Learning to Zoom: A Saliency-Based Sampling Layer for Neural Networks

Adrià Recasens[1](✉), Petr Kellnhofer[1](✉), Simon Stent[2], Wojciech Matusik[1], and Antonio Torralba[1]

[1] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
{recasens,pkellnho,wojciech,torralba}@csail.mit.edu
[2] Toyota Research Institute, Cambridge, MA 02139, USA
simon.stent@tri.global

Abstract. We introduce a saliency-based distortion layer for convolutional neural networks that helps to improve the spatial sampling of input data for a given task. Our differentiable layer can be added as a preprocessing block to existing task networks and trained altogether in an end-to-end fashion. The effect of the layer is to efficiently estimate how to sample from the original data in order to boost task performance. For example, for an image classification task in which the original data might range in size up to several megapixels, but where the desired input images to the task network are much smaller, our layer learns how best to sample from the underlying high resolution data in a manner which preserves task-relevant information better than uniform downsampling. This has the effect of creating distorted, caricature-like intermediate images, in which idiosyncratic elements of the image that improve task performance are zoomed and exaggerated. Unlike alternative approaches such as spatial transformer networks, our proposed layer is inspired by image saliency, computed efficiently from uniformly downsampled data, and degrades gracefully to a uniform sampling strategy under uncertainty. We apply our layer to improve existing networks for the tasks of human gaze estimation and fine-grained object classification. Code for our method is available in: http://github.com/recasens/Saliency-Sampler.

Keywords: Task saliency · Image sampling · Attention
Spatial transformer · Convolutional neural networks · Deep learning

1 Introduction

Many modern neural network models used in computer vision have input size constraints [1–4]. These constraints exist for various reasons. By restricting the input resolution, one can control the time and computation required during both training and testing, and benefit from efficient batch training on GPU. On certain datasets, limiting the input feature dimensionality can also empirically increase performance by improving training sample coverage over the input space.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11213, pp. 52–67, 2018.
https://doi.org/10.1007/978-3-030-01240-3_4

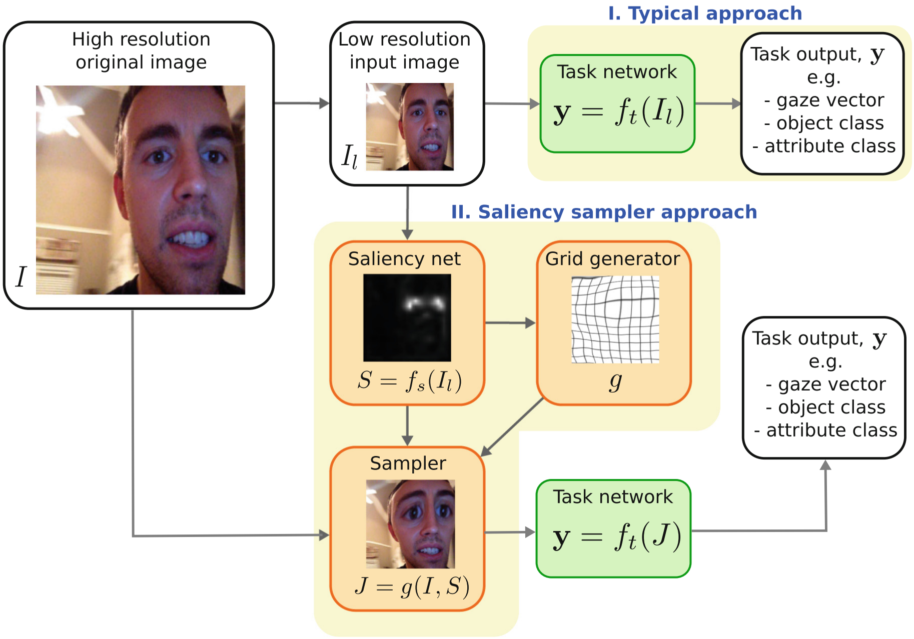

Fig. 1. Outline of our proposed saliency-based sampling layer. Numerous tasks in computer vision are solved by a task network (shown in green), operating on an image I_l which has been downsampled (for performance reasons) from a much larger original image I. For such tasks, where I is available but unused, we show that using a saliency sampler to downsample the image (rather than uniform downsampling) can lead to significant improvement in the task network performance for an identical architecture, as well as beating alternative sampling approaches such as bounding box proposals or spatial transformer networks. Our sampler is differentiable and can be trained end-to-end. The effect of the sampler is to discover and zoom in on (or sample more densely) those regions which are particularly informative to the task. In the case of gaze estimation, as seen here, the sampler locates the eyes as task-salient regions (S) and enlarges them in the resampled image (J). (Color figure online)

When the target input size is smaller than the images in the original dataset, the standard approach is to uniformly downsample the input images. Perhaps the best-known example is the commonly used 224 × 224 pixel input when training classifiers on the ImageNet Large Scale Visual Recognition Challenge [5], despite the presence of a range of image sizes – up to several megapixels – within the original dataset.

