Bottom-Up and Top-Down Reasoning with Hierarchical Rectified Gaussians

Peiyun Hu
UC Irvine

peiyunh@ics.uci.edu

Deva Ramanan
Carnegie Mellon University

dramanan@cs.cmu.edu

Abstract

Convolutional neural nets (CNNs) have demonstrated remarkable performance in recent history. Such approaches tend to work in a "unidirectional" bottom-up feed-forward fashion. However, practical experience and biological evidence tells us that feedback plays a crucial role, particularly for detailed spatial understanding tasks. This work explores "bidirectional" architectures that also reason with top-down feedback: neural units are influenced by both lower and higher-level units.

We do so by treating units as rectified latent variables in a quadratic energy function, which can be seen as a hierarchical Rectified Gaussian model (RGs) [39]. We show that RGs can be optimized with a quadratic program (QP), that can in turn be optimized with a recurrent neural network (with rectified linear units). This allows RGs to be trained with GPU-optimized gradient descent. From a theoretical perspective, RGs help establish a connection between CNNs and hierarchical probabilistic models. From a practical perspective, RGs are well suited for detailed spatial tasks that can benefit from top-down reasoning. We illustrate them on the challenging task of keypoint localization under occlusions, where local bottom-up evidence may be misleading. We demonstrate state-of-the-art results on challenging benchmarks.

1. Introduction

Hierarchical models of visual processing date back to the iconic work of Marr [31]. Convolutional neural nets (CNN's), pioneered by LeCun *et al.* [27], are hierarchical models that compute progressively more invariant representations of an image in a bottom-up, feedforward fashion. They have demonstrated remarkable progress in recent history for visual tasks such as classification [25,38,43], object detection [8], and image captioning [22], among others.

Feedback in biology: Biological evidence suggests that *vision at a glance* tasks, such as rapid scene categorization [48], can be effectively computed with feedforward hierarchical processing. However, *vision with scrutiny* tasks,
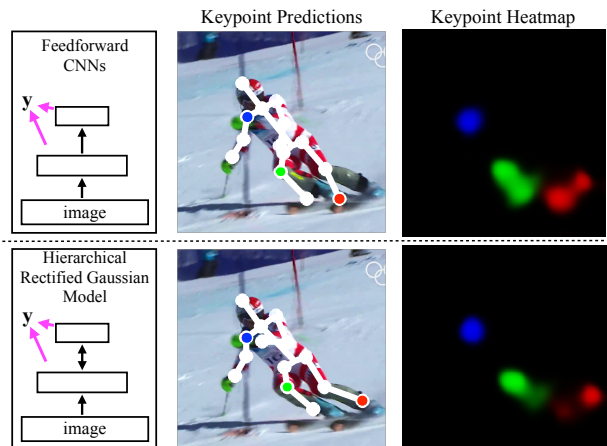

Figure 1: On the **top**, we show a state-of-the-art multi-scale feedforward net, trained for keypoint heatmap prediction, where the blue keypoint (the right shoulder) is visualized in the blue plane of the RGB heatmap. The ankle keypoint (red) is confused between left and right legs, and the knee (green) is poorly localized along the leg. We believe this confusion arises from bottom-up computations of neural activations in a feedforward network. On the **bottom**, we introduce hierarchical Rectified Gaussian (RG) models that incorporate top-down feedback by treating neural units as latent variables in a quadratic energy function. Inference on RGs can be unrolled into recurrent nets with rectified activations. Such architectures produce better features for "vision-with-scrutiny" tasks [17] (such as keypoint prediction) because lower-layers receive top-down feedback from above. Leg keypoints are much better localized with top-down knowledge (that may capture global constraints such as kinematic consistency).

such as fine-grained categorization [23] or detailed spatial manipulations [19], appear to require feedback along a "reverse hierarchy" [17]. Indeed, most neural connections in the visual cortex are believed to be feedback rather than feedforward [4, 26].

Feedback in computer vision: Feedback has also played a central role in many classic computer vision models. Hierarchical probabilistic models [20, 28, 55], allow

random variables in one layer to be naturally influenced by those above and below. For example, lower layer variables may encode edges, middle layer variables may encode parts, while higher layers encode objects. Part models [5] allow a face object to influence the activation of an eye part through top-down feedback, which is particularly vital for occluded parts that receive misleading bottom-up signals. Interestingly, feed-forward inference on part models can be written as a CNN [9], but the proposed mapping does not hold for feedback inference.

Overview: To endow CNNs with feedback, we treat neural units as nonnegative latent variables in a quadratic energy function. When probabilistically normalized, our quadratic energy function corresponds to a Rectified Gaussian (RG) distribution, for which inference can be cast as a quadratic program (QP) [39]. We demonstrate that coordinate descent optimization steps of the QP can be "unrolled" into a recurrent neural net with rectified linear units. This observation allows us to discriminatively-tune RGs with neural network toolboxes: *we tune Gaussian parameters such that, when latent variables are inferred from an image, the variables act as good features for discriminative tasks*. From a theoretical perspective, RGs help establish a connection between CNNs and hierarchical probabilistic models. From a practical perspective, we introduce RG variants of state-of-the-art deep models (such as VGG16 [38]) that require no additional parameters, but consistently improve performance due to the integration of top-down knowledge.

