Improved Visual Relocalization by Discovering Anchor Points

Soham Saha[1, 2]
soham.saha@research.iiit.ac.in
Girish Varma[1]
girish.varma@iiit.ac.in
C V Jawahar[1]
jawahar@iiit.ac.in

[1] Kohli Center for Intelligent Systems, Center for Visual Information Technology, IIIT Hyderabad, India.

[2] Flipkart Internet Private Limited, Bangalore, India.

Abstract

We address the visual relocalization problem of predicting the location and camera orientation or pose (6DOF) of the given input scene. We propose a method based on how humans determine their location using the visible landmarks. We define anchor points uniformly across the route map and propose a deep learning architecture which predicts the most relevant anchor point present in the scene as well as the relative offsets with respect to it. The relevant anchor point need not be the nearest anchor point to the ground truth location, as it might not be visible due to the pose. Hence we propose a multi task loss function, which discovers the relevant anchor point, without needing the ground truth for it. We validate the effectiveness of our approach by experimenting on Cambridge Landmarks (large scale outdoor scenes) as well as 7 Scenes (indoor scenes) using various CNN feature extractors. Our method improves the median error in indoor as well as outdoor localization datasets compared to the previous best deep learning model known as PoseNet (with geometric re-projection loss) using the same feature extractor. We improve the median error in localization in the specific case of Street scene, by over 8m.

1 Introduction

The visual relocalization problem is an essential component of many practical systems like autonomous navigation, augmented reality, drone navigation etc. Although GPS sensors could be used for localization in these applications, it is often noisy and will not work in indoor environments. Hence an independent source of location information is essential for safety as well as applicability in a wide variety of environments. Though location information could be accurately estimated using 3D point cloud data, gathering and processing such data can be expensive and slow. The visual relocalization problem is defined as estimating the location as well as camera pose, given just the observed camera frame, without using any other sensor data.

Early solutions modeled this problem as an image retrieval problem. However, these solutions required features for a large collection of images to be stored. Also, a nearest neighbour search was needed at test time. Thus the memory and runtime increased proportionately with the dataset size. The most accurate approaches require 3D point cloud data

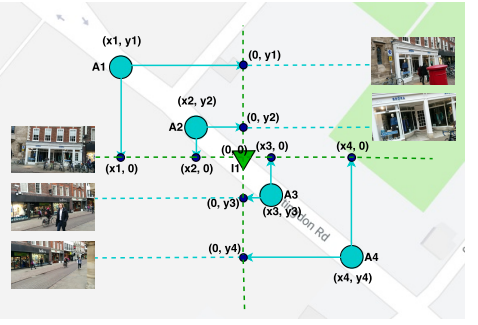

Figure 1: An example of anchor point allocation for a sample road. The blue points (A1-A4) denote the anchor points which are predefined uniformly. The green triangle(I1) is the input frame which denotes the current location. The coordinates of the anchor points are shown relative to the coordinate of the current location (denoted as (0,0)). The nearest anchor point A3 or A2, might not be visible properly from the current location. Our system can find the most relevant anchor point and the relative offsets from it, giving a more accurate prediction.

of the region [20], which is expensive to gather and process. PoseNet proposed to model this problem directly as a regression problem and used a deep neural network as the image feature extractor [12]. Extensions to this work aim at modeling the uncertainty in estimating the location and pose by using a Bayesian neural network [11]. Furthermore, a spatial LSTM based approach to regress the pose was found to improve performance [23]. A more recent approach has been to improve points through a geometric loss function, which tries to minimize the reprojection error [11].