While uniform downsampling is simple and effective in many situations, it can be lossy for tasks which require information from different spatial resolutions and locations. In such cases, sampling the salient regions at the necessary (and possibly diverse) scales and locations is essential. Humans perform such tasks by saccading their gaze in order to gather the necessary information with a mixture of high-acuity foveal vision and coarser peripheral vision. Attempts have also been made to endow machines with similar forms of sampling behavior. One popular example from traditional computer vision is SIFT [6], in which keypoints are localised within space and image scale before feature extraction. More recently, region proposal networks have been used widely in object detection [7]. Mimicking the human vision system more closely, mechanisms for task-dependent sequential attention are being developed to allow numerous scene regions to be processed in high resolution (see e.g. [8–10]). However, these approaches surrender some of the processing speed that makes machine vision attractive, and add complexity for proposal generation and evaluating task completion.

In this work we introduce a saliency-based sampling layer: a simple plug-in module that can be appended to the start of any input-constrained network and used to improve downsampling in a task-specific manner. As shown in Fig. 1, given a target image input size, our saliency sampler learns to allocate pixels in that target to regions in the underlying image which are found to be particularly important for the task at hand. In doing so, the layer warps the input image, creating a deformed version in which task-relevant parts of the image are emphasized and irrelevant parts are suppressed, similar to how a caricature of a face tries to magnify those parts of a person's identity which make them stand out from the average.

Our layer consists of a saliency map estimator connected to a sampler which varies sampling density for image regions depending on their relative saliency values. Since the layer is designed to be fully differentiable, it can be inserted before any conventional network and trained end-to-end. Unlike sequential attention models [9–12], the computation is performed in a single pass of the saliency sampler at constant computational cost.

We apply our approach to tasks where the discovery of small objects or fine-grained details is important (see Fig. 2), and consistently find that adding our layer results in performance improvements over baseline networks.

2 Related Work

We divide the related work into three main categories: attention mechanisms, saliency-based methods, and adaptive image sampling methods.

Attention Mechanisms: Attention has been extensively used to improve the performance of CNNs. Jaderberg et al. [13] introduced the Spatial Transformer Network (STN), a layer that estimates a parametrized transformation from an input image in an effort to undo nuisance image variation (such as from object pose in the task of rigid object classification) and thereby improve model generalization. In their work, the authors proposed three types of transformation that could be learned: affine, projective and thin plate spline (TPS). Although our method also applies a transformation to the input image, our application is quite different: we do not attempt to undo variation such as local translation or rotation; rather we try to vary the resolution dynamically to favor regions of the input image which are more task salient. While our method could be

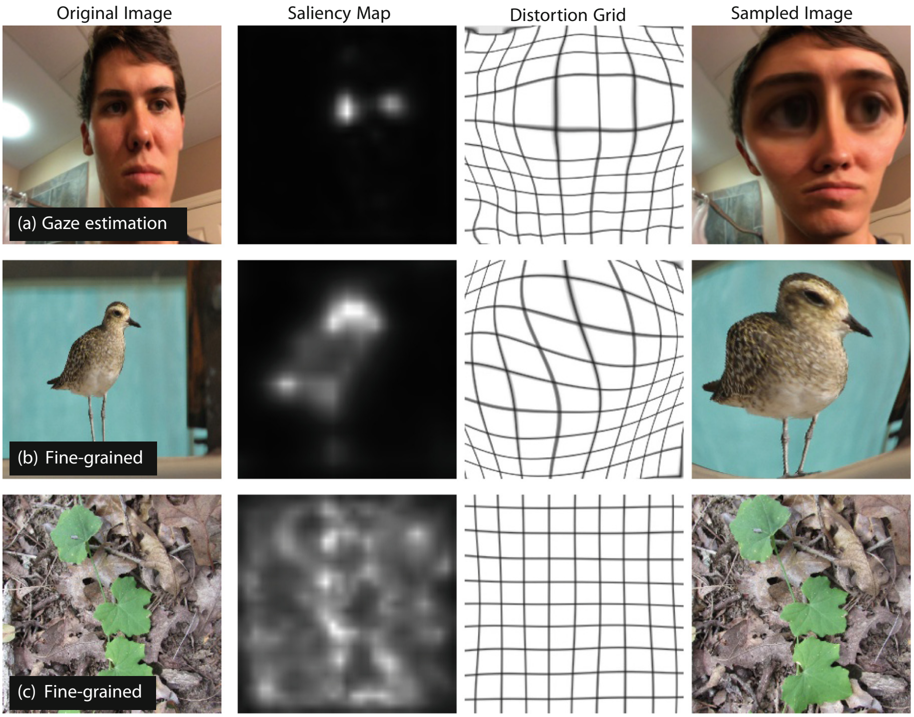

Fig. 2. Examples of resampled input images for various tasks using our proposed saliency sampler. Our module is able to discover saliency according to the task: for gaze estimation in (a), the sampler learns to zoom in on the subject's eyes to allow for higher precision gaze estimation; for fine-grained classification in (b), the sampler zooms in important parts of the bird's anatomy while cropping out much of the empty image; in (c), when no clear salient area is detected, the sampler defaults to a near-uniform sampling.

encapsulated within the TPS approach of [13], we implicitly prevent extreme transformations and fold-overs, which can easily occur for a TPS-based spatial transformer (and which also makes direct estimation of a non-parametrized sampling map intractable). We believe that this helps to prevent dramatic failures and therefore helps to make the module easier to learn.

