Capturing of Detailed and Very Large Photograph and Localization Within

Bc. Pavol Dubovec*
Supervised by: prof. Adam Herout PhD.†

Department of computer graphics and multimedia
Brno University of Technology
Brno / Czech Republic

Abstract

This paper presents a new technique for locating a photograph within a larger one, with the aim of enhancing the speed and accuracy of conventional methods. The proposed technique utilises a CNN architecture to extract multiple embeddings from the query image[1], which are then used to perform an approximate search within a database of embeddings from the large photograph. Two main models were trained on a large dataset. The first model used a triplet loss function, while the second model used a cross-entropy loss function. Conventional methods were used to determine the location of the images in the training set and to generate a large image. A database of embeddings was created by partitioning the large photograph with a certain sampling frequency (in pixels) using the trained model. The database is queried for K-nearest sub-query[2] embeddings. These embeddings are generated by partitioning the query image into equal-sized pieces as CNN inputs. The optimal homography model is determined through random sampling based on the positions of four sub-query images and their corresponding positions in the large image. The model homography with the lowest harmonic mean embedding distance is selected as the resulting position. The method demonstrates satisfactory accuracy and good speed on the generated test datasets. The best model achieved a top-1 accuracy of 97.71% and a top-3 accuracy of 99.17%. Future research will investigate the method's performance with increasing surface heterogeneity, the potential for automating video retrieval to obtain a large dataset of photos, and its effectiveness for photo localization in cases where conventional methods fail due to a lack of key points.

Keywords: Image Localization, Homography Estimation, Approximate Search

*xdubov02@stud.fit.vutbr.cz
†herout@fit.vut.cz

[1] query image – image to be localized
[2] sub-queries – patches of query image with same size as inputs of NN

1 Introduction

This article discusses solutions to two common problems in computer vision and graphics: image localization and image stitching. The conventional method for addressing the localization problem involves detecting keypoints, extracting local features (descriptors) around these keypoints, matching the extracted features from the query image with features from a large image (map), and then using the matched keypoints to estimate homography. This homography can then be refined using bundle adjustment, which minimises the reprojection error. The methods themselves are explained in the sections 2 and 4. These methods rely on handcrafted keypoints and homography matrices, which use robust fitting methods such as RANSAC[7] or LMS[20]. However, they may perform poorly when the percentage of inliers falls below 50% [13]. These methods often use Hough transform [11, 2] to overcome this problem. This problem is particularly significant in cases where it is necessary to locate a very small picture within a much larger one, especially when there are very few common features.The aim of this study is to explore a novel approach to determine position based on a CNN-generated model to create an embedding database for a large image. This database is then used to locate a photography within the large photography. The process involves dividing the input image into smaller sub-queries, determining the embedding for each sub-query, and using a random sampling algorithm to create a homogra-

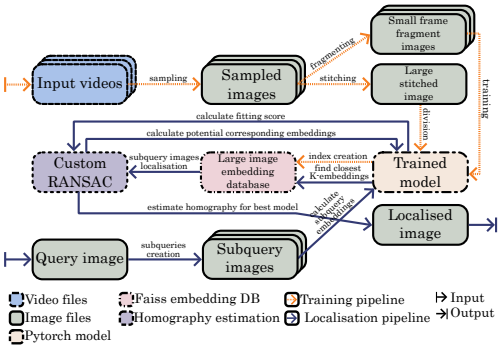

Figure 1: Model of training and localisation pipelines.

phy model. Preliminary results indicate that this method is both feasible and efficient, demonstrating promising speed and accuracy. Section 2 provides a comprehensive review of the relevant literature. The used methodology for the creation of the dataset is explained in Section 3. The process of stitching to create a large photograph is clarified in Section 4. Section 5 describes the architecture of the trained neural network models. The construction of the large image embedding database is explicated in Section 6. The procedure for localizing the query image is detailed in Section 7, followed by a presentation and discussion of the results in Section 8. The paper concludes in Section 9, where a summary of the findings is provided along with potential avenues for future research.

2 Related Work

The main focus of this paper is on image localization, local feature descriptors, and image retrieval, all of which are discussed below. This paper aims to apply these three methods to the localisation of query photography within large photography.

