Robust Optimization for Deep Regression

Vasileios Belagiannis¹, Christian Rupprecht¹, Gustavo Carneiro², and Nassir Navab¹,³

¹Computer Aided Medical Procedures, Technische Universität München, Germany
²Australian Centre for Visual Technologies, The University of Adelaide, Australia
³Johns Hopkins University, USA

{belagian, christian.rupprecht, navab}@in.tum.de, {gustavo.carneiro}@adelaide.edu.au

Abstract

Convolutional Neural Networks (ConvNets) have successfully contributed to improve the accuracy of regression-based methods for computer vision tasks such as human pose estimation, landmark localization, and object detection. The network optimization has been usually performed with L2 loss and without considering the impact of outliers on the training process, where an outlier in this context is defined by a sample estimation that lies at an abnormal distance from the other training sample estimations in the objective space. In this work, we propose a regression model with ConvNets that achieves robustness to such outliers by minimizing Tukey’s biweight function, an M-estimator robust to outliers, as the loss function for the ConvNet. In addition to the robust loss, we introduce a coarse-to-fine model, which processes input images of progressively higher resolutions for improving the accuracy of the regressed values. We demonstrate faster convergence and better generalization of our robust loss function for the task of human pose estimation, on four publicly available datasets. We also show that the combination of the robust loss function with the coarse-to-fine model produces comparable or better results than current state-of-the-art approaches in these datasets.

1. Introduction

Deep learning has played an important role in the computer vision field in the last few years. In particular, several methods have been proposed for challenging tasks, such as classification [18, 21], detection [13], categorization [47], segmentation [26], feature extraction [36] and pose estimation [9]. State-of-the-art results in these tasks have been achieved with the use of Convolutional Neural Networks (ConvNets) trained with backpropagation [23]. Moreover, the majority of the tasks above are defined as classification problems, where the ConvNet is trained to minimize a softmax loss function [9, 18, 21]. Besides classification, ConvNets have been also trained for regression tasks such as human pose estimation [25, 42], object detection [40], facial landmark detection [39] and depth prediction [10]. In re-

Figure 1: Comparison of L2 and Tukey’s biweight loss functions: We compare our results (Tukey’s biweight loss) with the standard L2 loss function on the problem of 2D human pose estimation (PARSE [46], LSP [17], Football [19] and Volleyball [3] datasets). On top, the convergence of L2 and Tukey’s biweight loss functions is presented, while on the bottom, the graph shows the mean pixel error (MPE) comparison for the two loss functions. For the convergence computation, we choose as reference error, the smallest error using L2 loss (blue bars in bottom graph). Then, we look for the epoch with the closest error in the training using Tukey’s biweight loss function.
gression problems, the training procedure usually optimizes
an $L_2$ loss function plus a regularization term, where the
goal is to minimize the squared difference between the esti-
mated values of the network and the ground-truth. However,
it is generally known that $L_2$ norm minimization is sensitive
to outliers, which can result in poor generalization depend-
ing on the amount of outliers present during training [15].
Without loss of generality, we assume that the samples are
drawn from an unknown distribution and outliers are sample
estimations that lie at an abnormal distance from other train-
sing samples in the objective space [27]. Within our con-
text, outliers are typically represented by uncommon sam-
ple that are rarely encountered in the training data, such as
rare body poses in human pose estimation, unlike facial
point positions in facial landmark detection or samples with
imprecise ground-truth annotation. In the presence of out-
liers, the main issue of using $L_2$ loss in regression problems
is that outliers can have a disproportionally high weight and
consequently influence the training procedure by reducing
the generalization ability and increasing the convergence
time.

In this work, we propose a loss function that is robust
to outliers for training ConvNet regressors. Our motivation
originates from Robust Statistics, where the problem of out-
liers has been extensively studied over the past decades, and
several robust estimators have been proposed for reducing
the influence of outliers in the model fitting process [15].
Particularly in a ConvNet model, a robust estimator can be
used in the loss function minimization, where training sam-
ple samples with unusually large errors are downweighted such that
they minimally influence the training procedure. It is worth
noting that the training sample weighting provided by the
robust estimator is done without any hard threshold between
inliers and outliers. Furthermore, weighting training sam-
ple also conforms with the idea of curriculum [5] and self-
paced learning [22], where each training sample has differ-
ent contribution to the minimization depending on its error.
Nevertheless, the advantage in the use of a robust estimat-
er, over the concept of curriculum or self-paced learning,
is that the minimization and weighting are integrated in a
single function.

