Learning to Localize Little Landmarks

Saurabh Singh, Derek Hoiem, David Forsyth
University of Illinois, Urbana-Champaign
http://vision.cs.illinois.edu/projects/litland/
{ss1, dhoiem, daf}@illinois.edu

Abstract

We interact everyday with tiny objects such as the door handle of a car or the light switch in a room. These little landmarks are barely visible and hard to localize in images. We describe a method to find such landmarks by finding a sequence of latent landmarks, each with a prediction model. Each latent landmark predicts the next in sequence, and the last localizes the target landmark. For example, to find the door handle of a car, our method learns to start with a latent landmark near the wheel, as it is globally distinctive; subsequent latent landmarks use the context from the earlier ones to get closer to the target. Our method is supervised solely by the location of the little landmark and displays strong performance on more difficult variants of established tasks and on two new tasks.

1. Introduction

The world is full of tiny but useful objects such as the door handle of a car or the light switch in a room. We call these *little landmarks*. We interact with many little landmarks everyday, often not actively thinking about them or even looking at them. Consider the door handle of a car (Figure 1), it is often the first thing we manipulate when interacting with the car. However, in an image it is barely visible; yet we know where it is. Automatically localizing such little landmarks in images is hard, as they don't have a distinctive appearance of their own. These landmarks are largely defined by their context. We describe a method to localize little landmarks by discovering informative context supervised solely by the location of the little landmark. We demonstrate the effectiveness of our approach on several datasets, including both new and established problems.

The target landmark may have a local appearance that is similar to many other locations in the image. However, it may occur in a consistent spatial configuration with some pattern, such as an object or part, that is easier to find and would resolve the ambiguity. We refer to such a pattern as a *latent landmark*. The latent landmark may itself be hard to localize, although easier than the target. Another latent landmark may then help localize the earlier one, which in turn localizes the target. Our method discovers a sequence of such landmarks, where every latent landmark helps find the next one, with the sequence ending at the location of the target.

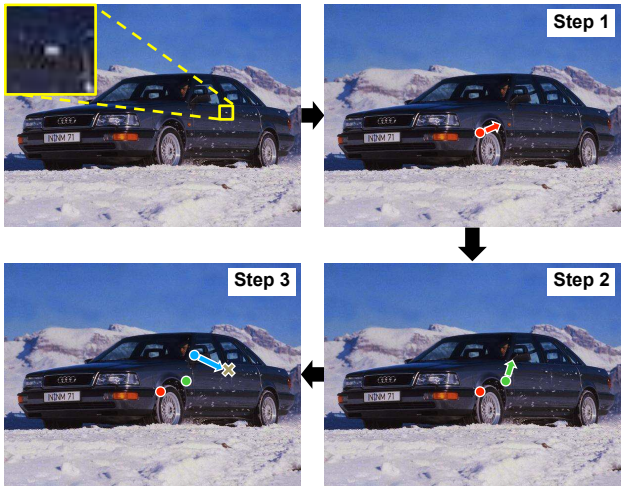

Figure 1. Several objects of interest are so tiny that they barely occupy few pixels (top-left), yet we interact with them daily. Localizing such objects in images is difficult as they do not have a distinctive local appearance. We propose a method that learns to localize such landmarks by learning a sequence of latent landmarks. Each landmark in this sequence predicts where the next landmark could be found. This information is then used to predict the next landmark and so on, until the target is found.

The first latent landmark in the sequence must be localizable on its own. Each subsequent landmark must be localizable given the previous landmark and predictive of the next latent landmark or the target. Our approach has to discover globally distinctive patterns to start the sequence and conditionally distinctive ones to continue it, while only being supervised by the location of the target. A detection of a latent landmark includes a set of positions, typically highly concentrated, and a prediction of where to look next. The

training loss function specifies that each of the first latent landmarks must predict the next latent landmark, and the last latent landmark must predict the target location. We train a deep convolutional network to learn all latent landmarks and predictions jointly. Our experiments on existing CUBS200 [43] and LSP [17] datasets and newly created car door handle and light switch datasets demonstrate the effectiveness of our approach. Code and datasets are available on the project webpage at http://vision.cs.illinois.edu/projects/litland/.

Contributions: We describe: 1) A novel and intuitive approach to localize little landmarks automatically. Our method learns to find useful latent landmarks and corresponding prediction models that can be used to localize the target; 2) A recurrent architecture using Fully Convolutional Networks that implements our approach; 3) A representation of spatial information particularly useful for robust prediction of locations by neural networks; 4) Two new datasets emphasizing practically important little landmarks that are hard to find.

2. Related Work

Landmark localization has been well-studied in the domain of human pose estimation [45, 42, 40, 12, 33, 2, 9, 39] as well as bird part localization [24, 25, 43, 34]. Localization of larger objects has similarly been well-studied [14, 15]. However, practically no work exists for localizing little landmarks.

