A Hybrid Probabilistic Model for Camera Relocalization

Ming Cai
ming.cai@adelaide.edu.au
Chunhua Shen
chunhua.shen@adelaide.edu.au
Ian Reid
ian.reid@adelaide.edu.au

School of Computer Science
The University of Adelaide
Adelaide, Australia

Abstract

We present a hybrid deep learning method for modelling the uncertainty of camera relocalization from a single RGB image. The proposed system leverages the discriminative deep image representation from a convolutional neural networks, and uses Gaussian Process regressors to generate the probability distribution of the six degree of freedom (6DoF) camera pose in an end-to-end fashion. This results in a network that can generate uncertainties over its inferences with no need to sample many times. Furthermore we show that our objective based on KL divergence reduces the dependence on the choice of hyperparameters. The results show that compared to the state-of-the-art Bayesian camera relocalization method, our model produces comparable localization uncertainty and improves the system efficiency significantly, without loss of accuracy.

1 Introduction

It is now well-established that Convolutional Neural Networks (CNNs) are the method of choice for extracting a variety of high-level knowledge from images, including classification, recognition and even regression tasks such as visual geometry (e.g, camera pose). However a significant disadvantage has remained that it is not straightforward – or even well understood – how to model the uncertainty of CNN outputs. Exploiting uncertainty in deep neural networks is therefore an eye-catching topic in the Machine Learning community, because the probability distribution of the prediction from a deep perception system can be used in a variety of ways. Most notably for our purposes the distribution over a regression result can be interpreted as its uncertainty, making the result amenable for use within the standard data fusion algorithms such as a Kalman Filter. For example in Simultaneous Localization and Mapping (SLAM) systems, the uncertainty of the pose estimate can be naturally fused with a state estimate [6] to improve the accuracy of localization over time.

In this paper, we aim to model the uncertainty of the deep structure that regresses the camera pose directly from a single RGB image. Closely related work has been done in Bayesian PoseNet [10], the probabilistic version of a learning-based camera relocalization method, PoseNet [12]. It uses CNN to extract features from an RGB image, and then performs linear regression to estimate the 6DoF pose of a moving camera. Thanks to the dropout

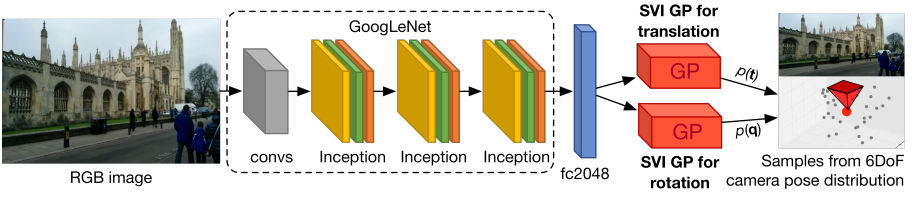


Figure 1: **The pipeline of the GPoseNet.** It takes a monocular RGB image as input. The high-level feature from *fc2048* layer of the CNN base is fed to two SVI GPs to perform probabilistic inference for translation and rotation. Our system outputs a distribution for camera relocalization. The red dot and pyramid indicate the point estimate of the 6DoF pose.

layers – i.e, different stochastic connections between neurons – before the regressors, multiple samples of camera pose are obtained during inference. Then the distribution of the 6DoF pose can be empirically summarized by these samples. The mathematics behind this approximation has been well studied in [8]. However, this distribution-from-samples method is not very resource-friendly, requiring a separate forward inference per sample.

Gaussian Process Regression (GPR) [18] is a probabilistic model that inherently provides a tractable predictive distribution for the output. But when one applies GPR to real world applications that involve large scale datasets, the low efficiency prevents GPR from being plug-and-play. This arises from four factors. First, the matrix inverse for computing precision matrices has complexity $O(n^3)$, which is prohibitive because n is the size of the dataset and usually a large number. Second, even with complexity reduction, the training of sparse GPR still involves all training samples in each optimization step. Third, the high-dimensional image data is not a straightforward input for GPR's kernel function. Forth, GPR is less commonly used for multivariate outputs. A solution to the first two factors is provided by Stochastic Variational Inference (SVI) [9], which treats the mean and covariance of a lower dimensional variational posterior as the global parameters, turning the variational GPs into a "parametric" model. As for the last two factors, the CNN can be an excellent feature extractor and Coregionalization kernel [2] enables GPR being suitable for vector-valued functions.

