Learning Global Features for Fast Visual Localisation

Nathaniel J. McAleese
University of Cambridge
nm583@cam.ac.uk

Abstract

In both structure from motion and simultaneous localisation and mapping (SLAM) systems that operate on large scale data, a key performance bottleneck is identifying the approximate location of an image. This work uses a deep metric learning approach to generate an index of compact global features that can be used to quickly identify candidate locations for previously unseen images. The localisation accuracy of these fast proposals in fact exceeds the final accuracy achieved by both recent deep learning approaches such as PoseNet as well as popular bag-of-word based approaches. The features may also be quantised, allowing for a database that stores only 128 bits per image whilst achieving similar accuracy.

1. Introduction

Simultaneous localisation and mapping (SLAM) and structure from motion (SfM) represent extremely exciting applications of computer vision, yet both technologies struggle to scale to truely large datasets. City-sized reconstructions take hours on clusters of hundreds of computers [2], and SLAM systems struggle to extend beyond kilometre journeys [16].

One core bottleneck in these tasks is determining approximate candidate locations of an image - after this step, existing methods for pose estimation based on RANSAC and epipolar geometry work very well [23], but it is well established that these geometric checks are too expensive to run for every image in a large database [17].

This paper proposes a novel combination of neural network based techniques to quickly provide candidate locations for a new image. These proposals may then be fed into a classical system for geometric verification. However the candidate proposal scheme works sufficiently well that simply selecting the first proposal as the predicted location for an unseen image outperforms recent work such as PoseNet and its extensions [11, 10], despite using many fewer parameters and providing faster inference.

The approach works by using training images from one or more existing SfM reconstructions to determine a relation R over the images - two images are in R if they have sufficient visual overlap and were captured from similar locations (see Section 3.3 for more details). A metric learning approach inspired by the state of the art in person reidentification [7] is then used to learn an embedding into a space in which cosine similarity is predictive of membership of R.

At inference time, a previously unseen image is embedded into the search space and standard approaches for quickly searching metric spaces [27, 19] can be applied to find similar images at known locations from a database.

In a somewhat surprising result, the resulting embeddings are also extremely amenable to quantisation. This allows for an accurate localisation system that stores only 128 bits per image in the database.

However, the approach is not without shortcomings. Results indicate that whilst the embeddings generalise well to unseen images of the same location, they do not generalise as well to similarity search in unseen locations. Whilst this issue is probably surmountable simply by acquiring more training data, it offers an exciting avenue for future work.

2. Related Work

The idea of fast proposals before a more expensive geometric check is not new. However COLMAP, ORBSLAM, and other state-of-the-art systems still use the bag of visual words (BoVW) approach developed more than ten years ago [23, 16, 15].

In BoVW, keypoint descriptors such as SIFT, SURF or ORB [14, 3, 21] are extracted from a large set of training images. These descriptors are then clustered (usually with k-means) into some predefined number of groups. The resulting clusters then constitute a "vocabulary" into which continuous feature descriptors can be quantised. A given image is then represented as a multi-set (bag) of quantised descriptors (visual words). The similarity between two images is then determined analogously to the similarity between two documents, using techniques such as Jaccard similarity and TF-IDF weighting of words [17].

Whilst this approach has some appeal, there are draw-

Figure 1. The top three results returned for the same query by the quantised 128-bit code proposed here and the DBoW2 library and the 1 million word vocabulary provided by ORBSLAM. Note that the location of these images is shown in Figure 2.