Deformable convolutional networks (DCNs), introduced by Dai et al. [14], follow a similar motivation to STNs. They show that convolutional layers can learn to dynamically adjust their receptive fields to adapt to the input features and improve invariance to nuisance factors. Their proposal involves the replacement of any standard convolutional layer in a CNN with a deformable layer which learns to estimate offsets to the standard kernel sampling locations, conditioned on the input. We note four main differences with our work. First, while their method samples from the same low-resolution input as the original CNN architecture, our saliency sampler is designed to sample from any available resolution, allowing it to take advantage of higher resolution data when available. Second, our approach estimates the sample field through saliency maps which have been shown to emerge naturally when training fully convolutional neural networks [15]. We found that estimating local spatial offsets directly, as in a DCN, is much harder. Third, our method can be applied to existing trained networks without modification, while DCNs require changing network configurations by swapping in the deformable convolutions. Finally, our approach produces human readable outputs in the form of the saliency map and the deformed image which allow for easy visual inspection and debugging. We note that our proposed saliency sampler and DCNs are not mutually exclusive: our saliency sampler is designed to sample efficiently across scale space and could potentially make use of deformable convolutional layers to help model local geometric variations. In the same spirit as deformable networks, Li et al. [16] propose an encoder-decoder structure to use non-squared convolutions. As in [13], they predict directly a parametrization of these transformations instead of using a saliency map.

Attending to multiple objects recursively has also been previously explored. Eslami et al. [11] proposed a method to iteratively attend to multiple objects in an image. In the same direction, [12] introduced a method for fine-grained classification which recursively locates an object in a low-resolution image followed by cropping from a high-resolution image. More recently, [17] expanded this idea to multiple attention locations in the image, instead of a single one. Finally, [18] describe a method where multiple crops are proposed and then filtered by a CNN. We note that these methods are designed specifically for classification and are not as general as our proposed sampling layer.

Saliency-Based Methods: CNNs have been shown to naturally direct attention to task-salient regions of the input data. Zhou et al. [15] found that CNNs use a restricted portion of the image to inform their decision in a classification task. They proposed the use of Class Activation Maps (CAM) as a mechanism for localizing objects in images without explicit location feedback during training. Rosendfeld et al. [19] proposed an iterative method to crop the relevant regions of the image for fine-grained classification. They generate a CAM to highlight the regions most used by the network to make the final decision. These regions are used to crop a part of the image and generate a new CAM, which then highlights the regions of the image used by the network to inform the final prediction. As presented in [15], the CAM requires the use of a particular fully convolutional architecture. To overcome this limitation, [20] introduces a gradient-based method to generate CAMs. Their method can be used to understand a wide variety of networks. In our work, we take advantage of the ability of CNNs to naturally localise task-salient regions, by encouraging the network to attend more to those regions.

Adaptive Image Sampling Methods: Another possible approach to our problem is to pre-design certain feature detectors in a multi-scale strategy. This approach is usually taken when solving a particular problem where the features to use are very clear for humans. For instance, to solve the problem of gaze-tracking on a mobile device display, Khosla et al. [21] proposed the iTracker method, a gaze estimation system based on RGB images. Their system uses the image from the device's front-facing camera, and extracts high resolution crops of both eyes and face using separate detectors. Another example along this line was presented by Wang et al. [22], who generate the features of the input image at different scales to then select the best features and produce the final output.

Adaptive image sampling is also used for image retargeting in computer graphics [23]. Unlike in our case where the sampled image only serves as an intermediate representation for solving another problem, the goal of retargeting is to deform an image to fit a new shape and preserve content important for human observer as well as avoid visible deformations. Similarly to our concept, this can be driven by saliency [24] and formulated as an energy minimization [25] or Finite Element Method [26] problem.

3 Saliency Sampler

Let I be a high-resolution image of an arbitrary size and let I_l be a low-resolution image bounded by size $M \times N$ pixels suitable for a task network f_t (Fig. 1). Typically, CNNs rescale the input image I to I_l without exploiting the relative importance of I's pixels. However, if our task requires information from a certain image region more than others, it may be advantageous to sample this region more densely. The saliency sampler executes this by first analyzing I_l before sampling areas of I proportionally to their perceived importance. In doing so, the model can capture some of the benefit of increased resolution without significant additional computational burden or risk of overfitting.

The sampling process can be divided into two stages. In the first stage, a CNN is used to produce a saliency map. This map is task specific, since different tasks may require focus on different image regions. In the second stage, the most important image regions are sampled according to the saliency map.

