, however, does not use the mask and will operate even on occluded landmarks. We divide images into train, validation and test sets by id (with no overlap in subjects between sets): train (ids 1 to 8), validation (ids 11 and 12), and test (ids 9 and 10). We end up with 573, 163, and 162 images for train, validation and test sets, respectively.

We use the architecture in Fig. 1 on this dataset with 6 convolutional layers of size 9×9, each having 64 maps, with the final layer producing 25 maps for landmarks. The input image to the model is 64×64 and gray-scaled. The classification section has 3 FC layers of size $(256, 256, 27)$. The first two layers have relu non-linearities with 50% dropout.

For training, we use the following parameters: $\beta = 0.001$, $\alpha = 0.3$, $\gamma = 10^{-5}$, $\lambda = 0.5$.

Accuracy of landmark detection on HGR1 dataset is measured by computing average RMSE metric in the image domain for every landmark and normalizing it by wrist width (the Euclidean distance between landmarks #1 and #25). We apply the transformation cost only on the images that do not have ground truth landmarks.

Table 2. Performance of architectures on HGR1 hands dataset. The error is Euclidean distance normalized by wrist width. Results are shown as percent; lower is better.

	Percentage of Images with Labeled Landmarks				
Model	5%	10%	20%	50%	100%
Seq-MT (L)	52.6	44.7	32.1	22.1	18.8
Seq-MT (L+C)	50.0	40.2	31.4	22.3	**18.4**
Seq-MT (L+T)	45.4	32.1	25.8	**19.6**	
Seq-MT (L+T+C)	**40.6**	**32.0**	**24.0**	19.8	
Comm-MT (L)	77.1	62.8	52.7	41.8	35.7
Comm-MT (L+C)	53.4	39.3	35.5	26.9	24.1
Heatmap-MT (L)	66.5	51.9	42.4	30.9	25.5
Heatmap-MT (L+C)	64.8	54.9	43.2	30.5	26.7

Table 2 shows results for landmark estimation for hands in the HGR1 test set[8]. All results were averaged over 5 seed, where at every seed we randomly select a subset of examples with provided landmarks. We observe: 1) applying sequential multitasking improves results for most experiments compared to using only landmarks (Seq-MT(L)) or other multi-tasking approaches, 2) the equivariant transformation cost (T) significantly improves results for all experiments, and 3) *Seq-MT (L+T+C)* compared to *Seq-MT (L)* can achieve the same performance with only half provided landmark labels (see 5%, 10%, 20%).

We show examples of landmark prediction with different models in Figure 11. transformation cost and class labels significantly improve results given a smaller fraction of data with annotated landmarks.

4.4. Multi-PIE Dataset

We next evaluate our proposed approach on the task of facial landmark estimation. Similar to Hands, most common face datasets including 300W [25], Helen [17], LFPW [3], and AFW [49] only provide landmark locations and no classes. MTFL [44] and MAFL [46] datasets provide attributes in addition to landmarks, however, these datasets have only 5 landmarks and it would be very difficult to infer the classes, e.g. gender or emotion, using just 5 landmarks, or to use those labels to guide the landmarks. Instead we

[8]To train all these models we apply these data augmentations: rotation [-20, 20], and scale and translation [-10%, 10%] of the face bounding box.

Figure 11. Examples of our model predictions on MultiPIE dataset. Models used, from top to bottom: 1) *Seq-MT (L)* with 100% labeled landmarks; 2) *Seq-MT (L+T)* with 20%; 3) *Seq-MT (L+T+C)* with 5%; 4) *Seq-MT (L)* with 5%. We observe close predictions by 1) and 2) indicating the effectiveness of our method even with only a small amount of labeled landmarks. Comparison between 3) and 4) shows significant improvement with the proposed transformation cost (T). Best viewed in color with zoom.

choose Multi-PIE [9] since it provides 68 landmark locations in addition to 6 emotions and 15 camera locations under 20 illuminations. The labeled emotions are neutral, surprise, smile, squint, disgust, and scream. We use emotion and camera as class labels to guide landmark prediction. We use images from 5 cameras (1 frontal, 2 with ± 15 degrees, and 2 with ± 30 degrees) and in each case a random illumination is selected. The images are then divided into subsets by id, with ids 1-150 in the train set, ids 151-200 in the valid set, and ids 201-337 in the test set. As a result, we have 1875, 579, and 1054 images in train, validation, and test sets, respectively.