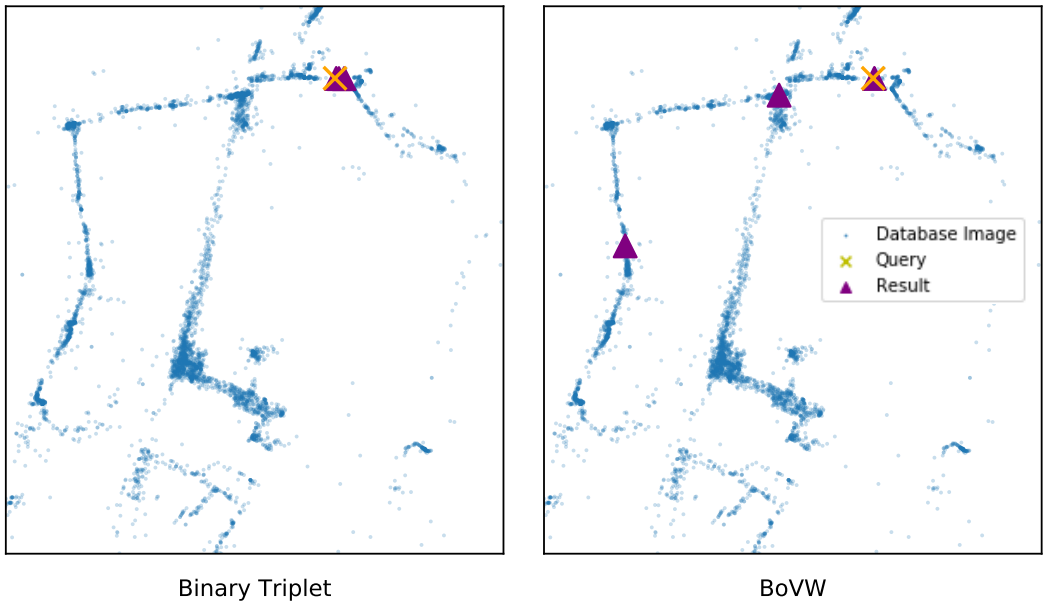

Figure 2. The spatial location of the top three results returned for the same query by the quantised 128-bit code proposed here (left) and the DBoW2 library and the 1 million word vocabulary provided by ORBSLAM. Note that these are the same query images as shown in Figure 1. Scale is 500x500 meters.

backs. For example, the BoVW representation of an image contains no geometric information. Another limitation is vocabulary size - in order to achieve usable results, vocabularies usually contain ten thousand to a million words [17, 16]. Thus, given the 3000 or so 128-dimensional features that might typically be extracted from a query image, even the task of quantising the extracted features (determining which word each is closest to) will be very computationally intensive if performed naively!

The most efficient approach to BoVW uses a vocabulary tree to combine the quantisation and indexing steps of search [17]. In this approach, heirarchical k-means is used to construct a tree of clusters. First, the key points are clustered into k clusters, then k-means is recursively applied to each of these clusters L times, resulting in a tree of depth L and branching factor k, giving a vocabulary of size k^L. This approach ensures that the quantisation of each feature can be achieved in kL dot products by determining which

of the k centroids is closest to it at each of the L levels of the tree. An inverted index is stored for each leaf node (visual word) in the tree, and so the quantisation of a particular image feature reveals the other images in which it occurs for scoring in a combined step.

This work develops an alternative approach to this proposal scheme that uses convolutional neural networks (CNNs). CNNs have previously been naively applied to loop closure in SLAM by directly using features extracted by the networks for similarity comparison, however this previous work did not attempt to learn a compact embedding; resulting in global features 72x larger than those presented here (without considering quantisation, which achieves a further 32x reduction in size). The most similar technique to the one proposed here is very recent work that learns a semantically meaningful embedding using a convolutional variational autoencoder to learn representative features without ground truth localisation data [24].

Other work has applied CNNs to similar problems at very different spatial resolution. For example PlaNet and deep attentive local features [18, 26] approach global scale localisation and landmark recognition respectively as classification problems. In PlaNet, the world is divided into 26,263 cells and localisation is treated as hierarchical classification. Deep Attentive Local Features presents a similar approach to landmark recognition by treating the problem as a fifteen-thousand-way classification problem. Whilst these approaches are interesting, they provide localisation to only a few hundred meters at best, and apply a fundamentally different approach.

Recent work has also investigated CNNs for the general task of 'reverse image search', however relevance in this context is usually defined with the semantics of object classes (e.g. given a photo of a cat, return images of similar cats), as opposed to the semantics of location (eg given a photo of a cat, return images taken at the same location) [28]. Our results section shows how the training procedure transforms a space preserving one notion of semantic similarity to another (see section 4.4).