We propose a visual relocalization system inspired by how humans determine location. We typically identify some landmarks which we deem as important. The coordinates of these landmarks are already known. Then, we try to estimate our position by determining the offset to our position relative to the landmark. The landmark we choose need not be the nearest one, as it might not be visible, due to the viewing angles. We refer to these landmarks as anchor points. Inspired by this idea, we propose a system which assigns anchor points relative to which the 6DOF (six degrees of freedom of the 3 spatial and 3 pose coordinates) can be predicted accurately. We do this by modeling the problem as a multi task problem, of classifying the input scene to an anchor point and then finding the offsets relative to that anchor point. However, a direct approach for training such a network requires ground truth of the anchor points for each image. The datasets typically provide only the ground truths of the 6-DOF and not for relative offsets from the anchor points. The anchor point visible in the image need not be the nearest one, because of the pose. Hence we propose a new loss function, which during training, automatically finds the appropriate anchor point relative to which the offsets needs to be regressed in an end to end fashion.

We benchmark our model on an outdoor dataset covering a large area (Cambridge landmarks [12] covering few 1000 m^2) as well as an indoor dataset (7 Scenes [22] covering few m^2) suited for small robot navigation. We improve the median error in all the scenes of both the datasets, from the previous best model, PoseNet (with geometric reprojection loss [11]), when using the same GoogleNet [24] feature extractor. We achieve <1.5m and <4° in localization performance in 4 out of the 6 outdoor scenes in Cambridge Landmarks. In the specific case of Street scene, our method improves the median error by over 8m. Furthermore, we localize to within <0.2m for all the indoor scenes of the 7 Scenes dataset which is a significant improvement over the previous deep learning based approaches. We also experiment with various feature extractors like DenseNet [8] (giving better accuracy) and MobileNet [7] (giving better runtime performance, while maintaining accuracies).

2 Related Works

In this section, we briefly survey the literature. For a detailed survey see [17].

Visual Place Recognition. The visual place recognition task has been traditionally solved by modeling it as an image retrieval problem [1], [5], [27], [21]. This enables Bag of Words (BOW) and VLAD [9] representations to be used in scalable retrieval approaches. More recently, deep learning models have also been used successfully for creating efficient image representations, which have been combined with retrieval methods [18], [6], [2], [26] [3]. However, retrieval based solutions require image features for the entire dataset to be stored. Also, a nearest neighbour search needs to be triggered at test time. Thus, the memory and runtime increased proportionately with the data set size. PlaNet [29] modeled the problem as a classification problem for localization at a world scale. However, estimation of a fine-grained 6-DOF localization requires continuous output and discrete methods have not worked so far for this task.

In contrast, metric localization (visual localization) techniques approach this as a 2D-3D mapping between the 2D coordinate system of the image space to the 3D coordinate system of the world space. This is typically done by matching the representations of the learnt image [4, 14, 15, 19, 20, 23] and adapting a nearest neighbour approach. The full 6-DOF pose of the camera can be estimated quite precisely. These methods however require a large database of features and efficient retrieval methods, which makes them suffer at test time since most retrieval methods have an additional feature matching latency.

PoseNet. PoseNet [12] addressed this idea by introducing a regression based deep neural network technique to estimate the metric localization parameters. It achieves a combination of strengths of place recognition and localization approaches and can localize without a prior estimate on the pose. This solves the disadvantage of storing the features in memory since they are learnt during training and estimated during testing.

Extensions to this work have been on RGB-D input images for localization [13], learning relative ego-motion of the camera [16], improving the context of features used for localization [25], using video sequence information for localization [10], modeling the uncertainly in localization using Bayesian Neural Networks [10], exploring a number of loss functions for learning camera pose which are based on geometry and scene re-projection error [11].

Although PoseNet and its extensions are robust and scalable, they lag behind some of the traditional methods [20] in estimating the pose . In this work, we propose a novel discrete anchor point assignment based regression technique which provides significant improvement over these methods while maintaining the scalability and robustness. We keep intact the advantages of PoseNet and its variants as well as improve over the nearest neighbour based methods, where feature vectors and required to be stored in memory entirely.