With these motivations, our main contribution is to show how to combine the CNN and GPR naturally, proposing a probabilistic framework to model the uncertainty in the regression of 6DoF camera pose based on an RGB image, while overcoming the complexity issues of naive GPR. We exploit the CNN to extract discriminative features and use the GPR to perform probabilistic inference. We show that the mean of our predictive pose distribution has the comparable accuracy to the state-of-the-art pure RGB based method for camera localization, and meanwhile the covariance is compatible with the uncertainty from Bayesian PoseNet, but with significant computational resource saving.

To achieve this combination we build an objective function for the whole framework that aims to minimize two Kullback-Leibler (KL) divergences between distributions. In the original PoseNet, the importance between rotational and translational penalization are balanced by grid-search over the network hyperparameters. Later, to avoid hyperparameters tuning, [11] embedded the translation and rotation into a single photometric loss via geometrical transformation and affine projection. However this is not a universal solution for some other multi-task CNNs that lack of underlying connections between tasks. Our use of

the KL-divergences not only results in a loss function that permits network optimisation in an end-to-end fashion, but furthermore, as shown in Section 4, it leads to greatly improved robustness to the choice of hyperparameters, obviating the need for expensive grid search during training.

The contribution of this paper includes: (a) We build a hybrid model of two main machine learning frameworks, CNN and GPR, and design an unified objective function that drives the model to be trained in an end-to-end fashion. (b) We provide the probability distribution for the 6DoF camera pose with one-time inference, saving time and resource compared to Bayesian PoseNet. (c) This work weakens the effect of hyperparameter selection for multi-task regression by using objectives from the variational sparse GPR. To the best of our knowledge, this is the first work that combines the CNN and GPR to perform probabilistic inference for large scale computer vision task.

2 Related work

There are mainly two groups of method for camera relocalization: geometry-based and appearance-based. The former are predicated on matching (typically hand-crafted) features such as SIFT [19] and ORB [16]) between a query image and the pre-built map database. Having established correspondences, the problem of 6DoF camera pose estimation is solved using the PnP algorithm. Appearance-based methods aim to find the 3D scene coordinates for all pixels in the image, and typically regress directly from an image to a 6D pose, without the extraction of local salience features. Examples include [3, 15, 20].

The first work that applies CNNs to regress from a monocular RGB image to 6DoF camera pose is PoseNet by Kendall *et al*. [12]. The network is a modified GoogLeNet [22] with an additional fully-connected (fc) layer (2048 units) before the two affine regressors. They use the Euclidean loss for both translational vector and rotational quaternion to supervise the learning process. The features from the penultimate layer are a high-level representation of the whole image, providing robustness to the light, weather and other dynamic changes in the mainly unchanged scene.

Several work improve the localization accuracy via different techniques, e.g. [24] uses LSTM units over the features from CNN to enhance it to be more correlated with the relocalization task, and [11] transfers the translation and rotation into a unified space, using photometric loss to supervise the training. Since in [11], the geometry (3D points of the scene) are brought in and the quantity of information for training is considerably increased, we categorize it to the geometry-based methods.

The dropout layers in PoseNet not only play an important role to prevent over-fitting, but also provide an alternative interpretation for CNNs with dropout as a Bayesian model approximation. In [8], Gal and Ghahramani prove that the widely-used dropout technique can be mathematically viewed as an approximation to the posterior of the deep Gaussian Process [5]. By running same test point through the model multiple times with the existence of dropout (which is often deactivated in most deterministic CNN models), the different connections between neurons lead to a set of Monte Carlo samples from the approximated variational posterior over the output. The Bayesian PoseNet [10] is a well-demonstrated application of dropout bayesian approximation. Without changing the training pipeline of PoseNet, it brings the camera relocalization to a probabilistic level. Whilst it improves the accuracy of the pose estimation, the uncertainty can be also obtained empirically with grounded theoretical support from [8].

Beyond the success of two versions of PoseNet, two problems can be further discussed. The first one is the computational efficiency of Bayesian PoseNet. To generate a accurate posteriors for camera pose, one often needs many prediction samples, which results in great computational cost. The second concerns the network hyperparameter β. It is responsible for balancing the weights between losses for translation and for rotation (for details, please refer to Fig. 2 in the PoseNet paper). Experimentally we found that it is heavily scene-dependent, and the optimal setting takes much effort to find. To this end, we propose to use the combination of CNN and GPR to improve the efficiency and eliminate hyperparameter tuning. A comprehensive explanation is given in section 4.

