# Joint Spatial and Layer Attention for Convolutional Networks

Tony Joseph[1]
tony.joseph@uoit.net
Konstantinos G. Derpanis[23]
www.scs.ryerson.ca/kosta
Faisal Z. Qureshi[1]
faculty.uoit.ca/qureshi

[1] Faculty of Science
Ontario Tech University
Oshawa, Canada

[2] Department of Computer Science
Ryerson University
Toronto, Canada

[3] Samsung AI Centre
Toronto, Canada

**Abstract**

In this paper, we propose a novel approach that learns to sequentially attend to different Convolutional Neural Networks (CNN) layers (i.e., "what" feature abstraction to attend to) and different spatial locations of the selected feature map (i.e., "where") to perform the task at hand. Specifically, at each Recurrent Neural Network step, both a CNN layer and localized spatial region within it are selected for further processing. We demonstrate the effectiveness of this approach on two computer vision tasks: (i) image-based six degrees of freedom camera pose regression and (ii) indoor scene classification. Empirically, we show that combining the "what" and "where" aspects of attention improves network performance on both tasks. We evaluate our method on standard benchmarks for camera localization (Cambridge, 7-Scenes, and TUM-LSI) and for scene classification (MIT-67 Indoor Scenes). For camera localization our approach reduces the median error by 18.8% for position and 8.2% for orientation (averaged over all scenes), and for scene classification, it improves the mean accuracy by 3.4% over previous methods.

## 1 Introduction

Convolutional Neural Networks (CNNs) [24] are central models in a broad range of computer vision tasks, e.g., [9, 10, 12, 22, 25]. Generally, the processing of input imagery consists of a series of convolutional layers interwoven with non-linearities (and possibly downsampling) that yield a hierarchical image representation. As deterministic processing proceeds in a CNN, both the spatial scope (i.e., the effective receptive field) and the level of feature abstraction [30, 48] of the representation gradually increase. Motivated by our understanding of human visual processing [32, 39] and initial success in natural language processing [9], an emerging thread in computer vision research consists of augmenting CNNs with an attentional mechanism. Generally, the goal of attention is to dynamically focus computational resources on the most salient features of the input image as dictated by the task.

In this paper, we present an approach that incorporates attention into a standard CNN in two ways: (i) a layer attention mechanism (i.e., "what" layer to consider) selects a CNN

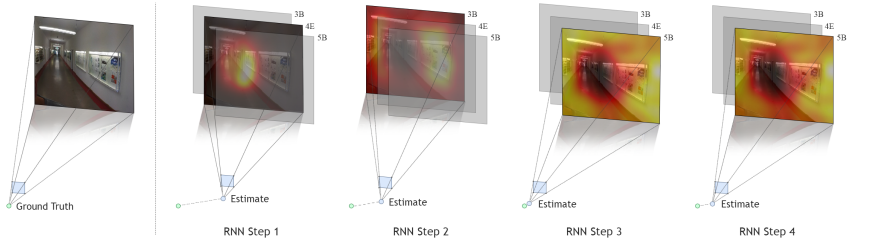


Figure 1: Overview of our approach to 6-DoF camera localization. Given a set of CNN feature layers (GoogLeNet [37] Conv-{3B, 4E, 5B} layers shown) our approach to attention uses an RNN to sequentially select a set of feature layers (highlighted by the non-grey images) and corresponding locations in the layers (highlighted by the heat maps, where darker regions indicate higher spatial weight). Finally, the processed attended features are used for regressing the camera position and orientation.

layer, and (ii) a spatial attention mechanism selects a spatial region within the selected layer (i.e., "where") for subsequent processing. Layer and spatial attention work in conjunction with a Recurrent Neural Network (RNN). At each time step, first a layer is selected and next spatial attention is applied to it. The RNN progressively aggregates the information from the attended spatial locations in the selected layers. The aggregated information is subsequently used for regression or classification. Our model is trained end-to-end, without requiring additional supervisory labels. Empirically, we consider both regression (i.e., six degree of freedom, 6-DoF, camera localization) and classification (i.e., scene classification) tasks. Figure 1 presents an overview of our approach to layer-spatial attention for 6-DoF camera localization.

The guiding intuition behind our approach is that the optimal feature set for a task may be distributed across a variety of feature abstraction levels and spatial regions. Here, we let an RNN identify the optimal features to aggregate. For instance, in the context of image-based localization, a scene may contain both a set of salient objects captured by high-level features, such as a window or door, and texture-like regions captured by low-level features. Prior localization methods have exclusively relied on either low-level features (e.g., [43]) or high-level ones, e.g., [2, 20]. Our approach considers the spectrum of feature abstractions in a unified manner. The project page can be found at: http://vclab.science.uoit.ca/projects/uan/uan.html

## 1.1 Contributions

This paper makes the following contributions:

1. We propose an attention model that learns to sequentially attend to different CNN layers (i.e., different levels of abstraction) and different spatial locations (i.e., specific regions within the selected feature map) to perform the task at hand.

