Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-020-05146-6

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Dynamic Time Warping of Deep Features for Place Recognition in Visually Varying Conditions

Saed Alqaraleh[1] · A. H. Abdul Hafez[1] · Ammar Tello[1]

Received: 28 April 2020 / Accepted: 11 November 2020

Abstract
This paper presents a new visual place recognition (VPR) method based on dynamic time warping (DTW) and deep convolutional neural network. The proposal considers visual place recognition in environments that exhibit changes in several visual conditions like appearance and viewpoint changes. The proposed VPR method belongs to the sequence matching category, i.e., it utilizes the sequence-to-sequence image matching to recognize the best matching to the current test image. This approach extracts the image's features from a deep CNN, where different layers of a two selected CNNs are investigated and the best performing layer along with the DTW is identified. Also, the performance of the deep features is compared to the one of classical features (handcrafted features like SIFT, HOG and LDB). Our experiments also compare the performance with other state-of-the-art visual place recognition algorithms, *Holistic*, *Only look once*, *NetVLAD* and *SeqSLAM* in particular.

Keywords CNN · Deep features · Dynamic time warping · Image sequence matching · Visual place recognition

1 Introduction

Visual place recognition (VPR) refers to how the robot can localize itself using only a visual input of a revisited place. In the last decade, the (VPR) or what called visual localization [1] received significant attention by the research community due to the importance of this task in the robotic field especially for autonomous robots and self-driving cars.

It is considered as a challenging problem as appearance can change for the same place over seasons and from day to night and even changes to the place itself, also the variation in viewpoint when the same place revisited again is a big challenge [2–5]. Even though there are other methods exist for localization task, like Global Positioning System (GPS)-based methods, VPR is still preferable due to the significant information that can be retrieved from images and also because of the lack of GPS info in terms of occlusion and absence of the signal. Overall, there are several works in the literature that approach the visual place recognition (VPR) problem as a content-based image retrieval problem (CBIR) [6,7].

✉ Saed Alqaraleh
saed.alqaraleh@hku.edu.tr

[1] Computer Engineering Department, Hasan Kalyoncu University, Gaziantep, Turkey

The main components of VPR systems, as described in [1] and depicted in Fig. 1, are as follows: (1) visual map: which is represented by the images of the visited place or generally the environment, and these images are considered as references, while the new coming images are called test images; (2) feature extraction: In this step, each image is represented by a descriptor that is formulated to find the most important representatives inside the image; and (3) localization: This component is responsible for finding the best matches between reference and test images, so, the robot can localize itself according to the place that the matched reference image referred to. Other components like visual perception, motion estimation and decision, i.e., the output of the system, can be available too.

The significant improvements in the visual localization topic lead to increasing the attention of the robotics community to the topic [1,2,4,8]. Furthermore, over the last decade, researchers are working on adapting image processing techniques, especially the deep learning-based features extracted from convolutional neural networks for improving the VPR systems. This is due to the fact that CNN was able to outperform the other state-of-the-art methods in many of the image retrieval tasks. It is worth mentioning that the last layer in a deep CNN is nothing but a classification layer that produces a distribution over the class labels. It is common these days to use layers known as Softmax layer, SVM layer as a classi-

Published online: 07 January 2021

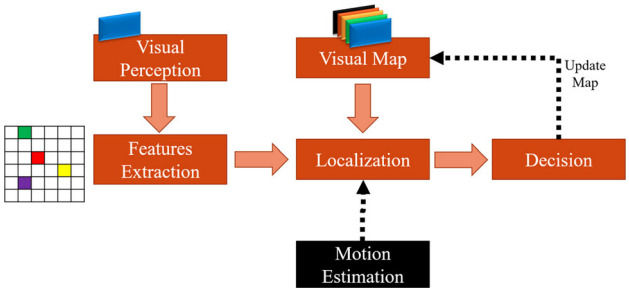

Fig. 1 Visual place recognition schematic diagram [1]

fier. An evidence was presented in [9,10] that SVM achieves a bit improvement as compared with the performance of the Softmax.

Dynamic time warping is known to be an efficient sequence alignment algorithm for recognition and matching applications [11,12]. Our work presented in [13] is the first to investigate the possibility of using the DTW as a classifier for recognition (decision-making), for VPR proposes. The work in [13] has investigated the DTW with handcrafted descriptors: SIFT, HOG and LDB. This work claims that the plain DTW in its simple form can outperform other state-of-the-art complex visual place recognition algorithms. Their results showed significant performance improvement as compared to the individual performance of the mentioned descriptors and the ABLE methods. Later, Lu et al. have presented in [14] a sequence place recognition using an improved DTW.

In this work, we propose to extend our work in Ref. [13] to utilize the deep features extracted from a CNN. This is done by using the DTW matching step to recognize the place corresponding to the query image instead of using the *Softmax* classifier as the last layer of the CNN. The DTW is used to align the sequences of the extracted feature maps for both test and reference images. Then, it works on defining an optimal path of matches for the two sequences. As a result, the robot will have the ability to localize itself according to the place referred by the matched reference image. It is worth mentioning that there is one category of place recognition that is based on a single image-to-image matching and another one that is a sequence-to-sequence image matching. We follow the latter paradigm in this paper.

The main contribution of the work presented in this paper is a new VPR algorithm that utilizes DTW of deep features for recognition in visually changing environments that show changes in illumination and pose for example. The feature maps are extracted from a selected convolution layer after applying the test and the reference image sequences as an input to the network. Then, the DTW algorithm aligns the two resulted sequences of features by matching each test image to an image from the reference sequence. Different layers from the very deep convolutional networks for large-scale image recognition (VGG-16) and residual network (ResNet50) are explored to identify the layers that are best performing with the DTW algorithm for visual place recognition.

The remaining of this paper is organized as follows: Sect. 2 presents related works and literature review. An overview of the whole algorithm and then the detailed steps are presented in Sect. 3.

The experimental evaluation and analysis are presented in Sect. 4. Finally, conclusions and future works are given and presented in Sect. 5.