2. Hierarchical Rectified Gaussians

In this section, we describe the Rectified Gaussian models of Socci and Seung [39] and their relationship with rectified neural nets. Because we will focus on convolutional nets, it will help to think of variables $z = [z_i]$ as organized into layers, spatial locations, and channels (much like the neural activations of a CNN). We begin by defining a quadratic energy over variables z:

$$S(z) = \frac{1}{2}z^T W z + b^T z \qquad (1)$$

$$P(z) \propto e^{S(z)}$$

Boltzmann: $\quad z_i \in \{0, 1\}, w_{ii} = 0$

Gaussian: $\quad z_i \in R, -W$ is PSD

Rect. Gaussian: $\quad z_i \in R^+, -W$ is copositive

where $W = [w_{ij}]$, $b = [b_i]$. The symmetric matrix W captures bidirectional interactions between low-level features (e.g., edges) and high-level features (e.g., objects). Probabilistic models such as Boltzmann machines, Gaussians, and Rectified Gaussians differ simply in restrictions on the latent variable - binary, continuous, or nonnegative. Hierarchical models, such as deep Boltzmann machines [36], can be written as a special case of a block-sparse matrix W
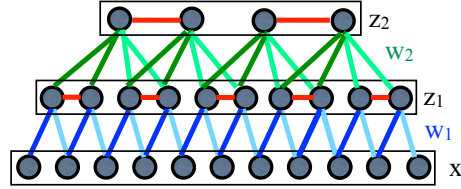

Figure 2: A hierarchical Rectified Gaussian model where latent variables z_i are denoted by circles, and arranged into layers and spatial locations. We write x for the input image and w_i for convolutional weights connecting layer $i-1$ to i. Lateral inhibitory connections between latent variables are drawn in red. Layer-wise coordinate updates are computed by filtering, rectification, and non-maximal suppression.

that ensures that only neighboring layers have direct interactions.

Normalization: To ensure that the scoring function can be probabilistically normalized, Gaussian models require that $(-W)$ be positive semidefinite (PSD) $(-z^T W z \geq 0, \forall z)$ Socci and Seung [39] show that Rectified Gaussians require the matrix $(-W)$ to only be *copositive* (-$z^T W z \geq 0, \forall z \geq 0$), which is a strictly weaker condition. Intuitively, copositivity ensures that the maximum of $S(z)$ is still finite, allowing one to compute the partition function. This relaxation significantly increases the expressive power of a Rectified Gaussian, allowing for multimodal distributions. We refer the reader to the excellent discussion in [39] for further details.

Comparison: Given observations (the image) in the lowest layer, we will infer the latent states (the features) from the above layers. Gaussian models are limited in that features will always be linear functions of the image. Boltzmann machines produce nonlinear features, but may be limited in that they pass only binary information across layers [33]. Rectified Gaussians are nonlinear, but pass continuous information across layers: z_i encodes the presence or absence of a feature, and if present, the strength of this activation (possibly emulating the firing rate of a neuron [21]).

Inference: Socci and Seung point out that MAP estimation of Rectified Gaussians can be formulated as a quadratic program (QP) with nonnegativity constraints [39]:

$$\max_{z \geq 0} \frac{1}{2} z^T W z + b^T z \qquad (2)$$

However, rather than using projected gradient descent (as proposed by [39]), we show that coordinate descent is particularly effective in exploiting the sparsity of W. Specifically, let us optimize a single z_i holding all others fixed. Maximizing a 1-d quadratic function subject to nonnegative constraints is easily done by solving for the optimum and clipping:

$$\max_{z_i \geq 0} f(z_i) \quad \text{where} \quad f(z_i) = \frac{1}{2} w_{ii} z_i^2 + (b_i + \sum_{j \neq i} w_{ij} z_j) z_i$$

$$\frac{\partial f}{\partial z_i} = w_{ii} z_i + b_i + \sum_{j \neq i} w_{ij} z_j = 0$$

$$z_i = -\frac{1}{w_{ii}} \max(0, b_i + \sum_{j \neq i} w_{ij} z_j) \qquad (3)$$

$$= \max(0, b_i + \sum_{j \neq i} w_{ij} z_j) \quad \text{for} \quad w_{ii} = -1$$

By fixing $w_{ii} = -1$ (which we do for all our experiments), the above maximization can solved with a rectified dot-product operation.

Layerwise-updates: The above updates can be performed for all latent variables in a layer in parallel. With a slight abuse of notation, let us define the input image to be the (observed) bottom-most layer $x = z_0$, and the variable at layer i and spatial position u is written as $z_i[u]$. The weight connecting $z_{i-1}[v]$ to $z_i[u]$ is given by $w_i[\tau]$, where $\tau = u - v$ depends only on the relative offset between u and v (visualized in Fig. 2):

$$z_i[u] = \max(0, b_i + top_i[u] + bot_i[u]) \quad \text{where} \quad (4)$$

$$top_i[u] = \sum_\tau w_{i+1}[\tau] z_{i+1}[u - \tau]$$

$$bot_i[u] = \sum_\tau w_i[\tau] z_{i-1}[u + \tau]$$

where we assume that layers have a single one-dimensional channel of a fixed length to simplify notation. By tying together weights such that they only depend on relative locations, bottom-up signals can be computed with cross-correlational filtering, while top-down signals can be computed with convolution. In the existing literature, these are sometimes referred to as deconvolutional and convolutional filters (related through a $180°$ rotation) [53]. It is natural to start coordinate updates from the bottom layer z_1, initializing all variables to 0. During the initial bottom-up coordinate pass, top_i will always be 0. This means that the bottom-up coordinate updates can be computed with simple filtering and thresholding. *Hence a single bottom-up pass of layer-wise coordinate optimization of a Rectified Gaussian model can be implemented with a CNN.*