We train a similar architecture to that used for Hands with the difference of using 68 feature-maps in the last convolutional layer for landmarks. The classification section has two branches, one for emotion and one for camera pose, with each branch receiving landmarks as inputs. Each has 3 FC layers of size $(256, 256, \#\text{ of classes})$. The first two layers have relu non-linearities with 25% dropout. For training we set the following parameters: $\beta = 0.001$, $\alpha = 0.3$, $\gamma = 10^{-5}$, $\lambda = 2$. Networks are learned with ADAM optimizer for 1000 epochs with early stopping on validation error. We use gray-scale images and apply the same data augmentation as for Hands. We evaluate the performance of this model with the same settings as in the Hands experiment: 1) only landmarks (L), 2) landmarks plus class labels (L+C), 3) landmarks plus transformation cost (L+T), 4) and all three (L+T+C). Similar to the previous experiments, we do not apply the transformation cost on the examples that provide true landmark locations, and only apply it to images that do not provide landmark locations.

Table 3. Performance of different architectures on MultiPIE dataset. The error is Euclidean distance normalized by ocular distance (as a percent; lower is better). The first 4 rows show Sequential Multi-tasking results. The last two rows show common multi-tasking results.

	Percentage of Images with Labeled Landmarks				
Model	5%	10%	20%	50%	100%
Seq-MT (L)	8.05	6.99	6.27	5.48	**5.07**
Seq-MT (L+C)	7.81	6.95	6.20	5.50	5.13
Seq-MT (L+T)	6.69	6.24	5.78	5.27	
Seq-MT (L+T+C)	**6.57**	**6.16**	**5.73**	**5.23**	
Comm-MT (L)	9.22	7.93	7.02	6.27	5.71
Comm-MT (L+C)	9.11	8.00	6.92	6.20	5.68
Heatmap-MT (L)	10.83	9.18	8.13	7.00	6.63
Heatmap-MT (L+C)	11.03	9.03	8.15	7.11	6.65

We compare our model with Comm-MT, Heatmap-MT architectures with and without class labels in Table 3. The error is Euclidean distance between true and predicted landmarks normalized by the distance between eye centers (inter-ocular distance). We observe: 1) the proposed Seq-MT outperforms other multitasking models with and without class labels, 2) in Seq-MT class labels improve the results for low percentage of labelled landmarks, 3) the transformation cost improves results considerably for all cases, further confirming the benefit of using the proposed ELT technique on unlabeled data, and 4) the best performance is obtained when the transformation cost and the class labels are used jointly, indicating both techniques can be used together to get the least error.

4.5. 300W Dataset

In order to compare our architecture with recent models and also evaluate it on natural images in the wild we use 300W [25] dataset. This dataset provides 68 landmarks and is composed of 3148 (337 AFW, 2000 Helen, and 811 LFPW) and 689 (135 IBUG, 224 LFPW, and 330 Helen) images in the training and test sets, respectively. Similar to RCN [12], we split the training set into 90% (2834 images) train-set and 10% (314 images) valid-set. Since this dataset does not provide any class label, we can evaluate our model in L and L+T cases. We use Seq-MT architecture with 8 convolutional layers (32 feature maps in each except the last one with 68) and kernels 9×9. [9]