3.1 Saliency Network

The saliency network f_s produces a saliency map S from the low resolution image: $S = f_s(I_l)$. The choice of network for this stage is flexible and may be changed depending on the task. For all choices of f_s, we apply a softmax operation in the final layer to normalize the output map.

3.2 Sampling Approach

Next, a sampler g takes as input the saliency map S along with the full resolution image I and computes $J = g(I, S)$ – that is, an image with the same dimensions as I_l, that has been sampled from I such that highly weighted areas in S are represented by a larger image extent (see Fig. 3). In this section, we will discuss the possible forms that g can take, and which one is more suitable for CNNs.

In all cases, we compute a mapping between the sampled image and the original image and then use the grid sampler introduced in [13]. This mapping can be written in the standard form as two functions $u(x, y)$ and $v(x, y)$ such that $J(x, y) = I(u(x, y), v(x, y))$.

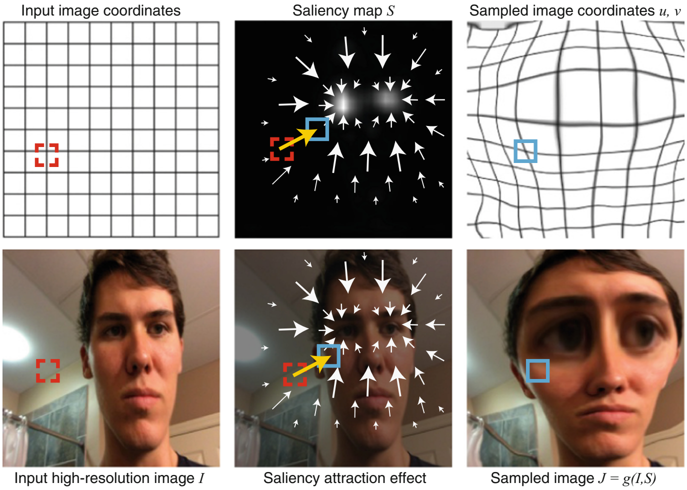
Input image coordinates
Saliency map S
Sampled image coordinates u, v
Input high-resolution image I
Saliency attraction effect
Sampled image $J = g(I,S)$

Fig. 3. Saliency sampler. The saliency map S (center, top) describes saliency as a mass attracting neighboring pixels (arrows). Each pixel (red square) of the output low-resolution image J samples from a location (cyan square) in the input high-resolution image I which is offset by this attraction (yellow arrow) as defined by the Saliency Sampler $g(I, S)$. This distorts the coordinate system of the image and magnifies important regions which get sampled more often than others.

The main goal for the design of u and v is to map pixels proportionally to the normalized weight assigned to them by the saliency map. Assuming that $u(x, y)$, $v(x, y)$, x and y range from 0 to 1, an exact approximation to this problem would be to find u and v such that:

$$\int_0^{u(x,y)} \int_0^{v(x,y)} S(x', y')dx'dy' = xy \tag{1}$$

However, finding u and v is equivalent to finding the change of variables that transforms the distribution set by $S(x, y)$ to a uniform distribution. This problem has been extensively explored and the usual solutions are computationally very costly [27]. For this reason, we need to take an alternative approach that is suitable for use in CNNs.

Our approach is inspired by the idea that each pixel (x', y') is pulling other pixels with a force $S(x', y')$ (see Fig. 3). If we add a distance kernel $k((x, y),(x', y'))$, this can be described as:

$$u(x,y) = \frac{\sum_{x',y'} S(x',y')k((x,y),(x',y'))x'}{\sum_{x',y'} S(x',y')k((x,y),(x',y'))} \tag{2}$$

$$v(x,y) = \frac{\sum_{x',y'} S(x',y')k((x,y),(x',y'))y'}{\sum_{x',y'} S(x',y')k((x,y),(x',y'))} \tag{3}$$

This formulation holds certain desirable properties for our functions u and v, notably:

Sampled Areas: Areas of higher saliency are sampled more densely, since those pixels with higher saliency mass will attract other pixels to them. Note that kernel k can act as a regularizer to avoid corner cases where all the pixels converge to the same value. In all our experiments, we use a Gaussian kernel with σ set to one third of the width of the saliency map, which we found to work well in various settings.

Convolutional Form: This formulation allows us to compute u and v with simple convolutions, which is key for the efficiency of the full system. This layer can be easily added in a standard CNN and preserve differentiability needed for training by backpropagation.

Note that the formulation in Eqs. 2 and 3 has an undesirable bias to sample towards the image center. We avoid this effect by padding the saliency map with its border values.