2.1 Homography Estimation

Homography estimation is a technique used in computer vision and image processing to find the relationship between two images of the same scene, but captured from different viewpoints. We can divide this process into few cathegories mainly by number of sources:

- **Single-source homography estimation** – source images are usually acquired by the same device from different viewpoints or at different times.
- **Multi-source homography estimation** – source image data with two or more different types of imaging mechanisms for the same scene or object.

The focus of this work is to work with a single camera so that we can focus on single source homography estimation techniques. These techniques can be divided into two main groups: feature-based and deep-learning methods.

2.1.1 Feature-Based Methods

In the feature-based homography estimation method, the feature points in the image are first detected by a feature extraction algorithm and then the similarity metric for the matching is calculated. The parameters of the homography matrix are then solved using the mapping relationship of the matched feature points.

1. **Conventional** – Conventional homography estimation methods rely on hand-designed feature extractors. The process conventional homography estimation is divided into three main steps:
   (a) **feature detection** – In this step, distinctive features are identified in both images. These features could be corners, edges, or other notable structures in the image,
   (b) **feature matching** – Finds matches between these features. This involves comparing each feature in one image with all features in the other image and finding the best match. The result of this step is a set of corresponding feature points between the two images,
   (c) **homography matrix estimation** – Using corresponding feature points the homography matrix [3] is estimated.

   Common conventional descriptors include:

   - **SIFT**[13] – A descriptor that is invariant to image scale and rotation, and robust to changes in viewpoint, noise, and illumination. It detects and describes local features in images based on the histograms of the gradient orientations within a local region around the feature.
   - **BEBLID**[22] – an efficient binary descriptor. It represents a small part of an image using a binary string of zeros and ones. In various benchmarks, it has been shown to significantly enhance other binary descriptors, such as ORB or BRISK, while maintaining the same level of efficiency.

2. **Learning-Based** – These methods utilise neural networks to replace feature extraction or matching in traditional algorithms. Traditional methods are then used to estimate the homography transformation parameters at subsequent steps. Common learning-based descriptors include:

- **LIFT**[25] – A novel Deep Network architecture that implements the full feature point handling pipeline, that is, detection, orientation estimation, and feature description. This technique implements hard negative mining techniques over the entire image to obtain more accurate descriptors.
- **SuperGlue**[21] – Introduces a flexible context aggregation mechanism based on attention, enabling it to reason about the underlying 3D scene and feature assignments jointly. Matches two groups of local features by collectively finding correspondences while rejecting non-matchable points.
- **MatchFormer**[24] – Hierarchical extract-and-match transformer. Interleave self-attention to extract features and cross-attention to match features.

[3] 3x3 matrix that describes the transformation from one image to another

2.1.2 Deep Learning-Based Methods

Methods with a unified homography estimation pipeline, handled by a deep neural network model. Network understands and handles complex image correspondences. Common deep learning-based methods include:

- **RHWF** [4] [4] – Supervised method. Combines homography-guided image warping and the focus transformer. Image warping improves feature consistency. Focus Transformer uses the attention focusing mechanism to aggregate the intra-inter correspondence into global, non-local and local. Has a relatively small number of parameters. There is an increase in computational cost due to the use of homography-guided image warping and attentional manipulation.

- **MS2CA-HENet** [5] [10] – Unsupervised method. The method uses different input sizes at different stages to deal with different scales of homography transformations between images. Lower error can be achieved when there are large changes in displacement between corresponding points.

2.2 Image Retrieval

Image retrieval is the process of retrieving relevant images from a large database based on a query image or query terms. Image retrieval methods can be divided into 2 categories:

- **Content-Based Image Retrieval** (CBIR)

- **Text-Based Image Retrieval** (TBIR)

This paper focuses on content-based image retrieval (CBIR). This technique uses the visual content of a picture like colours, shapes, textures and spatial layout to represent and index the picture. In CBIR, the features of each image stored in the database are extracted and compared with the features of the queried image. It involves two steps:

1. **Feature extraction** – In this step, features such as colour histogram, texture, shape, etc. are extracted from the image.

2. **Similarity measurement** – After extracting the features, the similarity between the extracted features and the features of the query image is calculated.

The current focus of research is on deep learning methods. We can divide them into two main categories [18]:

- **Off-the-Shelf models** – Pre-trained deep learning models which are used as-is for image retrieval without further training or modification. However, they may not perform optimally for specific retrieval tasks due to domain shifts-differences between the data they were trained on and the new target data.