We argue that training a ConvNet using a loss function
that is robust to outliers results in faster convergence and
better generalization (Fig. 1). We propose the use of Tukey’s
biweight function, a robust M-estimator, as the loss function
for the ConvNet training in regression problems (Fig. 4).
Tukey’s biweight loss function weights the training samples
based on their residuals (notice that we use the terms resid-
ual and error interchangeably, even if the two terms are not
identical, with both standing for the difference between the
true and estimated values). Specifically, samples with un-
usually large residuals (i.e. outliers) are downweighted and
consequently have small influence on the training proce-
dure. Similarly, inliers with insignificant residuals are also
downweighted in order to prevent instabilities around local
minima. Therefore, samples with residuals that are not too
high or too small (i.e. inliers with significant residuals) have
the largest influence on the training procedure. In our Con-
vNet training, this influence is represented by the gradient
magnitude of Tukey’s biweight loss function, where in the
backward step of backpropagation, the gradient magnitude
of the outliers is low, while the gradient magnitude of the
inliers is high except for the ones close to the local mini-
mum. In Tukey’s biweight loss function, there is no need
to define a hard threshold between inliers and outliers. It
only requires a tuning constant for suppressing the residu-
als of the outliers. We normalize the residuals with the me-
dian absolute deviation (MAD) [44], a robust approxima-
tion of variability, in order to preassign the tuning constant
and consequently be free of parameters.

To demonstrate the advances of Tukey’s biweight loss
function, we apply our method on 2D human pose estima-
tion in still images, where our regression method is based on
a novel coarse-to-fine model. The first stage in this model
is based on an estimation of all output variables using the
input image, and the second stage relies on an estimation
of different subsets of the output variables using higher res-
olution input image regions extracted using the results of
the first stage. In the experiments, we evaluate our method
on four publicly available datasets (PARSE [46], LSP [17],
Football [19] and Volleyball [3]) in order to show that:
1. the proposed robust loss function allows for faster con-
vergence and better generalization compared to the $L_2$ loss;
and 2. the proposed coarse-to-fine model produces compa-
rible to better results than the state-of-the-art in the four
datasets above.

2. Related Work

In this section, we discuss deep learning approaches for
regression-based computer vision problems. In addition, we
review the related work on human pose estimation, since it
comprises the evaluation of our method. We refer to [35] for
an extended overview of deep learning and its evolution.

Regression-based deep learning. A large number of
regression-based deep learning algorithms have been re-
cently proposed, where the goal is to predict a set of in-
terdependent continuous values. For instance, in object and
text detection, the regressed values correspond to a bound-
ing box for localisation [16, 40], in human pose estima-
tion, the values represent the positions of the body joints
on the image plane [25, 31, 42], and in facial landmark
detection, the predicted values denote the image locations
of the facial points [39]. In all these problems, a ConvNet has
been trained using an $L_2$ loss function, without consider-
ing its vulnerability to outliers. It is interesting to note that
some deep learning based regression methods combine the
$L2$-based objective function with a classification function,
which effectively results in a regularization of $L2$ and in-
creases its robustness to outliers. For example, Zhang et
al. [48] introduce a ConvNet that is optimized for landmark
detection and attribute classification, and they show that the
combination of softmax and $L2$ loss functions improves the
network performance when compared to the minimization
of $L2$ loss alone. Wang et al. [45] use a similar strategy
for the task of object detection, where they combine the
bounding box localization (using an $L2$ norm) with object
segmentation. The regularization of the $L2$ loss function
has been also addressed by Gkioxari et al. [14], where the
function being minimized comprises a body pose estimation
term (based on $L2$ norm) and an action detection term.
Finally, other methods have also been proposed to improve
the robustness of the $L2$ loss to outliers, such as the use
of complex objective functions in depth estimation [10] or
multiple $L2$ loss functions for object generation [1]. How-
ever, to the best of our knowledge, none of the proposed
approaches handles directly the presence of outliers during
training with the use of a robust loss function, like we pro-
pose in this paper.

3. Robust Deep Regression

In this section, we introduce the proposed robust loss
function for training ConvNets on regression problems. In-
spired by M-estimators from Robust Statistics [6], we pro-
pose the use of Tukey’s biweight function as the loss to be
used in the network training.

The input to the network is an image $x : \Omega \to \mathbb{R}$
and the output is a real-valued vector $\hat{y} = (y_1, y_2, \ldots, y_N)$
of $N$ elements, with $y_i \in \mathbb{R}$. Given a training dataset
$\{(x_s, y_s)\}_{s=1}^{S}$ of $S$ samples, our goal is the training of a
ConvNet, represented by the function $\phi(\cdot)$, under the mini-
mization of Tukey’s biweight loss function with backprop-
agation [34] and stochastic gradient descent [7]. This train-
ing process produces a ConvNet with learnt parameters $\theta$
that is effectively a mapping between the input image $x$ and
output $y$, represented by:

$$\hat{y} = \phi(x; \theta),$$

where $\hat{y}$ is the estimated output vector. Next, we present
the architecture of the network, followed by Tukey’s bi-
weight loss function. In addition, we introduce a coarse-
to-fine model for capturing features in different image res-
lutions for improving the accuracy of the regressed values.