3.3 Training with the Saliency Sampler

The saliency sampler can be plugged into any convolutional neural network f_t where more informative subsampling of a higher resolution input is desired. Since the module is end-to-end differentiable, we can train the full pipeline with standard optimization techniques. Our complete pipeline consists of four steps (see Fig. 1):

1. We obtain a low resolution version I_l of the image I.
2. This image is used by the saliency network f_s to compute a saliency map $S = f_s(I_l)$, where task-relevant areas of the image are assigned higher weights.
3. We use the deterministic grid sampler g to sample the high resolution image I according to the saliency map, obtaining the resampled image $J = g(I, S)$ which has the same resolution as I_l.
4. The original task network f_t is used to compute our final output $y = f(J)$.

Both f_s and f_t have learnable parameters and so can be trained jointly for a particular task. We found helpful to blur the resampled input image of the task network for some epochs at the beginning of the training procedure. It forces the saliency sampler to zoom deeper into the image in order to further magnify small details otherwise destroyed by the consequent blur. This is beneficial even for the final performance of the model with the blur removed.

4 Experiments

In this section we apply the saliency sampler to two important problems in computer vision: gaze-tracking and fine-grained object recognition. In each case, we examine the benefit of augmenting standard methods on commonly used datasets with our sampling module. We also compare against the closest comparable methods. As an architecture for the saliency network f_s, in all the tasks we use ablations of ResNet-18 [4] pretrained on the ImageNet Dataset [28] and one final 1×1 convolutional layer to reduce the dimensionality of the saliency map S. We found this network to work particularly well for classification and regression problems.

4.1 Gaze Tracking

Gaze tracking systems typically focus, for obvious reasons, on the eyes. Most of the state-of-the-art methods for gaze tracking rely on eye detection, with eye patches provided to the model at the highest possible resolution. However, in this experiment we show how with fewer inputs we are able to achieve similar performance to more complex engineered systems that aim to only solve the gaze-tracking task. We benchmark our model against the iTracker dataset [21], and show how their original model can be simplified by using the saliency sampler.

As the task network f_t we use a standard AlexNet [1] with the final layer changed to regress two outputs and a sigmoid function as the final activation.

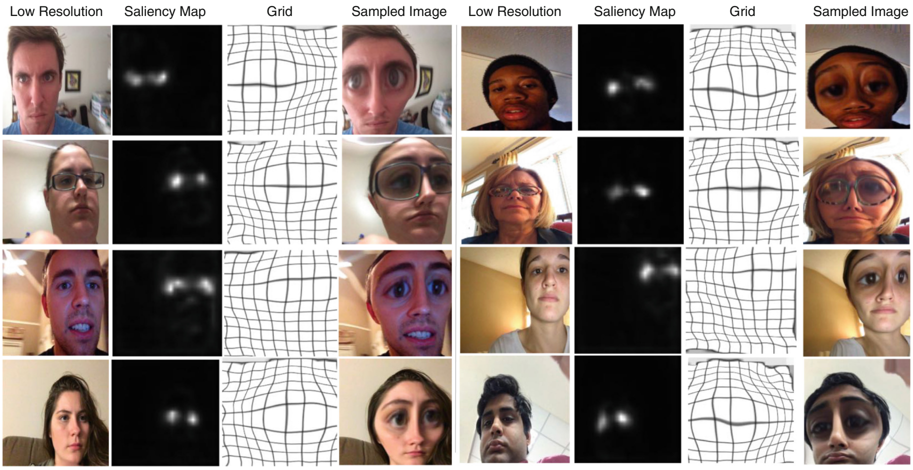

Fig. 4. Visualization of sampler behavior for iTracker gaze-tracking task. We show the low-resolution input image I_l, the saliency map S estimated by f_s, the sampling grid g, and the resampled image J. Note that the saliency network naturally discovers the eyes to be the most informative regions in the image to infer subject gaze, but also learns to preserve the approximate position of the head in the image, which is a further useful cue for estimating gaze position on a mobile device.

Table 1. Performance comparison on GazeCapture dataset. The table reports distance errors in cm for our models and benchmarks on the GazeCapture dataset.

Model	iPad (cm)	iPhone (cm)
iTracker	3.31	2.04
Plain AlexNet (AN)	5.44	2.63
AN + Deformable Convolutions	5.21	2.62
AN + STN TPS	4.44	2.39
AN + STN	4.33	2.25
AN + Grid Estimator	3.91	2.20
AN + Saliency Sampler (ours)	**3.29**	**2.03**

We choose AlexNet in order to be directly comparable to the iTracker system of [21], which is one of the state-of-the-art models for gaze tracking. The model has four inputs: two crops for both eyes, one crop for the face and a coarse encoding for the location of the face in the image. As saliency network f_S we use the initial 10 layers of ResNet-18. We aim to prove that our simple saliency sampler can allow a normal network to deal with the complexity of the four-input iTracker model by just magnifying the correct parts of a single input image.