2. We augment a standard CNN architecture, GoogLeNet [37], with our attention model and empirically demonstrate its efficacy on both regression and classification tasks: 6-DoF camera localization regression and indoor scene classification. We evaluate the proposed architecture on standard benchmarks: (a) Cambridge landmarks [20], 7-scenes [35], and TU Munich large-scale indoor (TUM-LSI) [41] for camera pose estimation; and (b) MIT-67 indoor scenes [31] for scene classification. For camera localization our approach re-

duced the overall median error by 12.3% for position and 13.9% for orientation on Cambridge Landmarks, 19.3% for position and 8.83% for orientation on 7-Scenes, and 25.1% for position and 1.79% for orientation on TUM-LSI over the baseline [41]. For indoor scene classification on MIT-67 our approach improves the mean accuracy by 3.4% over the baseline [8]. In both tasks, the baseline methods use the *same* base convolutional network.

## 2 Related works

**Attention.** Attention is a mechanism that dynamically allocates computational resources to the most salient features of the input signal. Attention has appeared in a variety of recent architectures [3, 15, 23, 29, 38, 40, 47]. A natural way to implement a sequential attentional probing mechanism is with a RNN or variant (e.g., Long Short-Term Memory, LSTM [11, 47]) in conjunction with a gating function [36, 42, 45] that yields a soft (e.g., softmax or sigmoid) or hard attention [42, 46]. The attentional policy is learned without an explicit training signal, rather the task-related loss alone provides the training signal for the attention-related weights. In this work, we incorporate both soft (spatial selection) and hard (layer selection) attention in an end-to-end trainable architecture. Most closely related to the current work are the soft and hard selection mechanisms proposed by Xu *et al*. [46] and Veit and Belongie [40], respectively. Xu *et al*. [46] proposed an end-to-end trainable soft spatial attention architecture for image captioning. We adapt this soft attention architecture for our purposes and further extend it to include hard attention. Veit and Belongie [40] proposed a dynamic convolutional architecture that selects whether or not information propagates through a given CNN layer during the forward pass. Similar to Veit and Belongie [40], we use the recently proposed Gumbel-Softmax to realize our discrete (hard) selection of layers.

**Image-based camera pose localization.** Low-level features (e.g., SIFT [27]) have dominated the camera pose localization literature, e.g., [2, 6, 26, 33]. An early example of using high-level features for camera localization appeared in Anati *et al*. [2], where object detections heatmaps were used for localization. More recently, high-level CNN features have garnered attention. These features can be considered as soft proxies to object detections. Kendall *et al*. [17, 20] proposed PoseNet, an image-based 6-DoF camera localization method. PoseNet regresses the camera position and orientation based on input provided by a CNN layer. Kendall and Cipolla [18] reconsidered the loss used in PoseNet to integrate additional geometric information. Walch *et al*. [41] extended the PoseNet approach by introducing an LSTM-based dimensionality reduction step prior to regression to avoid overfitting. In each case, the networks rely on features from a single manually selected layer, located relatively high in the feature hierarchy. In contrast, we propose an attentional network that is capable of dynamically integrating the most salient features across the spectrum of feature abstractions (capturing potentially texture-like and object-related features as necessary).

**Indoor scene classification.** To demonstrate the generality of our approach we also consider a classification task, indoor scene classification. Here, a wealth of research has considered both handcrafted (e.g., [8, 16]) and learned deep features, e.g., [8, 34]. In this work, we compare our approach using a standard deep architecture, GoogLeNet [37], which we also use as the base network for our layer-spatial attention method.