The GPR [18] is a fully Bayesian non-parametric model that elegantly estimates the posterior distribution of the target function. However, the exact posterior requires $O(n^3)$ to compute, where n is the size of training set. It hinders the efficiency of GPR. A group of sparse GPs that aim to reduce the complexity of the full GPs are well-researched, such as DTC [4], FITC [21], PTIC [17], VEF [23] etc. These methods downscale the complexity to $O(nm^2)$, where m is the number of induced pseudo training points. Nevertheless the joint training of pseudo points and kernel parameters still requires to take all training samples into consideration, which is intractable for big datasets. To solve this, Stochastic Variational Inference (SVI) [9] treats the mean and covariance of the variational distribution as global parameters, making the variational GPs a "parametric" model. The model then can be trained via the Stochastic Gradient Descend (SGD) based on batch data. We make use of this method to enable the connection between CNN and GPR in our framework.

3 The Bayesian knowledge revisited

Road to SVI for GPs. We start with introducing GPR for 1-d functions, bringing in the notation used in this paper. Let $\{\mathbf{x}_i, y_i\}_{i=1}^n$ denote the whole dataset. $\mathbf{x}_i \in \mathbf{X}$ is a sample from all points $\mathbf{X}$ in the feature domain $\mathbb{R}^F$, and $y_i \in \mathbf{y}$ is the observation of a function f at point $\mathbf{x}_i$ with independent Gaussian noise σ^2. GPR assumes a joint smooth Gaussian prior over the function space. The covariance of the prior, $\mathbf{K}$, is defined by the kernel function k, which will be discussed later. Given a test point $\mathbf{x}^\star$, the conditional posterior of the test function $f^\star$ can be inferred via multivariate Gaussian theorem [18]:

$$p(f^\star|\mathbf{y}) = \mathcal{N}\left(f^\star|\mathbf{K}_{\star n}(\mathbf{K}_{nn}+\sigma^2\mathbf{I})^{-1}\mathbf{y}, K_{\star\star}-\mathbf{K}_{\star n}(\mathbf{K}_{nn}+\sigma^2\mathbf{I})^{-1}\mathbf{K}_{n\star}\right). \tag{1}$$

To avoid the complexity of $(\mathbf{K}_{nn}+\sigma^2\mathbf{I})^{-1}$ ($O(n^3)$), variational methods [23] are proposed by Titsias to find a smaller set of points – called *inducing points* ($\mathbf{Z} \in \mathbb{R}^{m\times F}$) – to approximate the whole set of the training samples. Denote the corresponding inducing variables as $\mathbf{u}$. By approximating the exact posterior $p(\mathbf{u}|\mathbf{y})$ with a variational distribution $q(\mathbf{u})$, the optimal inducing points $\hat{\mathbf{Z}}$ can be found via maximizing the variational lower bound (ELBO) of the log of marginal likelihood $p(\mathbf{y})$, which is given as

$$\log p(\mathbf{y}) \geq \mathcal{L}(q,\mathbf{Z}) = \int q(\mathbf{u})\left\{\mathbb{E}_{<p(\mathbf{f}|\mathbf{u})>}(\log p(\mathbf{y}|\mathbf{f})) + \log\frac{p(\mathbf{u})}{q(\mathbf{u})}\right\}d\mathbf{u}, \tag{2}$$

In [23], Titsias proves the conclusion that the final formulation of the ELBO to $\log p(\mathbf{y})$ is

$$\mathcal{L}(\mathbf{Z}) = \log\mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}+\sigma^2\mathbf{I}\right) - \frac{1}{2\sigma^2}Tr\left(\mathbf{K}_{nn}-\mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}\right), \tag{3}$$

and the optimal variational posterior $q^*(\mathbf{u})$ is with mean $\mu = \sigma^{-2}\mathbf{K}_{mm}\Sigma^{-1}\mathbf{K}_{mn}\mathbf{y}$ and covariance $\Lambda = \mathbf{K}_{mm}\Sigma^{-1}\mathbf{K}_{mm}$, where $\Sigma = \mathbf{K}_{mm} + \sigma^{-2}\mathbf{K}_{mn}\mathbf{K}_{nm}$. The complexity is now $O(nm^2)$.