We compare our model to various competitive baselines. First, we replace the top three convolutional layers of an AlexNet network (pretrained in the ImageNet dataset [1]) with three deformable convolution layers [14] (`Deformable Convolutions`). Second, we test the Spatial Transformer Network baseline [13] with the affine parametrization (`STN`) and TPS parametrization (`STN TPS`). As a localization network, we use a network similar to f_s for fairness. Third, we modify the network f_s to directly estimate the sampling grid functions u and v without the saliency map (`Grid Estimator`). We also compare against the system in [21], which was engineered specifically for the task (`iTracker`). As an error metric, we take the average distance error of the predicted to ground truth gaze location in the screen space of the iPhone/iPad devices on which the dataset was captured.

In Table 1, we present the performance of our model and baselines. Our model achieves a performance similar to iTracker, which enjoys the advantage of four different inputs with 224×224 pixels for each, while our system compresses all the information into a single image of 227×227 pixels. Our approach also improves performance over the Deformable Convolutions, both of the STN variants and the Grid Estimator by a difference ranging from 0.62 to 1.92 cm for iPad and 0.17 cm to 0.59 cm for iPhone. The STNs as well as the Deformable Convolutions have a hard time finding a transformation useful for the task, while the grid estimator is not able to find the functions u and v directly without the aid of a saliency map. The intermediate outputs of our method are shown in Fig. 4.

4.2 Fine-Grained Classification

Fine-grained classification problems pose a very particular challenge: the information to distinguish between two classes is usually hidden in a very small part of the image, sometimes unresolvable at low resolution. In this scenario, the saliency sampler can play an important role: magnify the important parts of the image to preserve as many of their pixels as possible and to help the final decision network. In this experiment, we study the problem using the iNaturalist dataset that contains 5,089 animal species [29]. Our evaluation was performed using the validation set, since the test set is private and reserved for a challenge (Fig. 5).

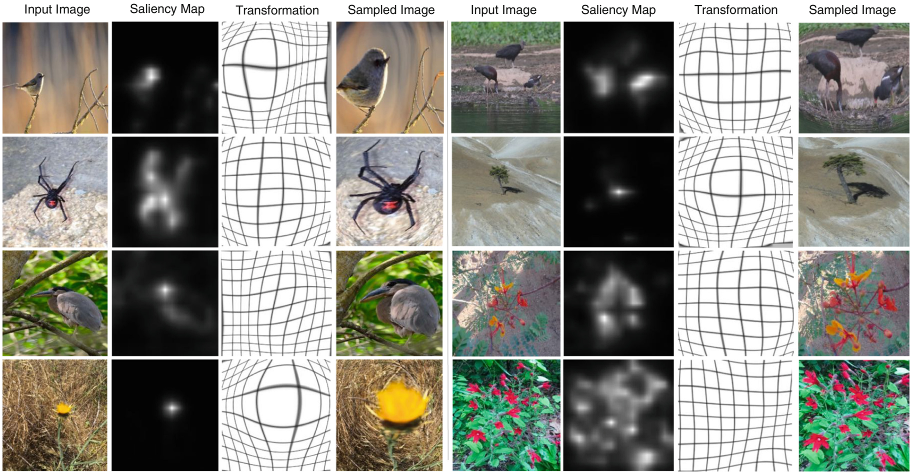

Fig. 5. Visualization of sampler behavior for iNat fine-grained classification task. Similarly to Fig. 4, the saliency network naturally discovers and zooms in on the most informative regions in the image, which tend to correspond to object parts.

In this experiment, we used the ResNet-101 [4] model pretrained on the ImageNet dataset [28] for the task network f_t as it has shown a very good performance in image classification. We used an input resolution of 227×227 for both the task and saliency networks, f_t and f_s. As saliency network f_S we use the initial 14 layers of ResNet-18, although the performance for other saliency networks can be found in Table 3.

As baselines for this task, we used the same methods as before, again with ResNet-101 as the base model. For the deformable convolutional network, we made the network modifications according to instructions in the original paper [14]. We also tested both the affine and the TPS version of STN (`STN Affine` and `STN TPS`) along with the direct grid estimator. Identically to our method, these baselines were allowed access to the original 800×800 pixel images in training time. In test time, the method had access to a center crop

Table 2. iNaturalist fine-grained classification results: top-1 and top-5 accuracy comparison on the validation set of the iNaturalist Challenge 2017 dataset.

Model	Top-1 (%)	[diff]	Top-5 (%)	[diff]
ResNet-101 227 (RN)	60	[-]	83	[-]
RN + Deformable Convolutions	44	[-16]	69	[-14]
RN + STN Affine	60	[0]	83	[0]
RN + Grid Estimator	61	[1]	83	[0]
RN + STN TPS	62	[2]	84	[1]
RN + CAM [15]	62	[2]	84	[1]
RN + Saliency Sampler (ours)	**65**	**[5]**	**86**	**[3]**

of 512 × 512 pixels. The localization networks were similar to f_s for fairness. To test whether a high-resolution input alone could improve the performance of the baseline Resnet-101 network, we also provided crops from the maximum activated regions for the ResNet-101 227 network, using the Class Activation Map method of [15] (`CAM`). We selected the class with the largest maximal activation, and computed the bounding box as in the original paper. We then cropped this region from the original input image and rescaled it to 227 × 227 resolution. These crops were used as inputs for the ResNet-101 227 × 227 network for the final classification.