Little landmarks are largely defined by their context. Thus, a successful method for localizing them will have to use this context. Use of context to improve performance has been studied (e.g. [27, 22]). In many problems, explicit contextual supervision is available. Felzenszwalb *et al*. [14] use contextual rescoring to improve object detection performance. Singh *et al*. [36] use context of easy landmarks to find the harder ones, Su *et al*. [37] use context from attributes in image classification. In contrast, our method has no access to explicit contextual supervision. Some methods do incorporate context implicitly e.g. Auto-Context [41], which iteratively includes information from an increasing spatial support to localize body parts. In contrast, our method learns to find a sequence of latent landmarks that are useful for finding the target little landmark without other supervised auxiliary tasks.

The work of Karlinsky *et al*. [19] is conceptually most related to our method. They evaluate keypoint proposals to choose an intermediate set of locations that can be used to form chains from a known landmark to a target. The target is predicted by marginalizing over evidence from all chains. In contrast, our approach does not use keypoint proposals and learns to find the first point in the chain as well.

Other closely related approaches are that of Alexe *et al*. [1] and Carreira *et al*. [6]. Alexe *et al*. learn a context driven search for objects. In each step, their method predicts a window that is most likely to contain the object given the previous windows and the features observed at those windows. This is done by a non-parametric voting scheme where the current window is matched to several windows in training images and votes are cast based on observed offsets to target object. Carreira *et al*. make a spatial prediction in each step and encode it by placing a gaussian at the predicted location. This is then used as a feature by the next step. Similar to Alexe *et al*., they supervise each step to predict the target. In addition, they constrain it to get closer to the target in comparison to previous step's prediction. In contrast, our method does not perform any matching with training images and does not supervise the intermediate steps with the target. Only the final step is directly supervised. The latent landmarks can be anywhere in the image as long as they are predictive of the next one in the sequence. Further, our method is trained end-to-end.

Reinforcement learning based methods bear some similarity to our method where they also operate in steps. Caicedo *et al*. [5] cast object detection in a reinforcement learning framework and learn a policy to iteratively refine a bounding box for object detection, Zhang *et al*. [46] learn to predict a better bounding box given an initial estimate. In comparison, our method does not have explicitly defined actions or a value function. Instead, it performs a fixed-length sequence of intermediate steps to find the target location.

Also related are methods that discover mid-level visual elements [35, 11, 18, 38] and use them as a representation for some task. The criterion for discovery of these elements is often not related to the final task they are used for. Some approaches have tried to address this by alternating between updating the representation and learning for the task [29]. In contrast, our method learns to find latent landmarks that are directly useful for localizing the target and is trainable end-to-end.

Our method has similarities with attention based methods that learn to look at a sequence of useful parts of the image [3, 44, 23, 4, 26]. An important difference is that an intermediate part is constrained to be spatially predictive of the next one.

3. Approach

The simplest scheme for finding a landmark looks at every location and decides whether it is the target landmark or not. We refer to this scheme as *Detection*. Training such a system is easy: we provide direct supervision for the target location. However this doesn't work well for little landmarks because they are not strongly distinctive. Now imagine using a single latent landmark to predict the location of the target, which could be far way. We refer to this scheme as *Prediction*. This is hard, because we don't have direct supervision for the latent landmark. Instead, the system must

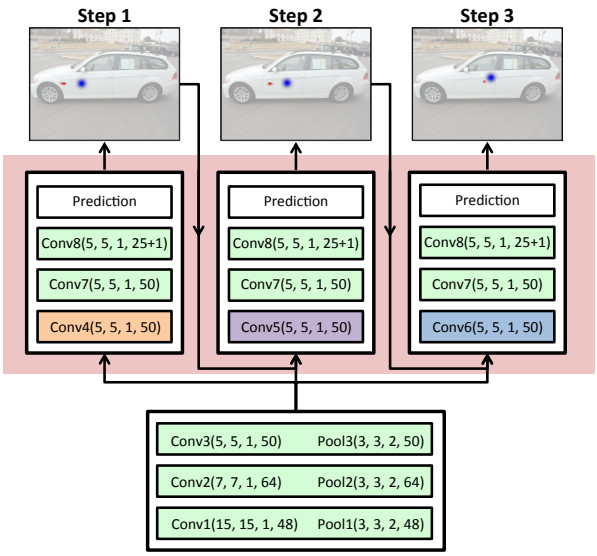

Figure 2. Model and Inference Overview. Our approach operates in steps, where, in each step, a latent landmark (red blobs at the top, best viewed with zoom) predicts the location of the latent landmark for the next step. This is encoded as a feature map with radial basis kernel (blue blob) and passed as a feature to the next step. This process repeats until the last step when the target is localized (door handle in above). Green boxes show layers and parameters that are shared across steps, while orange, purple and blue show step specific layers. Format for a layer is layer_name(height, width, stride, num_output_channels).

infer where these landmarks are. Furthermore, it must learn to find these landmarks *and* to use them to predict where the target is. While this is clearly harder to learn than detection, we describe a method that is successful and outperforms *Detection* (§ 4.1). Note that *Prediction* reduces to *Detection* if the latent landmark is forced to lie on the target.