2 Related Works and Literature Review

In the following, we have summarized recent researches, developments and solutions related to the proposed method.

For quite a long period, the handcrafted features were the main and state-of-the-art methods for building efficient VPR systems [8,15,16]. Up to now, FAB-MAP [16] and SeqSLAM [15] are considered as the state-of-the-art handcraft-based VPR systems. In general, FAB-MAP uses the SURF descriptor to extract the image features. Then, the extracted features are encoded using the BoW. On the other hand, and unlike most of the existing approaches that use the image similarity, SeqSLAM uses image differences for matching. In addition, in order to solve the localization problem, SeqSLAM searches for all possible matches in the visual map.

However, with the impressive improvements achieved by CNN models in many fields such as image classification [17,18] the CNN-based VPR approaches [19–22] were able to outperform the existing handcraft-based VPR ones. Related to deep features, we would like to point out that as shown in the extensive experimental work of Refs. [17,18,23], the fully connected layers outperform other layers in image classification tasks, while the convolutional layers are the best choice for image retrieval and visual place recognition tasks [17,18,23]. In such situation, the last CNN's layer is usually used as the classifier or decision-maker. For instance, the work in Ref. [9] suggested that using the output of the last fully connected layer (the 3rd) and the SVM classifier as an output layer improves the performance of CBIR. On the other hand, in Ref. [20], a multi-scale feature encoding method has been employed to generate features that can overcome the condition and viewpoint changes. Two CNN architectures were used and trained for the place recognition task in Refs. [20]. Overall, they achieved promising results using a model consisting of six convolutional layers followed by two fully connected layers, where the Softmax is used as an output layer.

As an alternative to the Softmax and SVM layers, the following two approaches use the cosine matrix [22,24]. The work proposed in Ref. [24] uses the image pixels as input for a CNN model, and the features vector is formulated from

the output of the pooling layer next to the fifth convolutional layer. Then, the performance of multiple pooling techniques such as max and average has been tested. As a result, the performance has been improved by using the hybrid pooling, which done by creating a vector that combines the output of both max and average pooling techniques. Finally, the cosine distance between the proposed representation of the query image and the references is calculated to find its most similar ones.

The work in Ref. [25] proposes an omnidirectional convolutional neural network (O-CNN), which tries to: (1) retrieve the closest place and (2) estimate the distance between the input (test image) and the closest place. This process is done by comparing directly the feature distances from the test image to all stored reference images.

In [26], a new approach that works on extracting the output of the max pooling of all convolutional layers in the VGG-16 was introduced. The second step in this approach is to combine all the extract vectors into one vector which represents the image's features. Finally, in order to find the most similar reference image, the produced vector of the test image features is fed into a visual similarity neural network that finds the similarity score with all reference images.

Another approach, named as Holistic visual place recognition, was presented in Ref. [9]. This approach obtains the feature maps' activations that will be used to identify the candidate of prominent regions (salient regions). Then, these regions are encoded using vector of locally aggregated descriptors (VLAD). This approach uses the cosine matrix instead of Softmax and SVM, where the matching is performed for each test image against all the reference images to select its best-matched reference image. In addition, the approach of [27], known as Only look once, that aims to overcome the viewpoint challenge uses the BOW [28] to encode the image's landmarks extracted from the output of the convolutional layers of a pre-trained VGG-16 deep neural network. In more detail, this approach tries to detect the most promising landmarks from the output of the convolutional layers of a pre-trained VGG-16 deep neural network. Then, the extracted features are fed into bag of words (BOW) to be encoded. Finally, the distance between images is calculated by cosine similarity to find the mutual matching of regions in the images. Overall, the investigation study of both approaches presented in Refs [22,27] showed their ability to achieve state-of-the-art performance.

On the other hand, in [29], a new approach named CoHOG, which is based on the handcrafted "HOG" descriptor, was developed. In more detail, this approach tries to eliminate the need to train a CNN model to extract the image's region of interest, by using the image entropy to identify its regions of interest; then, the HOG descriptor is created for each of the detected regions. Finally, this approach uses the max pooling to find the best-matched candidate region in the reference images for each of the query regions of interest, i.e., the query image is matched with all reference images and the reference image with the highest overall matching score is selected as the best match.

Unlike all existing approaches, the developed algorithm employs the DTW algorithm as a classifier for the VPR system, which significantly improves the overall performance and speeds up the process. In addition, in order to improve the performance, approaches such as Only look once [27], Holistic [22] and CoHOG [29] suggested extracting the region of interests then encode them using either BOW or VLAD, while other approaches such as [26] work on extracting the features of all the convolutional layers; our system, thanks to the powerful performance of the DTW, has the ability to outperform the mentioned approaches, while directly using the output of one convolutional layer, which is significantly less computational cost.

In addition to the aforementioned contributions, critical points lead us to select the DTW, as it is more efficient in handling the outlier points and dealing with the different length sequences as compared to the Euclidean distance when both are used for measuring the similarity between two of one-dimensional series [30]. Such advantages encourage us to use DTW for the real-world VPR scenarios. For instance, DTW can overcome situations when the length of the reference images sequence is not equal to the length of the test sequence. This may happen if the test sequence is collected through different frequencies in comparison with the reference, which leads to having the test and reference sequences unaligned. Also, when some abnormal images are collected in the test phase, the DTW is still able to detect such an image and get back to the right path.

3 Visual Place Recognition by Deep Features and DTW

The proposed place recognition method has been built based on the assumption that the visual localization (or place recognition) problem is considered as an image matching problem. The proposed method employs DTW and deep features to achieve the localization (recognition) task.

The principal concept is that the reference set of images are presented as an input to the CNN. The feature maps are collected from a selected layer and stored for later matching with the test. As soon as a test sequence of images is available, it is presented to the input of the network, and the corresponding feature maps are collected consequently. We employ the concept of learning transfer in order to produce our features. We use two different CNNs in our experiments, they are VGG-16 [31] and ResNet50 [32]. Both of them are initially trained using the famous ImageNet dataset [33]. We use the same trained VGG and ResNet architectures without

any retraining or fine-tuning. In fact, we are interested in the visual features that are involved in the middle layers, while the later fully connected and classification layers are discarded.