Table 2 shows the classification accuracy for the various models compared. Our model is able to significantly outperform the ResNet-101 baseline by 5% and 3% for top-1 and top-5 accuracies respectively. The performance of the CAM-based method is closer to our method which is expected as it benefits from the same idea of emphasizing image details. However, our method still performs several points better, perhaps because of its greater flexibility to focus on local image regions non-uniformly and selectively zoom-in on certain features more than others. It also has a major benefit of being able to zoom in on an arbitrary number of non-collocated image locations, whereas doing so with crops involves determining the number of crops beforehand or having a proposal mechanism.

The spatial transformers and the grid estimator perform similar or a bit better than the ResNet-101 baseline. Like our method, those methods benefit from the ability to focus attention on a particular region of the image. However, the affine version of the spatial transformers applies a uniform deformation across the whole image, which may not be particularly well suited to the task, while the more flexible TPS version and the grid estimator, which in theory could more closely mimic the sampling introduced by our method, were found to be harder to optimize and were consistently found to perform worse.

Finally, the deformable convolutions method performs significantly worse than the ResNet-101 architecture. Despite our best efforts, we were not able to make the model converge to a competitive performance. This could be attributed to the difficulty of training the complex parametrization utilized in its design, which usually end up in a local minima. In contrast, our method benefits from

Table 3. Saliency network ablation: we measure the effect of different depths of saliency network f_s on the iNaturalist fine-grained classification task.

	None (no f_s)	6-layer	10-layer	14-layer
Top-1 (%)	60	62	64	**65**
Top-5 (%)	83	84	85	**86**

the fact that neural networks have a natural ability to predict salient image elements [30] and thus the optimization may be significantly easier.

To justify our claim that the saliency sampler can benefit different task network architectures, we repeat our experiment using a Inception V3 architecture [31]. The original performance is already very high (64% and 86% for top-1 and top-5 respectively) as it uses higher resolution (299) and a deeper network, but our sampler still results in a performance of 66% in top-1 and 87% in top-5.

Saliency Network Importance: In Table 3, we retrained ResNet-101 with different depths of saliency network f_s. We used different ablations of ResNet-18 with 6, 10 or 14 layers (which corresponds to adding one block at a time to build ResNet-18) for the experiment. The performance of the overall network increases with the complexity of the saliency model but with diminishing returns.

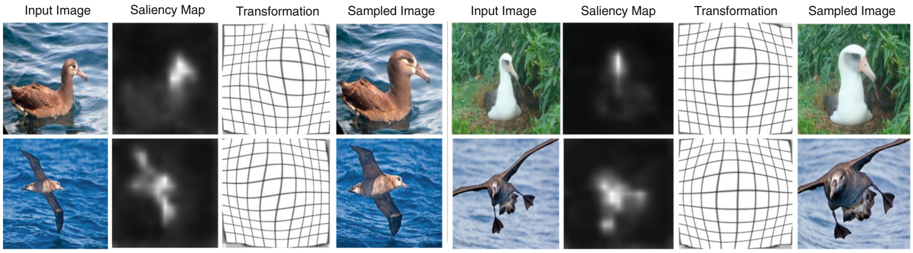

Fig. 6. Visualization of sampler behavior for the CUB-200 dataset: We show the sampled images for a ResNet-50 trained with the saliency sampler in the CUB-200 dataset. The saliency amplifies relevant image regions such as the bird's head.

4.3 CUB-200

To further prove that our model is useful across different datasets, we evaluated it in the CUB-200 dataset [32] (Table 4). Although the CUB-200 is also a fine-grained recognition dataset, it is significantly smaller and the images are better framed around subjects than in the iNaturalist dataset (see Fig. 6).

We used ResNet-50 as our task network and the initial 14 layers of ResNet-18 as our saliency network. By adding our sampling layer we achieve a 2.9% accuracy boost, which is less than the boost in iNaturalist, perhaps because

Table 4. Performance improvements from the addition of our sampling layer on the CUB-200 dataset [32]. **Res. (px)** refers to the input image resolution to the model.

	RN-50	RN-50+SS	DT-RAM	DT-RAM
Res. (px)	227	**227**	224	448
Top-1(%)	81.6	**84.5**	82.8	86.0

objects of interest are more tightly cropped in CUB-200. Compared to DT-RAM [33], one of the top performing models in CUB-200, our approach outperforms the comparable 224 × 224 version of RN-50 DT-RAM by 1.7%, using a simpler model. Our method is not as accurate as the 448 × 448 resolution version of DT-RAM, but the latter uses approximately 2 passes through a RN-50 on average and a larger input size leading to a higher computational cost.