Top-down feedback: We add top-down feedback simply by applying additional coordinate updates (4) in a top-down fashion, from the top-most layer to the bottom. Fig. 3 shows that such a sequence of bottom-up and top-down updates can be "unrolled" into a feed-forward CNN with "skip" connections between layers and tied weights. One can interpret such a model as a recurrent CNN that is capable of feedback, since lower-layer variables (capturing say, edges) can now be influenced by the activations of high-layer variables (capturing say, objects). Note that we
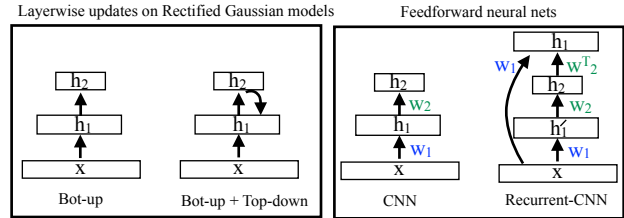

Figure 3: On the **left**, we visualize two sequences of layer-wise coordinate updates on our latent-variable model. The first is a bottom-up pass, while the second is a bottom-up + top-down pass. On the **right**, we show that bottom-up updates can be computed with a feed-forward CNN, and bottom-up-and-top-down updates can be computed with an "unrolled" CNN with additional skip connections and tied weights (which we define as a recurrent CNN). We use T to denote a $180°$ rotation of filters that maps correlation to convolution. We follow the color scheme from Fig. 2.

make use of recurrence along the depth of the hierarchy, rather than along time or spacial dimensions as is typically done [14]. When the associated weight matrix W is copositive, an infinitely-deep recurrent CNN *must* converge to the solution of the QP from (2).

Non-maximal suppression (NMS): To encourage sparse activations, we add lateral inhibitory connections between variables from same groups in a layer. Specifically, we write the weight connecting $z_i[u]$ and $z_i[v]$ for $(u, v) \in$ group as $w_i[u, v] = -\infty$. Such connections are shown as red edges in Fig. 2. For disjoint groups (say, non-overlapping 2x2 windows), *layer-wise updates correspond to filtering, rectification* (4)*, and non-maximal suppression (NMS) within each group.*

Unlike max-pooling, NMS encodes the spatial location of the max by returning 0 values for non-maximal locations. Standard max-pooling can be obtained as a special case by replicating filter weights w_{i+1} across variables z_i within the same group (as shown in Fig. 2). This makes NMS independent of the top-down signal top_i. However, our approach is more general in that NMS can be guided by top-down feedback: high-level variables (e.g., car detections) influence the spatial location of low-level variables (e.g., wheels), which is particularly helpful when parsing occluded wheels. Interestingly, top-down feedback seems to encode spatial information without requiring additional "capsule" variables [15].

Approximate inference: Given the above global scoring function and an image x, inference corresponds to $\text{argmax}_z S(x, z)$. As argued above, this can be implemented with an infinitely-deep unrolled recurrent CNN. However, rather than optimizing the latent variables to completion, we perform a fixed number (k) of layer-wise coordinate descent updates. This is guaranteed to report back finite variables z^* for any weight matrix W (even when not

copositive):

$$z^* = \mathbf{QP}_k(x, W, b), \quad z^* \in R^N \qquad (5)$$

We write $\mathbf{QP}_k$ in bold to emphasize that it is a *vector-valued function* implementing k passes of layer-wise coordinate descent on the QP from (2), returning a vector of all N latent variables. We set $k = 1$ for a single bottom-up pass (corresponding to a standard feed-forward CNN) and $k = 2$ for an additional top-down pass. We visualize examples of recurrent CNNs that implement $\mathbf{QP}_1$ and $\mathbf{QP}_2$ in Fig. 4.

Output prediction: We will use these N variables as features for M recognition tasks. In our experiments, we consider the task of predicting heatmaps for M keypoints. Because our latent variables serve as rich, multi-scale description of image features, we assume that simple linear predictors built on them will suffice:

$$y = V^T z^*, \quad y \in R^M, V \in R^{N \times M} \qquad (6)$$

Training: Our overall model is parameterized by (W, V, b). Assume we are given training data pairs of images and output label vectors $\{x_i, y_i\}$. We define a training objective as follows

$$\min_{W,V,b} R(W) + R(V) + \sum_i \text{loss}(y_i, V^T \mathbf{QP}_k(x_i, W, b)) \qquad (7)$$

where R are regularizer functions (we use the Frobenius matrix norm) and "loss" sums the loss of our M prediction tasks (where each is scored with log or softmax loss). We optimize the above by stochastic gradient descent. Because $\mathbf{QP}_k$ is a deterministic function, its gradient with respect to (W, b) can be computed by backprop on the k-times unrolled recurrent CNN (Fig. 3). We choose to separate V from W to ensure that feature extraction does not scale with the number of output tasks ($\mathbf{QP}_k$ is independent of M). During learning, we fix diagonal weights ($w_i[u, u] = -1$) and lateral inhibition weights ($w_i[u, v] = -\infty$ for $(u, v) \in$ group).

Related work (learning): The use of gradient-based backpropagation to learn an unrolled model dates back to 'backprop-through-structure' algorithms [11, 40] and graph transducer networks [27]. More recently, such approaches were explored general graphical models [41] and Boltzmann machines [12]. Our work uses such ideas to learn CNNs with top-down feedback using an unrolled latent-variable model.

Related work (top-down): Prior work has explored networks that reconstruct images given top-down cues. This is often cast as unsupervised learning with autoencoders [16,32,49] or deconvolutional networks [53], though

supervised variants also exist [29, 34]. Our network differs in that all nonlinear operations (rectification and max-pooling) are influenced by both bottom-up and top-down knowledge (4), which is justified from a latent-variable perspective.