3 Approach

The visual relocalization problem involves inferring the 3D location coordinates and the camera pose (given by 3 angles) from just the image. In the previous works [10, 11, 12], a deep neural network based approach was proposed which directly predicts the 6-DOF by regressing against the ground truth, relative to a global coordinate system. Here, we propose a system which assigns anchor points relative to which the 6-DOF can be predicted accurately. We do this by modeling the problem as a multi-task problem. The first task is

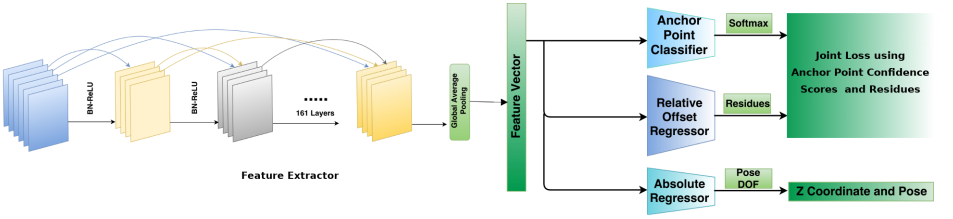

Figure 2: Block diagram of the proposed architecture. We experiment with different CNN feature extractors including GoogleNet, DenseNet and MobileNet. The CNN feature extractor is followed by a global average pooling layer to get the final feature vector. The feature vector is fed to an anchor point classifier, relative offset regressor and an absolute regressor for pose.

of classifying the input camera image to one of the anchor points and the second task is of finding the offsets relative to that anchor point. We use standard CNN feature extractors for image feature extraction. The feature extractors are typically followed by a Global Average Pooling (GAP) layer in order to get a single vector instead of a feature map. This is illustrated in Figure 3. Our proposed method then branches this output into 3 heads.

Anchor Point Classification. Given a route map where visual relocalization is desired, we subdivide the route into equally spaced intervals. The end points of the intervals are designated as anchor points for the 2D spatial coordinates. Note that the global coordinates of these anchor points are known, since they are fixed a-priori. We then model the problem of finding the most relevant anchor point as an image classification problem. Given an image, the predicted class should be the most prominent anchor point in the image. Note that the most prominent anchor point need not always be the one nearest to the location of the image. Hence, we frame our problem in such a way that we need not have the ground truth information about the prominent anchor point directly as part of the training data set. We treat the predicted probabilities as a confidence score for each of the anchor points. We combine this information in our loss with the relative offsets. Furthermore, we experiment by considering the nearest anchor point to the image, as the ground truth, and incorporate the corresponding cross entropy loss in our overall loss function. We report results for both cases.

The Anchor Point Classification head is obtained by mapping the global average pooling output of the feature extractor to the number of predefined classes using a Fully Connected (FC) layer. In our case, the number of classes is equal to the number of anchor points defined during preprocessing. The output of the FC layer is then subjected to a softmax classifier to get a probability mapping. However, instead of using these values as probability predictions, we use them as confidence scores in our experiments as will be explained in more detail in Section 3.1.

Relative Offset Regression. Along with anchor point classification, our system also produces the relative offsets from the anchor point. We train these offsets through the regression loss. However, we face the same problem of not having the ground truth for the most relevant anchor point. Hence our system produces the relative offsets with respect to all the anchor points and our model tries to regress these relative offsets. Note that since the global coordinates of all the anchor points are known, the relative offset ground truth information for all the images can also be calculated from the global ground truth information. The output of the GAP layer of the feature extractor is passed through a fully connected neural network layer to obtain the regressing head whose task is to predict the X,Y Coordinate offset values relative to each anchor point defined previously. Thus if there were N anchor points, say, the

output of this head would be a $2N$ dimensional vector.

Absolute Offset Regression for Z and Pose. Our approach is motivated from human scene recognition through relative identification. Hence, having the anchor points can be efficient only for the X and Y coordinates and not for the Z and camera angle coordinates. Predicting relative offsets for Z coordinate and pose is not carried out due to the following reasons:

1. The Z coordinate and the camera pose of a scene is independent of the previous scene. It is thus counter intuitive and does not capture the continuity of the scene like the X,Y coordinates do.