Prediction is hard when the best latent landmark for a target is itself hard to find. Here, we can generalize to a sequential prediction scheme (referred to as *SeqPrediction*). The system uses a latent landmark to predict the location of another latent landmark; uses that to predict the location of yet another latent landmark; and so on, until the final step predicts the location of the target. Our method successfully achieves this and outperforms *Prediction* (§ 4.1).

Note that another generalization that is natural but not useful is an alternating scheme. One might estimate some mid-level pattern detectors, learn a prediction model, and then re-estimate the detectors conditioned on the current target estimates, etc. This scheme is unhelpful when the landmark is itself hard to find. First, re-estimates tend to be poor. Second, it is tricky to learn a sequential prediction as one would have to find conditionally distinctive patterns.

Our approach discovers latent landmarks that are directly useful for the localization of a target, as it is supervised only by this objective and can be trained end-to-end. Our method thus *learns to find a sequence of latent landmarks each with a prediction model to find the next in sequence*. In the following, we first provide an overview of the model, followed by the prediction scheme, and finally the training details.

3.1. Model and Inference

Figure 2 provides an overview of the model and how it is used for the inference. Our method operates in steps where each step $s \in \{1, \ldots, S\}$ corresponds to a *Prediction*. Each step predicts the location of the next latent landmark using the image features and the prediction from the previous step. The final step predicts the location of the target landmark. To make the prediction, each step finds a latent landmark (Figure 2, red blob) and makes an offset prediction to the next latent landmark. This prediction is encoded as a feature map (blue blob) and passed on to the next step. Note that the spatial prediction scheme is of key importance for the system to work. We describe it in Section 3.2.

Our system uses a fully convolutional network architecture, sliding a network over the image to make the prediction at each location. In Figure 2, the green boxes indicate the layers with parameters shared across various steps. Other colored boxes (orange, purple and blue) show layers that have step specific parameters. Note that this configuration of not sharing parameters for the layer that operates directly on features from previous step worked better than sharing all parameters and a few other alternatives (§ 5). The step specific parameters allow the features of a step to quickly adapt as estimates of underlying landmarks improve. Our model is trained using stochastic gradient descent on a robust loss function. Our loss function encourages earlier steps to be informative for the later steps by penalizing disagreement between the predicted and later detected latent landmark locations.

3.2. Prediction Scheme for a Step

Since our model is fully convolutional, images of different sizes produce feature maps of different sizes. To make a single prediction for the whole image we view the image as a grid of locations $l_i, i \in \{1, \ldots, L\}$. Each location can make a prediction using the sliding neural network and the combined prediction is a weighted average of these.

Each step s produces a summary estimate of the position of the next latent landmark $P^{(s)}$. Each location l_i separately estimates this position as $p_i^{(s)}$ with a confidence $c_i^{(s)}$. Each $p_i^{(s)}$ is estimated using a novel representation scheme with several nice properties (§ 3.3). The individual predictions are then combined as

$$P^{(s)} = \sum_{i=1}^{L} c_i^{(s)} p_i^{(s)} \quad (1)$$

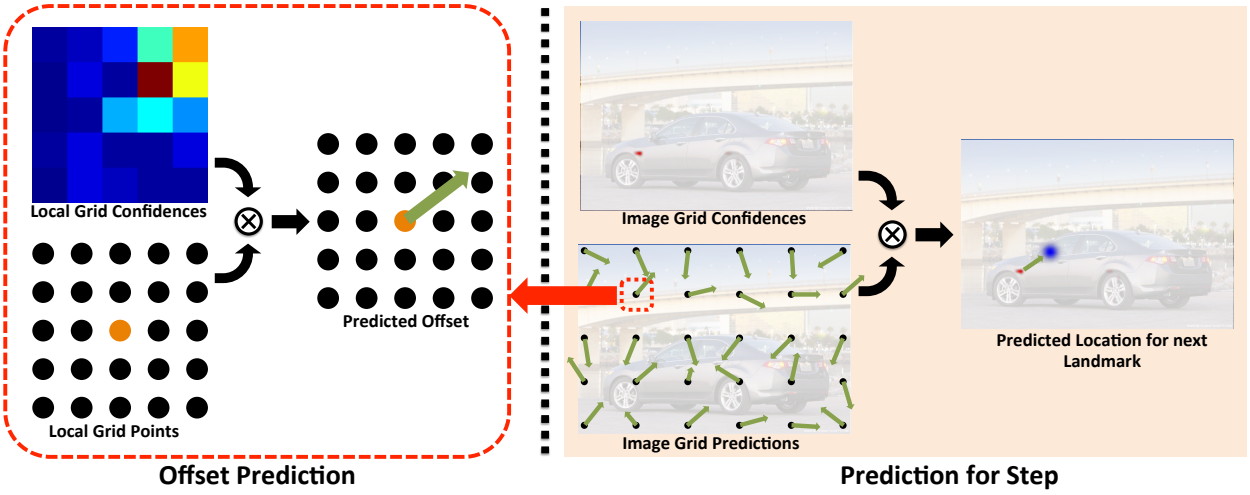

Figure 3. Prediction Scheme for a Step: On the left, we visualize how the offset prediction is made at each location. The model predicts confidences for the points on a local grid around a location of interest. The offset is then computed as a weighted average of the local grid points using the confidences. On the right, we visualize the prediction scheme for the whole image given the individual predictions. Model predicts a confidence in each offset prediction (red blob in the top image, best viewed with zoom). Individual offset predictions are then averaged using the confidences as weights to produce the final prediction. The prediction is then encoded as a radial basis kernel centered at the prediction (blue blob).