Human pose estimation The problem of human pose es-
   timation from images can be addressed by regressing a
   set of body joint positions. It has been extensively studied
   from the single- and multi-view perspective, where the
   standard ways to tackle the problem are based on
   part-based models [2, 4, 11, 32, 37, 46] and holistic ap-
   proaches [8, 12, 28, 33]. Most of the recent proposals
   using deep learning approaches have extended both part-
   based and holistic models. In part-based models, the body
   is decomposed into a set of parts and the goal is to infer
   the correct body configuration from the observation. The
   problem is usually formulated using a conditional random
   field (CRF), where the unary potential functions include,
   for example, body part classifiers, and the pairwise poten-
   tial functions are based on a body prior. Recently, part-
   based models have been combined with deep learning for
   2D human pose estimation [9, 30, 41], where deep part de-
   tectors serve as unary potential functions and also as image-
   based body prior for the computation of the pairwise po-
   tential functions. Unlike part-based models, holistic pose
   estimation approaches directly map image features to body
   poses [12, 28, 33]. Nevertheless, this mapping has been
   shown to be a complex task, which ultimately produced less
   competitive results when compared to part-based models.
   Holistic approaches have been re-visited due to the recent
   advances in the automatic extraction of high level features
   using ConvNets [25, 31, 42]. More specifically, Toshev et
   al. [42] have proposed a cascade of ConvNets for 2D hu-
   man pose estimation in still images. Furthermore, temporal
   information has been included to the ConvNet training for
   more accurate 2D body pose estimation [31] and the use of
   ConvNets for 3D body pose estimation from a single im-
   age has also been demonstrated in [25]. Nevertheless, these
   deep learning methods do not address the issue of the pres-
   ence of outliers in the training set.

The main contribution of our work is the introduction of
Tukey’s biweight loss function for regression problems
based on ConvNets. We focus on 2D human pose estima-
tion from still images (Fig. 2), and as a result our method
 can be classified as a holistic approach and is close to the
cascade of ConvNets from [42]. However, we optimize a
robust loss function instead of the $L2$ loss of [42] and em-
pirically show that this loss function leads to more efficient
   (i.e faster convergence) and better generalization
   results.
3.1. Convolutional Neural Network Architecture

Our network takes as input an RGB image and regresses a $N$-dimensional vector of continuous values. As it is presented in Fig. 3, the architecture of the network consists of five convolutional layers, followed by two fully connected layers and the output that represents the regressed values. The structure of our network is similar to Krizhevsky’s [21], but we use smaller kernels and fewer filters in the convolutional layers. Our fully connected layers are smaller as well, but as we demonstrate in the experimental section, the smaller number of parameters is sufficient for the regression tasks considered in this paper. In addition, we apply local contrast normalization, as proposed in [21], before every convolutional layer and max-pooling after each convolutional layer in order to reduce the image size. We argue that the benefits of max-pooling, in terms of reducing the computational cost, outweighs the potential negative effect in the output accuracy for regression problems. Moreover, we use dropout [38] in the fourth convolutional and first fully connected layers to prevent overfitting. The activation function for each layer is the rectified linear unit (ReLU) [29], except for the last layer, which uses a linear activation function for the regression. Finally, we use our robust loss function for training the network of Fig. 3.

3.2. Robust Loss Function

The training process of the ConvNet is accomplished through the minimization of a loss function that measures the error between ground-truth and estimated values (i.e. the residual). In regression problems, the typical loss function used is the $L^2$ norm of the residual, which during backpropagation produces a gradient whose magnitude is linearly proportional to this difference. This means that estimated values that are close to the ground-truth (i.e. inliers) have little influence during backpropagation, but on the other hand, estimated values that are far from the ground-truth (i.e. outliers) can bias the whole training process given the high magnitude of their gradient, and as a result adapt the ConvNet to these outliers while deteriorating its performance for the inliers. Recall that we consider the outliers to be estimations from uncommon training samples that lie at an abnormal distance from other sample estimations in the objective space. This is a classic problem addressed by Robust Statistics [6], which is solved with the use of a loss function that weights the training samples based on the residual magnitude. The main idea is to have a loss function that has low values for small residuals, and then usually grows linearly or quadratically for larger residuals up to a point when it saturates. This means that only relatively small residuals (i.e. inliers) can influence the training process, making it robust to the outliers that are mentioned above.