The localization component of the proposed method takes the preprocessed input stream of visual data and the visual map to generate a belief on the current place. This is done by filling out the similarity matrix between the reference and test sequences as shown in Algorithm 1. The matrix C represents the cumulative similarity, i.e., the sum of the similarity between the current two images being matched, and the maximum of the cumulative similarity of the neighboring images is calculated as well. Then, the optimal path, which is the path consisted of elements from matrix C that has a maximum sum of cost values $C(i, j)$ is estimated using Algorithm 2. This path can be found by tracing backward in matrix C choosing the previous elements with the highest cumulative similarity. Finally, the system has a decision on whether it is a prior visited place or a new place. As an example, Fig. 2 shows the output images using different kinds of features corresponding to the same input image.

In the following subsections, we discuss the DTW of a sequence of image feature vectors, the deep feature extraction step and the matching criterion used by the matching process. These steps are depicted in Fig. 3 as well.

3.1 Image Sequence Alignment Using DTW

This section presents the DTW-based image sequence matching. Let us have a test image sequence Y and the priori annotated reference sequence X. Here, the reference sequence is given as

$$X = [x_1, x_2, x_3, \ldots, x_n],$$

and the test sequence is given as

$$Y = [y_1, y_2, y_3, \ldots, y_m].$$

Hence, n is the number of reference images and m is the number of test images and their features vectors are represented by

$$A_x = [Ax_1, Ax_2, Ax_3, \ldots, Ax_n]$$

and

$$A_y = [Ay_1, Ay_2, Ay_3, \ldots, Ay_m],$$

respectively. DTW-based image sequence matching can be formulated as to construct a warp path

$$W = [w_1, \ldots, w_l, \ldots, w_L]$$

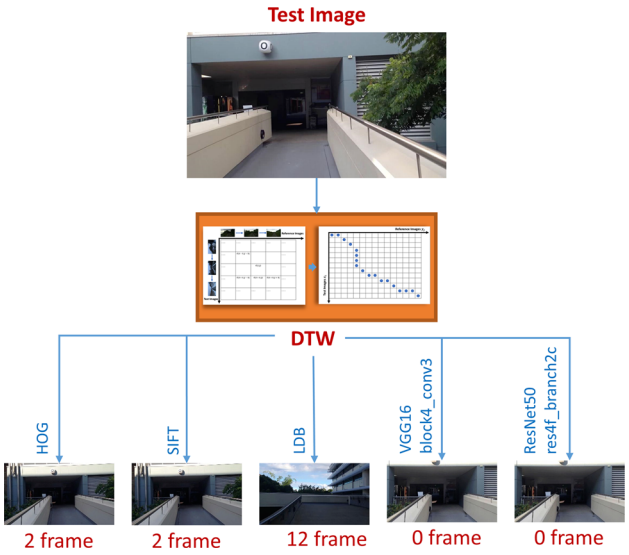

Fig. 2 Example on visual place recognition using DTW. The output of the algorithm is given using different type of features along with the difference from the exact match given in red

where L is the length of the warp path. The lth element of this path is $w_l = (i, j)$. Of course, i and j are the indices of the image sequences X and Y, respectively. It is worth noticing here that the length L of the warp path satisfies the following inequality $\max(n, m) \leq L \leq n + m, and$ it also starts at $w_l = (1, 1)$ and ends at $w_L = (n, m)$. For a certain point $w_l = (i, j)$ of this path, the image feature vector Ax_i matches the image feature Ay_j. An example of such a path is depicted in the "Optimal Path" part of Fig. 3. It is well common in the DTW literature to use a distance metric to measure how much an image descriptor vector Ax_i is close to an image descriptor vector Ax_i, like the cosine distance $Dist(i, j) = 1-\cos(Ax_i, Ay_j)$. Other metrics like city block and the Euclidean distance metrics are also used. The path W then represents a set of matches whose sum of distances $Q_D(W) = \sum_{l=1}^{L} Dist(i_l, j_l)$ is minimal. Notice that the length L of the path is independent on the sum $Q_D(W)$ of the distances involved in the path.

We observed in our VPR applications that measuring the similarity between the image feature vectors produce better performance than using distance function. We use in this paper the cosine similarity function given as

$$S(i, j) = cos(Ax_i, Ay_j) = \frac{{Ax_i}^T \cdot Ay_j}{\|Ax_i\| \|Ay_j\|} \quad (1)$$

It is empirically observed that the similarity function exhibits more discrimination capabilities, considering the higher performance and efficacy when used with deep features [13,21,22,24]. The reason for that is that distance functions are defined to have positive values that make it suitable to be used with vectors of features like HOG and SIFT whose