The local scheme for producing each $p_i^{(s)}$ looks at both the image features and the predicted location by the previous step, $P^{(s-1)}$. The confidence $c_i^{(s)}$ is a softmax over all locations, computed as $c_i^{(s)} = e^{z_i^{(s)}} / \sum_i e^{z_i^{(s)}}$, where $z_i^{(s)} \in \mathbb{R}$ is the output from the network for confidence at l_i in step s. The right half of Figure 3 visualizes the prediction scheme. Locations with high confidences can be seen as a red blob for each of the steps in figure 2 and 3 (best viewed with zoom).

$P^{(s)}$ is then encoded as a feature map that is passed on to the subsequent step together with the image feature maps. The encoding is done by placing a radial basis kernel of fixed bandwidth, $\beta = 15$, centered at the predicted location (blue blob, Figure 2). Note that encoding the prediction as a feature map instead of as a rigid constraint for the next step allows it to easily ignore the prediction from the previous step if necessary. This flexibility is specially helpful in early stages of the training when we do not have reliable estimates of the latent landmarks or their prediction models.

Furthermore, the scheme of $P^{(s)}$ as a weighted average of several individual predictions is robust to individual variances because it averages over redundant information from several locations. With proper initialization at the beginning of the training, we can ensure that all the locations have non-zero weights and thus are explored as potential latent landmarks.

3.3. The Estimate at a Location

We need a prediction $p_i^{(s)}$ from location l_i at step s. Pure regression works poorly because it is sensitive to learning rate and weight scales, and it is difficult to confine predictions to a range.

Instead, we place a local grid of G points over each l_i (Figure 3, left). Each grid point has coordinates g_j relative to l_i. We train the network to produce G confidence values $o_{j,i}^{(s)}$ for $j \in \{1, \ldots, G\}$ and at each location. These $o_{j,i}^{(s)}$ are a softmax of network outputs which themselves depend on the image as well as the feature map representing $P^{(s-1)}$. Each l_i then produces the estimate

$$p_i^{(s)} = l_i + \sum_{j=1}^{G} o_{j,i}^{(s)} g_j \tag{2}$$

Our scheme has several strengths. The network is required to predict confidences, rather than locations, and so deals with well-scaled values. By construction, each prediction is within the range specified by the local grid. Finally, redundancy helps control the variance of the prediction.

In our experiments we use a local 5×5 grid with $g_j(x), g_j(y) \in \{-50, -25, 0, 25, 50\}$ pixels.

3.4. Training

For regression using neural networks, the usual choice of L_2 loss requires careful tuning of learning rate. Setting it too high results in an explosion of gradients at the beginning and too low slows down the learning in later epochs.

Instead, we use Huber loss (eq. 3) for robustness.

$$\mathcal{H}(x) = \begin{cases} \frac{x^2}{2\delta}, & \text{if } |x| < \delta. \\ |x| - \frac{\delta}{2}, & \text{otherwise.} \end{cases} \quad (3)$$

For a vector $\mathbf{x} \in \mathbb{R}^D$ we define Huber loss as $\mathcal{H}(\mathbf{x}) = \sum_{i=1}^{D} \mathcal{H}(x_i)$. Robustness arises from the fact that the gradients are exactly one for large loss values ($|x| > \delta$), and less than one for smaller values ensuring stable gradient magnitudes. We use $\delta = 1$.

Assume that we know the regression target $y_*^{(s)}$ for step s. Then, given the prediction $P^{(s)}$, we define the loss for step s as following

$$\mathcal{L}^{(s)} = \mathcal{H}(P^{(s)} - y_*^{(s)}) + \gamma \sum_{i=1}^{L} c_i^{(s)} \mathcal{H}(p_i^{(s)} - y_*^{(s)}) \quad (4)$$

The first term enforces that the prediction $p^{(s)}$ coincides with the target $y_*^{(s)}$. The second term enforces that the individual predictions for each location also fall on the target, but the individual losses are weighted by their contribution to the final prediction. We found that the use of this term with a small value of $\gamma = 0.1$ consistently leads to solutions that generalize better.

The regression target for the final step S is the known ground truth location y_*. But we do not have supervision for the intermediate steps. We would like our step s to predict the location of the latent landmark of the next step $s+1$. Note that the latent landmark for the next step is considered to be the set of locations in the image that the model considers to be predictive and therefore assigns high confidences $c_i^{(s+1)}$. We set $y_*^{(s)} = \sum_{i=1}^{L} c_i^{(s+1)} l_i$, i.e. as the centroid of locations with confidence weights $c_i^{(s+1)}$ in the next step. This setting encourages the prediction from step s to coincide with the locations that are predictive in next step.