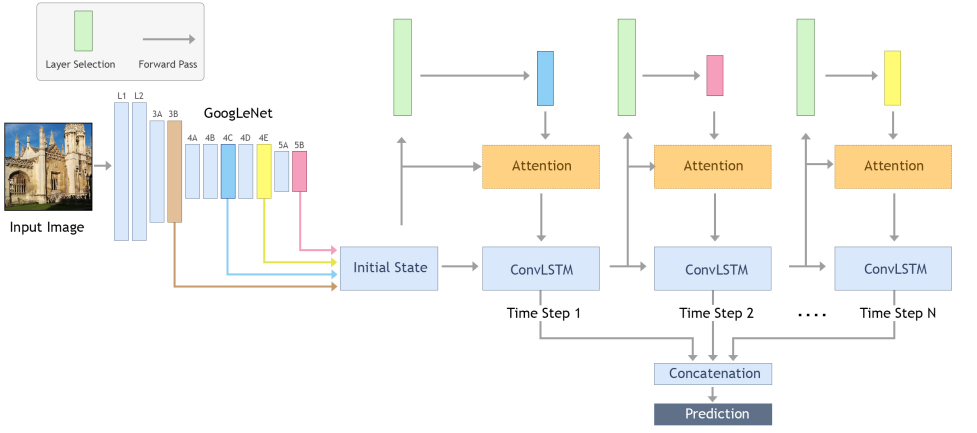


Figure 2: Overview of our layer-spatial attention architecture. Layer-spatial attention is realized within a Conv-LSTM framework, where the layer attention uses the previous hidden state, and spatial attention uses both the selected layer and the previous hidden state. After a constant number of Conv-LSTM steps, the $N$ hidden states from all steps are concatenated and used for regression or classification.

# 3 Technical approach

Our layer-spatial attention network sequentially probes the input signal over a fixed number of steps. It is comprised of a soft attention that selects a spatial location within the selected layer (Sec. 3.1) and a hard selection mechanism that selects a CNN layer (Sec. 3.2). The attention network is realized using a convolutional LSTM (Conv-LSTM) [45]. Figure 2 provides an overview of our architecture. At each Conv-LSTM step, the layer attention selects a CNN layer and spatial attention localizes a region within it. After $N$ recurrent steps, the Conv-LSTM hidden states for all steps are concatenated and used for classification or regression.

## 3.1 Where: Spatial Attention

We adapt the recurrent model from Xu *et al.* [46] with soft spatial attention as the foundation of our method. At each time step $t$, the spatial attention mechanism receives as input the selected layer $\mathbf{f} \in \mathbb{R}^{h_f \times w_f \times d_f}$ (see Sec. 3.2) and the recurrent hidden state $\mathbf{h}_t \in \mathbb{R}^{h_h \times w_h \times d_h}$ from the previous step. The soft attention layer is implemented as follows:

$$\begin{aligned} \mathbf{h}_{att} &= \mathbf{h}_t * \mathbf{E}_h \\ \mathbf{f}'_{att} &= \mathrm{ReLU}(\mathbf{h}_{att} + \mathbf{f}) \\ \mathbf{f}_{att} &= \mathbf{f}'_{att} * \mathbf{E}_{att} \\ \mathbf{O}_{att} &= \mathrm{softmax}(\mathbf{f}_{att} * \mathbf{C}_A) \odot \mathbf{f}, \end{aligned} \tag{1}$$

where $*$ denotes the convolutional operator and $\odot$ is element-wise multiplication. The attention layer consists of three convolutional layers, $\mathbf{E}_h$, $\mathbf{E}_{att}$, and $\mathbf{C}_A$, which compute two embeddings and (unscaled) attention mask, respectively. The embedding layer, $\mathbf{E}_h$, is used to transform the hidden state channel dimension to bring it equal to the input layer's channel

dimension. The second convolutional layer, $\mathbf{E}_{att}$, is a hidden layer before the last convolutional layer, $\mathbf{C}_A$. The $\mathbf{C}_A$ layer computes the unscaled attention mask with dimensions $h_f \times w_f \times 1$. The final attention mask is computed by taking the softmax of the unscaled attention mask. The output of the attention layer $\mathbf{O}_{att}$ is obtained by taking an element-wise multiplication between the features in each channel and attention map.

## 3.2 What: Layer Attention

In layer attention (i.e., "what" features to attend) a CNN layer is selected whose feature map is deemed to contain the most salient information at the current recurrent step. Our layer attention involves a discrete (hard) selection of a CNN layer. Here, we use the recently proposed continuous relaxation of the Gumbel-Max trick [7], the Gumbel-Softmax [14, 28], to realize the discrete selection of layers.

Gumbel-Max provides a simple and efficient way to draw samples from a categorical (discrete) distribution:

$$z = \text{one\_hot}(\arg\max[g_i + \log \pi_i]), \tag{2}$$

where, $g_1, ..., g_k$ are i.i.d. samples drawn from the Gumbel$(0,1)$ distribution, and $\pi_i$ are unnormalized probabilities. Samples $g$ are drawn using the following procedure: (i) draw sample $u \sim \text{Uniform}(0,1)$; and (ii) set $g = -\log(-\log(u))$. In the forward pass (and during testing), we compute the arg max of the unnormalized probabilities. In contrast, in the backward pass the arg max is approximated with a softmax function:

$$y_i = \frac{\exp\left(\frac{\log(\pi_i)+g_i}{\tau}\right)}{\sum_{j=1}^{k} \exp\left(\frac{\log(\pi_j)+g_j}{\tau}\right)}, \tag{3}$$

where $k$ is the number of CNN layers that are considered for selection, $i \in [1,k]$, and $\tau$ represents temperature. (This approach is the straight-through version of the Gumbel-Softmax estimator proposed in [14].) During training the temperature, $\tau$, is progressively lowered. As the temperature approaches zero, samples from the Gumbel-Softmax distribution closely approximate those drawn from a categorical distribution.