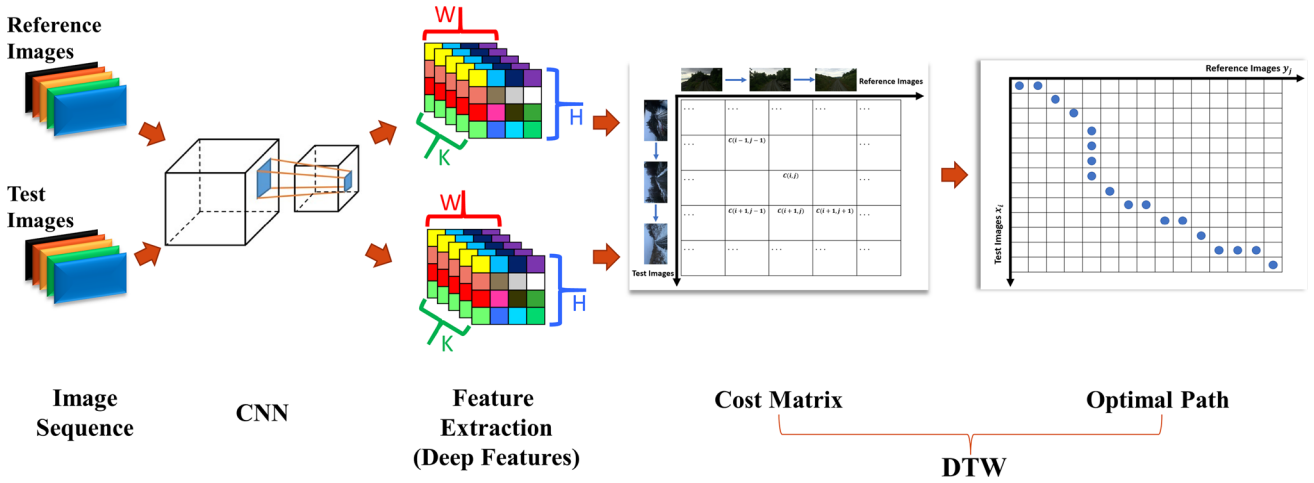

Fig. 3 Visual place recognition using DTW. In this proposed system, features are extracted from the test and reference images through a deep CNN. After that, the alignment of both set of images is done through calculating a cost matrix and finding an optimal path through this cost matrix

descriptor vector has only positive values. In contrast, deep features are represented using descriptors that contain positive and negative values. This reduces the discriminative property of the features. In addition, the cosine similarity function produces values in the range $[-1, +1]$ that are more expressive of the discrimination capabilities of the deep features.

The similarity value between the image y_j from the test sequence and the image x_i from the reference sequence is stored in S(i,j), whereas as mentioned before Ax_i and Ay_j refer to the descriptors of the reference and test images, respectively, and $\|Ax_i\|$ and $\|Ay_j\|$ refer to the magnitude of the descriptors.

In classical DTW, the distance measure is minimized while searching for the best warp path. This means images with minimum inter-distance are matched. This, of course, means that these two images are the most similar. This is reflected in our case to maximize the similarity measure, since the higher the similarity means the better the match. We modified the classical DTW to find the maximum similarity instead of finding the minimum distance. However, it is shown in [34] that minimizing the sum of distances is equivalent to maximizing the sum of similarities and produce the same warp path W. Indeed, the optimal path can be interpreted as maximizing the sum of the similarities values between the image descriptor vectors Ax_i and Ax_i involved in the path. This sum is denoted with respect to the path W as $Q_S(W)$, the function that represents the sum of similarities between the matched elements. The path W is defined as $W = w_l(i, j)$ where i and j are optimized in such a way that maximizes the sum of similarities between the matched image descriptor vectors that is

$$\max_{i,j} Q_S(W) = \sum_{l=1}^{L} S(i_l, j_l). \tag{2}$$

Finding such a path is in an exponential complexity considering the huge number of samples that could exist in the test and reference sequences. Utilizing the dynamic programming (DP) approach in solving this optimization problem reduces the complexity to $O(nm)$. DP is represented in DTW by the calculation of the cumulative matrix, where Eq. (3) is used to fill the cost matrix with the accumulated elements of the similarity matrix. The maximum value between the above left and upper left accumulated neighbors for each element is considered to be added as follows

$$C(i, j) = S(i, j) + \max \begin{cases} C(i-1, j), \\ C(i, j-1), \\ C(i-1, j-1), \end{cases} \tag{3}$$

Note that the similarity matrix is filled in using elements calculated according to Eq. (1).

The details about building the accumulated matrix are described in Algorithm 1. When the matrix C is filled out, DTW works on defining an optimal path of matches W, which is the result of backward tracing in the matrix C choosing the previous elements with the highest cumulative similarity, as shown in Algorithm 2.

To make a clear discrimination of our method that maximizes a similarity objective function from this DTW that minimizes a distance objective function, let us have further analysis on that. The length of the desired path is bounded by the lengths of the test and reference sequences, n and m, respectively. Indeed, the length cannot grow infinitely or toward larger value than the bound. DTW traces the accumulative matrix inversely starting from the match (n, m) toward the match $(1, 1)$. This is illustrated in Algorithm 1. It iter-

Algorithm 1 AccumulatedMatrix(X,Y,S)

```
n ← |X|      //number of reference images
m ← |Y|      //number of test images
C ← new  array[n * m]
C(1, 1) ← 0      //Fil the first element with 0
for i = 2; i ≤ m; i++ do
    C(i, 1) ← C(i − 1, 1) + S(i, 1)      //Fill the first column
end for
for j = 2; j ≤ n; j++ do
    C(1, j) ← C(1, j − 1) + S(1, j)      //Fill the first row
end for
for i = 2; i ≤ m; i++ do
    for j = 2; j ≤ n; j++ do
        C(i, j) ← S(i, j)+max(C(i−1, j), C(i, j−1), C(i−1, j−
        1))
    end for
end for

return C
```

Algorithm 2 BestPath(C)

```
path ← new  array[]
i ← rows(C)      //Assign number of rows in C into i
j ← columns(C)//Assign number of columns in C into j

while (i > 1) & (j > 1) do
    //Reach to the first row
    if i == 1 then
        //Iterate over columns only
        j = j − 1
        //Reach to the first column
    else if j == 1 then
        //Iterate over rows only
        i = i − 1
    else
        //The maximum element is on the same row
        if C(i−1, j) == max(C(i−1, j), C(i, j−1), C(i−1, j−1))
        then
            i = i − 1
            //The maximum element is on the same column
        else if C(i, j − 1) == max(C(i − 1, j), C(i, j − 1), C(i −
        1, j − 1)) then
            j = j − 1
            //The maximum element is on the diagonal
        else
            i = i − 1
            j = j − 1
        end if
        //Add the element to the path
        path.add(C(i,j))
    end if
end while

return path
```

ates over all possibilities of rows and columns to add a new match to the path which maximize the sum of similarities. The output of this algorithm is the best path with maximum total similarity. Longer path in the matrix does not necessarily have the larger cumulative similarity. The rationale behind the DTW is that if a sub-path has a maximum value of the objective function, then adding a new path segment with a maximum objective value produces a path with a maximum total objective value. In classical DTW works that minimize a distance function, the rationale works but talking about minimizing the objective values instead. In fact, it was shown in [34] that minimizing a distance objective function is equivalent to maximizing a probability objective function like the HMM case. It is worth also mentioning that the distance function $Dist(Ax, Ay) = 1 - cos(Ax_i, Ay_j)$ has a local minimum and the function $S(i, j) = cos(Ax_i, Ay_j)$ has local maximum at the same point (Ax, Ay); this is from a mathematical point of view. This actually applicable as well to maximizing a similarity objective function. For more details about DTW algorithm, the reader is referred to [11,12,34,35].