We define the full loss for a given sample as a weighted sum of the losses from individual steps as following

$$\mathcal{L} = \sum_{s=1}^{S} \lambda_s \mathcal{L}^{(s)} + \mathcal{R}(\theta) \quad (5)$$

We use $\lambda_s = 0.1$, except for the final step S where $\lambda_S = 1$, assigning more weight to the target prediction. $\mathcal{R}(\theta)$ is a regularizer for the parameters of the network. We use L_2 regularization of network weights with a multiplier of 0.005.

Training Details: We train our model through back-propagation using stochastic gradient descent with momentum. The errors are back-propagated across the steps through the radial basis kernel based feature encoding of the latent landmark prediction in each step. Since our model is recurrent, we found that the use of gradient scaling makes the optimization better behaved [30]. We do this by scaling the gradients with combined L_2 norm more than 1000, back down to 1000 for all filters of each layer individually. We initialize the weights using the method suggested by Glorot *et al*. [16].

We augment the datasets by including left-right flips as well as random crops near the border. The images are scaled to make the longest side 500 pixels.

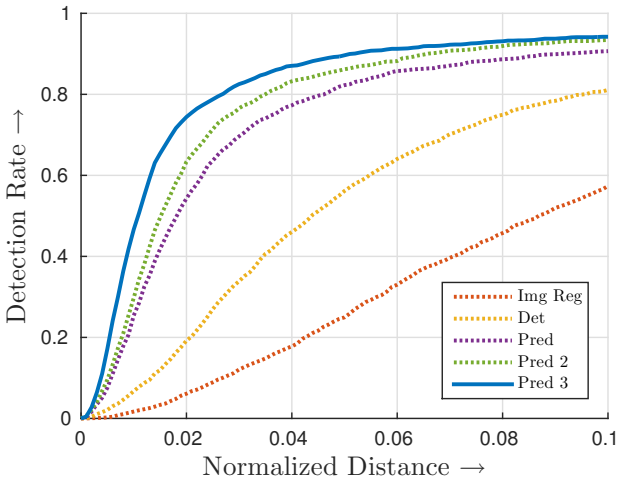

Figure 4. Our method localizes car door handles very accurately. Pred 3, the three step *SeqPrediction* scheme, outperforms the other schemes. Det, Pred and Pred 2 are *Detection*, *Prediction* and two step *SeqPrediction* schemes respectively. Img Reg is a baseline that replaces classification layer by a regression layer in the VGG-M model. Table 1 reports detection rates at a fixed normalized distance of 0.02.

				Seq. Prediction	
Method	Img Reg	Det	Pred	Pred 2	Pred 3
Det. Rate	6.1	19.2	54.3	63.3	**74.4**

Table 1. Detection rates for the Car Door Handle Dataset at the fixed normalized distance of 0.02. The three step scheme (Pred 3) performs significantly better than the alternatives. Refer to Figure 4 for more details.

4. Experiments

We present two new datasets: The Light Switch Dataset (LSD) and the Car Door Handle Dataset (CDHD) that emphasize practically important and hard to find little landmarks. Further, we evaluate our method on more difficult variants of two established tasks: 1) beak localization on the Caltech UCSD Birds Dataset (CUBS); 2) wrist localization on the Leeds Sports Dataset (LSP). Note that we refer to the three step *SeqPrediction* as *Ours* in the following unless specified otherwise .

Evaluation Metric: We adopt the generally accepted met-

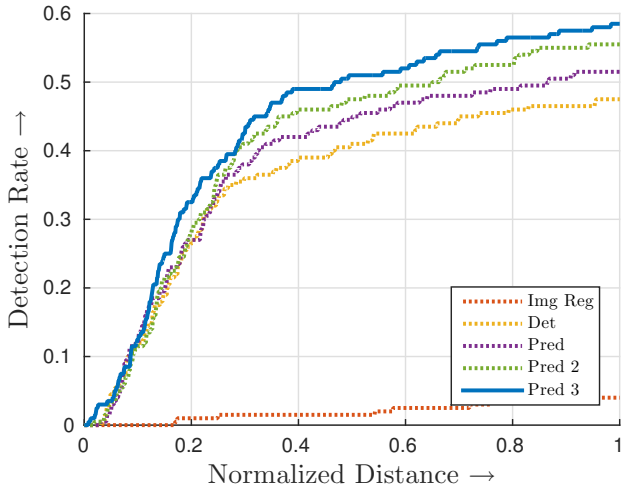

Figure 5. Our method localizes light switches relatively well in comparison to the baselines. The baselines are the same as the ones for Car Door Handle dataset. Table 2 reports detection rates at a fixed normalized distance of 0.5.

ric of plotting detection rate against normalized distance from ground truth for all datasets except CUBS, where PCP is used. Normalization is based on torso height for LSP, car bounding box height for CDHD, and switch board height for LSD. For CUBS we report PCP as used in [24]. It is computed as detection rate for an error radius defined as $1.5 \times \sigma_{human}$, where σ_{human} is the standard deviation of the human annotations.