Near-identical image matching via "image hashing" has also been investigated, but is not appropriate for this use as it is insufficiently flexibel to account for changes in lighting, season and other varying scene content.

PoseNet and its extensions aim to use CNNs to replace the entire localisation pipeline, as opposed to just the proposal step. It directly regresses from an image to its location and pose in 3D space. The work is extremely interesting because it is surprising that CNNs are sufficiently powerful to learn such a mapping and generalise it to unseen data. This work improves upon the localisation accuracy of PoseNet whilst using roughly 10x fewer parameters and substantially less computing power at inference time. It also provides more interpretable results by returning a set of related images as opposed to a single prediction of location.

Other approaches to visual localisation have eschewed the image retrieval problem entirely and instead focus on "direct matching" between a 3D model of the world from an SfM pipeline and a newly presented image [22]. Whilst these approaches have reported higher accuracy, even their proponents profess that this comes at cost of increased memory consumption [22]. This is prohibitive of their use on mobile devices.

3. Learning short codes

3.1. Triplet loss

We aim to learn a function $f_\theta(x) : \mathbb{R}^F \mapsto \mathbb{R}^D$ such that xs which are similar under our notion of semantic similarity are close together in $\mathbb{R}^D$ under some metric D. It is useful to shorten $D(f_\theta(x_i)), f_\theta(x_j))$ to $D_{i,j}$. The standard triplet loss, first introduced in FaceNet [25]

$$\mathcal{L}_{\text{Trip}}(\theta) = \sum_{\substack{a,p,n \\ (a,p)\in R \\ (a,n)\notin R}} [m + D_{a,p} - D_{a,n}]_+ \qquad (1)$$

Where $[n]_+$ denotes $\min(n, 0)$. As originally presented, R is the transitive relation "x and y share the same label", but transitivity is not intrinsically required by the nature of the loss. Intuitively, the expression above says that we consider all triplets of an anchor a, a related example p and an unrelated example n and try to ensure that the positive example is closer to the negative example by at least the margin m (where m is a hyper-parameter).

The obvious drawback of this approach is that the number of triplets grows cubically with the number of data points. Worse still, most empirical evidence [7] suggests that, in the case of relations on images, f_θ will quickly learn the correct relative embedding for most trivial examples, slowing learning to a crawl. Solutions to this problem have been proposed that include "mining" triplets of the appropriate difficulty. A more straightforward approach known as the "Batch Hard" triplet loss has recently been proposed that mitigates this issue and achieved a new state of the art for person re-identification [7]. Described here with a slightly modified form to account for our non-transitive relation, we consider P examples from the dataset. We then select K related examples for each of these P, giving a batch size of PK. The loss for a minibatch B containing each of these PK examples is then given by:

$$\mathcal{L}_{\text{BH}}(\theta; B) = \sum_{a\in B} [m + \max_{(a,p)\in R \,\wedge\, p\in B} D_{a,p} - \min_{(a,n)\notin R \,\wedge\, n\in B} D_{a,n}]_+ \qquad (2)$$

Figure 3. No images of snowfall are present in the training set, yet the model proves quite robust. Note also that these images also demonstrate substantial variation in camera technology (having been taken at least 6 years after the original dataset) and include HDR.

In other words for each example in the batch, we consider the most distant related example (the max term) and the closest unrelated example (the min term). If this hardest positive example is not closer to the anchor than the hardest negative one by more than m, the model is penalised. It might seem curious not to use all the information available in the batch, by considering the sum over all (a, p, n) triples, but it has been shown empirically that this results in slower convergence and worse overall performance [7]. Intuitively this is because the information from the moderate, "Batch Hard" examples is more useful, but this remains an active area of exploration. Another common optimisation, which was applied here, is to replace the hinge function $[x]_+$ with softplus, $f(x) = \ln(1 + e^x)$, to decrease the sparsity of the gradients. This is known as soft-margin. Cosine similarity was used as for D.