3.2 Image Representation

For the current work, the features vectors Ax_i and Ay_j are considered as handcrafted features and deep CNN-based features.

The image is represented using the output of a specific Layer from a CNN model. Layers from the VGG-16 and the ResNet50 networks [31,32] have been considered in this work. The deep features are detailed in the next subsection.

Related to the handcrafted features, they represent the image using the information present in itself [36]. SIFT, HOG and LDB, which are commonly used [37–39], are examples of efficient and frequently used handcrafted descriptors. In more detail, SIFT firstly works on detecting the image key points. Then, the appearance of the extracted key points is characterized using a 3-D spatial histogram to produce the image descriptor. In addition, HOG mainly decomposes the image into small squared cells. Then, for each cell, a histogram of oriented gradients is computed, and the result is normalized using a blockwise pattern to produce a descriptor for that cell. Furthermore, LDB uses simple intensity and gradient difference tests on pairwise grid cells for each of the image's patches to produce a binary string as the image descriptor. Also, LDB applies multiple gridding strategies to capture the distinct patterns of the patch at different spatial granularities.

3.3 Features from Deep CNNs

CNN composed of a number of layers such as convolution, pooling, ReLU and fully connected layers. The section name 'experiments' is mentioned in the caption of Fig. 4, but there is no such section heading in the manuscript. Please check.Convolution layer detects local features from the previous layer and maps them to the next layer. The pooling layer is responsible for reducing the size of the activation maps. The ReLU layer aims to combine nonlinearity and rectifica-

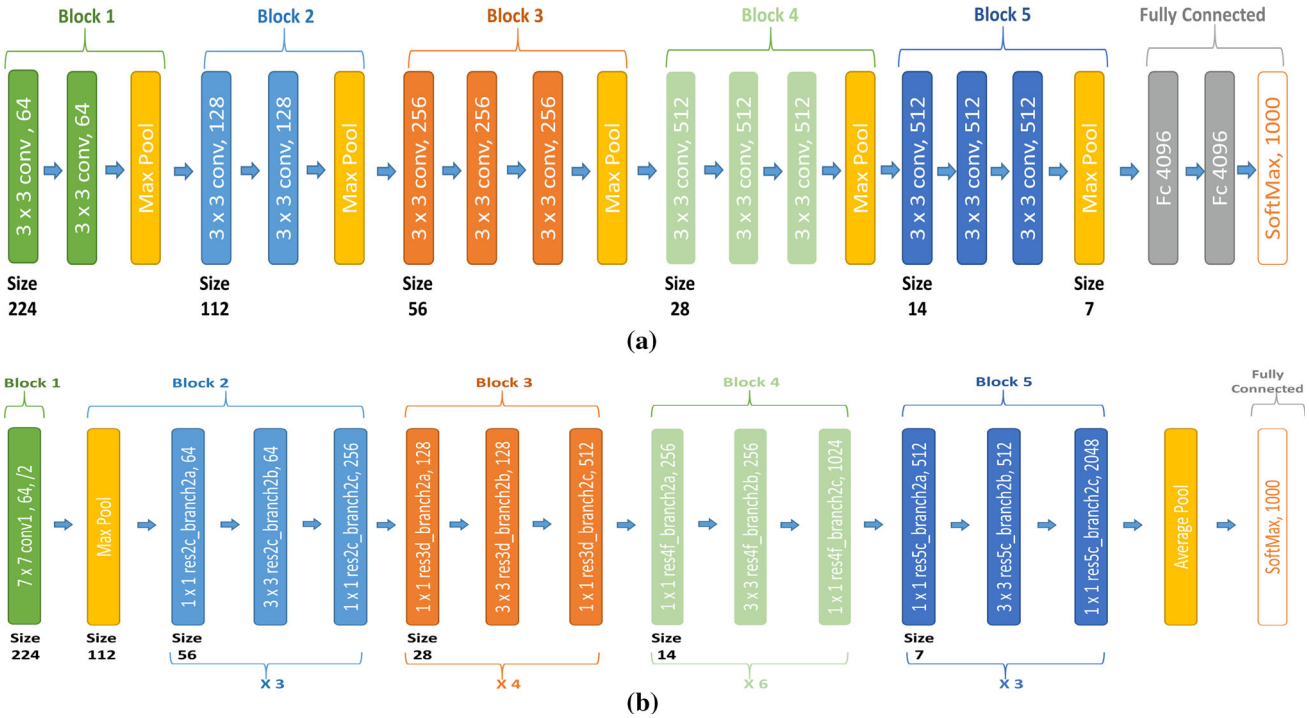

Fig. 4 **a** A visualization of the VGG-16 architecture. The network consists of five convolutional blocks and three FC layers with a final Softmax classifier. **b** A summarized visualization of ResNet50 with five blocks where each block consists of three layers repeated multiple times. Feature vector with the best performance. P–R curves resulted from this investigation are shown in Figs. 6 and 7 from the experiments section

tion layers. Nowadays, several CNN models are considered as a good choice for improving any image retrieval systems.