3. Implementation

In this section, we provide details for implementing $\mathbf{QP}_1$ and $\mathbf{QP}_2$ with existing CNN toolboxes. We visualize our specific architecture in Fig. 4, which closely follows the state-of-the-art VGG-16 network [38]. We use 3x3 filters and 2x2 non-overlapping pooling windows (for NMS). Note that, when processing NMS-layers, we conceptually use 6x6 filters with replication after NMS, which in practice can be implemented with standard max-pooling and 3x3 filters (as argued in the previous section). Hence $\mathbf{QP}_1$ *is* essentially a re-implementation of VGG-16.

$\mathbf{QP}_2$: Fig. 5 illustrates top-down coordinate updates, which require additional feedforward layers, skip connections, and tied weights. Even though $\mathbf{QP}_2$ is twice as deep as $\mathbf{QP}_1$ (and [38]), *it requires no additional parameters*. Hence top-down reasoning "comes for free". There is a small notational inconvenience at layers that decrease in size. In typical CNNs, this decrease arises from a previous pooling operation. Our model requires an explicit $2\times$ subsampling step (sometimes known as strided filtering) because it employs NMS instead of max-pooling. When this subsampled layer is later used to produce a top-down signal for a future coordinate update, variables must be zero-interlaced before applying the $180°$ rotated convolutional filters (as shown by hollow circles in Fig. 5). Note that is *not* an approximation, but the mathematically-correct application of coordinate descent given subsampled weight connections.

Supervision y: The target label for a single keypoint is a sparse 2D heat map with a '1' at the keypoint location (or all '0's if that keypoint is not visible on a particular training image). We score this heatmap with a per-pixel log-loss. In practice, we assign '1's to a circular neighborhood that implicitly adds jittered keypoints to the set of positive examples.

Multi-scale classifiers V: We implement our output classifiers (7) as multi-scale convolutional filters defined over different layers of our model. We use upsampling to enable efficient coarse-to-fine computations, as described for fully-convolutional networks (FCNs) [29] (and shown in Fig. 4). Specifically, our multi-scale filters are implemented as 1×1 filters over 4 layers (referred to as fc7, pool4, pool3, and pool2 in [38]). Because our top (fc7) layer is limited in spatial resolution (1x1x4096), we define our coarse-scale filter to be "spatially-varying", which can alternatively be thought of as a linear "fully-connected" layer that is reshaped to predict a coarse (7x7) heatmap of
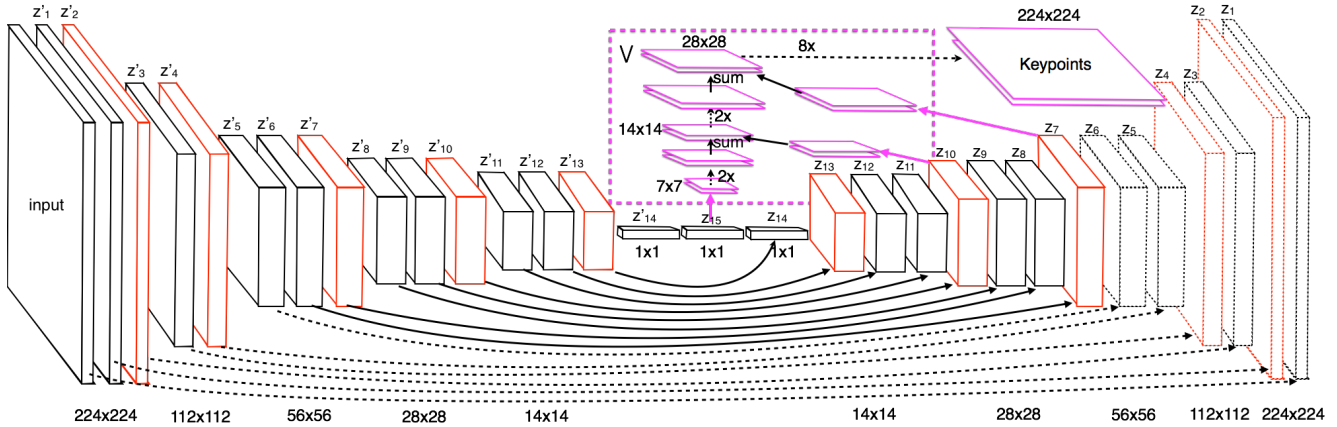
Figure 4: We show the architecture of $\mathbf{QP}_2$ implemented in our experiments. $\mathbf{QP}_1$ corresponds to the left half of $\mathbf{QP}_2$, which essentially resembles the state-of-the-art VGG-16 CNN [38]. $\mathbf{QP}_2$ is implemented with an 2X "unrolled" recurrent CNN with transposed weights, skip connections, and zero-interlaced upsampling (as shown in Fig. 5). Importantly, $\mathbf{QP}_2$ does not require any additional parameters. Red layers include lateral inhibitory connections enforced with NMS. Purple layers denote multi-scale convolutional filters that (linearly) predict keypoint heatmaps given activations from different layers. Multi-scale filters are efficiently implemented with coarse-to-fine upsampling [29], visualized in the purple dotted rectangle (to reduce clutter, we visualize only 3 of the 4 multiscale layers). Dotted layers are not implemented to reduce memory.
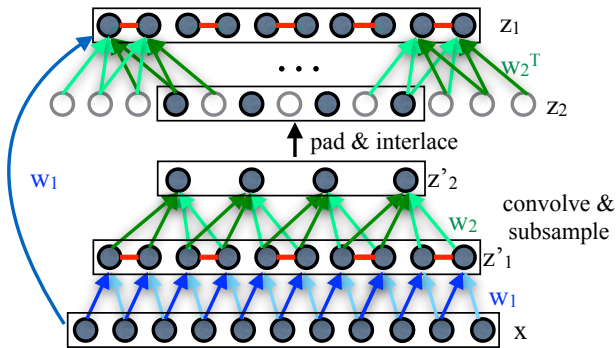

Figure 5: Two-pass layer-wise coordinate descent for a two-layer Rectified Gaussian model can be implemented with modified CNN operations. White circles denote 0's used for interlacing and border padding. We omit rectification operations to reduce clutter. We follow the color scheme from Fig. 2.

keypoint predictions given fc7 features. Our intuition is that spatially-coarse global features can still encode global constraints (such as viewpoints) that can produce coarse keypoint predictions. This coarse predictions are upsampled and added to the prediction from pool4, and so on (as in [29]).