Note that when computing $\mathcal{L}(\mathbf{Z})$ and $q^*(\mathbf{u})$ during optimization, the existence of $\mathbf{K}_{nm}$ (or $\mathbf{K}_{mn}$) and $\mathbf{y}$ makes the algorithm need to use all training samples. This is disadvantageous for tasks with large datasets. In SVI for GPs [9], Hensman *et al*. propose to use a parametric variational Gaussian posterior for $\mathbf{u}$, such that the mean and covariance of $q_g(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m},\mathbf{S})$ act as the *global* variational parameters across all training samples. This enables the joint training of $\mathbf{m}$, $\mathbf{S}$ and $\mathbf{Z}$ via batch data to perform SGD. The lower bound of the log marginal likelihood then changes from (2) to

$$\begin{aligned}\mathcal{L}_{svi}(\mathbf{m},\mathbf{S},\mathbf{Z}) &= \int q_g(\mathbf{u})\left\{\mathbb{E}_{<p(\mathbf{f}'|\mathbf{u})>}\left(\log p\left(\mathbf{y}'|\mathbf{f}'\right)\right) + \log\frac{p(\mathbf{u})}{q_g(\mathbf{u})}\right\}d\mathbf{u} \\ &= \log\mathcal{N}(\mathbf{y}'|\mathbf{K}_{n'm}\mathbf{K}_{mm}^{-1}\mathbf{m},\sigma^2\mathbf{I}) - \frac{1}{2\sigma^2}Tr\left(\mathbf{K}_{n'n'} - \mathbf{K}_{n'm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn'}\right) \\ &\quad - \frac{1}{2}Tr\left(\mathbf{S}\Lambda'^{-1}\right) - \mathrm{KL}(q_g(\mathbf{u})||p(\mathbf{u}))\end{aligned} \tag{4}$$

where $\mathrm{KL}(q_g(\mathbf{u})||p(\mathbf{u}))$ is the KL divergence between the variational posterior and the exact prior $p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0},\mathbf{K}_{mm})$. Note that in equation (4), $(\cdot)'$ means that it is from/for the batch data. We use this lower bound as the objective of our GP regressors.

Coregionalization kernel. Matrix $\mathbf{K}$ in previous section is the covariance of the function values in the Gaussian prior. It is built from the kernel function $k(\mathbf{x}_i,\mathbf{x}_j)$ that describes the similarity between two points. The most commonly used kernel in GPR is the Radial Basis Function (RBF) kernel $k_{rbf} = \sigma_k^2\exp(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2l^2})$. For 1-d GPR, the output of k_{rbf} is a scalar and the kernel matrix for the whole training points $\mathbf{K}_{nn} = K(\mathbf{X},\mathbf{X})$ is a $n\times n$ Positive Semi-definite (PSD) matrix. However in our task, both of the translation and quaternion are vector-valued functions, and have correlation between their entries. The scalar does not satisfy the need of storing this interrelationship. [2] reviews the *cokriging* from geostatistics, and formally introduces the kernel for the vector-valued function GPs, the Coregionalization kernel. The key idea of coregionalization is to have a PSD $\mathbf{B}\in\mathbb{R}^{D\times D}$ act as a learnable parameter to represent the correlation between these functions, where D is the dimension of the output. To ensure the PSD property, a coregionalization kernel is built by $\mathbf{B} = \mathbf{W}\mathbf{W}^T + \mathrm{diag}(\kappa)$, where $\mathbf{W}\in\mathbb{R}^{D\times R}$ and $\kappa\in\mathbb{R}^D$. R is the rank of $\mathbf{B}$ (specified by the user at algorithm design time). For example, the full kernel matrix for inducing points $\mathbf{Z}$ in SVI is given as $\mathbf{K}_{mm}^c = K^c(\mathbf{Z},\mathbf{Z}) = \mathbf{B}\otimes\mathbf{K}_{mm}$, where $\otimes$ is the Kronecker product of matrices. In the following, the superscript c of the coreg-augmented kernel is omitted for brevity.

4 Modelling uncertainty for camera relocalization

Problem formulation. Given an RGB image $I_i\in\mathcal{I}$, our goal is to build a probabilistic model to predict the multivariate distribution of the translational vector $\mathbf{t}\in\mathbb{R}^3$ and rotational quaternion $\mathbf{q}\in\mathbb{R}^4$. Denote the CNN feature extractor as $N(I_i,\theta_N)$. It takes RGB image I_i as input, and has learnable parameters θ_N. To model the uncertainty of prediction, we assume two independent Gaussian priors for $\{\mathbf{t}_i\}_{i=1}^n$ and $\{\mathbf{q}_i\}_{i=1}^n$, and consider the output from $N(I_i,\theta_N)$ as the shared input features for both of the SVI GP regressors. Based on the Bayesian knowledge in the previous section, the predictive distribution for translation

component is

$$\begin{aligned} p(\mathbf{t}^*) &= \int p(\mathbf{t}^*|\mathbf{u}) q_g(\mathbf{u}) d\mathbf{u} \\ &= \mathcal{N}(\mathbf{t}^*|\mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{m}_t, \mathbf{K}_{**} - \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*} + \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{S}_t^{-1}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*}). \end{aligned} \tag{5}$$

Note in equation (5), $\mathbf{K}_{**} = K(N(I^*, \theta_N), N(I^*, \theta_N))$ is the kernel matrix built on the features from the CNN base. Since the priors and likelihoods are all Gaussians, the predictive distribution for translation and quaternion are two multi-variate Gaussians. This distribution has learnable parameters $\mathbf{Z}_t$, $\mathbf{m}_t$, $\mathbf{S}_t$ and parameters for the kernel function θ_k. Similar results for rotation can be obtained by replacing the subscript t with q.