AlexNet [40] was one of the first deep networks that defeat classification traditional methodologies. In general, it consists of 5 convolutional layers followed by 3 fully connected layers. VGGNet [31] consists of 16 layers, i.e., 13 convolutional layers followed by 3 fully connected layers; it is considered as one of the most preferred choices for extracting features from images. In addition, the weight configuration of the VGGNet is publicly available and has been used in many other applications. GoogleNet [41] consists of a 22 layers. Although it is much deeper, it has significantly reduced the number of used parameters by using several very small convolutions. This in turns leads to reducing the number of parameters to 4 million. It has achieved performance close to the human level. More recently, ResNet [32] has been proposed with a novel architecture based on skip connections and features. Heavy batch normalization was introduced. The system can be trained using 152 layers while still having lower complexity than other models. In this work, we have used the VGG-16 network [31] and the ResNet50 [32].

The VGG-16 network is shown in Fig. 4a, it consists of five convolutional blocks and three fully connected (FC) layers with a final Softmax classifier. In other words, VGG-16 can be considered as an improved version of the AlexNet that replaced the large kernel-sized filters with multiple sequential smaller kernel-sized filters.

The input image is set to a fixed size of 224 × 224 RGB, Fig. 4a. Then, after passing the image through first and second convolutional layers, known as "block1_conv1" and "block1_conv2," respectively, each of which has 64 filters of size 3 × 3 and applied with the stride of the pooling set to 1. The dimensions of the produced features at these layers are $224 \times 224 \times 64$. Then the maximum pooling layer or subsampling layer reduces the image dimensions to $112 \times 112 \times 128$. In the second block ("Block 2"), there are two convolutional layers, i.e., "block2_conv1" and "block2_conv2," with 128 filters of size 3 × 3 and a stride of 1, and its pooling layer has 256 feature maps that reduce the output to 56 × 56 × 256. In the third block, three convolutional layers ("block3_conv1," "block3_conv2" and "block3_conv3") are followed by a maximum pooling layer with filter size 3 × 3, a stride of 2, and have 512 feature maps. The following two blocks, i.e., "Block 4" and "Block 5," consist of 3 convolutional layers that each has 512 filters of size 3 × 3 and a stride of 1, followed by a maximum pooling layer which reduces the size to 7 × 7 × 512. The output of the last convolutional layer, named as "block5_conv3", is flattened through a fully connected layer with 25088 feature maps each of size 1 × 1,

where the other two fully connected layers have 4096 feature maps. Finally, there is a Softmax output layer.

On the other hand and as depicted in Fig. 4b, the ResNet50 model consists of five stages each has a convolution and identity blocks, where each of the convolution and the identity block has 3 convolution layers. In the case of using the Resnet, the input RGB image is also from size 224 × 224, Fig. 4b. The input image is subsampled into $112 \times 112 \times 64$ as an output of the first block with a kernel size 7×7; this output is fed into the next block, i.e., "Block 2" which contains two stages: the first one uses the max pooling with a kernel size of 3×3, and then, three convolutional layers with different parameters formulate the second stage where each of them is repeated three times, the first and second layers, each of which has 64 feature maps with a 1×1 kernel size for the first one and a 3×3 for the second. The third one has 256 feature maps and 1×1 kernel size. The output of the second block is pooled into $56 \times 56 \times 256$, and this output is fed into the next block which gives an output with a size of $28 \times 28 \times 512$. In this block, each of the three types of the convolutional layers is repeated 4 times, the first and second ones have 128 feature maps for each and the third one has a 512 feature maps, and the last layer in this block as depicted in "Fig 4b" is named as "res3d_branch2c." The fourth block gives an output with $14 \times 14 \times 1024$ size from its last layer "res4f_branch2c," where each convolutional layer in this is repeated 6 times, the first and second ones contain 256 feature maps and the last one has 1024 feature maps. The last convolutional block gives a $7 \times 7 \times 2048$ output with 512 feature maps for the first two convolutional layers and 2048 for the third one, which known as "res5c_branch2c," with 3 repetitions for each. The last layer is a decision layer that uses the Softmax function to represents the output.

It is worth mentioning that, for the first convolutional layer of the third, fourth and fifth blocks, the stride is set to 2. Hence, although the mentioned layers do not have pooling layer, the used stride value leads to decrease the dimensionality of the following layers to the half.

In this work, we derive the image representations from a convolutional layer, i.e., "res5c_branch2c," the layer from block 5 in the ResNet50 architecture. In addition, to ensure the robustness of the obtained result we have also used the layer from block 4 ("res4f_branch2c"). Both layers have shown superior performance with respect to other layers as shown in the experimental evaluation and analysis section.

4 Experimental Evaluation and Analysis

We present in this section the experiments that have been carried out to (1) investigate the proposed DTW place recognition method and (2) evaluate the performance of using multiple handcrafted features like SIFT, HOG and LDB, and deep features extracted from the VGG-16 and ResNet50 networks. We use the precision–recall curve (P–R curve) to evaluate the performance. In addition, multiple datasets like "Berlin_A100" [42], Nordland [43] and "Garden Point" [1] are used.

In our experiments, we firstly studied the efficacy of the DTW algorithm with handcrafted features, particularly SIFT, HOG and LDB. Then, we investigated both the performance of features extracted from different layers from VGG-16 [31] and ResNet50 [32] networks, and the effect of injecting the output of the studied layers into DTW to get the best matching images. As a result of this experiment, we detected the best layer that obtains the best performance when integrated with DTW for place recognition, and it was used in the remaining experiments. After that, we compare the performance of the DTW with SVM and Softmax using deep features. In the last two experiments, the performance of the proposed DTW algorithm was compared with the state-of-the-art approaches, i.e., *Holistic* [22], *Only look once* [27], *NetVLAD* [19] and *SeqSLAM* [15].

4.1 Datasets and Metrics of Evaluation

The "Berlin_A100" is a dataset collected from a platform called Mapillary where images of the same route were collected by different users with a variation in viewpoint and appearance. In this work, we have used the sub-dataset of "Berlin_A100" that was constructed by [22] where the reference set has consisted of 85 images and the test set consisted of 81 images. Note that the ground truth of this sub-dataset was made by matching the images which have the same position in terms of GPS. In addition, the "Garden Point" dataset is used in the evaluation process. It is a dataset that captures the changes in the pose and lighting conditions through the Garden point campus. Furthermore, the Nordland dataset[43] was also used. In general, it is a 40 h video record over the four seasons (10 h for each) that have been aligned frame to frame. In our work, 800 images from the summer are considered as the reference sequence, and their corresponding winter images are used as the test sequence.