In Table 4 we compare Seq-MT with other models in the literature. Seq-MT model is outperforming many models including CDM, DRMF, RCPR, CFAN, ESR, SDM, ERT, LBF and CFSS, and is only doing worse than few recent models with complicated architectures, eg. RCN [12] with multiple branches, RAR [36] with multiple refinement procedure and Lv et. al [19] with multiple steps. In Table 6 we

[9] We apply these data augmentations: rotation [-30, 30], and scale and translation [-10%, 10%] of the face bounding box.

compare recent models on their training conditions. RAR, TCDCN, [19], and CFSS do not use an explicit validation set. This makes comparison with these models more difficult for two reasons: 1) These models do hyper-parameter (HP) selection on the test-set, which makes them overfit on the test-set; and 2) Due to not using any explicit validation set, their affective training size is bigger. The first three models use extra datasets, either through pre-trained models (RAR, [19]) or additional labeled data (TCDCN). RCN is the only model that performs better than our model and does not leverage extra dataset. It also uses a validation set for HP selection.

Note that the originality of Seq-MT is not in the specific architecture used for the first part of the network that localizes landmarks, but rather in its multi-tasking architecture (specifically in its usage of the class labels to enhance landmark localization) and also leveraging the transformation cost. The landmark localization part of Seq-MT can be replaced with more complex models. To verify this, we use the RCN model, with publicly available code, [12] and replace the original softmax layer with a soft-argmax layer in order to apply the transformation cost. We refer to this model as RCN$^+$ and it is trained with these hyperparameters: $\beta = 1.0$, $\alpha = 0.5$, $\gamma = 0$, $\lambda = 1.0$. The result is shown as RCN$^+$(L) when using only landmark cost and RCN$^+$(L+T) when using landmark plus the transformation cost. On 300W dataset we apply the transformation cost to samples with or without labelled landmarks to observer how much improvement can be obtained when used on all data. We can further reduce RCN error from 5.54 to 5.1 by applying the transformation cost and soft-argmax. This is a new state of the art without any data-augmentation.

In Table 5 we compare Seq-MT with Heatmap-MT and Common-MT [10] on different percentage of labelled landmarks. Seq-MT outperforms the other two models. We also demonstrate the improvement that can be obtained by using RCN$^+$. Note that the transformation cost improves the results when applied to two different landmark localization architectures (Seq-MT, RCN). Moreover, it considerably improves the results on IBUG test-set that contains more difficult examples than the training set. Figure 12 shows the improvement obtained by using transformation cost on some samples.

5. Conclusion

We present a new architecture and training procedure for semi-supervised landmark localization. Our contributions are twofold; We first propose an unsupervised technique for equivariant landmark transformation without requiring labeled landmarks. This technique helped improving the landmark localization on all dataset. In addition we propose an architecture to improve landmark estimation using auxiliary labels such as classes by backpropagating errors through the landmark localization components of the model. Our experiments show we can achieve high accuracy with far less labelled training data. We achieve state of the art performance on public benchmark dataset for fiducial points, 300W, when training from scratch without extra data.

[10] We trained these models with these parameters: $\beta = 0.01$, $\alpha = 2.0$, $\gamma = 10^{-5}$, $\lambda = 2.0$.

Table 4. Performance of different architectures on 300W test-set using 100% labeled landmarks. The error is Euclidean distance normalized by ocular distance (as a percent; lower is better).

Model	Common	IBUG	Fullset
CDM [39]	10.10	19.54	11.94
DRMF [2]	6.65	19.79	9.22
RCPR [4]	6.18	17.26	8.35
CFAN [42]	5.50	16.78	7.69
ESR [5]	5.28	17.00	7.58
SDM [37]	5.57	15.40	7.50
ERT [5]			6.40
LBF [24]	4.95	11.98	6.32
CFSS [48]	4.73	9.98	5.76
TCDCN* [45]	4.80	8.60	5.54
RCN [12]	4.70	9.00	5.54
RCN + denoising [12]	4.67	8.44	5.41
RAR [36]	**4.12**	8.35	**4.94**
Lv et. al [19]	4.36	**7.56**	4.99
Heatmap-MT (L)	6.18	13.56	7.62
Comm-MT (L)	5.68	11.04	6.73
Seq-MT (L)	4.93	10.24	5.95
Seq-MT (L+T)	4.84	9.53	5.74
RCN$^+$ (L)	4.47	8.47	5.26
RCN$^+$ (L+T)	4.34	8.20	5.10