3.2. Model

In our case, f_θ is a convolutional neural network (CNN). As is standard in cases when training data is limited, an "off the shelf" network architecture was taken and adapted to the task at hand so that pre-trained weights from a large image classification task (ImageNet [4]) could be used as a good starting point for the optimisation. Because localisation is most useful on mobile devices, MobileNet [9] was chosen as the base architecture. This is a model engineered to minimise its number of parameters and computational cost. It offers a tuning parameter α that allows the accuracy/computational cost tradeoff to be set when doing model selection; experiments here use $\alpha = 1$ unless otherwise specified. When compared to AlexNet, another common baseline CNN architecture that was used in the development of PoseNet, it uses 14x fewer parameters, 60% of the FLOPs and achieves a much higher top-1 accuracy on ImageNet (70% as opposed to 57%). Setting $\alpha = 0.5$ further reduces the computational cost to 10x less than AlexNet whilst maintaining superior accuracy. The output of the pre-trained ImageNet classifier is then captured at the penultimate layer, giving a 1024 dimensional representation of each image. Two additional fully connected layers and a nonlinearity then reduce the size of this representation to 128 dimensions. The precise architecture used is illustrated in Figure 4. RMSProp [8] with a learning rate of 10^{-3} was used for optimisation with no learning rate scheduling. A 20% validation set was held out to monitor loss, after 100 epochs the model checkpoint with the lowest validation loss was selected for testing. The weights of the MobileNet module were initialised by pre-training on ImageNet, the additional two layers randomly.

3.3. What relation to learn

An SfM reconstruction was used to the target relation on images. Image pairs occur in R if they share at least one identical point in 3D space and were taken within five meters of each other. The requirement of visual overlap prevents training the model on impossible pairs, such as photos taken at identical locations but in opposite orientations. Whilst an SfM reconstruction is not necessarily needed to determine appropriate image pairs, it provides useful approximate ground truth data for evaluation and comparison to other techniques. Future work could extend the procedure to learn from more readily available data, such as Google StreetView images.

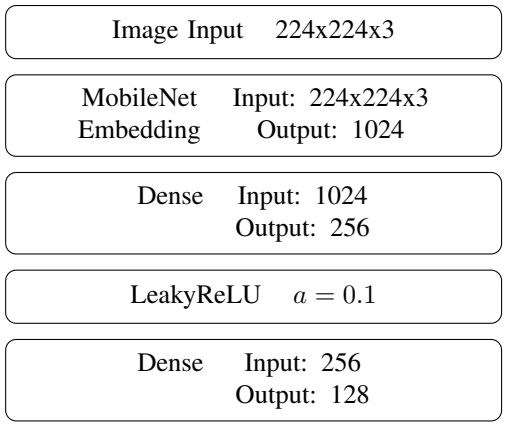

Figure 4. The structure of the embedding network. LeakyReLU is given by $\sigma(x) = x$ if $x > 0$ else ax and reduces gradient sparsity when compared to ReLU [6].

4. Results

4.1. Dataset

The Dubrovnik6K dataset is a collection of six thousand images of the old city of Dubrovnik and an associated SfM reconstruction [12]. It covers an area of 2.25 square kilometres and has 800 predetermined query images to standardise testing. Because there are a small number of outlier images in the canonical dataset (due to reconstruction error), it is standard to report median values for localisation results on this dataset [10]. The images in the dataset constitute real photographs from the internet taken over an extended period of time and thus are a challenging test of robustness to variation in device, lighting conditions, time of day and season. However, it is worth noting that all the images in the dataset were originally correctly localised by an expensive offline SfM procedure. Thorough and quantitative testing of the methods' robustness to even more extreme variation merits further work, but as a preliminary result Figure 3 exploits recent and extremely rare bad weather in Dubrovnik. No images of snowfall are included in the training dataset, yet the model returns reasonable results for the small set of manually gathered queries.

The 800 query images used at test time are not included during training.