Architecture. To make a fair comparison between linear regressors in [10][12] and the SVI GP regressors in our framework, we use the same deep feature encoder as PoseNet. We remove the last two fc layers of PoseNet and replace them with two SVI GP regressors. They take the output from the previous fc2048 layer as input, and perform Bayesian inference to obtain the distribution of translation and rotation for camera pose. The "loss function" of this framework is the sum of the objectives of two GPs, which will be further addressed in the next paragraph. We use the coregionalization kernel in our system. Since the main goal of this work is to compare the capability of this hybrid architecture and the pure CNN structure, we did not tune the type of kernel function to seek for better performance. We use RBF as the base kernel function for simplicity, and leave the selection of kernel function to future work. Fig. 1 illustrates the structure of our framework.

Objective function. To train the whole structure in an end-to-end fashion, we design a multi-component objective function that combines CNN loss and the ELBOs of two log marginal likelihoods as follow

$$L = \beta_{g_t}\mathcal{L}_{svi}(\mathbf{m}_t, \mathbf{S}_t, \mathbf{Z}_t) + \beta_{g_q}\mathcal{L}_{svi}(\mathbf{m}_q, \mathbf{S}_q, \mathbf{Z}_q) + \beta_{n_t}\left\|\hat{\mathbf{t}} - \mathbf{t}\right\|_2 + \beta_{n_q}\left\|\hat{\mathbf{q}} - \mathbf{q}\right\|_2. \tag{6}$$

Hyperparameters. The main issue of the multi-task CNNs is that the norm-based losses for these targets are not always at the same scale, therefore they need different weights to penalize. In the proposed objective function (6), however, the first two components are the objectives from the SVI GPs. Maximizing the ELBO of $\log p(\mathbf{y})$ is mathematically equivalent to minimizing the KL divergence between the exact posterior $p(\mathbf{f}|\mathbf{y})$ and the variational distribution $q(\mathbf{f})$ [23]. We observed that this measure between two distributions can reduce the dependence on the choice of hyperparameters.

For a clear understanding of this advantage, we replace these two multivariate normal distributions with two univariate Gaussians over a same random variable, $p_1(x) = \mathcal{N}(x|\mu_1, \sigma_1^2)$ and $p_2(x) = \mathcal{N}(x|\mu_2, \sigma_2^2)$. The KL divergence between them is $\mathrm{KL}(p_1(x)||p_2(x)) = \log\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$. It is straightforward to see that $(\mu_1 - \mu_2)^2$ and σ_2^2 have the same scale. The scale is canceled out in this equation. It means that the unit of the KL divergence is always 1, hence the choice of the hyperparameters β_{g_t} and β_{g_q} becomes easier. In the following experiments, we always keep them equal.

The last two components of equation (6) are the losses from shallower depth of the CNN base. They supervise the learning of the low-level and middle-level features from the RGB image (See [12]). We keep these two weights (β_{n_t} and β_{n_q}) same as the PoseNet.