- **Fine-tuned models** – Pre-trained models that are further fine-tuned on a specific dataset related to the retrieval task. The model's weights need to be adjusted to better suit the particular characteristics of the new dataset, which will improve retrieval performance. However, this requires additional data and computational resources.

On the **off-the-shelf** side Mohedano et al. [15] proposed that both fully-connected layer and last convolutional layer can be used as feature extractors. **Fully-connected layer method** lack spatial information and a lack local geometric invariance. A. Razavian et al. [18] proposed very efficient single feed forward pass technique where features are used for direct similarity measurement without further processing. The need for more accurate image retrieval has led to a surge of multiple feed forward pass techniques. Although these techniques are more time-consuming, they can lead to more accurate results. Also discriminative features from the image patches better retain spatial information [18]. Multi-scale image patches could be obtained using sliding windows Y.Gong et al. [8] or spatial pyramid Y. Liu et al. [12]. These methods have problems with retrieval efficiency so Cao et al. [3] introduced merging image patches into larger regions with different hyper parameters. Random or dense creation of image patches may not be ideal so Zitnick et al. [29] proposed method where region proposals can be generated using object detectors instead. **The last convolutional layer method** preserves more structural details, which is particularly advantageous for instance-level retrieval [19]. The convolutional layer effectively organizes spatial information and generates location-specific features [28]. Razavian et al. [19] were the first to attempt spatial max pooling on the feature maps of an off-the-shelf DCNN model. They also apply max pooling on the convolutional features for retrieval to improve the discrimination of deep features. Yue et al. [17] and are the first to encode local features into VLAD [26] features. R. Arandjelović et al. [1] used VLAD as a layer plugged into the last convolutional layer. In addition to pooling agregation techniques, it is possible to embed the convolutional feature maps into a high-dimensional space to obtain compact features. Commonly used embedding methods include Bag of Words (BoW can be used with other metrics, such as Hamming distance [23]), Vector of Locally Aggregated Descriptors (VLAD [26]), and Fisher Vector (FV [7]).

[4] Recurrent Homography Estimation Using Homography-Guided Image Warping and Focus Transformer

[5] Multiscale Multi-stage based Content-Aware Homography Estimation method

3 Dataset creation

The study utilises a large collection of photographs depicting various indoor surfaces. The carpet dataset is the most extensive and commonly used dataset, and is considered the reference due to its 11 possible light conditions (data was collected at various times throughout the day and/or under artificial lighting). The dataset contains 21457 / 68961 / 142489 images based on the output size of the image used. It can be used to test independence for reflection symmetry in both directions of the axis. Other datasets include: laminate1, laminate2, carpet2, stone1, stone2, rusty_sheet_metal and wood. These are smaller (5000 – 10000 images). The dataset was created with several assumptions that made it easier to create, including:

- The camera scanning the ground is parallel to it,
- The camera scanning speed is constant,
- The objects to be captured should not move,
- The diversity of the dataset was mainly achieved by changing the lighting conditions.

The process of capturing the desired material or object involves using a custom-built cart that is designed to hold the camera parallel to the ground. This process must be repeated several times to capture the object in different lighting environments. By doing so, the dataset becomes more generalised and less impacted by lighting changes. The process of extracting fragments out of video frames consist of:

1. Sampling of videos by given sampling size,
2. Dividing large image into smaller fragments,
3. Localising fragments in sampled frames using conventional methods.

If a classification approach is used, the fragments will be grouped into classes based on their spatial position within the image. This means that there is a limiting factor to this approach, which determines how many surface datasets can be used for training (the number of classes should be finite and not very large). In the classification method, the name of the fragment file also means the class to which it belongs, and the two numbers are the x and y coordinates of the midpoint of that class. For triplet loss, each sample is created as a triplet. This triplet consists of an anchor, a positive image (from the same class) and a negative image (from a different class). Triplets are created by generating csv file from all image files that consists of paths to files and index of this file and class. This information is then used in semi-/hard negative mining. All fragments in datasets undergo custom augmentations, like shown in table 3.