4.2. Localisation Performance & Comparison with DBoW2

One of the most popular implementations of BoVW is the C++ library DBoW2 [5]. It provides a vocabulary tree implementation and is used by ORBSLAM and other high-quality localisation libraries. It was thus chosen as a baseline with which to compare performance. When using BoVW, the initial vocabulary must be determined from a dataset of images. For the purposes of these experiments, a vocabulary of 1 million ORB features distributed with ORBSLAM and a vocabulary extracted from the Dubrovnik dataset itself (naturally with the query images excluded) were used.

To evaluate the quality of the localisation proposals, we consider the median distance to the first result and also the median minimum distance in the top-k results. This is motivated by the idea of passing proposals to a more expensive verification step - if that step is capable of rejecting bad proposals, then what matters is the best (closest) proposal in the top-k.

Figure 5 shows the performance of these approaches alongside another experiment in which Triplet model was trained on a different location (images of Rome, sourced from [13]) that tests the generalisability of the features learned by this framework to new locations. The chart shows both the localisation accuracy of the first proposal (bar chart on the left) as well as the closest result in the top-k proposals for k up to 30. Interestingly, the features learned on Rome provide better performance than the generic ORB vocabulary, but worse than that provided by a Dubrovnik-specific one. All methods vastly outperform random selection (not shown), which gives a median distance of more than 220 meters that improves slowly as additional random results are returned.

In order to confirm that the localisation results presented here were due to the triplet training procedure proposed, the Triplet model was also compared to the raw usage of CNN features extracted by MobileNet. It is notable that all the CNN based methods except Triplet-Rome outperformed the BoVW approach quite substantially in terms of the quality of results. This is shown in Figure 8, with all methods returning top-1 proposals within a median distance of 10 meters, and three outperforming PoseNet's median performance of 7.9 meters. The model trained with the triplet loss results in a better initial proposal, a more rapid improvement with the number of returned results and a better final accuracy at 50 (and 100, not shown) results. This is despite using a 128-dimensional embedding, as opposed to the raw 1024D embedding initially extracted by MobileNet.

4.3. Localisation time

The use of MobileNet affords fast embeddings - even without straightforward optimisations such as 16-bit arithmetic that can improve speed with little accuracy cost, inference on a single CPU thread takes only 30ms for the triplet model. Setting $\alpha = 0.5$ also allows a faster model at to be trained a relatively small initial performance penalty (median top-1 localisation error of 6.8m, as opposed to 5.7m), and allows for feature extraction in 12ms on a single CPU thread.

However the vast majority of the computation in large-scale image search and localisation is not in the extraction

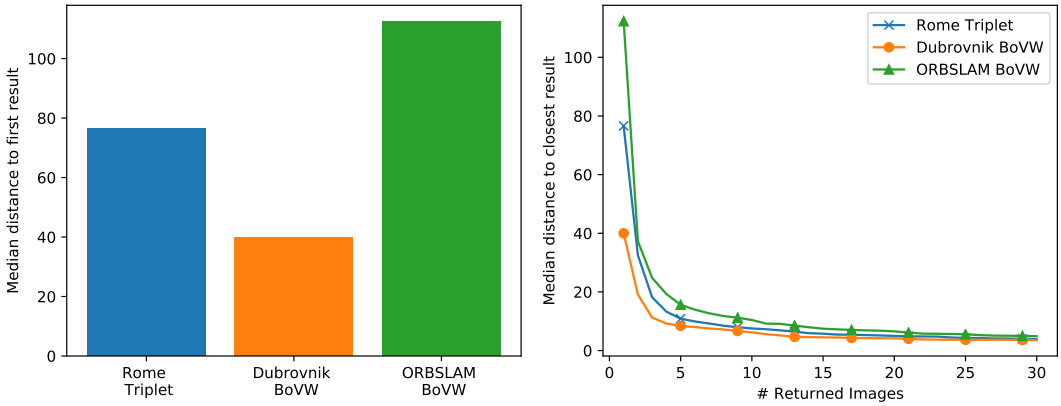

Figure 5. This figure illustrates the relative performance of transfer learning in the different approaches. Training the model on a different city (Rome) produces features that outperform the generic ORBSlam vocabulary, but does not achieve the same performance as a city-specific vocabulary.