Table 5. Performance of different architectures on 300W test-set. The error is Euclidean distance normalized by ocular distance (eye-centers). Error is shown as a percent; lower is better.

		Percentage of Images with Labeled Landmarks				
	Model	5%	10%	20%	50%	100%
Fullset	Heatmap-MT (L)	13.47	11.68	9.85	8.18	7.62
	Comm-MT (L)	16.73	9.66	8.61	7.39	6.73
	Seq-MT (L)	9.82	8.30	7.26	6.28	5.95
	Seq-MT (L+T)	8.23	7.28	6.62	6.10	5.74
	RCN$^+$ (L)	7.26	6.48	5.91	5.52	5.26
	RCN$^+$ (L+T)	**7.22**	**6.32**	**5.88**	**5.45**	**5.10**
IBUG	Heatmap-MT (L)	26.36	22.77	18.46	14.94	13.56
	Comm-MT (L)	28.64	16.17	14.56	12.16	11.04
	Seq-MT (L)	18.74	16.21	13.41	11.20	10.24
	Seq-MT (L+T)	14.68	12.73	11.39	10.37	9.53
	RCN$^+$ (L)	15.36	12.74	11.82	10.12	8.47
	RCN$^+$ (L+T)	**12.54**	**10.35**	**9.56**	**8.67**	**8.20**

Acknowledgment

We would like to thanks Compute Canada and Calcul Quebec for providing computational resources. This work was partially funded by NVIDIA's NVAIL program.

Table 6. Comparison of recent models on their training conditions. RAR, Lv et. al [19], CFSS, and TCDCN do not use an explicit validation subset. This makes hyper-parameter (HP) selection based on the test-set and also brings extra training data (by using the validation-set). RAR and Lv et. al [19] initialize their models by pre-trained parameters (an ImageNet and a VGG-19 pre-trained model, respectively), which is equivalent to using extra data. TCDCN uses 20,000 extra labelled data for pre-training. Finally, RAR adds manual samples by occluding images with sunglasses, medical masks, phones, etc to make them robust to occlusion. Similar to RCN, Seq-MT and RCN$^+$ both have an explicit validation set for HP selection and therefore use a smaller training set. Neither use any extra data, either through pre-trained models or explicit external data. For data-augmentation we follow RCN and apply only affine transformation on the training set and use a basic occlusion through applying random black rectangles on top of images.

	Model					
Feature	RAR[36]	Lv et. al [19]	TCDCN[45]	CFSS[48]	RCN[12]	Seq-MT / RCN$^+$
Hyper-parameter selection dataset	Test-set	Test-set	Test-set	Test-set	Valid-set	Valid-set
Training on entire train-set?	Yes	Yes	Yes	Yes	No	No
Using Extra Dataset?	Yes	Yes	Yes	No	No	No
Manually augmenting the training set?	Yes	No	No	No	No	No

Figure 12. Examples of our model predictions on 300W test-set. Models used, from top to bottom: 1) *Seq-MT (L)*; 2) *Seq-MT (L+T)*; 3) *RCN $^+$(L)*; 4) *RCN $^+$(L+T)*. The green and red dots show ground truth and model predictions, respectively and the yellow lines show the error. These examples illustrate the improved accuracy obtained by using transformation cost (T). The rectangles indicate the regions that landmarks are mostly improved.