Multi-scale training: We initialize parameters of both $\mathbf{QP}_1$ and $\mathbf{QP}_2$ to the pre-trained VGG-16 model [38], and follow the coarse-to-fine training scheme for learning FCNs [29]. Specifically, we first train coarse-scale filters, defined on high-level (fc7) variables. Note that $\mathbf{QP}_1$ and

$\mathbf{QP}_2$ are equivalent in this setting. This coarse-scale model is later used to initialize a two-scale predictor, where now $\mathbf{QP}_1$ and $\mathbf{QP}_2$ differ. The process is repeated up until the full multi-scale model is learned. To save memory during various stages of learning, we only instantiate $\mathbf{QP}_2$ up to the last layer used by the multi-scale predictor (not suitable for $\mathbf{QP}_k$ when $k > 2$). We use a batch size of 40 images, a fixed learning rate of 10^{-6}, momentum of 0.9 and weight decay of 0.0005. We also decrease learning rates of parameters built on lower scales [29] by a factor of 10. Batch normalization [18] is used before each non-linearity. Both our models and code are available online [1].

Prior work: We briefly compare our approach to recent work on keypoint prediction that make use of deep architectures. Many approaches incorporate multi-scale cues by evaluating a deep network over an image pyramid [44, 46, 47]. Our model processes only a single image scale, extracting multi-scale features from multiple layers of a single network, where importantly, fine-scale features are refined through top-down feedback. Other approaches cast the problem as one of regression, where (x,y) keypoint locations are predicted [54] and often iteratively refined [3, 42]. Our models predict heatmaps, which can be thought of as *marginal distributions* over the (x,y) location of a keypoint, capturing uncertainty. We show that by thresholding the heatmap value (certainty), one can also produce *keypoint visibility* estimates "for free". Our comments hold for our bottom-up model $\mathbf{QP}_1$, which can be thought of as a FCN tuned for keypoint heatmap prediction, rather than seman-

_____

[1] https://github.com/peiyunh/rg-mpii

tic pixel labeling. Indeed, we find such an approach to be a surprisingly simple but effective baseline that outperforms much prior work.

4. Experiment Results

We evaluated fine-scale keypoint localization on several benchmark datasets of human faces and bodies. To better illustrate the benefit of top-down feedback, we focus on datasets with significant occlusions, where bottom-up cues will be less reliable. All datasets provide a rough detection window for the face/body of interest. We crop and resize detection windows to 224x224 before feeding into our model. Recall that $\mathbf{QP}_1$ is essentially a re-implementation of a FCN [29] defined on a VGG-16 network [38], and so represents quite a strong baseline. Also recall that $\mathbf{QP}_2$ adds top-down reasoning *without any increase in the number of parameters*. We will show this consistently improves performance, sometimes considerably. Unless otherwise stated, results are presented for a 4-scale multi-scale model.

AFLW: The AFLW dataset [24] is a large-scale real-world collection of 25,993 faces in 21,997 real-world images, annotated with facial keypoints. Notably, these faces are not limited to be responses from an existing face detector, and so this dataset contains more pose variation than other landmark datasets. We hypothesized that such pose variation might illustrate the benefit of bidirectional reasoning. Due to a lack of standard splits, we randomly split the dataset into training (60%), validation (20%) and test (20%). As this is not a standard benchmark dataset, we compare to ourselves for exploring the best practices to build multi-scale predictors for keypoint localization (Fig. 6). We include qualitative visualizations in our supplementary material.

COFW: Caltech Occluded Faces-in-the-Wild (COFW) [2] is dataset of 1007 face images with severe occlusions. We present qualitative results in Fig. 7 and Fig. 8, and quantitative results in Table 1 and Fig. 9. Our bottom-up $\mathbf{QP}_1$ already performs near the state-of-the-art, while the $\mathbf{QP}_2$ significantly improves in accuracy of visible landmark localization and occlusion prediction. In terms of the latter, our model even approaches upper bounds that make use of ground-truth segmentation labels [7]. Our models are not quite state-of-the-art in localizing occluded points. We believe this may point to a limitation in the underlying benchmark. Consider an image of a face mostly occluded by the hand (Fig. 7). In such cases, humans may not even agree on keypoint locations, indicating that a keypoint *distribution* may be a more reasonable target output. Our models provide such uncertainty estimates, while most keypoint architectures based on regression cannot.

Pascal Person: The Pascal 2011 Person dataset [13] consists of 11,599 person instances, each annotated with a
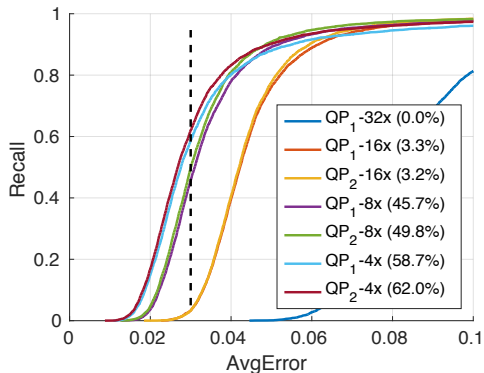