2. Regressing relative offset for the Z coordinate as well as the pose complicates the regression task further thereby leading to inferior results as we found out in our experiments.

Hence, our model separately predicts the the remaining 4-DOF in accordance with the global coordinate system. The third and final head is assigned the task of predicting the absolute values of the Z coordinate and the remaining DOF values for determining the camera pose. Therefore, the GAP layer of the feature extractor is mapped to a 5 dimensional vector in this case. This head also consists of a fully connected layer for regressing the absolute values of the remaining DOF (i.e Z coordinate and pose).

3.1 Loss Function

We train our model with a loss function which is the sum of 3 components. Let us denote the output of the anchor point classification head to be $\hat{C}$, which gives confidence scores for each of the N anchor points, for the current input image. X, Y denotes the vectors of dimension N of ground truth offsets of the frame in the horizontal plane, with respect to each of the N anchor points and Z, the vertical distance. The ground truths X, Y, though not provided with the dataset, can easily be computed as a preprocessing step, by changing the origin of the coordinate system to the respective anchor points. P denote the three angles of the camera. Let $\hat{X}, \hat{Y}, \hat{Z}, \hat{P}$, be the predictions of the model.

Since we do not have ground truths for the most relevant anchor point in the image, we cannot train the anchor point classification head with a cross entropy loss. However, we formulate a joint loss function that uses $\hat{C}$ and the offsets $\hat{X}, \hat{Y}$ from each of anchor points. We weight the squared loss of offsets with confidence scores of the respective anchor points. The resultant loss function therefore looks like the following:

$$\sum_i \left[\left(X_i - \hat{X}_i\right)^2 + \left(Y_i - \hat{Y}_i\right)^2 \right] \hat{C}_i \tag{1}$$

This is motivated by the fact that the accuracy is determined by the offset of the most relevant anchor point. If an anchor point is completely irrelevant ($\hat{C}_i = 0$), then we do not need the corresponding offset predictions.

Now, we include the predictions of the absolute regressor which is responsible for predicting the Z coordinate values and the remaining angular DOF for determining the pose. In order to do this, add the following component to the loss function:

$$\sum_i \left[\left(Z_i - \hat{Z}_i\right) \right]^2 + \left\| P_i - \frac{\hat{P}_i}{\|\hat{P}_i\|} \right\|^2 \tag{2}$$

Additionally, we also experiment with adding a cross entropy loss with the ground truth being nearest anchor point.

$$\text{Confidence loss} = H\left(C_i, \hat{C}_i\right) \tag{3}$$

All the 3 components of our loss function mentioned in equations 1, 2, 3 are then assigned weights (hyperparameter) and the summed up to get the overall loss for our task. Therefore, we now have the following resultant loss function:

$$\alpha_1 H(C_i, \hat{C}_i) + \alpha_2 \sum_i \left[\left(X_i - \hat{X}_i\right)^2 + \left(Y_i - \hat{Y}_i\right)^2\right] C_i + \alpha_3 \sum_i \left[\left(Z_i - \hat{Z}_i\right)\right]^2 + \left\| P_i - \frac{\hat{P}_i}{\|\hat{P}_i\|} \right\|^2 \tag{4}$$

where $\alpha_1, \alpha_2, \alpha_3$ are weights assigned to the separate components of the loss function.

4 Dataset and Experiments

We benchmark our method on an outdoor and as well as indoor localization dataset. A summary of the datasets can be found in Table 1. We rescale all the images to 224×224 since our convolutional feature extractors are trained on those dimensions.

As described in Section 3, we initialize our feature extractor (DenseNet architecture) pretrained on the Imagenet 1K dataset trained on the image classification task. This gives better performance for outdoor scenes than using a network pre-trained on the Places dataset as shown in earlier work by Kendall et.al [12].

Dataset	Type	Scale	Imagery	Scenes	Train Images	Test Images	Spatial Area
Cambridge Landmarks [12]	Outdoor	Street	Mobile phone camera	6	8,380	4,841	100×500 m
7 Scenes [23]	Indoor	Room	RGB-D sensor (Kinect)	7	26,000	17,000	4×3 m

Table 1: Summary of localization datasets used for benchmarking.