4.1. Car Door Handle Dataset

Our method finds car door handles very accurately (Figure 4 and 7 and Table 1), with superior performance to various baselines. Det is the *Detection* method, Pred is the *Prediction* method and Pred 2 and Pred 3 are two and three step *SeqPrediction* methods respectively (§ 3). Use of *Prediction* instead of *Detection* gives considerable performance improvement, while *SeqPrediction* provides additional improvement. Img Reg is a baseline implemented by taking the VGG-M model [7] that was pre-trained on ImageNet [10], removing the top classification layer and replacing it by a 2D regression layer. The learning rate for all the layers was set to 0.1 times the learning rate λ_r for the regression layer. The model performed best with a learning rate $\lambda_r = 0.01$, chosen by trying values in $\{0.1, 0.01, 0.001\}$. We noticed that the baseline generalized poorly in all the experiments. This is likely due to a combination of VGG-M model being relatively large in comparison to the dataset size, task being regression instead of classification and hyper-parameter range explored being suboptimal.

Dataset details: To collect a dataset with car door handles, 4150 images of the Stanford Cars dataset for fine grained categorization [21] were annotated. Annotators were asked to annotate the front door handle of the visible side. The handle was marked as hidden for frontal views of the car when it was not visible. We use the training and test split of the original dataset.

				Seq. Prediction	
Method	Img Reg	Det	Pred	Pred 2	Pred 3
Det. Rate	1.5	41.0	44.5	47.5	**51.0**

Table 2. Detection rates for the Light Switch Dataset at the fixed normalized distance of 0.5. Again, the three step scheme performs better than alternatives.

Method	PCP
Liu *et al*. [24]	49.0
Liu *et al*. [25]	61.2
Shih *et al*. [34]	51.8
Ours	**64.1**

Table 3. Our method outperforms several state of the art methods for localizing beaks on the Caltech UCSD Birds Dataset. Note that this comparison is biased against our method; others are trained with *all* landmarks while ours is supervised only by beak.

4.2. Light Switch Dataset

Our method finds light switches with reasonable accuracy (Figure 5 and 7 and Table 2). Again, the three step scheme performs better than the alternatives. The baselines are the same as the ones for the Car Door Handle dataset. Img Reg baseline again generalizes poorly with LSD being significantly smaller than CDHD.

Dataset details: With the aim of building a challenging single landmark localization problem, we collected the Light Switch Dataset (LSD) with 822 annotated images (622 train, 200 test). Annotators were asked to mark the middle points of the top and bottom edge of the switch board. The location of the light switch is approximated as the mean of these. These two points also provide approximate scale information used for normalization in evaluation. This dataset is significantly harder than the Car Door Handles dataset as context around light switches exhibits significant variation in appearance and scale.

4.3. Caltech UCSD Birds Dataset - Beaks

Caltech-UCSD Birds 200 (2011) dataset (CUBS 200) [43] contains 5994 training and 5794 testing images with 15 landmarks for birds. We evaluate our approach on the task of localizing *beak* as the target landmark. We chose beak because it is one of the hardest landmarks and several state of the art approaches do not perform well on this. We used the provided train and test splits for the dataset.

Our method, *while having access only to the beak lo-*

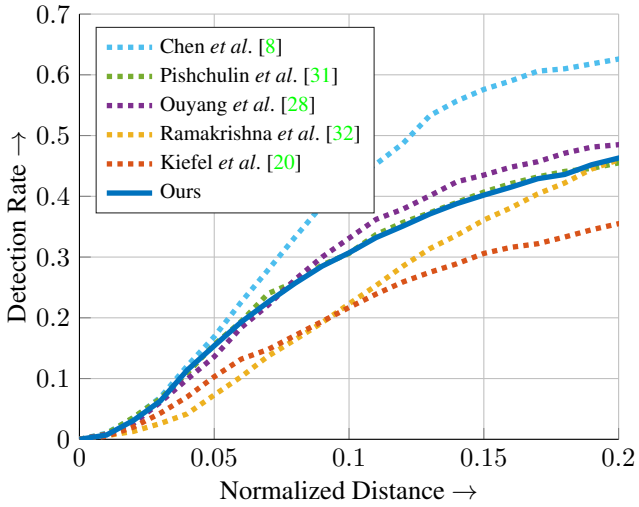

Figure 6. Our method, while supervised only by the location of left wrist, performs competitively against several state of the art methods for localizing the left wrist landmark on the Leeds Sports Dataset.

cation during training (all other methods are trained using other landmarks as well), outperforms several state of the art methods (Table 3).

4.4. Leeds Sports Dataset - Left Wrist

Leeds Sports Dataset (LSP) [17] contains 1000 training and 1000 testing images of humans in difficult poses with 14 landmarks. We choose left wrist as the target landmark as wrists are known to be difficult to localize. We use the Observer Centric (OC) annotations [13] and work with provided training/test splits.

Our method (Figure 6) performs competitively with several recent works all of which train their method using other landmarks.