There are many robust loss functions that could be used, but we focus on Tukey’s biweight function [6] because of its property of suppressing the influence of outliers during backpropagation (Fig. 4) by reducing the magnitude of their gradient close to zero. Another interesting property of this loss function is the soft constraints that it imposes between inliers and outliers without the need of setting a hard threshold on the residuals. Formally, we define a residual of the $i^{th}$ value of vector $y$ by:

$$r_i = y_i - \hat{y}_i,$$  \hspace{1cm} (2)

where $\hat{y}_i$ represents the estimated value for the $i^{th}$ value of $y$, produced by the ConvNet. Given the residual $r_i$, Tukey’s biweight loss function is defined as:

$$\rho(r_i) = \begin{cases} \frac{c^2}{\pi^2} \left[ 1 - (\frac{c}{r_i})^2 \right]^3 , & \text{if } |r_i| \leq c \\ \frac{c^2}{2} & \text{otherwise} \end{cases},$$  \hspace{1cm} (3)

where $c$ is a tuning constant, which if is set to $c = 4.6851$, gives approximately 95% asymptotic efficiency as $L^2$ minimization on the standard normal distribution of residuals. However, this claim stands for residuals drawn from a distribution with unit variance, which is an assumption that does not hold in general. Thus, we approximate a robust measure of variability from our training data in order to scale
the residuals by computing the median absolute deviation (MAD) [15]. MAD measures the variability in the training data and is estimated as:

$$MAD_i = \text{median}_{k \in \{1,...,S\}} \left( \left| r_{i,k} - \text{median}_{j \in \{1,...,S\}} (r_{i,j}) \right| \right),$$

for $i \in \{1,...,N\}$ and the subscripts $k$ and $j$ index the training samples. The $MAD_i$ estimate acts as a scale parameter on the residuals for obtaining unit variance. By integrating $MAD_i$ to the residuals, we obtain:

$$r_i^{MAD} = \frac{y_i - \hat{y}_i}{1.4826 \times MAD_i},$$

where we scale $MAD_i$ by 1.4826 in order to make $MAD_i$ an asymptotically consistent estimator for the estimation of the standard deviation [15]. Then, the scaled residual $r_i^{MAD}$ in Eq. (5) can be directly used by Tukey’s biweight loss function Eq. (3). We fix the tuning constant based on $MAD$ scaling and thus our loss function is free of parameters. The function $E$ in Fig. 4, which shows the loss function and its derivative as a function of the training samples. The $MAD_i$ estimate is given by:

$$E = \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{N} \rho \left( r_{i,s}^{MAD} \right).$$

We illustrate the functionality of Tukey’s biweight loss function in Fig. 4, which shows the loss function and its derivative as a function of sample residuals in a specific training problem. This is an instance of the training for the LSP [17] dataset that is further explained in the experiments.

### 3.3. Coarse-to-Fine Model

We adopt a coarse-to-fine model, where initially a single network $\phi(.)$ of Eq. (1) is trained from the input images to regress all $N$ values of $\hat{y}$, and then separate networks are trained to regress subsets of $\hat{y}$ using the output of the single network $\phi(.)$ and higher resolution input images. Effectively, the coarse-to-fine model produces a cascade of ConvNets, where the goal is to capture different sets of features in high resolution input images, and consequently improve the accuracy of the regressed values. Similar approaches have been adopted by other works [10, 41, 42] and shown to improve the accuracy of the regression. Most of these approaches refine each element of $\hat{y}$ independently, while we employ a different strategy of refining subsets of $\hat{y}$. We argue that our approach constrains the search space more and thus facilitates the optimization.

More specifically, we define $C$ image regions and subsets of $\hat{y}$ that are included in these regions (Fig. 3). Each image region $x^c$, where $c \in \{1,...,C\}$, is cropped from the original image $x$ based on the output of the single ConvNet of Eq. (1). Then the respective subset of $\hat{y}$ that falls in the image region $c$ is transformed to the coordinate system of this region. To define a meaningful set of regions, we rely on the specific regression task. For instance, in 2D human pose estimation, the regions can be defined based on the body anatomy (e.g. head and torso or left arm and shoulder); similarly, in facial landmark localization the regions can be defined based on the face structure (e.g. nose and mouth). This results in training $C$ additional ConvNets $\{\phi^c(.)\}_{c=1}^{C}$ whose input is defined by the output of the single ConvNet $\phi(.)$ of Eq. (1). The refined output values from the cascade of ConvNets are obtained by:

$$\hat{y}_{ref} = \text{diag}(z)^{-1} \sum_{c=1}^{C} \phi^c(\mathbf{x}^c; \theta^c, \hat{y}(l^c)),$$

where $l^c \in \{1,2,...,N\}$ indexes the subset $c$ of $\hat{y}$, the vector $z \in \mathbb{N}^N$ has the number of subsets in which each element of $\hat{y}$ is included and $\theta^c$ are the learnt parameters. Every ConvNet of the cascade regresses values only for the dedicated subset $l^c$, while its output is zero for the other elements of $\hat{y}$. To train the ConvNets $\{\phi^c(.)\}_{c=1}^{C}$ of the cascade, we use the same network structure that is described in Sec. 3.1 and the same robust loss function of Eq. (6). Finally, during inference, the first stage of the cascade uses the single ConvNet $\phi(.)$ of Eq. (1) to produce $\hat{y}$, which is refined by the second stage of the cascade with the ConvNets $\{\phi^c(.)\}_{c=1}^{C}$ of Eq. (7). The predicted values $\hat{y}_{ref}$ of the refined regression function are normalized back to the coordinate system of the image $x$.

![Figure 4: Tukey’s biweight loss function](image-url)

**Figure 4:** Tukey’s biweight loss function (left) and its derivative (right) as a function of the training sample residuals.

### 3.4. Training Details

The input RGB image to the network has resolution $120 \times 80$, as it is illustrated in Fig. 3. Moreover, the input images are normalized by subtracting the mean image estimated from the training images. We also use data augmentation in order to regularize the training procedure. To that end, each training sample is rotated and flipped (50 times)
as well as a small amount of Gaussian noise is added to the ground-truth values $y$ of the augmented data. Furthermore, the same data is shared between the first cascade stage for training the single ConvNet $\phi(\cdot)$ and second cascade stage for training the ConvNets $\{\phi^c(\cdot)\}_{c=1}^C$. Finally, the elements of the output vector of each training sample are scaled to the range $[0, 1]$. Concerning the network parameters, the learning rate is set to 0.01, momentum to 0.9, dropout to 0.5 and the batch size to 230 samples.

The initialisation of the ConvNets’ parameters is performed randomly, based on an unbiased Gaussian distribution with standard deviation 0.01, with the result that many outliers can occur at the beginning of training. To prevent this effect that could slow down the training or exclude samples at all from contributing to the network’s parameter update, we increase the MAD values by a factor of 7 for the first 50 training iterations (around a quarter of an epoch). Increasing the variability for a few iterations helps the network to quickly reach a more stable state. Note that we have empirically observed that the number of iterations needed for this MAD adjustment does not play an important role in the whole training process and thus these values are not hard constraints for convergence.

4. Experiments

We evaluate Tukey’s biweight loss function for the problem of 2D human pose estimation from still images. For that purpose, we have selected four publicly available datasets, namely PARSE [46], LSP [17], Football [19] and Volleyball [3]. All four datasets include sufficient amount of data for training the ConvNets, except for PARSE which has only 100 training images. For that reason, we have merged LSP and PARSE training data, similar to [17], for the evaluation on the PARSE dataset. For the other three datasets, we have used their training data independently. In all cases, we train our model to regress the 2D body skeleton as a set of joints that correspond to pixel coordinates (Fig. 8).

We assume that each individual is localized within a bounding box with normalized body pose coordinates. Our first assumption holds for all four datasets, since they include cropped images of the individuals, while for the second we have to scale the body pose coordinates in the range $[0, 1]$. Moreover, we introduce one level of cascade using three parallel networks ($C = 3$) based on the body anatomy for covering the following body parts: 1) head - shoulders, 2) torso - hands, and 3) legs (see Fig. 3). In the first part of the experiments, a baseline evaluation is presented, where Tukey’s biweight and the standard $L2$ loss functions are compared in terms of convergence and generalization. Then, we compare the results of our proposed coarse-to-fine model with state-of-the-art methodologies.

Experimental setup: The experiments have been conducted on an Intel i7 machine with a GeForce GTX 980 graphics card. The training time varies slightly between the different datasets, but in general it takes 2-3 hours to train a single ConvNet. This training time scales linearly for the case of the cascade. Furthermore, the testing time of a single ConvNet is 0.01 seconds per image. Regarding the implementation of our algorithm, basic operations of the ConvNet such as convolution, pooling and normalization are based on MatConvNet [43].