As a pre-processing step, we first divide the scene space into several anchor points by uniformly selecting every n-th frame from among all the frames in a scene video. The distance of x,y coordinates for each image is then calculated relatively from each of these anchor points. This is the ground truth information for the Relative Offset Regressor. We illustrate some of the selected anchor points for 3 scenes, King's College, Shop Facade and Street in Figure 4.

Our proposed loss function has 3 separate components. We assign separate weights to each of these components, all of which are hyperparameters, and are subject to be optimized through grid-search or any other hyperparameter optimization technique. We found specific ranges for these weights for outdoor scene localization task. The first part of the loss function is the cross-entropy calculated between the classifier output and the nearest anchor point to an input scene, which acts as its label. For this classification loss, we use a weight value ranging between 1-3 across all the scenes. The second component of the loss function is the Relative Offset (Translation) Regressor which predicts the relative distance of the x,y coordinates of the input scene from each of the anchor points. As a weight for this component, we use a value ranging between 4-30. Finally, the pose regressor predicts the remaining 5-DOF and a weight of 0.1-2 is assigned to it.

We use a learning rate varying from 0.00005 to 0.0005 for the scenes in the dataset. The Adam optimizer is used for optimization with a scheduler to decay the learning rate by half after every 30th epoch. The evaluation of accuracy is a subjective procedure. We consider a prediction for the scene to be accurate if the prediction for position is <2m and

the corresponding pose is <5°. Other thresholds for accuracy in evaluation will naturally produce variations in performance calculation.

5 Results and Discussions

For validating the effectiveness of our approach, we first compare the median errors for 6-DOF pose in the spatial and angular dimensions using our approach, as well as the previous best PoseNet model [12] . For fairness of comparison, we experiment with using the same GoogleNet [24] feature extractor as the PoseNet result. Secondly, we also experiment with newer feature extractors like DenseNet [8] and MobileNet [7] to drastically improve the accuracy and runtime speed. This also demonstrates that our approach generalizes to a variety of feature extractors. Furthermore, we provide a more detailed analysis of anchor points as well as qualitative results.

Comparison with PoseNet on GoogleNet. We compare our method with PoseNet. We use the same GoogleNet feature extractor used in the PoseNet results, so that we can observe the improvement due to our approach rather than using a improved feature extractor. We report results without the cross entropy loss applied to the classification head since letting the network discover relevant the anchor points gave better results. We report the median error for all the scenes in the Cambridge Landmarks and the 7 Scenes datasets in Table 2. Also, it can be seen from Table 2, that we are doing better than the previous best method of PoseNet (with geometric reprojection loss) [11] in all the scenes, and improve the localization median error by around 8m in the Street scene. We also improve upon the previous best performance for each of the indoor scenes in the 7 Scenes dataset.