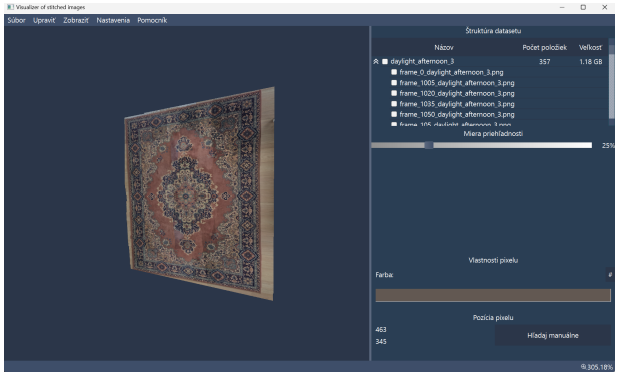

Figure 2: GUI application for visualising the homography of frames in a large image. This application is able to show the frames of the dataset in their correct positions.

Custom augmentations	
translation	$\pm 25px$
rotation	$\pm 180°$
homography	$\pm 25°$
center crop	224×224 pixels

4 Image stitching

To locate a query image in a large image, the first step is to retrieve the large image (map). The images for this map were created during the dataset creation process, explained in Section 3, specifically in the sampling section. This map is created using the image stitching technique, which involves the following steps:

1. **Keypoint Detection and Matching** – To detect the key points in each image, algorithm SIFT [13] is used. This is followed by the FLANN [16] key point matching algorithm, which allows the identification of correspondences between key points in different images.
2. **Homography Estimation** – RANSAC [7] algorithm is used to estimate the homography between pairs of images. Homography maps points in one image to corresponding points in another image.
3. **Image Warping and Blending** – Images are transformed to align them for stitching once homography has been estimated. Then, multiband image blending [27] is used to create one seamless large image.
4. **Extraction of Largest Inner Rectangle** – identifies and extracts the largest possible rectangle within the stitched image.

Numerous experiments have been conducted using various subdatasets to optimize image blending quality. Although most of the process has produced satisfactory results, a few minor artefacts require further investigation for refinement.

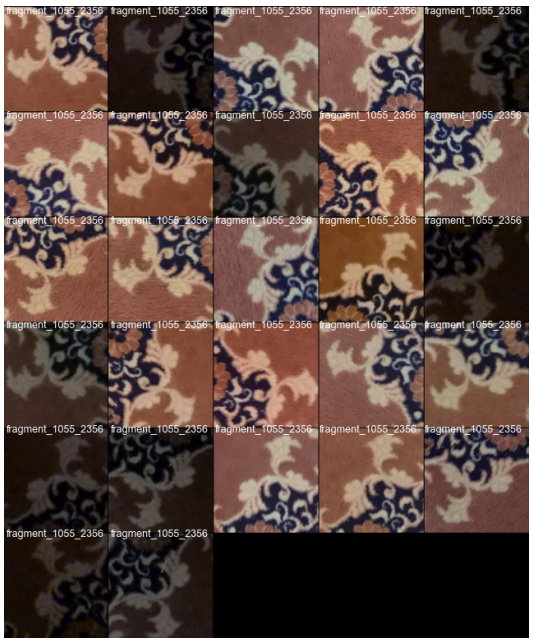

Figure 3: Examples of fragments from the same class.

5 Models architecture

The architecture of the models used in this study is based on the ResNet50 [9] model. The models are as follows:

- **Classification model** – The first model uses ResNet50 for classification tasks. The final layer is a fully connected layer with a softmax activation function, which outputs the probabilities for each class. In addition to the class probabilities, the model also returns the embeddings from the forward method. These embeddings are the output of the layer before the final fully connected layer. They represent high-dimensional learned features of the input data that the model uses for classification.

Model details for classification model	
Dimensions of embeddings	2048
Linear layers – carpet	**in**=2048, **out**=231
Linear layers – all	**in**=2048, **out**=914
Trainable parameters	24108389

- **Triplet model** – The second model also uses ResNet50, but it's trained with a triplet loss function. A triplet consists of an anchor, a positive, and a negative sample. The batch of data is used to extract the anchor, positive, and negative images. The anchor and positive images belong to the same class, while the negative image belongs to a different class. The model is used to obtain embeddings, which are vector representations of the images. Finally, the distances between the anchor and positive embeddings, and the anchor and negative embeddings, are calculated using a distance function. The model should select either hard or semi-hard negatives based on the chosen method. The valid triplets' embeddings are then selected for further processing. The calculation of the triplet loss involves the chosen embeddings. Finally, the loss is backpropagated and the model parameters are updated. The purpose of this design is to learn embeddings in such a way that the distance between an anchor image and a positive image (belonging to the same class) in the embedding space is smaller than the distance between the anchor image and a negative image (belonging to a different class).