References

[1] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, et al. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. 4

[2] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, pages 3444–3451, 2013. 9

[3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2930–2940, 2013. 7

[4] X. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *ICCV*, pages 1513–1520, 2013. 9

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *IJCV*, 107(2):177–190, 2014. 9

[6] N. Dardas, Q. Chen, N. D. Georganas, and E. M. Petriu. Hand gesture recognition using bag-of-features and multi-class support vector machine. In *Haptic Audio-Visual Environments and Games (HAVE), 2010 IEEE International Symposium on*, pages 1–5. IEEE, 2010. 1

[7] D. Datcu and S. Lukosch. Free-hands interaction in augmented reality. In *Proceedings of the 1st symposium on Spatial user interaction*, pages 33–40. ACM, 2013. 1

[8] T. Devries, K. Biswaranjan, and G. W. Taylor. Multi-task learning of facial landmarks and expression. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 98–103. IEEE, 2014. 1, 2, 5

[9] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, 2010. 4, 8

[10] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015. 2, 6

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[12] S. Honari, J. Yosinski, P. Vincent, and C. Pal. Recombinator networks: Learning coarse-to-fine feature aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5743–5752, 2016. 1, 6, 8, 9, 10, 16

[13] K. Hu, S. Canavan, and L. Yin. Hand pointing estimation for human computer interaction based on two orthogonal-views. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3760–3763. IEEE, 2010. 1

[14] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, Ç. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari, et al. Combining modality specific deep neural networks for emotion recognition in video. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 543–550. ACM, 2013. 1

[15] M. Kawulok, J. Kawulok, J. Nalepa, and B. Smolka. Self-adaptive algorithm for segmenting skin regions. *EURASIP Journal on Advances in Signal Processing*, 2014(1):170, 2014. 4, 6

[16] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representation*, 2017. 2

[17] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *European Conference on Computer Vision*, pages 679–692. Springer, 2012. 7

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2, 6

[19] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 8, 9, 10

[20] K. A. F. Mora and J.-M. Odobez. Gaze estimation from multimodal kinect data. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 25–30. IEEE, 2012. 1

[21] J. Nalepa and M. Kawulok. Fast and accurate hand shape classification. In *International Conference: Beyond Databases, Architectures and Structures*, pages 364–373. Springer, 2014. 6

[22] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016. 2

[23] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015. 2

[24] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, pages 1685–1692, 2014. 9

[25] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV Workshop*, pages 397–403, 2013. 4, 7, 8

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2016. 2

[27] A. Sinha, C. Choi, and K. Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4150–4158, 2016. 6

[28] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2456–2463, 2013. 6

[29] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015. 6

[30] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014. 1

[31] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 1

[32] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014. 6

[33] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015. 1, 6

[34] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014. 6

[35] J. Weston, F. Ratle, H. Mobahi, and R. Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. 2

[36] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *European Conference on Computer Vision*, pages 57–72. Springer, 2016. 1, 8, 9, 10

[37] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, pages 532–539, 2013. 9

[38] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representation*, 2016. 1

[39] X. Yu, J. Huang, S. Zhang, W. Yan, and D. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *ICCV*, pages 1944–1951, 2013. 9

[40] X. Yu, F. Zhou, and M. Chandraker. Deep deformation network for object landmark localization. In *European Conference on Computer Vision*, pages 52–70. Springer, 2016. 1

[41] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 1036–1041. IEEE, 2014. 1, 2, 5

[42] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *ECCV*, pages 1–16. 2014. 9

[43] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2015. 1

[44] Z. Zhang, P. Luo, C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, pages 94–108, 2014. 1, 2, 5, 7

[45] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. In *PAMI*, 2015. 9, 10

[46] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *IEEE transactions on pattern analysis and machine intelligence*, 38(5):918–930, 2016. 1, 2, 5, 7

[47] J. J. Zhao, M. Mathieu, R. Goroshin, and Y. LeCun. Stacked what-where auto-encoders. In *International Conference on Learning Representation - Workshop Track*, 2016. 1

[48] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, pages 4998–5006, 2015. 9, 10

[49] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012. 7

Supplementary Information for
Improving Landmark Localization with Semi-Supervised Learning

5.1. Extra Experiments on Multi-PIE Dataset

In Table 7 we show how much the landmark localization accuracy changes using different classes in MultiPIE dataset. We find camera and emotion to be better helping on landmark localization compared to id and therefore we use these two classes for our experiments in Section 4.4.