Figure 6: We plot the fraction of recalled face images whose average pixel localization error in AFLW (normalized by face size [56]) is below a threshold (x-axis). We compare our $\mathbf{QP}_1$ and $\mathbf{QP}_2$ with varying numbers of scales used for multi-scale prediction, following the naming convention of FCN [29] (where the Nx encodes the upsampling factor needed to resize the predicted heatmap to the original image resolution.) Single-scale models ($\mathbf{QP}_1$-32x and $\mathbf{QP}_2$-32x) are identical but perform quite poorly, not localizing any keypoints with 3.0% of the face size. Adding more scales dramatically improves performance, and moreover, as we add additional scales, the relative improvement of $\mathbf{QP}_2$ also increases (as finer-scale features benefit the most from feedback). We visualize such models in Fig. 10.
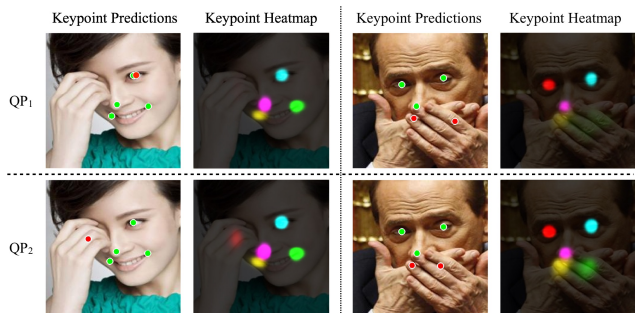

Figure 7: Visualization of keypoint predictions by $\mathbf{QP}_1$ and $\mathbf{QP}_2$ on two example COFW images. Both our models predict both keypoint locations and their visibility (produced by thresholding the value of the heatmap confidence at the predicted location). We denote (in)visible keypoint predictions with (red)green dots, and also plot the raw heatmap prediction as a colored distribution overlaid on a darkened image. Both our models correctly estimate keypoint visibility, but our bottom-up models $\mathbf{QP}_1$ misestimate their locations (because bottom-up evidence is misleading during occlusions). By integrating top-down knowledge (perhaps encoding spatial constraints on configurations of keypoints), $\mathbf{QP}_2$ is able to correctly estimate their locations.

bounding box around the visible region and up to 23 human keypoints per person. This dataset contains significant occlusions. We follow the evaluation protocol of [30]
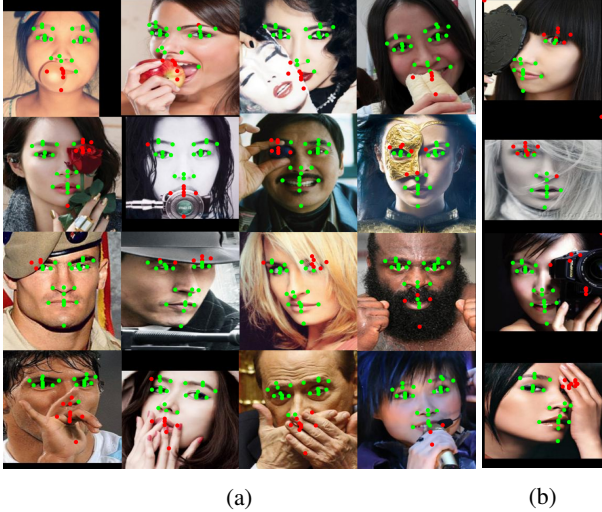
(a)   (b)

Figure 8: Facial landmark localization and occlusion prediction results of $\mathbf{QP}_2$ on COFW, where red means occluded. Our bidirectional model is robust to occlusions caused by objects, hair, and skin. We also show cases where the model correctly predicts visibility but fails to accurately localize occluded landmarks (**b**).

	Visible Points	All Points
RCPR [2]	-	8.5
RPP [51]	-	7.52
HPM [6]	-	7.46
SAPM [7]	5.77	6.89
FLD-Full [50]	5.18	**5.93**
$\mathbf{QP}_1$	5.26	10.06
$\mathbf{QP}_2$	**4.67**	7.87

Table 1: Average keypoint localization error (as a fraction of inter-ocular distance) on COFW. When adding top-down feedback ($\mathbf{QP}_2$), our accuracy on visible keypoints significantly improves upon prior work. In the text, we argue that such localization results are more meaningful than those for occluded keypoints. In Fig. 9, we show that our models significantly outperform all prior work in terms of keypoint visibility prediction.

and present results for localization of visible keypoints on a standard testset in Table 2. Our bottom-up $\mathbf{QP}_1$ model already significantly improves upon the state-of-the-art (including prior work making use of deep features), while our top-down models $\mathbf{QP}_2$ further improve accuracy by 2% without any increase in model complexity (as measured by the number of parameters). Note that the standard evaluation protocols evaluate only visible keypoints. In supplementary material, we demonstrate that our model can also accurately predict keypoint visibility "for free".

MPII: MPII is (to our knowledge) the largest available articulated human pose dataset [1], consisting of 40,000 people instances annotated with keypoints, visibility flags, and activity labels. We present qualitative results in Fig. 11
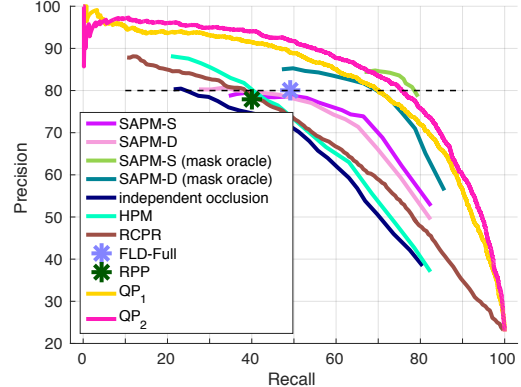

Figure 9: Keypoint visibility prediction on COFW, measured by precision-recall. Our bottom-up model $\mathbf{QP}_1$ already outperforms all past work that does not make use of ground-truth segmentation masks (where acronyms correspond those in Table 1). Our top-down model $\mathbf{QP}_2$ even approaches the accuracy of such upper bounds. Following standard protocol, we evaluate and visualize accuracy in Fig. 8 at a precision of 80%. At such a level, our recall (76%) significantly outperform the best previously-published recall of FLD [50] (49%).