For layer attention, we realize the (layer) selection scores (i.e., unnormalized probabilities) at each recurrent step as the output of a fully connected layer computed using the previous hidden state. During the forward pass we perform layer selection using Eq. 2 and in the backward pass gradients are computed using Eq. 3 to keep our architecture end-to-end trainable.

## 3.3 Tasks

After $N$ Conv-LSTM steps, the hidden states are concatenated, average pooled, and passed onto a fully connected layer for (regression/classification) prediction. To ensure that our comparisons are meaningful, and that any differences in the performance of our method to those posted by previous methods are due to our attention mechanism, we use the exact same losses as those used by our baselines.

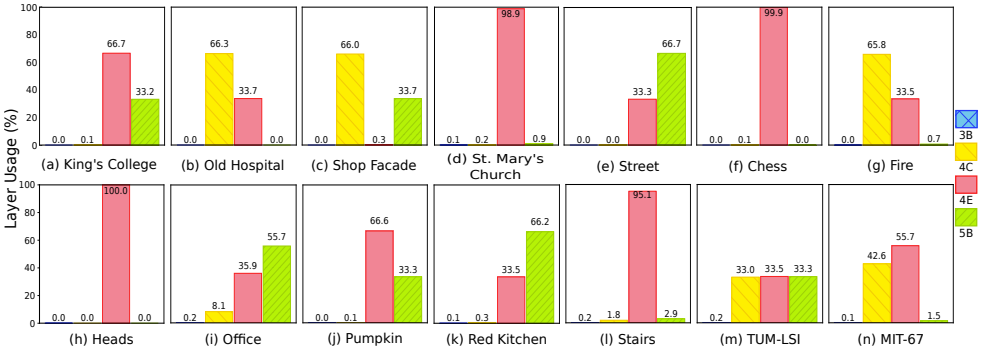


Figure 3: Layer Selection Frequencies on all four datasets on the test set. (a) - (e) are Cambridge Landmarks scenes, (f) - (l) are scenes from 7-Scenes, (m) and (n) are TUM-LSI, and MIT-67 dataset, respectively. The bins refer to the GoogLeNet[37] Conv-{3B, 4C, 4E, 5B} layers. The vertical axis represents layer usage percentages.

### 3.3.1 Camera pose estimation

The proposed camera localization network takes an RGB image as input and outputs camera position and orientation $[\hat{\mathbf{x}}, \hat{\mathbf{q}}]^\top$, where $\hat{\mathbf{x}} \in \mathbb{R}^3$ and $\hat{\mathbf{q}} \in \mathbb{R}^4$ represented by a quaternions. Camera pose is defined relative to an arbitrary reference frame. We use the same regression loss as our baselines [19, 20, 41] to facilitate direct empirical comparison:

$$\mathcal{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2 + \beta \|\mathbf{q} - \frac{\hat{\mathbf{q}}}{\|\hat{\mathbf{q}}\|_2}\|_2, \tag{4}$$

where $[\mathbf{x}, \mathbf{q}]^\top$ represent ground truth position $\mathbf{x}$ and orientation $\mathbf{q}$. $\beta$ is a scalar hyperparameter that determines the relative weighting between the positional and orientation errors.

### 3.3.2 Indoor scene classification

Consistent with our scene classification baseline [37], we use the standard cross-entropy classification loss:

$$\mathcal{L} = -\mathbf{y}_c^\top \log(\hat{\mathbf{y}}_c), \tag{5}$$

where $\mathbf{y}_c$ is a one-hot encoded class label for class $c$, and $\hat{\mathbf{y}}_c$ is the output of the softmax classifier.