Model details for triplet model	
Dimensions of embeddings	512
Linear layers – carpet / all	**in**=2048 **out**=512
Trainable parameters	24675111

6 Large image embedding database

A database of embeddings is created from the large image by traversing the entire image with a specified step in pixels and extracting the image to obtain its embedding. Currently, a step of 5 pixels in both axis directions is used. The created embeddings are then utilised to generate a Feiss index with the Feiss [6] library. In addition, the JSON file also stores the bounding box position of each index used in the matching process. Although this process is computationally and time-consuming, it is only performed once for a single large image.

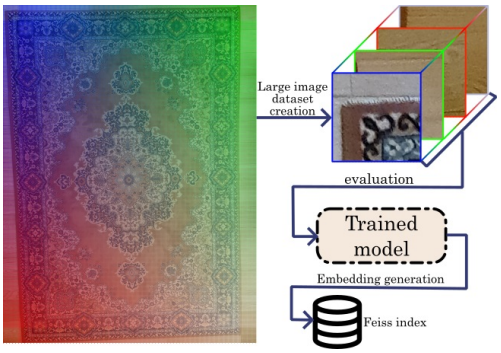

Figure 4: Feiss index creation process.

7 Localization

The final stage of the localisation pipeline is the localisation process. Its purpose is to identify a specific image within a larger one. The process begins by dividing the query image into smaller sub-queries, each of the same size as the inputs to our CNN model. If the image is large

Figure 5: Image showing all valid hypotheses for image patches, where small solid coloured squares are patches in the query image, dotted squares represent potential patch positions and the solid coloured squares in the large image represent the K-nearest embedding patch. The solid red rectangle represents the resulting homography without optimisations.

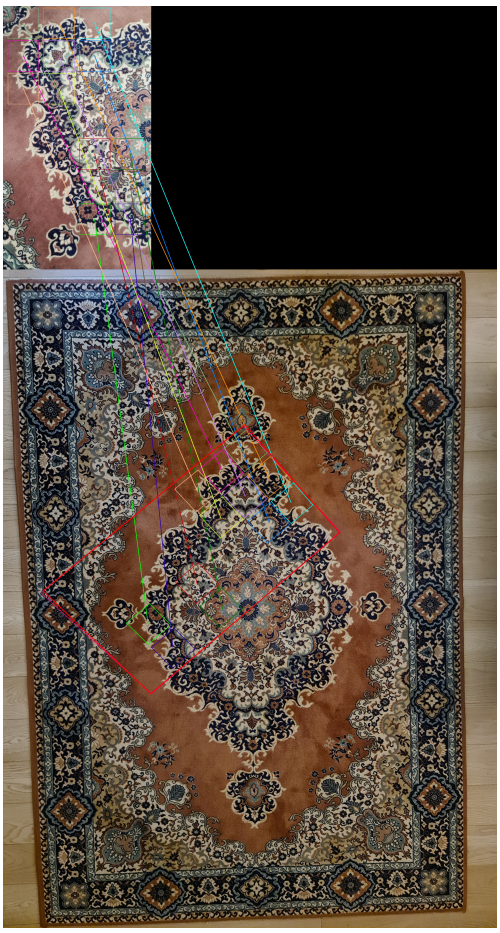

Figure 6: Result of the localisation. The red line is the original homography, while the pink line is the refined one. The solid squares are the sub-query inliers. The dotted ones are their localised potential correspondences.