The precision recall curve (P–R curve), area under the curve (AUC), F1 and error mean were used to evaluate the performance during the multiple experiments conducted in this work.

– The P–R curve is obtained by finding the frame from the training sequences that best match each test frame, and the formula to calculate the precision (P) is given as

[1] http://tinyurl.com/gardenspointdataset.

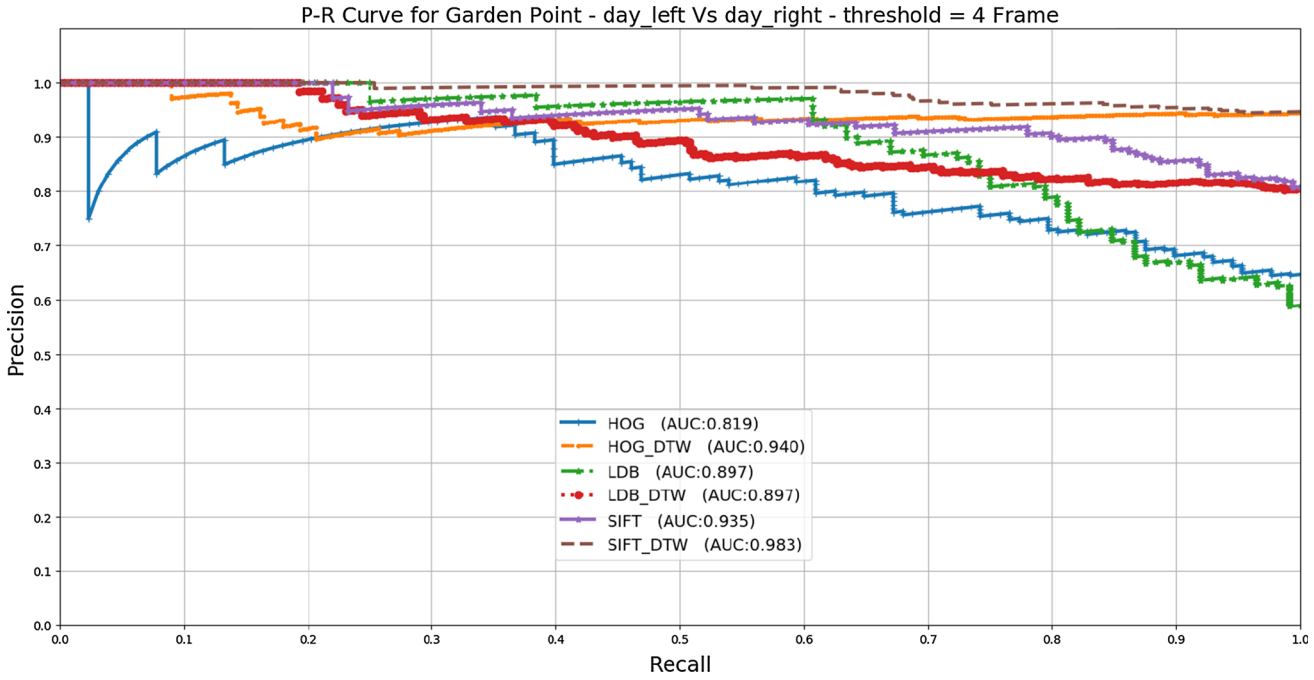

Fig. 5 Precision–recall curves comparing the matching performance of the DTW algorithm with direct matching using the cosine cost matrix. Direct matching is noted as HOG, SIFT and LDB, while curves resulted from using the DTW are noted as HOG_DTW, SIFT_DTW and LDB_DTW. Using DTW has clearly improved the matching performance

$$P = \frac{\text{TP}}{(TP + FP)} \tag{4}$$

while the formula to calculate the recall (R) is given as

$$R = \frac{TP}{(TP + FN)} \tag{5}$$

Here, TP stands for the number of matched images (true positives), FP refers to the number of queries matched with the wrong reference images (false positives), and FN represents the images classified as non-matched despite the fact they have corresponding images in the reference set (false negative).

– The $F1$ score is a weighted average of precision and recall, that considers both false positives and false negatives into account was also used as an evaluation matrix. The $F1$ is calculated using

$$F1 = 2 \times \frac{(P \times R)}{(P + R)}. \tag{6}$$

– The AUC can be calculated using the trapezoidal rule

$$AUC = \sum_{i=1}^{n-1} \frac{P_i^{min} + P_{i+1}^{max}}{2} (R_{i+1} - R_i) \tag{7}$$

where $P_i^{\min}$ is the minimum precision corresponding to R_i, $P_i^{\max}$ is the maximum precision corresponding to R_i and n is the considered number of recalls.

Note that every match between i and j frames is considered as a positive if the visual similarity $S(i, j)$, given in Eq. (1), is bigger than a threshold t. Otherwise, the match is considered as negative matches. It is worth mentioning here that to decide whether a match is true or false, we use the difference in the number of frames from the correct frame in the training sequence.

4.2 Dynamic Time Warping with Handcrafted Features

The proposed visual place recognition method has been initially tested for the performance with some well-known handcrafted descriptors, in particular, HOG, SIFT and LDB. The experiment has been initially conducted by matching the selected two sequences by directly initiating the matrix of the cosine distances, and the match with the minimum distance is selected(without DTW). Then, the experiment has been repeated by achieving the matching selection using the DTW algorithm. The P–R curve obtained from both experiments are depicted in Fig. 5. As a result, it is clear that using DTW has outperformed and improved the performance of all used handcrafted descriptors.