Evaluation metrics: We rely on the mean pixel error (MPE) to measure the performance of the ConvNets. In addition, we employ the PCP (percentage of correctly estimated parts) performance measure, which is the standard metric used in human pose estimation [11]. We distinguish two variants of the PCP score according to the literature [32]. In strict PCP score, the PCP score of a limb, defined by a pair of joints, is considered correct if the distance between both estimated joint locations and true limb joint locations is at most $50\%$ of the length of the ground-truth limb, while the loose PCP score considers the average distance between the estimated joint locations and true limb joint locations. During the comparisons with other methods, we explicitly indicate which version of the PCP score is used (Table 1).

4.1. Baseline Evaluation

In the first part of the evaluation, the convergence and generalization properties of Tukey’s biweight loss functions are examined using the single ConvNet $\phi(\cdot)$ of Eq. (1), without including the cascade. We compare the results of the robust loss with $L2$ loss using the same settings and training data of PARSE [46], LSP [17], Football [19] and Volleyball [3] datasets. To that end, a 5-fold cross validation.
has been performed by iteratively splitting the training data of all datasets (none of the datasets includes by default a validation set), where the average results are shown in Fig. 5. Based on the results of the cross validation which is terminated by early stopping [24], we have selected the number of training epochs for each dataset. After training by using all training data for each dataset, we have compared the convergence and generalization properties of Tukey’s biweight and L2 loss functions. For that purpose, we choose the lowest MPE of L2 loss and look for the epoch with the closest MPE after training with Tukey’s biweight loss function. The results are summarized in Fig. 1 for each dataset. It is clear that by using Tukey’s biweight loss, we obtain notably faster convergence (note that on the PARSE dataset it is 20 times faster). This speed-up can be very useful for large-scale regression problems, where the training time usually varies from days to weeks. Besides faster convergence, we also obtain better generalization, as measured by the error in the validation set, using our robust loss function (see Fig. 1). More specifically, we achieve 12% smaller MPE error using Tukey’s biweight loss functions in two out of four datasets (i.e., PARSE and Football), while we are around 8% better with the LSP and Volleyball datasets. In addition, we present the full body PCP scores in Fig. 6, since this is the most common evaluation metric in human pose estimation, and similar conclusions can be drawn compared to the MPE.

### 4.2. Comparison with other Methods

In this part, we evaluate our robust loss functions using the coarse-to-fine model represented by the cascade of ConvNets (Fig. 3), presented in Sec. 3.3, and compare our results with the state-of-the-art from the literature, on the four aforementioned datasets (PARSE [46], LSP [17], Football [19] and Volleyball [3]). For the comparisons, we use the strict and loose PCP scores, depending on which evaluation metric was used by the state-of-the-art. The results are summarized in Table 1, where the first row of each evaluation shows our result using a single ConvNet $\phi(.)$ of Eq. (1) and the second row, the result using the cascade of ConvNets $\{\phi^c(.)\}_{c=1}^C$ of Eq. (7), where $C = 3$.

#### PARSE: This is a well-known dataset to assess 2D human pose estimation methodologies and thus we are able to show the results from most of the current state-of-the-art, as displayed in Table 1a. While our result is 68.5% for the full body regression using a single ConvNet, our final score is improved by around 5% with the cascade of ConvNets. We achieve the best score in the full body regression as well as in most body parts. Closer to our performance is another deep learning method by Ouyang et al. [30] that builds on part-based models and deep part detectors. The rest of the compared methods are also part-based, but our holistic model is simpler to implement and at the same time is shown to perform better (Fig. 2 and 7).

#### LSP: Similar to PARSE, this is another standard dataset to assess human pose estimation methodologies. In LSP dataset, our approach shows a similar performance, compared to the PARSE dataset, using a single ConvNet or a cascade of ConvNets. In particular, the PCP score using one ConvNet increases again by around 5% with the cascade of ConvNets, from 63.9% to 68.8% for the full body evaluation (Table 1b). The holistic approach of Toshev et al. [42] is also a cascade of ConvNets, but it relies on L2 loss minimization and a different network structure. On the other hand, the Tukey’s biweight loss being minimized in the training of our network brings better results in combination with the cascade. Note also that we have used 4 ConvNets in total for our model in comparison to the 29 networks used by Toshev et al. [42]. Moreover, considering

![Figure 6: PCP Comparison of L2 and Tukey’s biweight loss functions](image)

![Figure 7: Model refinement](image)
the performance with respect to body parts, the best PCP scores are shared between our method and the one of Chen & Yuille [9]. The part-based model of Chen & Yuille [9] scores best for the full body, head, torso and arms, while we obtain the best scores on the upper and lowers legs. We show some results on this dataset in Fig. 7 and 8.

**Football:** This dataset has been introduced by Kazemi et al. [19] for estimating the 2D pose of football players. In this dataset, our results (Table 1c) using one ConvNet are almost optimal (with a PCP score of 95.8%) and thus the improvement using the cascade is smaller in comparison to the two datasets above. However, it is important to notice that effective refinements are achieved with the use of the cascade of ConvNets, as demonstrated in Fig. 7 and 8.