## 3.4 Implementation details

To realize our layer-spatial attention model we use the same basic architecture as Xu *et al.* [46] for sequential spatial attention. We augment this network with hard attention for layer selection. To avoid overfitting, we replace the LSTM layers with ConvLSTM [45] layers that reduce the network weight parameterization. The hidden state size is set to 96. In this work we use a multi-convolutional layer modeled after the Inception module [37] for layer-spatial selection. All experiments use GoogLeNet [37] as the feature extractor to maintain meaningful comparisons with the baseline methods. It is conceivable that using a different base network may yield improved results; however, the focus of our experiments is to study the impact of our proposed layer-spatial attention mechanism. For practical reasons we selected

a sparse set of layers (Conv-{3B, 4C, 4E, 5B}) that capture a range of abstractions. It is straightforward to extend the network to select any layer; however, it will considerably increase the training time. Another consideration is that the layers often have different channel dimensions, which necessitates additional weights for embedding layers.

For camera based image localization, separate mean images were computed for each channel and the images were mean subtracted per channel, similar to [20] and [41]. The $\beta$ in Eq. 4 used to balance the position and orientation loss during training was set to the same values as [20, 41]. The $\beta$ values were set between 250 to 2000 and 120 to 750, for Cambridge landmarks and 7-Scenes datasets, respectively, and to 1000 for TUM-LSI dataset. BatchNorm [13] with default parameters is applied to both spatial attention and layer selection network. Our code is implemented using TensorFlow 1.4 [1]. All models were trained end-to-end using the ADAM [21] optimizer. Additional details about our architecture are provide in the supplementary material.

# 4 Empirical evaluation

## 4.1 Datasets

We evaluate our layer-spatial attention model on a variety of publicly available standard datasets. For 6-DoF camera localization we evaluate on Cambridge Landmarks [20], 7-Scenes [35], and TUM-LSI [41]. For scene classification we evaluate on MIT-67 Indoor Scenes [31]. (Additional information on these datasets can be found in the supplementary material). For camera pose estimation, we resize the images to $256 \times 455$ pixels. As done in our localization baselines [20, 41], separate mean images are computed for each colour channel and the images are mean subtracted per channel. For indoor scene classification, we resize the images to $256 \times 256$. For indoor scene classification images were mean subtracted using the Places dataset [49] image mean. For all experiments, we use crops of $224 \times 224$ pixels (random crops during training and center crops during testing). For indoor scene classification we also used random horizontal flips during training.

## 4.2 Results

Figure 3 shows the frequencies of the GoogLeNet feature layers selected for each dataset on their respective test sets. As can be seen, the datasets predominately utilize more than one layer. Furthermore, the layers most frequently selected differ widely amongst the datasets. We found that for image-based camera localization using three Conv-LSTM steps worked best, after which, the error increases. In the case of indoor scene classification two Conv-LSTM steps performed best. Additional experiments using five recurrent steps are present in the supplementary material for both tasks.

### 4.2.1 Results for Camera localization

Table 1 compares our proposed method against other image-based camera pose regression methods [17, 20, 41]. These methods use GoogLeNet as the source of features for regression, with the baselines limiting features from layer Conv-5B. In terms of the individual scenes, our method achieves the least error in both translation and rotation in the majority of cases at step three. Considering the aggregate results over the respective datasets, we see our method