Scene	Spatial Extent(m)	Geometry-based: Active Search (SIFT) [15]	Geometry-based: Geometric loss PoseNet[11]	Geometry-based: Bayesian PoseNet[10]	Pure-RGB-based: PoseNet Spatial LSTM[24]	Pure-RGB-based: GPoseNet (Ours)
King's College	140×40	0.42m, 0.55°	0.88m, 1.04°	1.74m, 4.06°	0.99m, 3.65°	1.61m, 2.29°
Old Hospital	50×40	0.44m, 1.01°	3.20m, 3.29°	2.57m, 5.14°	1.51m, 4.29°	2.62m, 3.89°
Shop Facade	35×25	0.12m, 0.40°	0.88m, 3.78°	1.25m, 7.54°	1.18m, 7.44°	1.14m, 5.73°
St Mary's Church	80×60	0.19m, 0.54°	1.57m, 3.32°	2.11m, 8.38°	1.52m, 6.68°	2.93m, 6.46°
Chess	3×2×1	0.04m, 1.96°	0.13m, 4.48°	0.37m, 7.42°	0.24m, 5.77°	0.20m, 7.11°
Fire	2.5×1×1	0.03m, 1.53°	0.27m, 11.3°	0.43m, 13.7°	0.34m, 11.9°	0.38m, 12.3°
Heads	2×0.5×1	0.02m, 1.45°	0.17m, 13.0°	0.31m, 12.0°	0.21m, 13.7°	0.21m, 13.8°
Office	2.5×2×1.5	0.09m, 3.61°	0.19m, 5.55°	0.48m, 8.04°	0.30m, 8.08°	0.28m, 8.83°
Pumpkin	2.5×2×1	0.08m, 3.10°	0.26m, 4.75°	0.61m, 7.08°	0.33m, 7.00°	0.37m, 6.94°
Red Kitchen	4×3×1.5	0.07m, 3.37°	0.23m, 5.35°	0.58m, 7.54°	0.37m, 8.83°	0.35m, 8.15°
Stairs	2.5×2×1.5	0.03m, 2.22°	0.35m, 12.4°	0.48m, 13.1°	0.40m, 13.7°	0.37m, 12.5°

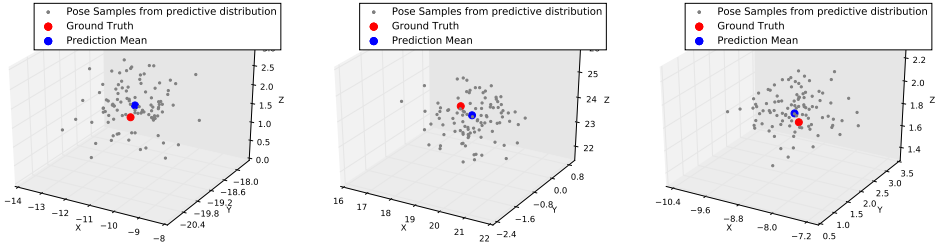
Table 1: **Median error of localization for *Cambridge Landmarks* and *7 Scenes* datasets**. We compare our method (GPoseNet) with the Spatial LSTM-PosNet [24], Bayesian PoseNet [10]. For *Cambridge Landmarks* dataset and *7 Scenes* datasets, the median pose error of our method is averagely (2.0*m*, 4.6°) and (0.3*m*, 9.9°). The overall results surpass Bayesian PoseNet and are comparable with Spatial LSTM-PoseNet [24], for which the average of median error for these two datasets are (1.3m, 5.5°) and (0.3m, 9.9°) respectively.

5 Experiments and results

To benchmark our model both on outdoor and indoor scenarios, we use two datasets for training and evaluation, the *Cambridge Landmarks* [12] and the *7 Scenes* [20] dataset. We follow the same traninging/test split in PoseNet. All the experiments are done on a NVIDIA GeForce GTX 1070 GPU. The batch size in training is 75. We optimize all the models in an end-to-end fashion with ADAM [13]. We also initialize the CNN base with pre-trained weights from ImageNet [7], suggested by [12] and [10]. The number of inducing points for SVI GPs is 10% of the image number in each training split. This number various w.r.t different scenes, from to 23 to 149 in the *Cambridge Landmarks* dataset. We initialize the inducing points $\mathbf{Z}$ with the results from k-means clustering over the features from 500 images. These images are randomly selected from training set. This initialization keeps the induced feature points and the deep features of the training images in the same domain, preventing the large – and meaningless – elements in the kernel matrix, which could rasie if $\mathbf{Z}$ is with random initialization around zero. Experiments show that this initialization also ensures a stable convergence. The learning rate is 10^{-4} for CNN base and 10^{-2} for GPs' parameters. We implement the CNN base with Tensorflow [1] and the GPs part with GPflow [14].

We evaluate our method from two perspectives, localization accuracy and predictive uncertainty. Overall, the results from the following experiments shows that our method can achieve comparable accuracy with the state-of-the-art pure RGB based method, Spatial LSTM PoseNet [24], and estimates the uncertainty in a more efficient way comparing to Bayesian PoseNet [10].

Localization accuracy. We use the mean of the translational and rotational predictive distributions as the point estimation for the 6DoF camera pose. The rotational vector is normalized to ensure that it is an unit vector. In Table 1, we compare the median error of localization in different scenes with the state-of-the-art methods. We can see that with the same CNN encoder, our SVI GP regressors outperform PoseNet and Bayesian PoseNet in every scene, and have similar results with Spatial LSTM PoseNet [24]. In the scene *Old*



(a) King's College (b) Old Hospital (c) Shop Facade

Figure 2: **Position samples from the predictive distribution.** We show 100 samples from three predictive pose distributions of our models from *Cambridge Landmarks*.