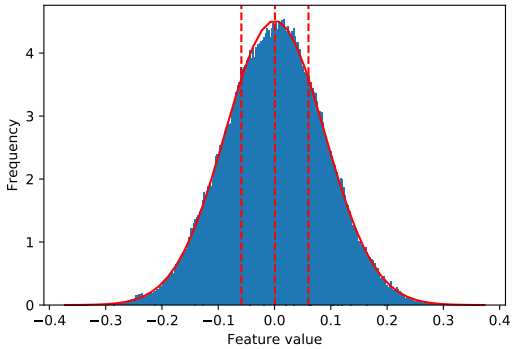

Figure 6. The features extracted by the model are very well approximated by a normal distribution, motivating a simple quantisation scheme. Here four equiprobable buckets are delimited, which would allow a 2-bit encoding.

of features. By extracting global features as opposed to hundreds of local descriptors we vastly decrease the cost of computing similarity; yet as it turns out there are further optimisations available. Inspecting the distribution of features, as shown in figure 6 shows that they are well approximated by a normal distribution, both in aggregate as shown and also on a per-feature level. This allows for a very simple quantisation scheme, in which equiprobable buckets are determined under the approximating normal and the features quantised into the buckets. Surprisingly, this results in little loss of accuracy, even when the features are aggressively quantised to one bit per dimension (eg greater than less than or less than the mean). This is shown in figure 7. This allows for further improvements in localisation time - with 128 bits per image, linear search is wholly viable, searching datasets around the size of Dubrovnik in well less than 2ms.

4.4. Interpretability & Example Proposals

One of the advantages of the proposed scheme, when compared to PlaNet and PoseNet, is that, given an image, the model localises it by telling us which images are similar. This set of similar images can always be inspected to learn about the model's decision making. However, it is also useful to determine why images are considered similar, in addition to what images are considered similar. The interpretability of neural models is an area of active research, and whilst simple tasks such as classification are amenable to LIME and related approaches [20], embedding tasks are not so straightforward. Figure 4.4 shows a visualisation specifically devised to test the embedding method trained here. Images x_i in the left column are embedded into the metric space, and a function is defined $s(x) = D(x, f_\theta(x_i))$. The gradient of this function is computed at x_i, thus determining the direction in image space that would most quickly increase the distance between x and the current position of the embedding. This allows us to determine what components of the image which, if altered, would most rapidly change its current location. Such results are often displayed as heat maps, which are often overly fuzzy and difficult to interpret. Thus the image is first segmented using SLIC [1], the mean gradient per image patch is computed, and the top 10 patches displayed. This results in a clearer representation of what regions matter most.

This approach is applied to randomly selected examples for both the trained triplet model and MobileNet with no fine tuning on Dubrovnik. As noted in the caption, this seems to provide compelling evidence that the untuned network is most influenced by the objects in the image, whilst the localisation network is influenced by the buildings and informative structure in the background.

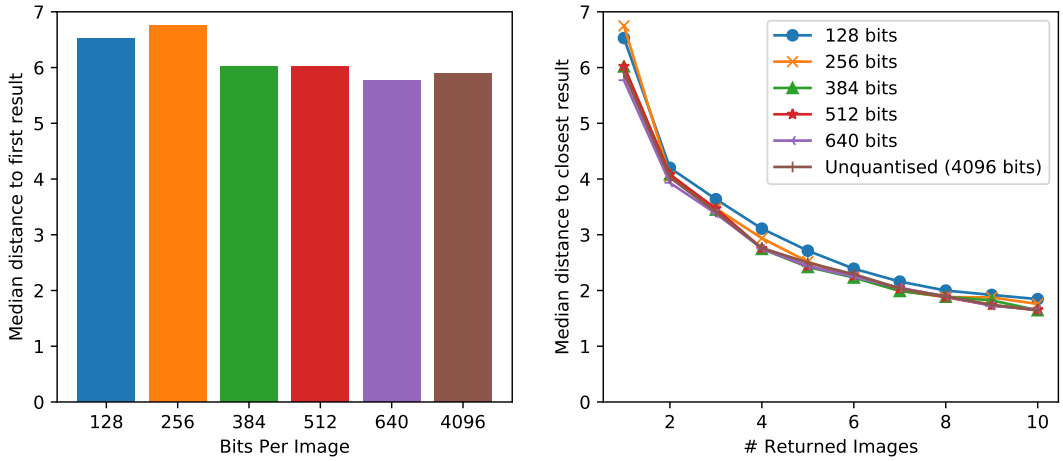

Figure 7. Embeddings can be aggresively quantised to further reduce search times.