enough to contain four non-overlapping sub-queries, the non-overlapping method is selected. This method divides the space between all the sub-queries equally. Otherwise, a method is selected where the sub-queries may overlap. It is important to ensure that the query image is smaller or at worst the same size as the large image to avoid computing homographies where the large image is inside a black area, resulting in missing information. For each sub-query, embeddings are extracted using one of the trained models. These embeddings capture the essential features of the sub-query. They are then used to search our database of embeddings from the large image. The user chooses K-nearest subquery embeddings to query the database. This allows us to find the most similar embeddings in a large index of FAISS large image embeddings for each sub-query. A random sampling algorithm with neighbourhood suppression is used to determine the best homography model based on the positions of four sub-query images and their corresponding positions in the large image. This generates a given number of homography hypotheses. Invalid homography hypotheses are marked to make the process faster, including those with unwanted properties such as area change, angle change, and scale ratio change. The cosine distance is computed between the sub-query image embeddings and the potential homography hypothesis embeddings. The harmonic mean of all distances is then calculated to evaluate the hypotheses. This process is repeated for all valid hypotheses, and the homography hypothesis with the lowest global distance is selected as the best model. This homography represents the location of the query photo within the large image. The homography is refined by applying LK optical flow to the best hypothesis. Inliers are searched for within the potential sub-query centers. The potential midpoints and their corresponding closest real point are found using the second Feiss index with midpoints, this time with L2 distance. This process allows for the computation of a new refined midpoint position for each inlier sub-query, which can be used to compute a better homography. Homography refinement can

be performed multiple times to improve accuracy. If the new homography has a better distance, it will be chosen as the new one. Otherwise, the original homography will be selected. Despite the best possible training of the neural network, there are still inconsistencies that cannot be corrected even with a higher number of nearest neighbours. Further work will aim to improve the localisation model to avoid such inconsistencies.

8 Results

For clarity, the results are illustrated with figures and tables. All results are based on test data sets. Homographies were mainly tested with a these metric techniques: Average Corner Error [5] and Point Matching Error [14]. The main problem with them is that the points where the homographies are given have to be given manually.

Percentage accuracy of models					
classification model			**triplet loss model**		
train	**val**	**test**	**train**	**val**	**test**
99.21	97.78	97.71	98.72	95.35	94.11

Table 1: Accuracy of the trained models for each dataset.

query:(899x1599) map:(4800x3600)	This solution	SIFT + RANSAC
Create index database (once per new map)	9-21h (size dep.)	0s
Processing time (once for all of images)	7.9s	0s
Query time (for every query)	5,131s	0s
Localization (for every query)	5.471s	111.7s

Table 2: Average time in each part of localization pipeline.

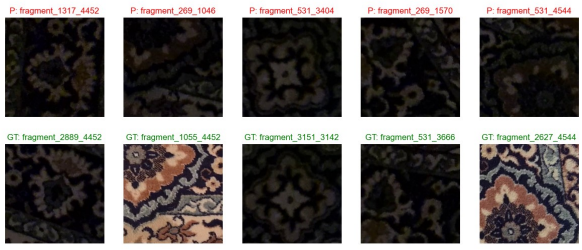

Figure 7: Examples of 10 misclassified images (CE). The two numbers represent the midpoint (x,y) of that particular fragment class.

9 Conclusions

In conclusion, this research presents a novel method for locating a specific photograph within a very large photograph. This method uses a convolutional neural network to extract embeddings from the query image, which are then used to perform an approximate search within a database of embeddings from the large photo. The results of this research demonstrate the effectiveness of this method, which is comparable to state-of-the-art localisation methods.

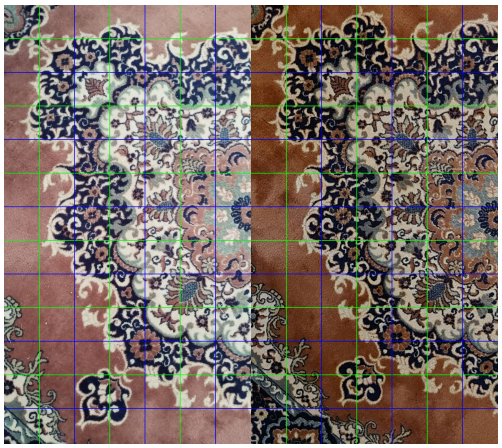

Figure 8: Query image next to the localised part of the large image after inverse homography transformation.