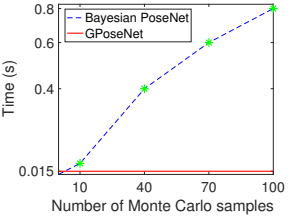
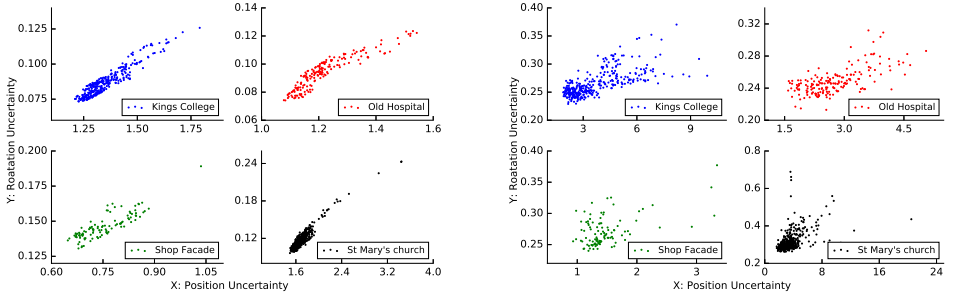


Figure 3: **Comparison of system efficiency.** For Bayesian PoseNet, the average time consumption for probabilistic inference is correlated to the number of Monte Carlo samples.



(a) GPoseNet (b) Bayesian PoseNet

Figure 4: **The correlation between uncertainty of translation and rotation.** This shows that the translation uncertainty is linearly correlated with rotation uncertainty, and the linearity is more obvious in our distribution compared to Bayesian PoseNet.

Hospital and *Stairs*, the proposed method produces similar accuracy with geometry-based method [].

This result shows that by replacing the L2-loss based pose regressors with the SVI GPs, our system improves the performance of the original PoseNet and Bayesian PoseNet. Since all of them use same CNN base, hence the advancement is contributed by the regressors. The comparable accuracy with Spatial LSTM PoseNet suggests that the effect of regressors replacement qualitatively equals the enhancement of the output feature from CNN.

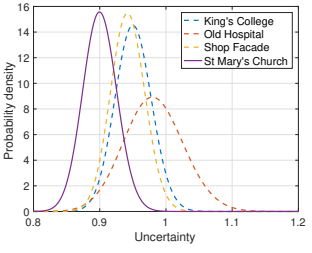
	King's College	Old Hospital	Shop Facade	St Mary's Church
King's College	78%	20%	1%	1%
Old Hospital	19%	61%	17%	3%
Shop Facade	0%	37%	63%	0%
St Mary's Church	13%	14%	2%	71%

(a) GPoseNet

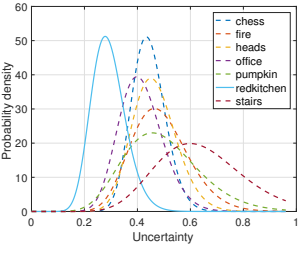
	King's College	Old Hospital	Shop Facade	St Mary's Church
King's College	75%	6%	12%	7%
Old Hospital	12%	80%	8%	0%
Shop Facade	5%	14%	76%	5%
St Mary's Church	2.5%	5%	3%	79.5%

(b) Bayesian PoseNet

Figure 5: **Confusion matrices of model uncertainty.** The test images from each dataset (row) are tested on the each model (column). We consider the model that the lowest uncertainty belongs to as the classified scene. We achieve about 69% accuracy for dataset *Cambridge Landmarks*, compared to 78% in Bayesian PoseNet.



(a) St Mary's Church



(b) Red Kitchen

Figure 6: **The gamma distribution of uncertainties on chosen model.** We evaluate all the test images on the model of scene *St Mary's Church* and *Red Kitchen* to obtain uncertainties. We plot the approximated Gamma distributions of test uncertainties. This shows that these two models produce smaller uncertainties on the test images from the corresponding scenes.

Uncertainty The "cherry on top" of this proposed framework is the ability to predict a distribution over the 6DoF pose without losing the localization accuracy. Fig. 2 shows the samples from estimated position distribution for three test images in different scenes.

Firstly, we compare the distributions of our method and Bayesian PoseNet in terms of efficiency. The pose distribution of Bayesian PoseNet is summarized from the Monte Carlo samples. The number of samples is also the number of inference times for one image. The more poses sampled, the more time consumed. In Fig. 3, we plot the average time for pose distribution estimation of one image against the number of samples[1]. If the number of samples is 40, it takes 0.4 second to estimate the pose distribution in average.