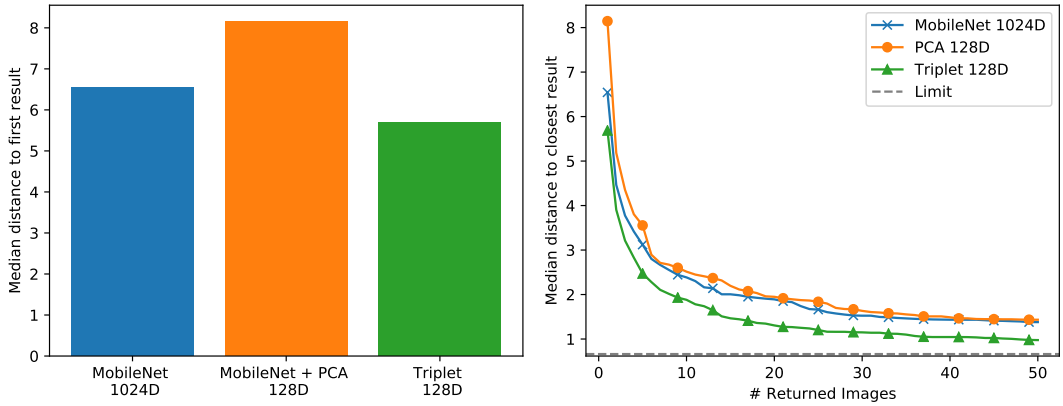

Figure 8. The proposed model returns closer initial results that also improve more rapidly than those provided by the untuned MobileNet model, despite using a smaller representation. A more naive dimensionality reduction technique, by comparison, degrades performance. The limit shown is that of returning the closest image in the dataset of every query.

5. Future work & Conclusion

The results of this initial exploration strongly suggest that the triplet loss is worth exploring further for localisation. The disappointing results in the case of transferring learnt embeddings from Rome to Dubrovnik (Figure 5) suggests that more data may be needed to acquire truly robust encoders, but success with extremely short 128 bit codes and under a variety of weather conditions is very promising. Once the proposal technique has been further honed, one next step will be to combine it with a fast geometric verification technique that is amenable to mobile deployment in order to determine final 6DoF pose. However other approaches also merit investigation. Because the system can be run at multiple frames per second, the candidate locations provided by the system could be effectively combined with a particle filter or similar smoothing scheme, potentially removing the need for geometric verification in cases that need only coordinate position as opposed to full pose. The most exciting next step, of course, is to empirically determine to what extent this approach can scale.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