5. Discussion

Figure 7 shows some qualitative results for various datasets. First thing to note is the pattern in the locations of the latent landmarks for each of the datasets. For cars, the system tends to find the wheel as the first latent landmark and then moves towards the door handle in subsequent steps. For light switches it relies on finding the edge of the door first. For birds, the first landmark tends to be on the neck, followed by one near the eye and the last tends to be outside at the curve of neck and beak. It is remarkable that these patterns emerge solely from the supervision of the target landmark. Also, note that these patterns are not rigid; they adapt to the image evidence. This is primarily due to the fact that our method does not impose any hard constraints. Later steps can choose to ignore the evidence from the earlier steps. This property allows our model to be trained effectively, especially in the beginning when the latent landmarks and their prediction models are not known.

Our method highlights the trade-off inherent in parts vs. larger template. Parts assume structure, reducing parameters and variance in their estimation. While larger templates support richer models, but with more parameters resulting in larger variance.

We explored two other architectures for propagating information from one step to the next and found that the current scheme performs the best in terms of the final performance. In the first scheme, step-specific weights were at the top instead of at the bottom of the recurrent portion of our model (Figure 2, middle block). In the second scheme, instead of passing the encoded prediction as a feature, it was used as a prior to modulate the location confidences of the next step.

6. Conclusion

We described a method to localize little landmarks by finding a sequence of latent landmarks and their prediction models. We demonstrated strong performance of our method on harder variants of several existing and new tasks. The success of our approach arises from the spatial prediction scheme and the encoding of information from one step to be used by the next. A novel and well behaved local estimation model coupled with a robust loss aids training. Promising future directions include localizing multiple targets, generalizing sequence of latent landmarks to directed acyclic graphs of latent landmarks, and accumulating all the information from previous steps to be used as features for the next step.

Acknowledgments: This work is supported in part by ONR MURI Awards N00014-10-1-0934 and N00014-16-1-2007. We would like to thank NVIDIA for donating some of the GPUs used in this work.

References

[1] B. Alexe, N. Heess, Y. W. Teh, and V. Ferrari. Searching for objects driven by context. In *Advances in Neural Information Processing Systems*, pages 881–889, 2012.

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

[3] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] J. Caicedo and S. Lazebnik. Semantic guidance of visual attention for localizing objects in scenes. In *ICCV*, 2015.