Scene	Area or Volume	Active Search (SIFT) [20]	Posenet Spatial LSTM[12]	Posenet sigma^2 weight [12]	Posenet Geom. Rep. [12]	Ours (DenseNet) (cross entropy)	Ours (DenseNet) (w/o cross entropy)	**Ours (GoogleNet) (w/o cross entropy)**
Great Court	8000m^2	-	-	7.00m, 3.65°	6.83m, 3.47°	5.85m, 3.61°	4.64m, 3.42°	**5.89m, 3.53°**
King's College	5600m^2	0.42m, 0.55°	0.99m, 3.65°	0.99m, 1.06°	0.88m, 1.04°	0.55m, 0.97°	0.57m, 0.88°	**0.79m, 0.95°**
Old Hospital	2000m^2	0.44m, 1.01°	1.51m, 4.29°	2.17m, 2.94°	3.20m, 3.29°	1.45m, 3.16°	1.21m, 2.55°	**2.11m, 3.05°**
Shop Facade	875m^2	0.12m, 0.40°	1.18m, 7.44°	1.05m 3.97°	0.88m, 3.78°	0.49m, 2.42°	0.52m, 2.27°	**0.77m, 3.25°**
St. Mary's Church	4800m^2	0.19m, 0.54°	1.52m, 6.68°	1.49m, 3.43°	1.57m, 3.32°	1.12m, 2.84°	1.04m, 2.69°	**1.22m, 3.02°**
Street	50000m^2	0.85m, 0.83°	-	20.7m, 25.7°	20.3m, 25.5°	8.19m, 25.5°	7.86m, 24.2°	**11.8m, 24.3°**
Chess	6m^2	0.04m, 1.96°	0.24m, 5.77°	0.14m, 4.50°	0.13m, 4.48°	0.06m, 3.95°	0.06m, 3.89°	**0.08m, 4.12°**
Fire	2.5m^2	0.03m, 1.53°	0.34m, 11.9°	0.27m, 11.8°	0.27m, 11.3°	0.16m, 10.4°	0.15m, 10.3°	**0.16m, 11.1°**
Head	1m^2	0.02m, 1.45°	0.21m, 13.7°	0.18m, 12.1°	0.17m, 13.0°	0.08m, 10.7°	0.08m, 10.9°	**0.09m, 11.2°**
Office	7.5m^2	0.09m, 3.61°	0.30m, 8.08°	0.20m, 5.77°	0.19m, 5.55°	0.11m, 5.24°	0.09m, 5.15°	**0.11m, 5.38°**
Pumpkin	5m^2	0.08m, 3.10°	0.33m, 7.00°	0.25m, 4.82°	0.26m, 4.75°	0.11m, 3.18°	0.10m, 2.97°	**0.14m, 3.55°**
Red Kitchen	18m^2	0.07m, 3.37°	0.37m, 8.83°	0.24m, 5.52°	0.23m, 5.35°	0.08m, 4.83°	0.08m, 4.68°	**0.13m, 5.29°**
Stairs	7.5m^2	0.03m, 2.22°	0.40m, 13.7°	0.37m, 10.6°	0.35m, 12.4°	0.13m, 10.1°	0.10m, 9.26°	**0.21m, 11.9°**

Table 2: Comparison of median error for 6-DOF pose. As reported in the last column, our model performs better than the best deep learning model PoseNet [11], making the gap with active search methods [20] (which uses 3D point cloud data unlike us) lesser. Use of improved feature extractors like DenseNet reduces the errors further.

Improved Accuracy using DenseNet and Abalation study. We then experiment with the state of the art CNN feature extractors. DenseNet [8] is known to give improved accuracies for image classification. In Table 2, we report the results of our proposed method with the DenseNet feature extractor and compare with all the previous methods. We report the results with and without the cross entropy term (see Section 3.1). For the Cambridge Landmarks, 5 out of the 6 scenes, the model performed better without the cross entropy loss, validating our approach of discovering the appropriate anchor point relative to which offsets needs to be calculated.

In Table 3, we also report the accuracy, the mean and the median distance localized to in orientation, and the median camera angle pose localization for each of the 6 scenes of the

Table 3: We report the mean and median distances for each scene of the Cambridge Landmarks dataset using the DenseNet feature extractor. The accuracy is calculated by considering the orientation localization within 2m and the pose localization with 5° to be a correct prediction.

Scene	Mean Distance	Median Distance	Accuracy (<2m,<5°)	#Anchor Points
Great Court	10.48m	5.85m	69.51 %	154
King's College	0.76m	0.57m	93.52 %	122
Old Hospital	2.93m	1.45m	85.94 %	110
Shop Facade	0.72m	0.52m	93.76 %	33
St. Mary's Church	1.62m	1.12m	88.95 %	146
Street	17.45m	9.89m	11.26 %	201

Cambridge Landmarks dataset. It can be seen from Table 3, that the accuracy for the Street scene is 11.26%. The Street scene is the most challenging scene from among the used scenes for the visual localization task since it covers more than $50000m^2$ in area and also contains multiple landmarks. We achieve state-of-the-art performance for translation and pose on the Street scene as well. Our method is able to significantly improve the translation and orientation localization for this particular scene from 20.3m to 7.86m (see Table 2) which is a considerable improvement.