Although the focus of this paper is on improving landmark localization, in order to observe the impact of each multi-tasking approach on classification accuracy, we report the classification results on emotion in Table 8 and on camera in Table 9. Results show that the classification accuracy improves by providing more labeled landmarks, despite having the number of <image, class label> pairs unchanged. It indicates that improving landmark localization can directly impact the classification accuracy. Landmarks are especially more helpful in emotion classification. On camera classification, the improvement is small and all models are getting high accuracy. Another observation is that Heatmap-MT performs better on classification tasks compared to the other two multi-tasking approaches. We believe this is due to passing high-level features to the classification network, which can pass more information from the image to the classification network compared to Seq-MT. However, this model is performing worse than Seq-MT on landmark localization. The Seq-MT model benefits from the landmark bottleneck to improve its landmark localization accuracy, which is the focus of this paper. In Tables 8 and 9 by adding the transformation cost the classification accuracy improves (in addition to landmarks) indicating the improved performance in landmark localization can enhance classification performance.

Figure 13 provides further localization examples on MultiPIE dataset. It illustrates the improved accuracy obtained with the proposed sequential mutli-tasking and ELT components when providing a small percentage of labeled landmarks.

5.2. Extra Experiments on hands Dataset

In Table 10 we show classification accuracy obtained using different multi-tasking techniques. Similar to the MultiPIE dataset, we observe increased accuracy by providing more labeled landmarks, showing the classification would benefit directly from landmarks. Also similar to MultiPIE, we observe better classification accuracy with Heatmap-MT. Comparing Seq-MT models, we observe improved classification accuracy by using the transformation cost. It demonstrates the impact of this component on both landmark localization and classification accuracy.

Table 7. Landmark error on MultiPIE using different class labels. Rows 2 to 8 correspond to using different class labels for L+C case. Camera and emotion classes on average improve better the landmark localization compared to id.

Model	Percentage of Images with Labeled Landmarks				
	5%	10%	20%	50%	100%
Seq-MT (L)	8.05	6.99	6.27	5.48	**5.07**
Seq-MT (L + id)	8.36	7.03	6.24	5.53	5.14
Seq-MT (L + emotion)	7.86	**6.94**	6.22	5.49	5.16
Seq-MT (L + camera)	7.80	6.96	6.26	**5.44**	**5.11**
Seq-MT (L + camera+emotion)	7.81	6.95	**6.20**	5.50	5.13
Seq-MT (L + id+emotion)	7.99	6.99	6.27	5.54	5.18
Seq-MT (L + id+camera)	8.32	7.07	6.27	5.50	5.15
Seq-MT (L + id+emotion+camera)	**7.78**	7.00	6.27	5.59	5.16

Table 8. Emotion classification error on MultiPIE dataset. In percent; higher is better.

Model	Percentage of Images with Labeled Landmarks				
	5%	10%	20%	50%	100%
Comm-MT (L+C)	74.67	79.90	83.76	86.37	86.83
Heatmap-MT (L+C)	**85.14**	**87.50**	**86.93**	**88.16**	**87.29**
Seq-MT (L+C)	78.78	82.62	84.69	84.03	84.86
Seq-MT (L+C+T)	82.90	84.57	84.85	86.48	

Table 9. Camera classification error on MultiPIE dataset. In percent; higher is better.