| Dataset | Area or Volume | PoseNet [11] | Bayesian PoseNet [10] | LSTM PoseNet [11] | Ours | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | Conv-LSTM Step-1 | Conv-LSTM Step-2 | Conv-LSTM Step-3 | Improvement (meter, degree) |
| Great Court | 8000 $m^2$ | - | - | - | - | - | - | - |
| Kings College | 5600 $m^2$ | 1.66 m, 4.86° | 1.74 m, 4.06° | 0.99 m, **3.65**° | 1.02 m, 4.22° | 1.00 m, 4.51° | **0.90** m, 3.70° | +9.09, -1.36 |
| Old Hospital | 2000 $m^2$ | 2.62 m, 4.90° | 2.57 m, 5.14° | 1.51 m, 4.29° | 1.62 m, 4.11° | 1.51 m, 4.02° | **1.36** m, **3.95**° | +9.93, +7.92 |
| Shop Facade | 875 $m^2$ | 1.41 m, 7.18° | 1.25 m, 7.54° | 1.18 m, 7.44° | 1.15 m, 5.45° | 0.95 m, 6.44° | **0.91** m, **5.29**° | +22.8, +28.8 |
| St. Marys Church | 4800 $m^2$ | 2.45 m, 7.96° | 2.11 m, 8.38° | 1.52 m, 6.68° | 1.62 m, 7.22° | 1.59 m, 5.94° | **1.42** m, **6.07**° | +6.57, +1.64 |
| Street | 50000 $m^2$ | - | - | - | 18.7m, 34.1° | 15.0 m, 30.3° | **13.9** m, **30.0**° | - |
| Average [11] | 3319 $m^2$ | 2.08 m, 6.83° | 1.92 m, 6.28° | 1.30 m, 5.52° | 1.35 m, 5.25° | 1.26 m, 5.22° | **1.14** m, **4.75**° | +12.3, +13.9 |
| Chess | 6.0 $m^3$ | 0.32 m, 6.08° | 0.37 m, 7.24° | 0.24 m, 5.77° | 0.17 m, 5.58° | 0.16 m, 5.27° | **0.15** m, **4.79**° | +37.5, +16.9 |
| Fire | 2.5 $m^3$ | 0.47 m, 14.0° | 0.43 m, 13.7° | 0.34 m, 11.9° | 0.32 m, 12.6° | 0.31 m, 11.7° | **0.23** m, **10.0**° | +32.3, +15.9 |
| Heads | 1.0 $m^3$ | 0.30 m, 12.2° | 0.31 m, 12.0° | 0.21 m, 13.7° | 0.18 m, 13.8° | 0.18 m, 14.1° | **0.18** m, **13.7**° | +14.2, +0.00 |
| Office | 7.5 $m^3$ | 0.48 m, 7.24° | 0.48 m, 8.04° | 0.30 m, 8.08° | 0.29 m, 7.63° | 0.29 m, 7.23° | **0.29** m, **8.02**° | +3.33, +0.74 |
| Pumpkin | 5.0 $m^3$ | 0.49 m, 8.12° | 0.61 m, 7.08° | 0.33 m, 7.00° | 0.25 m, 5.46° | 0.25 m, 5.76° | **0.26** m, **6.16**° | +21.2, +12.0 |
| Red Kitchen | 18 $m^3$ | 0.58 m, 8.31° | 0.58 m, 7.51° | **0.37** m, 8.83° | 0.43 m, 8.03° | 0.37 m, 7.49° | 0.39 m, **8.20**° | -2.00, +5.77 |
| Stairs | 7.5 $m^3$ | 0.48 m, 13.1° | 0.48 m, 13.1° | 0.40 m, 13.7° | 0.32 m, 9.98° | 0.31 m, 10.5° | **0.29** m, **12.0**° | +27.5, +12.4 |
| Average All | 6.9 $m^3$ | 0.44 m, 9.01° | 0.46 m, 9.81° | 0.31 m, 9.85° | 0.28 m, 9.01° | 0.26 m, 8.86° | **0.25** m, **8.98**° | +19.1, +9.10 |
| TUM-LSI | 5575 $m^2$ | 1.87 m, 6.14° | - | 1.31 m, 2.79° | 1.32 m, 3.82° | 1.26 m, 3.69° | **0.98** m, **2.74**° | +25.1, +1.79 |

Table 1: Camera localization results. Median localization error achieved by the proposed attention model over three steps on Cambridge Landmarks, 7-Scenes, and TUM-LSI. Bold values indicate the lowest error achieved for each row. Improvement is reported with respect to LSTM-PoseNet [11]. A dash (-) indicates that no result is reported.

| CNNaug-SVM [32] | $S^2$ICA [8] | GoogLeNet [37] | Ours | | | |
|---|---|---|---|---|---|---|
| | | | Conv-LSTM Step-1 | Conv-LSTM Step-2 | Conv-LSTM Step-3 | Improvement (%) |
| 69.0 % | 71.2 % | 73.7 % | 74.5 % | **77.1** % | 76.0 % | +3.4 |

Table 2: Mean accuracy results for indoor scene classification on MIT-67. The proposed method achieves the highest accuracy (shown in boldface). Improvement is reported with respect to the GoogLeNet [37] baseline.

yields significant improvements over the baselines, ranging between 12.3 and 25.1 percent for translation and 1.79 and 13.9 percent for rotation. The TUM-LSI dataset contains large textureless surfaces and repetitive scene elements covering over 5,575 $m^2$. Active search or SIFT-based approaches have been previously shown to perform poorly on this dataset [11]. Our method achieves improvements over the (deep) regression baselines, suggesting that the ability to attend to different CNN layers over successive LSTM steps helps. Figure 4 (top row) shows qualitative results for camera localization. For outdoor scenes, it appears our attention mechanism captures both low-level (e.g., corners) and high-level structures (e.g., rooftops and windows).