Next, we perform an experiment with DenseNet as the feature extractor connected to a 6-DOF regressor directly. This is a sort of control for our proposed method and we do not use an anchor point classifier here. We provide a comparison of our proposed method and directly using the regressor in the Cambridge Landmarks dataset. It is observed that our proposed method is able to perform much better than simply regressing the DOF from the DenseNet feature extractor. The results are summarized in Table 4. As can be seen our method gives lesser errors. This verifies that the improvement in median errors is indeed happening due to our approach and not due to the feature extractors only.

Scene	Median Dist. (DenseNet + DOF Regressor)	Median Dist. (Our method)	Accuracy (DenseNet + DOF Regressor)	Accuracy (Our Method)
Shop Facade	1.32m	**0.52m**	82.64%	**93.76%**
King's College	1.45m	**0.57m**	81.80%	**93.52%**

Table 4: Comparison between the DenseNet feature extractor followed by a simple regressor and our proposed method.

Runtime Analysis. The MobileNet [7] feature extractor is known to be significantly faster while maintaining accuracies. We illustrate the results when using it in Table 5. For some scenes in Cambridge Landmarks dataset, we empirically compare the performance and efficiency of 3 popular feature extractors, namely GoogleNet, DenseNet and MobileNet, when used with our proposed method. We observe from Table 5 that MobileNet provide a considerable better runtime performance, while still giving errors lesser than GoogleNet.

Scene	DenseNet (Feature Extractor)		GoogleNet (Feature Extractor)		MobileNet (Feature Extractor)	
	Performance	FLOPs	Performance	FLOPs	Performance	FLOPs
Kings College	0.57m, 0.88°	5998 M	0.79m, 0.95°	760 M	0.67m, 0.94°	569 M
Shop Facade	0.52m, 2.27°		0.77m, 3.25°		0.60m, 2.31°	

Table 5: Comparison of different feature extractors on the basis of performance and FLOPs using our proposed anchor point based method for visual relocalization on the Cambridge landmarks dataset.

Analysis of Anchor Points and Qualitative Results. An important hyperparameter for our approach is the number of frames between consecutive anchor point which is required for assigning anchor points as a preprocessing step. The outcome for this selection also determines the number of classes the classifier should predict since it is equal to the number of anchor points. We therefore plot the behavior of the median distance localization for

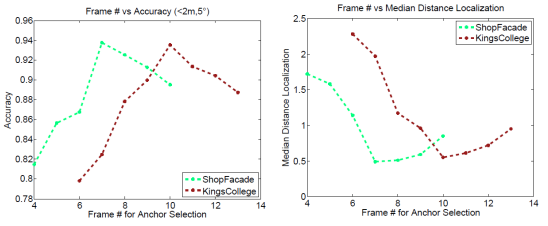

Figure 3: Plots showing how accuracy and median distance varies with different choices of frames interval between anchor points. We chose the optimum frame number for the dataset for anchor point selection.

translation and the overall accuracy, with the varying frame number selection (say k). This means that for a particular scene, we select every k^{th} frame. We observe in Figure 3, that we are able to get an optimum accuracy for a specific frame number which in turn makes the task easy of deciding the number of classes since it depends on k.

Finally, we showcase some qualitative results for the Cambridge Landmark scenes as well as for the 7 scenes dataset. It can demonstrated from those results that not only is the learned anchor point different from the nearest one but also better, in terms of camera angle and in avoiding occlusion.