5 Discussion

Adding our saliency sampler is most beneficial for image tasks where the important features are small and sparse, or appear across multiple image scales. The deformation introduced in the vicinity of the magnified regions could potentially discourage the network from strong deformations if another point of interest would be affected. This could be harmful for tasks such as text recognition. In practice, we observed that the learning process is able to deal with such situations well as it was capable of magnifying both collocated eyes without hindering the gaze prediction performance. That is particularly interesting as this task requires preservation of geometric information in the image. The method proved to be easier to train than other approaches which modify spatial sampling, such as Spatial Transformer Networks [13] or Deformable Convolutional Networks [14]. These methods ofter performed closer to the baseline as they failed to find suitable parameters for their sampling strategy. The non-uniform approach to the magnification introduced by our saliency map also enables variability of zoom over the spatial domain. This together with the end-to-end optimization results in a performance benefit over uniformly magnified area-of-interest crops as observed in our fine-grained classification task. Unlike in the case of the iTracker [21], we do not require prior knowledge about the relevant image features in the task.

6 Conclusion

We have presented the saliency sampler – a novel layer for CNNs that can adapt the image sampling strategy to improve task performance while preserving memory allocation and computational efficiency for a given image processing task. We have shown our technique's effectiveness in locating and focusing on image features important for the tasks of gaze tracking and fine-grained object recognition. The method is simple to integrate into existing models and can be efficiently trained in an end-to-end fashion. Unlike some of the other image transformation techniques, our method is not restricted to a predefined number or size of important regions and it can redistribute sampling density across the entire image domain. At the same time, the parametrization of our technique by a single scalar attention map makes it robust against irrecoverable image degradation due to fold-overs or singularities. This leads to a superior performance in problems that require the recovery of small image features such as eyes or subtle differences between related animal species.

Acknowledgment. This research was funded by Toyota Research Institute. We acknowledge NVIDIA Corporation for hardware donations.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Conference on Neural Information Processing Systems (2012)
2. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
5. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) **115**(3), 211–252 (2015)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV) **60**(2), 91–110 (2004)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing System, pp. 91–99 (2015)
8. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
9. Mnih, V., Heess, N., Graves, A.: Recurrent models of visual attention. In: Advances in Neural Information Processing System, pp. 2204–2212 (2014)
10. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755 (2014)
11. Eslami, S.A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Hinton, G.E., et al.: Attend, infer, repeat: fast scene understanding with generative models. In: Advances in Neural Information Processing Systems, pp. 3225–3233 (2016)
12. Fu, J., Zheng, H., Mei, T.: Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition. In: Conference on Computer Vision and Pattern Recognition (2017)
13. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in Neural Information Processing System, pp. 2017–2025 (2015)
14. Dai, J., et al.: Deformable convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 764–773 (2017)
15. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929. IEEE (2016)

16. Li, J., Chen, Y., Cai, L., Davidson, I., Ji, S.: Dense transformer networks, May 2017. arXiv:1705.08881 [cs, stat] arXiv: 1705.08881
17. Zheng, H., Fu, J., Mei, T., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: IEEE International Conference on Computer Vision (ICCV) (2017)
18. Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., Zhang, Z.: The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 842–850. IEEE (2015)
19. Rosenfeld, A., Ullman, S.: Visual concept recognition and localization via iterative introspection. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (eds.) ACCV 2016. LNCS, vol. 10115, pp. 264–279. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54193-8_17
20. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 618–626 (2017)
21. Khosla, A., et al.: Eye tracking for everyone. In: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, June 2016
22. Wang, S., Luo, L., Zhang, N., Li, J.: AutoScaler: scale-attention networks for visual correspondence. In: British Machine Vision Conference (BMVC) (2017)
23. Rubinstein, M., Gutierrez, D., Sorkine, O., Shamir, A.: A comparative study of image retargeting. In: ACM Transactions on Graphics (TOG), vol. 29, p. 160. ACM (2010)
24. Wolf, L., Guttmann, M., Cohen-Or, D.: Non-homogeneous content-driven video-retargeting. In: IEEE International Conference on Computer Vision (ICCV), pp. 1–6. IEEE (2007)
25. Karni, Z., Freedman, D., Gotsman, C.: Energy-based image deformation. In: Computer Graphics Forum, vol. 28, pp. 1257–1268. Wiley Online Library (2009)
26. Kaufmann, P., Wang, O., Sorkine-Hornung, A., Sorkine-Hornung, O., Smolic, A., Gross, M.: Finite element image warping. In: Computer Graphics Forum, vol. 32, pp. 31–39. Wiley Online Library (2013)
27. Chen, R., Freedman, D., Karni, Z., Gotsman, C., Liu, L.: Content-aware image resizing by quadratic programming. In: Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1–8. IEEE (2010)
28. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
29. Van Horn, G., et al.: The inaturalist challenge 2017 dataset. arXiv preprint arXiv:1707.06642 (2017)
30. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
32. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-UCSD birds-200-2011 dataset (2011)
33. Li, Z., Yang, Y., Liu, X., Zhou, F., Wen, S., Xu, W.: Dynamic computational time for visual attention. arXiv preprint arXiv:1703.10332 (2017)