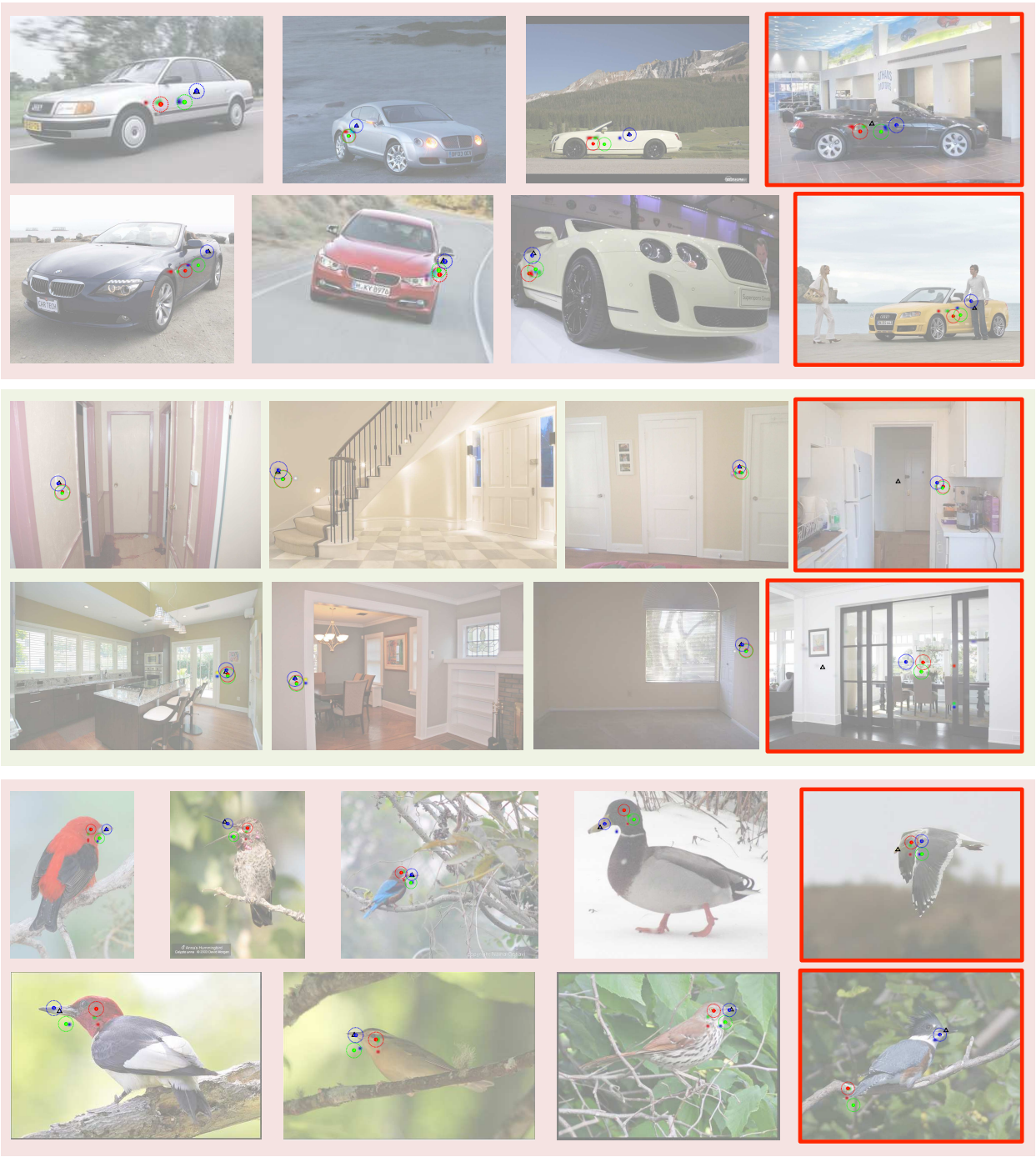

Figure 7. Qualitative results for the Car Door Handle Dataset (top two rows), Light Switch Dataset (middle two rows) and the Caltech UCSD Birds Dataset (last two rows) (best viewed with zoom). Ground truth locations are shown as a black triangle. Step 1, Step 2 and Step 3 are color coded as Red, Green and Blue respectively. Colored blobs show the locations of the latent landmarks for each step. Solid circles with numbers show the predicted location of the next latent landmark by each step. Dotted circles show the bandwidth of the radial basis kernel used to encode the predictions. Note the patterns in the locations of the latent landmarks. For cars, the first latent landmark tends to be on the wheel and later ones get closer to the door handle. For light switches it relies on finding the edge of the door first. For birds, the first landmark tends to be on the neck, followed by one near the eye and the last tends to be outside at the curve neck and beak. Rightmost column shows failure/interesting cases for each dataset (red border). It is evident that the latent landmarks tend to be close to the prediction from the previous step, though they are not constrained to do so (bottom right bird image). Typical failure modes include clutter, circumstantial contextual signal (door frame) and rarer examples (e.g. flying bird).

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

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[8] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. *arXiv preprint arXiv:1407.3399*, 2014.

[9] M. Dantone, J. Gall, C. Leistner, and L. van Gool. Human pose estimation from still images using body parts dependent joint regressors. In *CVPR*. IEEE, 2013. to appear.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[11] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012.

[12] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *ICCV*, 2009.

[13] M. Eichner and V. Ferrari. Appearance sharing for collective human pose estimation. In *ACCV*, 2012.

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection a nd semantic segmentation. In *CVPR*. IEEE, 2014.

[16] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[17] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. doi:10.5244/C.24.12.

[18] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013.

[19] L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman. The chains model for detecting parts by their context. In *CVPR*, pages 25–32, 2010.

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

[21] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013.

[22] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *ICCV*, 2005.

[23] H. Larochelle and G. E. Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *NIPS*, pages 1243–1251, 2010.

[24] J. Liu and P. N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In *ICCV*, 2013.

[25] J. Liu, Y. Li, and P. N. Belhumeur. Part-pair representation for part localization. In *ECCV 2014*, 2014.

[26] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *NIPS*, pages 2204–2212, 2014.

[27] K. Murphy, A. Torralba, W. Freeman, et al. Using the forest to see the trees: a graphical model relating features, objects and scenes. *Advances in neural information processing systems*, 2003.

[28] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *CVPR*. IEEE, 2014.

[29] S. N. Parizi, A. Vedaldi, A. Zisserman, and P. Felzenszwalb. Automatic discovery and optimization of parts for image classification. *ICLR*, 2014.

[30] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*, 2012.

[31] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *ICCV*, pages 3487–3494. IEEE, 2013.

[32] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014.

[33] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*, 2010.

[34] K. J. Shih, A. Mallya, S. Singh, and D. Hoiem. Part localization using multi-proposal consensus for fine-grained categorization. *arXiv preprint arXiv:1507.06332*, 2015.

[35] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012.

[36] S. Singh, D. Hoiem, and D. Forsyth. Learning a sequential search for landmarks. In *Computer Vision and Pattern Recognition*, 2015.

[37] Y. Su and F. Jurie. Improving image classification using semantic attributes. *International journal of computer vision*, 100(1):59–77, 2012.

[38] J. Sun and J. Ponce. Learning discriminative part detectors for image classification and cosegmentation. In *International Conference on Computer Vision*, 2013.

[39] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *arXiv preprint arXiv:1312.4659*, 2013.

[40] D. Tran and D. Forsyth. Improved human parsing with a full relational model. In *ECCV*, 2010.

[41] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *PAMI*, 32(10):1744–1757, 2010.

[42] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *CVPR*, 2011.

[43] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[44] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[45] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *PAMI*, 2013.

[46] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. *arXiv preprint arXiv:1504.03293*, 2015.