In contrast, the number of inference in our method for pose distribution is only one. As showen in Fig. 3, it takes 0.015 second to perform the distribution prediction, which is lower than the Bayesian PoseNet when the sample number is 10. Since the distribution is not from the sampling method, the time consumption is not related to the number of samples. If we leverage the parallel computing power of GPU, the average time for a single image inference could be less. This significant improvement of efficiency makes our system ready for real time relocalization with uncertainty.

To qualitatively evaluate the uncertainty of our model, we use the same measure in Bayesian PoseNet [10], which is the trace of the covariance matrix for each pose component. In the following evaluation, the term *uncertainty* stands for this trace.

[1] Due to the performance of our GPU, the time consumption of Bayesian PoseNet inference in this paper is more than [10]. However we perform all the experiments using the same hardware to ensure a fair comparison.

The translational uncertainty and rotational uncertainty from our model are strongly linear correlated. In Fig. 4, we show these two uncertainties from scenes in *Cambridge Landmarks* datasets. This linear correlation is in accordance with the results from Bayesian PoseNet, but with a more consistent linearity.

The uncertainty of camera pose can be interpreted as a measure for scene classifier. [10] defines a normalized metric, Z-score, to compare the uncertainties of different models. Firstly, all the test images from the scene that the model was trained on are inferred by the model, and then they fit a Gamma distribution over the uncertainties of the test results. At last, when a *new* image (from scene's test images split or images from other scene) is inferred by the model, a percentile of the *new* uncertainty in the Gamma distribution is defined as the Z-score for the image. We evaluate the test split from one scene on all models. The images are classified to one scene if the uncertainties inferred by that model is lowest. Fig. 5 compares the confusion matrices of our method and [10]. It shows that the smallest predictive uncertaintiy of a test image is majoritively produced by the model that is trained on the analogous training split.

The uncertainty of our model also indicates the confidence of the pose prediction. The confusion matrices in Fig. 5 is from a setting we call 'same-image-for-different-models'. This means the comparison is done by applying different models on the same images. We also do an experiment on the 'different-images-for-same-model' setting, which is an intuitive evaluation scheme for each individual model. To to so, we evaluate the test splits from all scenes on one of the models, and plot the probability density function of the approximated Gamma distribution of uncertainties in Fig. 6. We can see that if the images are from the test split of the dataset that this model is trained on, the model tends to estimate a lower uncertainty. This observation corroborates the conclusion of the vanilla GPR.

It means that the uncertainty from our model is a practicable indicator for the confidence of the inferred result. We suggest that this confidence also can be used in other tasks beyond pose regression, such as image classification or object detection, for which the most of the deterministic CNNs trust the prediction with absolute certainty.

6 Conclusion and future work

We show how to combine the deterministic CNN and probabilistic GPR together to accomplish real time camera relocalization with modelling the uncertainty. This is done by replacing the traditional L2 norm loss based linear regressor with KL divergence based SVI GPs regressor. It improves the system efficiency of method based on bayesian approximate CNN without losing accuracy. In the future, we would like to exploit the different forms of kernel functions and other non-Gaussian likelihood to improve the performance. And the usage of the model uncertainty could also be considered to track the 6DoF pose of camera in real time, as well as in other tasks.

7 Acknowledgement

This work was supported by the Australian Research Council through the Australian Centre for Robotic Vision (CE140100016), the ARC Future Fellowship (FT120100969) to C. Shen, and the ARC Laureate Fellowship (FL130100102) to I. Reid.

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: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

[3] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372, 2016.

[4] Lehel Csató and Manfred Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

[5] Andreas Damianou and Neil Lawrence. Deep Gaussian processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 207–215, 2013.

[6] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(6):1052–67, 2007.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[9] James Hensman, N Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 282–290, 2013.

[10] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[11] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[14] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research (JMLR)*, 18(40):1–6, 2017.

[15] Lili Meng, Jianhui Chen, Frederick Tung, James J Little, Julien Valentin, and Clarence W. de Silva. Backtracking regression forests for accurate camera relocalization. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 6886–6893, 2017.

[16] Raúl Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31 (5):1147–1163, 2015.

[17] Joaquin Quiñonero-candela, Carl Edward Rasmussen, and Ralf Herbrich. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research (JMLR)*, 6:1935–1959, 2005.

[18] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

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

[20] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013.

[21] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1257–1264, 2006.

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

[23] Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 567–574, 2009.

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