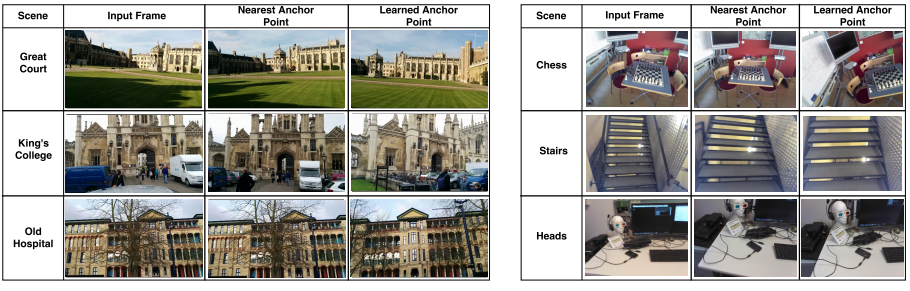

Figure 4: (Left) We contrast the nearest anchor point and the learned anchor point for an input query for the Cambridge dataset. Note that for the Old Hospital scene, the relevant anchor point learned is not blocked by a tree while the nearest anchor point is. Generalizing, learned anchor points gives a clear view of a landmark as compared to the nearest anchor point, validating our approach of discovering the relevant anchor point. (Right) Learned anchor points from the 7 Scenes dataset. In this case we observe a more zoomed in version of the input image is learned as the reference anchor point.

6 Conclusion

We propose a novel approach for solving the visual relocalization problem, inspired by how humans estimate their location, by observing suitable landmarks. We model it as a multi task problem of classification and relative offset regression. We propose a deep learning architecture and loss function which automatically discovers the anchor point relative to which accurate offset estimates can be predicted. We do not require each image to be tagged with the relevant landmark to train the classification part. Through our experiments, we show that our method achieves an improvement over PoseNet and its extensions in all scenes of the Cambridge Landmarks dataset as well as the indoor scenes of the 7 Scenes dataset. We achieve 1.5m and 4° in localization performance in 4 out of the 6 outdoor scenes in Cambridge Landmarks and 0.2m localization for the 7 indoor Scenes. Furthermore, our method outperforms simple replacement of the feature extractor followed by regression which further showcases the advantages of an anchor point classification and relative offset regression based method for the visual localization task.

References

[1] Relja Arandjelović and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *ACCV*, pages 188–204. Springer, 2014.

[2] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *ICCV*, pages 1269–1277, 2015.

[3] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *ECCV*, pages 584–599. Springer, 2014.

[4] Siddharth Choudhary and PJ Narayanan. Visibility probability structure from sfm datasets and applications. In *ECCV*, pages 130–143. Springer, 2012.

[5] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6): 647–665, 2008.

[6] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, pages 392–407. Springer, 2014.

[7] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. in cvpr, 2017.

[9] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 9(34):1704–1716, 2012.

[10] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Robotics and Automation, ICRA, 2016*, pages 4762–4769. IEEE, 2016.

[11] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017.

[12] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, pages 2938–2946, 2015.

[13] Ruihao Li, Qiang Liu, Jianjun Gui, Dongbing Gu, and Huosheng Hu. Indoor relocalization in challenging environments with dual-stream convolutional neural networks. *IEEE Transactions on Automation Science and Engineering*, 2017.

[14] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, pages 791–804. Springer, 2010.

[15] Yunpeng Li, Noah Snavely, Daniel P Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *Large-Scale Visual Geo-Localization*, pages 147–163. Springer, 2016.

[16] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 675–687. Springer, 2017.

[17] Nathan Piasco, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition, PR*, 74:90–109, 2018.

[18] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016.

[19] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, pages 752–765. Springer, 2012.

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

[21] 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 *CVPR*, pages 2930–2937, 2013.

[22] 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 *CVPR*, pages 2930–2937, 2013.

[23] Linus Svarm, Olof Enqvist, Magnus Oskarsson, and Fredrik Kahl. Accurate localization and pose estimation for large 3d models. In *CVPR*, pages 532–539, 2014.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[26] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[27] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *CVPR*, pages 883–890, 2013.

[28] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization with spatial lstms. *ICCV*, 2016.

[29] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *ECCV*, pages 37–55. Springer, 2016.