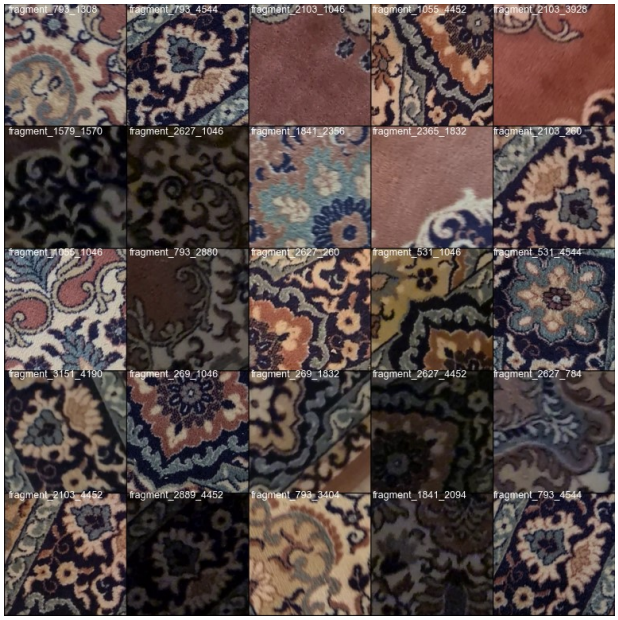

Figure 9: Examples of 25 images from testing dataset.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly super-

vised place recognition, 2016.

[2] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[3] J. Cao, L. Liu, P. Wang, Z. Huang, C. Shen, and H. T. Shen. Where to focus: Query adaptive matching for instance retrieval using convolutional feature maps. *CoRR*, abs/1606.06811, 2016.

[4] S.-Y. Cao, R. Zhang, L. Luo, B. Yu, Z. Sheng, J. Li, and H.-L. Shen. Recurrent homography estimation using homography-guided image warping and focus transformer. In *2023 IEEE/CVF Conf. on Comp. Vis. and Pattern Rec. (CVPR)*, pages 9833–9842, 2023.

[5] D. DeTone, T. Malisiewicz, and A. Rabinovich. Deep image homography estimation, 2016.

[6] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The faiss library, 2024.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.

[8] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. *CoRR*, abs/1403.1840, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[10] B. Hou, J. Ren, and W. Yan. Unsupervised multi-scale-stage content-aware homography estimation. *Electronics*, 12(9), 2023.

[11] P. V. Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654.

[12] Y. Liu, Y. Guo, S. Wu, and M. Lew. Deepindex for accurate and efficient image retrieval. In *ACM International Conference on Multimedia Retrieval (ICMR)*, pages 43–50, Shanghai, China, 06 2015.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, nov 2004.

[14] Y. Luo, X. Wang, Y. Wu, and C. Shu. Detail-aware deep homography estimation for infrared and visible image. *Electronics*, 11(24), 2022.

[15] E. Mohedano, A. Salvador, K. McGuinness, F. Marqués, N. E. O'Connor, and X. G. i Nieto. Bags of local convolutional features for scalable instance search. *Proceedings of the 2016 ACM on Int. Conf. on Multimedia Ret.*, 2016.

[16] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Int. Conf. on Comp. Vis. Theory and Applications*, 2009.

[17] J. Y.-H. Ng, F. Yang, and L. S. Davis. Exploiting local features from deep networks for image retrieval, 2015.

[18] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[19] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks, 2016.

[20] P. Rousseeuw. Least median of squares regression. *Journal of The American Statistical Association - J AMER STATIST ASSN*, 79:871–880, 12 1984.

[21] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. *CoRR*, abs/1911.11763, 2019.

[22] I. Suárez, G. Sfeir, J. M. Buenaposada, and L. Baumela. Beblid: Boosted efficient binary local image descriptor. *Pattern Recognition Letters*, 133:366–372, 2020.

[23] F. Wang, W.-L. Zhao, C.-W. Ngo, and B. Merialdo. A hamming embedding kernel with informative bag-of-visual words for video semantic indexing. *ACM Trans. Multimedia Comput. Commun. Appl.*, 10(3), apr 2014.

[24] Q. Wang, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching, 2022.

[25] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform, 2016.

[26] J. Zhang, Y. Cao, and Q. Wu. Vector of locally and adaptively aggregated descriptors for image feature representation. *Pattern Recognition*, 116:107952, 2021.

[27] Y. Zhao, W. Qian, and D. Xu. Fast multi-band blending using run-length encoding. In *2015 14th Int. Conf. on Computer-Aided Design and Comp. Graph. (CAD/Graphics)*, pages 224–225, 2015.

[28] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian. Good practice in cnn feature transfer, 2016.

[29] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 391–405, Cham, 2014. Springer International Publishing.