Model	Percentage of Images with Labeled Landmarks				
	5%	10%	20%	50%	100%
Comm-MT (L+C)	96.98	97.53	98.30	98.63	98.80
Heatmap-MT (L+C)	**98.46**	**98.99**	**98.99**	**98.98**	**98.98**
Seq-MT (L+C)	97.97	98.31	98.50	98.96	98.92
eq-MT (L+C+T)	98.41	98.53	98.47	98.43	

Table 10. Classification error on hands dataset. In percent; higher is better.

Model	Percentage of Images with Labeled Landmarks				
	5%	10%	20%	50%	100%
Comm-MT (L+C)	60.86	69.64	69.20	76.03	73.42
Heatmap-MT (L+C)	**83.74**	**87.86**	**87.55**	**90.29**	**89.27**
Seq-MT (L+C)	69.08	70.14	72.26	77.07	75.92
Seq-MT (L+C+T)	74.64	75.01	73.90	79.10	

Figure 14 provides further landmark localization examples on hands dataset.

5.3. Extra Information on 300W Dataset

In Figure 15 we show the architecture of RCN^+ used for 300W dataset. In Figure 16 we illustrate further samples from 300W dataset. The samples show the improved accuracy obtained in both *Seq-MT* and RCN^+ by using the ELT technique.

Figure 13. Extra examples of our model predictions on MultiPIE dataset. Models used, from top to bottom: 1) *Seq-MT (L)* with 100% labeled landmarks; 2) *Seq-MT (L+T)* with 20%; 3) *Seq-MT (L+T+C)* with 5%; 4) *Seq-MT (L)* with 5%. We observe close predictions by 1) and 2) indicating the effectiveness of our proposed transformation cost even with only a small amount of labeled landmarks. Comparison between 3) and 4) shows the improvement obtained with both the ELT technique and the sequential multitasking architecture when using a small percentage of labeled landmarks. Note that the model trained with ELT technique preserves better the joint distribution over the landmarks even with a small number of labeled landmarks. The last two examples on the bottom row show examples with high errors. Best viewed in color with zoom.

Figure 14. Extra examples of our model predictions on the test set of the HGR1 dataset. GT represents ground-trust annotations, while numbers 100, 50, and 20 indicate the percentage of the training set with labeled landmarks. Results are computed with Seq-MT (L+T+C) model (denoted *) and Seq-MT (L). Examples illustrate improvement of the landmark prediction by using the class label and the transformation cost in addition to the labeled landmarks. The last three examples on the bottom row show examples with high errors. Best viewed in color with zoom.

Figure 15. The ReCombinator Networks (RCN) [12] architecture used for experiments on 300W dataset. P indicates a pooling layer. All pooling layers have stride of 2. C indicates a convolutional layer. The number written below C indicates the convolution kernel size. All convolutions have stride of 1. U indicates an upsampling layer, where each feature map is upsampled to the next (bigger) feature map resolution. K indicates concatenation, where the upsampled features are concatenated with features of the same resolution before a pooling is applied to them. The dashed arrows indicate the feature maps are carried forward for concatenation. The solid arrows following each other, e.g. P, C, indicate the order of independent operations that are applied. The number written above feature maps in $n@w \times h$ format indicate number of feature maps n and the width w and height h of the feature maps.

Figure 16. Extra examples of our model predictions on 300W test-set. Models used, from top to bottom: 1) *Seq-MT (L)*; 2) *Seq-MT (L+T)*; 3) *RCN* $^{+}$*(L)*; 4) *RCN* $^{+}$*(L+T)*.The first two columns depict examples where all models get accurate predictions, The next 5 columns illustrate the improved accuracy obtained by using ELT technique in two different architectures (Seq-MT and RCN). The last two columns show difficult examples where error is high. The rectangles indicate the regions that landmarks are mostly affected. The green and red dots show ground truth (GT) and model predictions (MP), respectively. The yellow lines show the error by connecting GT and MP. Note that the ELT technique improves predictions in both architectures. Best viewed in color with zoom.