**Volleyball:** Similar to the Football dataset [19], our results on the Volleyball dataset are already quite competitive using one ConvNet (Table 1d), with a PCP score of 81.7%. On this dataset, the refinement step has a negative impact to our results (Table 1d). We attribute this behaviour to the interpolation results of the cropped images, since the original images have low resolution (last row of Fig. 8).

5. Conclusion

We have introduced Tukey’s biweight loss function for the robust optimization of ConvNets in regression-based problems. Using 2D human pose estimation as testbed, we have empirically shown that optimizing with this loss function, which is robust to outliers, results in faster convergence and better generalization compared to the standard L2 loss, which is a common loss function used in regression problems. We have also introduced a cascade of ConvNets that improves the accuracy of the final regression result. The combination of our robust loss function with the cascade of ConvNets produces comparable or better results than the state-of-the-art approaches in all four public datasets on 2D human pose estimation.

| Method                     | Head | Torso | Upper Legs | Lower Legs | Upper Arms | Lower Arms | Full Body |
|----------------------------|------|-------|------------|------------|------------|------------|-----------|
| Ours                       | 78.5 | 95.6  | 82.0       | 75.6       | 61.5       | 36.6       | 68.5      |
| Ours (cascade)             | **91.7** | **98.1** | **84.2**   | **79.3**   | **66.1**   | **41.5**   | **73.2**  |
| Andriluka et al. [2]       | 72.7 | 86.3  | 66.3       | 60.0       | 54.6       | 35.6       | 59.2      |
| Yang & Ramanan [46]        | 82.4 | 82.9  | 68.8       | 60.5       | 63.4       | 42.4       | 63.6      |
| Pishchulin et al. [32]     | 77.6 | 90.7  | 80.0       | 70.0       | 59.3       | 37.1       | 66.1      |
| Johnson et al. [17]        | 76.8 | 87.6  | 74.7       | 67.1       | 67.3       | 45.8       | 67.4      |
| Ouyang et al. [30]         | 89.3 | 89.3  | 78.0       | 72.0       | **67.8**   | **47.8**   | 71.0      |

(a) **PARSE Dataset** The evaluation metric on PARSE dataset [46] is the strict PCP score.

| Method                     | Head | Torso | Upper Legs | Lower Legs | Upper Arms | Lower Arms | Full Body |
|----------------------------|------|-------|------------|------------|------------|------------|-----------|
| Ours                       | 72.0 | 91.5  | 78.0       | 71.2       | 56.8       | 31.9       | 63.9      |
| Ours (cascade)             | 83.2 | 92.0  | **79.9**   | **74.3**   | 61.3       | 40.3       | 68.8      |
| Toshev et al. [42]         | -    | -     | 77.0       | 71.0       | 56.0       | 38.0       | -         |
| Kiefel & Gehler [20]       | 78.3 | 84.3  | 74.5       | 67.6       | 54.1       | 28.3       | 61.2      |
| Yang & Ramanan [46]        | 79.3 | 82.9  | 70.3       | 67.0       | 56.0       | 39.8       | 62.8      |
| Pishchulin et al. [32]     | 85.1 | 88.7  | 78.9       | 73.2       | 61.8       | 45.0       | 69.2      |
| Ouyang et al. [30]         | 83.1 | 85.8  | 76.5       | 72.2       | 63.3       | 46.6       | 68.6      |
| Chen & Yuille [9]          | **87.8** | **92.7** | **77.0**   | **69.2**   | **69.2**   | **55.4**   | **75.0**  |

(b) **LSP Dataset** The evaluation metric on LSP dataset [17] is the strict PCP score.

| Method                     | Head | Torso | Upper Legs | Lower Legs | Upper Arms | Lower Arms | Full Body |
|----------------------------|------|-------|------------|------------|------------|------------|-----------|
| Ours                       | 97.1 | 99.7  | 99.9       | 98.1       | 96.2       | 87.1       | 95.8      |
| Ours (cascade)             | **98.3** | **99.7** | **99.9**   | **98.1**   | **96.6**   | **88.7**   | **96.3**  |
| Yang & Ramanan [46]        | 97.0 | 99.9  | 94.0       | 80.0       | 92.0       | 66.0       | 86.0      |
| Kazemi et al. [19]         | 96.0 | 98.0  | 97.0       | 88.0       | 93.0       | 71.0       | 89.0      |

(c) **Football Dataset** The evaluation metric on Football dataset [19] is the loose PCP score.