α	0.10	0.20
CNN+prior [30]	47.1	-
$\mathbf{QP}_1$	66.5	78.9
$\mathbf{QP}_2$	**68.8**	**80.8**

Table 2: We show human keypoint localization performance on PASCAL VOC 2011 Person following the evaluation protocol in [30]. PCK refers to the fraction of keypoints that were localized within some distance (measured with respect to the instance's bounding box). Our bottom-up models already significantly improve results across all distance thresholds ($\alpha = 10, 20\%$). Our top-down models add a 2% improvement without increasing the number of parameters.

and quantitative results in Table 3. Our top-down model $\mathbf{QP}_2$ appears to outperform all prior work on full-body keypoints. Note that this dataset also includes visibility labels for keypoints, even though these are not part of the standard evaluation protocol. In supplementary material, we demonstrate that visibility prediction on MPII also benefits from top-down feedback.

TB: It is worth contrasting our results with TB [45], which implicitly models feedback by (1) using a MRF to post-process CNN outputs to ensure kinematic consistency between keypoints and (2) using high-level predictions from a coarse CNN to adaptively crop high-res features for a fine CNN. Our single CNN endowed with top-down feedback is slightly more accurate without requiring any additional parameters, while being 2X faster (86.5 ms vs TB's 157.2 ms). These results suggest that top-down reasoning may elegantly capture structured outputs and attention, two active areas of research in deep learning.
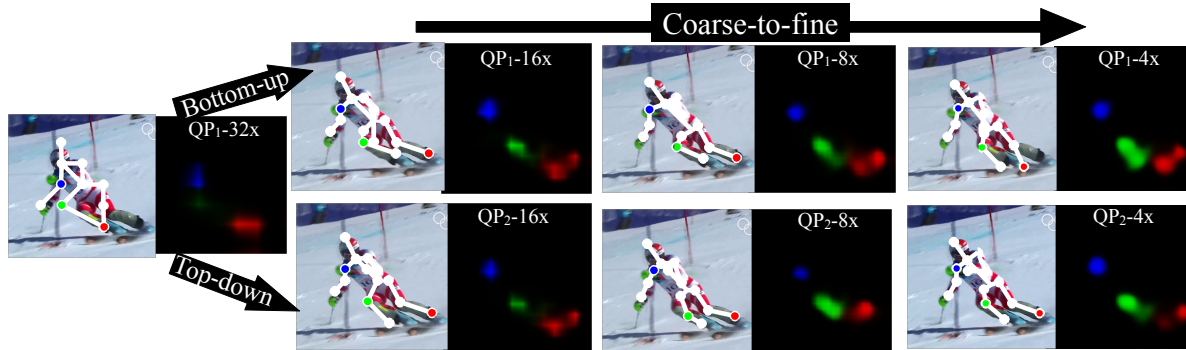
Figure 10: We visualize bottom-up and top-down models trained for human pose estimation, using the naming convention of Fig. 6. Top-down feedback ($\mathbf{QP}_2$) more accurately guides finer-scale predictions, resolving left-right ambiguities in the ankle (red) and poor localization of the knee (green) in the bottom-up model ($\mathbf{QP}_1$).
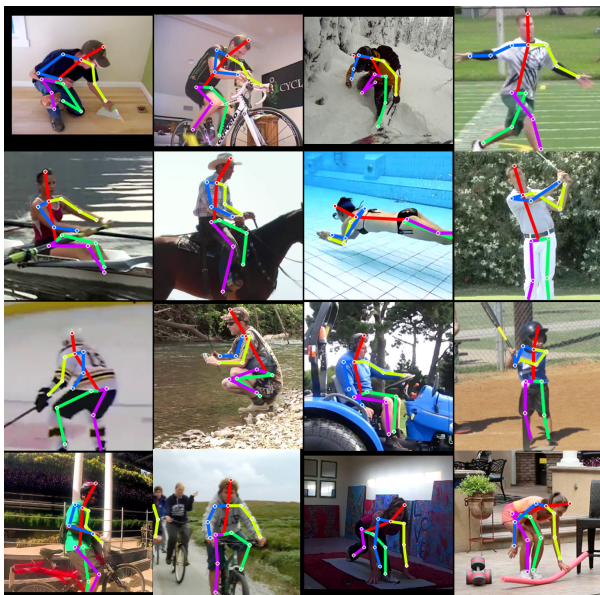

Figure 11: Keypoint localization results of $\mathbf{QP}_2$ on the MPII Human Pose testset. We quantitatively evaluate results on the validation set in Table 2. Our models are able to localize keypoints even under significant occlusions. Recall that our models can also predict visibility labels "for free", which are evaluated in supplementary material.

	Head	Shou	Elb	Wri	Hip	Kne	Ank	Upp	Full
GM [10]	-	36.3	26.1	15.3	-	-	-	25.9	-
ST [37]	-	38.0	26.3	19.3	-	-	-	27.9	-
YR [52]	73.2	56.2	41.3	32.1	36.2	33.2	34.5	43.2	44.5
PS [35]	74.2	49.0	40.8	34.1	36.5	34.4	35.1	41.3	44.0
TB [45]	96.1	91.9	83.9	77.8	80.9	72.3	64.8	**84.5**	82.0
$\mathbf{QP}_1$	94.3	90.4	81.6	75.2	80.1	73.0	68.3	82.4	81.1
$\mathbf{QP}_2$	95.0	91.6	83.0	76.6	81.9	74.5	69.5	83.8	**82.4**

Table 3: We show PCKh-0.5 keypoint localization results on MPII using the recommended benchmark protocol [1].