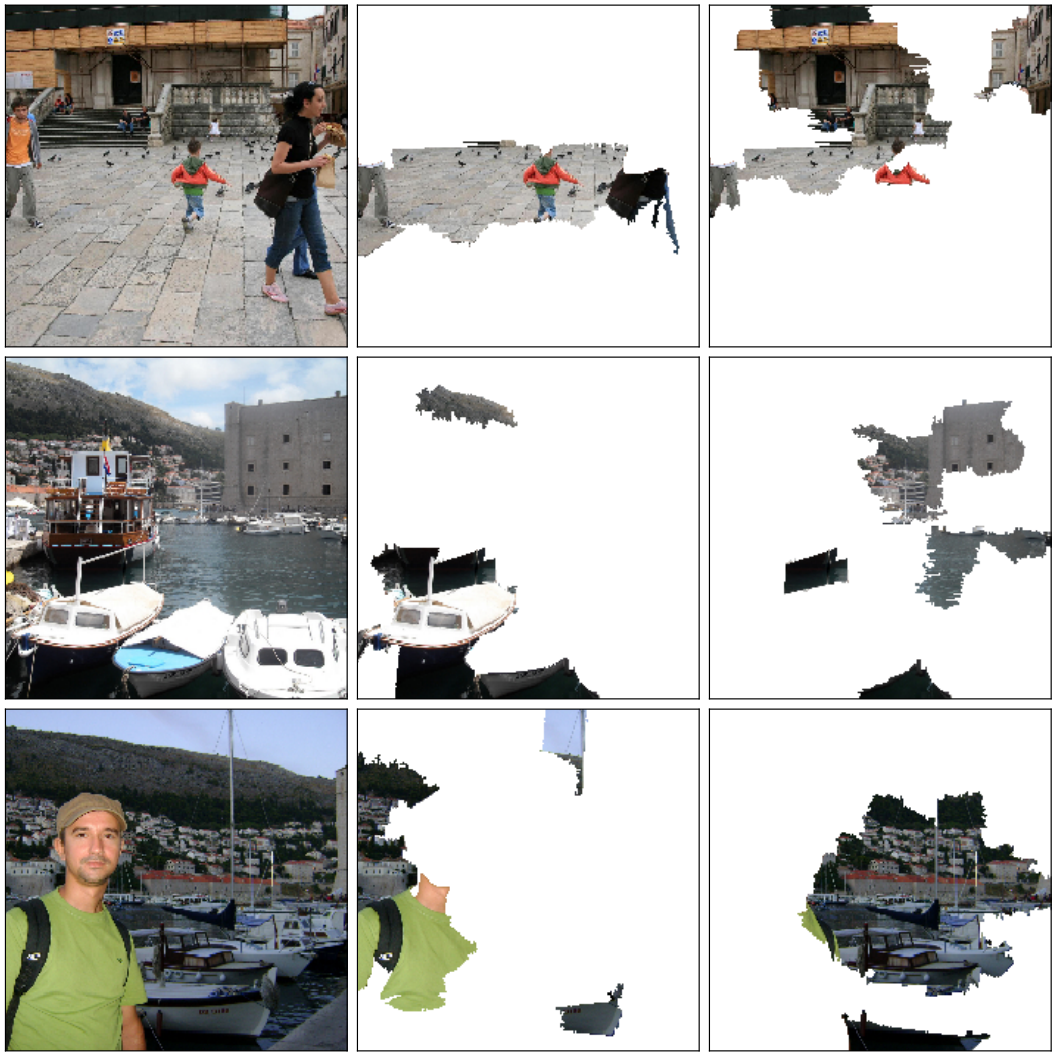

Figure 9. Saliency analysis: the left column shows the original image. Center, the regions that most influence the embedding produced by a model trained on ImageNet. Notice that the child, boat and person objects are more influential than in the right column. This shows the important regions for the localisation model, which focuses on the church, harbour and distinctive background buildings.

[5] D. Gálvez-López and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[7] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[8] G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning-lecture 6a-overview of mini-batch gradient descent.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proc. CVPR*, volume 3, page 8, 2017.

[11] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2938–2946. IEEE, 2015.

[12] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European conference on computer vision*, pages 791–804. Springer, 2010.

[13] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European conference on computer vision*, pages 791–804. Springer, 2010.

[14] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[15] A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning vocabularies over a fine quantization. *International journal of computer vision*, 103(1):163–175, 2013.

[16] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[17] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. Ieee, 2006.

[18] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3456–3465, 2017.

[19] M. Norouzi, A. Punjani, and D. J. Fleet. Fast search in hamming space with multi-index hashing. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3108–3115. IEEE, 2012.

[20] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[21] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[22] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, volume 1, page 4, 2012.

[23] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[24] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic visual localization. *arXiv preprint arXiv:1712.05773*, 2017.

[25] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[26] T. Weyand, I. Kostrikov, and J. Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[27] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, volume 93, pages 311–321, 1993.

[28] L. Zheng, Y. Yang, and Q. Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 2017.