### 4.2.2 Results for Indoor scene classification

Table 2 compares our proposed layer-spatial attention method against three baselines [8, 32, 37]. The proposed method achieves best performance after two recurrent steps. Figure 4 (bottom row) shows several qualitative results for indoor scene classification. The layer-spatial attention seems to capture objects and physical scene structures present in the scene. For the Concert Hall image, the attention mechanism appears to focus on the entire image, perhaps focusing on the scene architecture. For the Dental Office image, spatial attention picks out the dental equipment (a permanent fixture) and correctly ignores the person (a

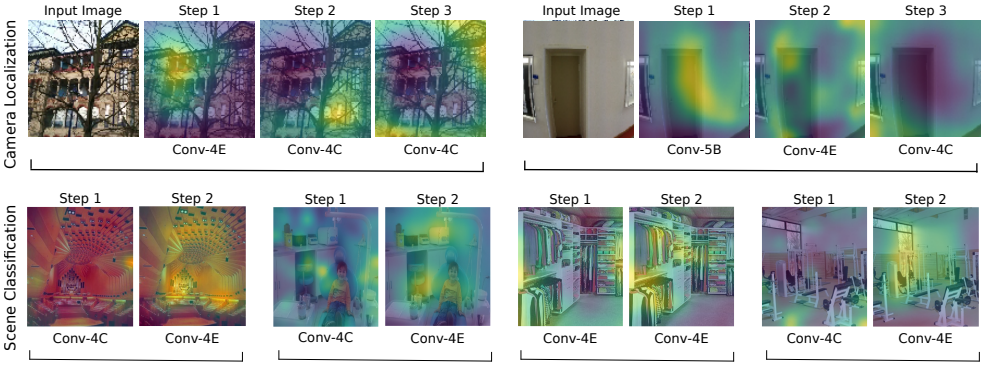


Figure 4: Qualitative results on camera pose localization (top row) and indoor scene classification (bottom row). Top row: input image along with the spatial attention superimposed on the input image for three Conv-LSTM steps. Bottom row: spatial attention superimposed on the input image for two Conv-LSTM steps. The labels underneath each image indicates the selected CNN layer.

| Dataset | Spatial Attention Only | | | Layer Selection Only | Spatial and Layer Attention |
|---|---|---|---|---|---|
| | Conv-3B | Conv-4E | Conv-5B | | |
| Camera-Pose Estimation | | | | | |
| Old Hospital | 1.49 m, 4.29° | 1.42 m, 4.37° | 1.76 m, 4.44° | 2.36 m, 6.28° | **1.36** m, **3.95**° |
| Office | 0.27 m, 7.37° | **0.26** m, **7.35**° | 0.28 m, 7.52° | 0.33 m, 7.97° | 0.29 m, 8.02° |
| TUM-LSI | 1.21 m, 3.26° | 1.13 m, 3.66° | 1.12 m, 3.66° | 5.27 m, 10.8° | **0.98** m, **2.74**° |
| Indoor-Scene Classification | | | | | |
| MIT-67 | 61.6 % | 74.5 % | 74.2 % | 76.4 % | **77.1** % |

Table 3: Ablation study on layer-spatial attention. In all cases, GoogLeNet [37] Conv-{3B, 4E, 5B} layers are used. Bold values indicate the best result achieved for each row.

transient entity). For the Closet image, clothes and cabinetry are selected. Finally, for the Gym image, the proposed attention mechanism selects the exercise equipment.

## 4.3 Ablation study

Table 3 summarizes an ablation study that we performed to gauge the impact of combining layer selection with spatial attention. We chose the Old Hospital (Cambridge Landmarks), Office (7-Scenes), TUM-LSI, and MIT-67 datasets for this ablation study. Old Hospital and Office were selected since we found these to be the most challenging for our proposed network.

We manually selected GoogLeNet's Conv-{3B, 4E, 5B} layers and applied spatial attention to each independently. (Note, the PoseNet results reported in Table 1 use layer Conv-5B without any form of attention for direct position-orientation regression). Our results confirm that it is sometimes beneficial to use layers other than the final CNN layer. Median localization errors, for example, improve for both Old Hospital and Office datasets when we use layers other than Conv-5B. Note that in previous camera pose localization works [17, 20, 41] Conv-5B was manually selected. For indoor scene classification, selecting Conv-4E yields the best result. The last column of Table 3 includes results obtained by combining layer selection and spatial attention. Notice that in three out of four cases shown, the network

achieves best performance (lowest errors in case of camera pose estimation, and highest accuracy in case of indoor scene classification) is achieved when using both layer selection and spatial attention. The second last column in Table 3 includes results when using layer selection alone. The network performance deteriorates when spatial attention is absent.

These results are consistent with our initial guiding intuition that salient information is distributed across the spectrum of feature abstractions, e.g., things vs. stuff. Our proposed layer-spatial attention mechanism exploits this aspect to achieve better performance.