| Method                     | Head | Torso | Upper Legs | Lower Legs | Upper Arms | Lower Arms | Full Body |
|----------------------------|------|-------|------------|------------|------------|------------|-----------|
| Ours                       | 90.4 | 97.1  | 86.4       | 95.8       | 74.0       | 58.3       | 81.7      |
| Ours (cascade)             | 89.0 | 95.8  | 84.2       | 94.0       | 74.2       | **58.9**   | 81.0      |
| Yang & Ramanan [46]        | 76.1 | 80.5  | 52.4       | 70.5       | 40.7       | 33.7       | 56.0      |
| Belagiannis et al. [3]     | **97.5** | **81.4** | **65.1**   | **81.2**   | **54.4**   | **19.3**   | **60.2**  |

(d) **Volleyball Dataset** The evaluation metric on Volleyball dataset [3] is the loose PCP score.

Table 1: **Comparison with other approaches:** We compare our results using one ConvNet (first row in each dataset) and the cascade of ConvNets (second row). The scores of the other methods are the ones reported in their original papers.
6. Acknowledgment

This work was partly funded by the DFG Project “Advanced Learning for Tracking and Detection in Medical Workflow Analysis” and Technische Universität München - Institute for Advanced Study (German Excellence Initiative - FP7 Grant agreement 291763).

References

[1] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. Technical report, arXiv, 2014. 3

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In CVPR, 2009. 3, 8

[3] V. Belagiannis, C. Amann, N. Navab, and S. Ilic. Holistic human pose estimation with regression forests. In Articulated Motion and Deformable Objects, 2014. 1, 2, 6, 7, 8

[4] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3D pictorial structures for multiple human pose estimation. In CVPR, 2014. 3

[5] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In ICML, 2009. 2

[6] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. IJCV, 1996. 3, 4

[7] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT 2010, pages 177–186. Springer, 2010. 3

[8] C. S. Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In ICCV, 2011. 3

[9] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In NIPS, 2014. 1, 3, 7, 8

[10] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In NIPS, 2014. 1, 3, 5

[11] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In CVPR, 2008. 3, 6

[12] D. M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. TPAMI, 2007. 3

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 1

[14] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. R-cnn for pose estimation and action detection. arXiv, 2014. 3

[15] P. I. Huber. Robust statistics. Springer, 2011. 2, 4, 5

[16] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. arXiv, 2014. 2

[17] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In BMVC, 2010. 1, 2, 5, 6, 7, 8

[18] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In CVPR, 2014. 1

[19] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan. Multi-view body part recognition with random forests. In BMVC, 2013. 1, 2, 6, 7, 8

[20] M. Kiefel and P. V. Gehler. Human pose estimation with fields of parts. In ECCV, 2014. 8

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012. 1, 4

[22] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In NIPS, 2010. 2

[23] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989. 1

[24] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Muller. Neural networks-tricks of the trade. Springer Lecture Notes in Computer Sciences, 1524:5–50, 1998. 7

[25] S. Li and A. B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In ACCV, 2014. 1, 2, 3

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015. 1

[27] D. S. Moore and G. P. McCabe. Introduction to the Practice of Statistics. WH Freeman/Times Books/Henry Holt & Co, 1989. 2

[28] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In ECCV, 2002. 3

[29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010. 4

[30] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In CVPR, 2014. 3, 7, 8

[31] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In ACCV, 2014. 2, 3

[32] L. Pishchulin, A. Jain, M. Andriluka, T. Thomайдenh, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In CVPR, 2012. 3, 6, 8

[33] G. Reger, J. Rihan, S. Ramalingam, C. Orrite, and P. H. Torr. Randomized trees for human pose detection. In CVPR, 2008. 3

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. Cognitive modeling, 5, 1988. 3

[35] J. Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015. 2

[36] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv, 2013. 1

[37] L. Sigal, M. Isard, H. Haussecker, and M. J. Black. Loose-leaf segmentation. In ECCV, 2012. 3

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. JMLR, 2014. 4

[39] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In CVPR, 2013. 1, 2
[40] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In NIPS, 2013. 1, 2
[41] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In NIPS, 2014. 3, 5
[42] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In CVPR, 2014. 1, 2, 3, 5, 7, 8
[43] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. CoRR, 2014. 6
[44] W. N. Venables and B. D. Ripley. Modern applied statistics with S. Springer Science & Business Media, 2002. 2
[45] X. Wang, L. Zhang, L. Lin, Z. Liang, and W. Zuo. Deep joint task learning for generic object extraction. In NIPS, 2014. 3
[46] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. TPAMI, 2013. 1, 2, 3, 6, 7, 8
[47] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In ECCV, 2014. 1
[48] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In ECCV, 2014. 3