More recurrence iterations: To explore $\mathbf{QP}_K$'s performance as a function of K without exceeding memory limits, we trained a smaller network from scratch on 56X56

K	1	2	3	4	5	6
Upper Body	57.8	59.6	58.7	**61.4**	58.7	60.9
Full Body	59.8	62.3	61.0	**63.1**	61.2	62.6

Table 4: PCKh(.5) on MPII-Val for a smaller network

sized inputs for 100 epochs. As shown in Table 4, we conclude: (1) all recurrent models outperform the bottom-up baseline $\mathbf{QP}_1$; (2) additional iterations generally helps, but performance maxes out at $\mathbf{QP}_4$. A two-pass model ($\mathbf{QP}_2$) is surprisingly effective at capturing top-down info while being fast and easy to train.

Conclusion: We show that hierarchical Rectified Gaussian models can be optimized with rectified neural networks. From a modeling perspective, this observation allows one to discriminatively-train such probabilistic models with neural toolboxes. From a neural net perspective, this observation provides a theoretically-elegant approach for endowing CNNs with top-down feedback – *without any increase in the number of parameters*. To thoroughly evaluate our models, we focus on "vision-with-scrutiny" tasks such as keypoint localization, making use of well-known benchmark datasets. We introduce (near) state-of-the-art bottom-up baselines based on multi-scale prediction, and consistently improve upon those results with top-down feedback (particularly during occlusions when bottom-up evidence may be ambiguous).

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[2] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1513–1520. IEEE, 2013.

[3] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. *arXiv preprint arXiv:1507.06550*, 2015.

[4] R. J. Douglas, C. Koch, M. Mahowald, K. Martin, and H. H. Suarez. Recurrent excitation in neocortical circuits. *Science*, 269(5226):981–985, 1995.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[6] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *CVPR*, pages 1899–1906, 2014.

[7] G. Ghiasi and C. C. Fowlkes. Using segmentation to predict the absense of occluded parts. In *BMVC*, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587. IEEE, 2014.

[9] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*. IEEE, 2015.

[10] G. Gkioxari, P. Arbeláez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *CVPR*, 2013.

[11] C. Goller and A. Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *ICNN*, volume 1, pages 347–352. IEEE, 1996.

[12] I. Goodfellow, M. Mirza, A. Courville, and Y. Bengio. Multi-prediction deep boltzmann machines. In *NIPS*, 2013.

[13] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[14] S. Haykin. *Neural networks and learning machines*, volume 3. Pearson Education Upper Saddle River, 2009.

[15] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *ICANN*, pages 44–51. Springer, 2011.

[16] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[17] S. Hochstein and M. Ahissar. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5):791–804, 2002.

[18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[19] M. Ito and C. D. Gilbert. Attention modulates contextual influences in the primary visual cortex of alert monkeys. *Neuron*, 22(3):593–604, 1999.

[20] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *CVPR*, 2006.

[21] E. R. Kandel, J. H. Schwartz, T. M. Jessell, et al. *Principles of neural science*, volume 4. McGraw-Hill New York, 2000.

[22] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[23] S. M. Kosslyn, W. L. Thompson, I. J. Kim, and N. M. Alpert. Topographical representations of mental images in primary visual cortex. *Nature*, 378(6556):496–498, 1995.

[24] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCV Workshops*, pages 2144–2151. IEEE, 2011.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[26] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *PAMI*, 35(8):1847–1871, 2013.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] T. S. Lee and D. Mumford. Hierarchical bayesian inference in the visual cortex. *JOSA A*, 20(7):1434–1448, 2003.

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

[30] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *NIPS*, pages 1601–1609, 2014.

[31] D. Marr. Vision: A computational approach, 1982.

[32] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*, pages 52–59. Springer, 2011.

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

[34] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *arXiv preprint arXiv:1505.04366*, 2015.

[35] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, pages 588–595, 2013.

[36] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *AISTATS*, pages 448–455, 2009.

[37] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013.

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[39] N. D. Socci, D. D. Lee, and H. Sebastian Seung. The rectified gaussian distribution. *NIPS*, pages 350–356, 1998.

[40] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, pages 129–136, 2011.

[41] V. Stoyanov, A. Ropson, and J. Eisner. Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. In *AISTATS*, 2011.

[42] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3476–3483. IEEE, 2013.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*. IEEE, June 2015.

[44] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. *arXiv preprint arXiv:1411.4280*, 2014.

[45] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015.

[46] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014.

[47] S. Tulsiani and J. Malik. Viewpoints and keypoints. *arXiv preprint arXiv:1411.6067*, 2014.

[48] R. VanRullen and S. J. Thorpe. Is it a bird? is it a plane? ultra-rapid visual categorisation of natural and artifactual objects. *Perception-London*, 30(6):655–668, 2001.

[49] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11:3371–3408, 2010.

[50] Y. Wu and Q. Ji. Robust facial landmark detection under significant head poses and occlusion. In *ICCV*, 2015.

[51] H. Yang, X. He, X. Jia, and I. Patras. Robust face alignment under occlusion via regional predictive power estimation. 2015.

[52] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392. IEEE, 2011.

[53] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, 2010.

[54] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014*, pages 94–108. Springer, 2014.

[55] L. L. Zhu, Y. Chen, and A. Yuille. Recursive compositional models for vision: Description and review of recent work. *Journal of Mathematical Imaging and Vision*, 41(1-2):122–146, 2011.

[56] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.