# 5 Conclusion

In this paper, we have presented an architecture that dynamically probes the convolutional layers of a CNN to aggregate and process the optimal set of features for a given task. We introduced an attention architecture that learns to sequentially attend to different CNN layers (i.e., levels of feature abstraction) and different spatial locations within the selected layer. In the context of two vision tasks, camera localization, and scene classification, we empirically showed that our approach to adaptive computation with layer-spatial attention improves regression and classification performance over manually selecting layers as used in our baselines. Our proposed approach to attention is general and may prove useful for other vision tasks.

# 6 Acknowledgments

Konstantinos Derpanis is supported by a Canadian NSERC Discovery grant. He contributed to this article in his personal capacity as an Associate Professor at Ryerson University. Faisal Z. Qureshi is supported by a Canadian NSERC Discovery grant. The authors would thank Kamyar Nazeri (Imaging lab, Ontario Tech University, Canada) for providing support on figures presented in this paper. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

# References


[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A System For Large-Scale Machine Learning. In *Proc. of the Operating Systems: Design and Implementation (OSDI)*, volume 16, pages 265–283, November 2016.

[2] Roy Anati, Davide Scaramuzza, Konstantinos G. Derpanis, and Kostas Daniilidis. Robot localization using soft object detection. In *Proc. of the IEEE Conference on International Conference on Robotics and Automation (ICRA)*, pages 4992–4999, May 2012.

[3] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. In *Proc. of the IEEE Conference on International Conference on Robotics and Automation (ICRA)*, May 2015.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. of the International Conference on Learning Representations (ICLR)*, May 2015.

[5] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Mid-level visual element discovery as discriminative mode seeking. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 494–502, December 2013.

[6] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 834–849, September 2014.

[7] Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*. US Govt. Print. Office, 1954.

[8] Munawar Hayat, Salman H Khan, Mohammed Bennamoun, and Senjian An. A spatial layout and scale invariant feature representation for indoor scene classification. *Proc. of the IEEE Transactions on Image Processing*, pages 4829–4841, August 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of the IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. of the IEEE Conference on International Conference on Computer Vision (ICCV)*, pages 2980–2988, October 2017.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPR)*, pages 1647–1655, June 2017.

[13] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 448–456, July 2015.

[14] Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *Proc. of the International Conference on Learning Representations (ICLR)*, April 2017.

[15] Dinesh Jayaraman and Kristen Grauman. Learning to look around: Intelligently exploring unseen environments for unknown tasks. *CoRR*, abs/1709.00507:1–11, 2017.

[16] Mayank Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 923–930, June 2013.

[17] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Proc. of the IEEE Conference on International Conference on Robotics and Automation (ICRA)*, pages 4762–4769, May 2016.

[18] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8–16, June 2017.

[19] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 5574–5584, December 2017.

[20] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2938–2946, June 2015.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conference on Learning Representations (ICLR)*, May 2015.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105, December 2012.

[23] Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 1243–1251, December 2010.

[24] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 396–404, November 1990.

[25] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPRW)*, pages 105–114, June 2017.

[26] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide Pose Estimation using 3D Point Clouds. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 15–29, October 2012.

[27] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[28] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 3086–3094, December 2014.

[29] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 2204–2212, December 2014.

[30] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

[31] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *Proc. of the IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 413–420, June 2009.

[32] Ronald A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7:17–42, 2000.

[33] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-Scale image-based localization. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(9):1744–1756, 2017.

[34] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPRW)*, pages 806–813, June 2014.

[35] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proc. of the IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, June 2013.

[36] Marijn F. Stollenga, Jonathan Masci, Faustino J. Gomez, and Jürgen Schmidhuber. Deep networks with internal selective attention through feedback connections. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 3545–3553, December 2014.

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. of the IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

[38] Yichuan Tang, Nitish Srivastava, and Ruslan R Salakhutdinov. Learning generative models with visual attention. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 1808–1816, December 2014.

[39] John K. Tsotsos. *A Computational Perspective on Visual Attention*. MIT Press, 2011.

[40] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 3–18, September 2018.

[41] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using LSTMs for structured feature correlation. In *Proc. of the IEEE Conference on International Conference on Computer Vision (ICCV)*, pages 627–637, October 2017.

[42] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proc. of the IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6458, July 2017.

[43] Junqiu Wang, Hongbin Zha, and Roberto Cipolla. Coarse-to-Fine Vision-Based Localization by Indexing Scale-Invariant Features. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 36(2):413–422, 2006.

[44] Ronald J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8(3–4):229–256, May 1992.

[45] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wangchun Woo. Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 802–810, December 2015.

[46] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of the International Conference on Machine Learning, (ICML)*, pages 2048–2057, July 2015.

[47] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, 2014.

[48] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 818–833, September 2014.

[49] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 487–495, December 2014.