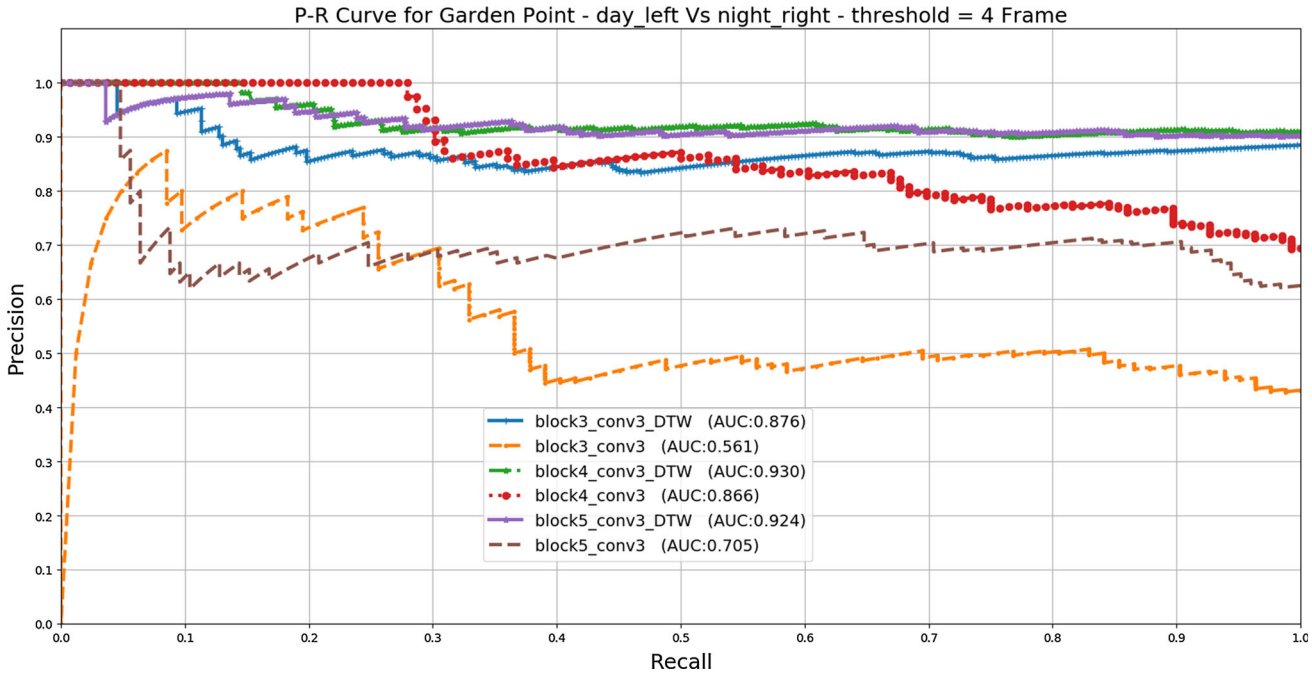

Fig. 6 Different precision–recall curves resulted from exploring features extracted from the different layers in the VGG-16 architecture. The layers "block4_conv3" and "block5_conv3" have better performance

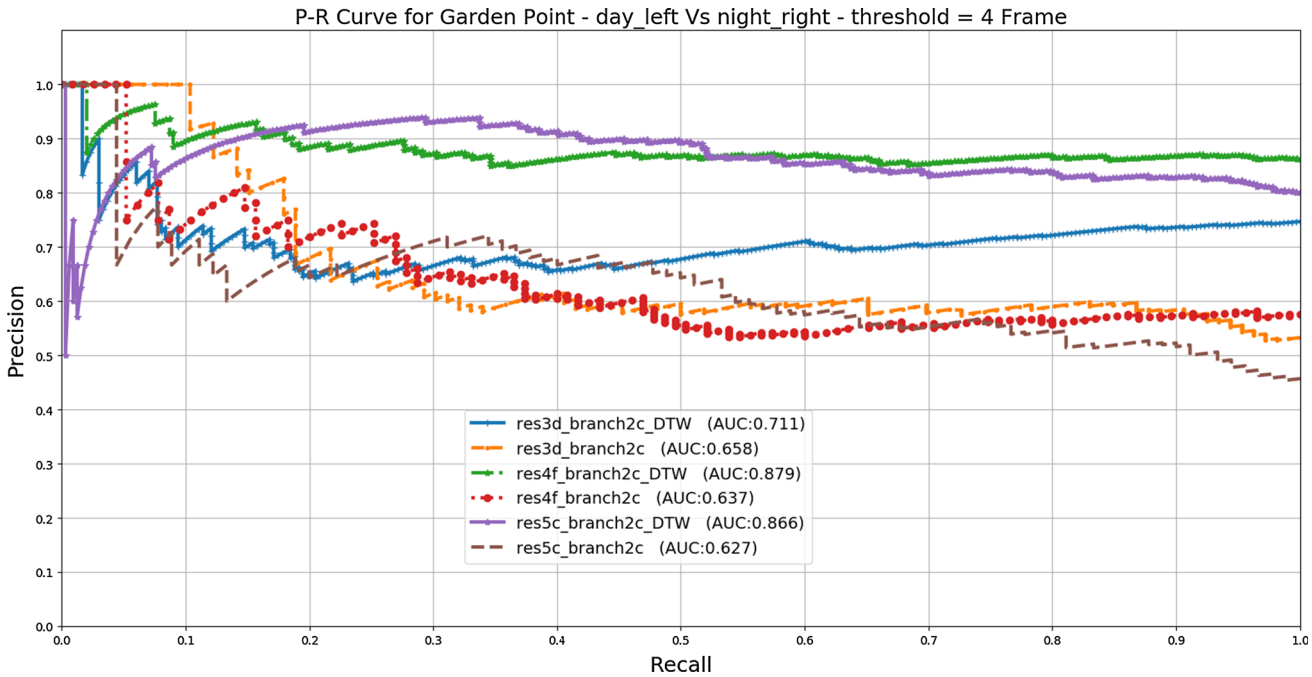

Fig. 7 Different precision–recall curves resulted from exploring features extracted from different layers in the ResNet50 architecture. Both layers "res5c_branch2c" and "res4f_branch2c" have clearly outperformed the others

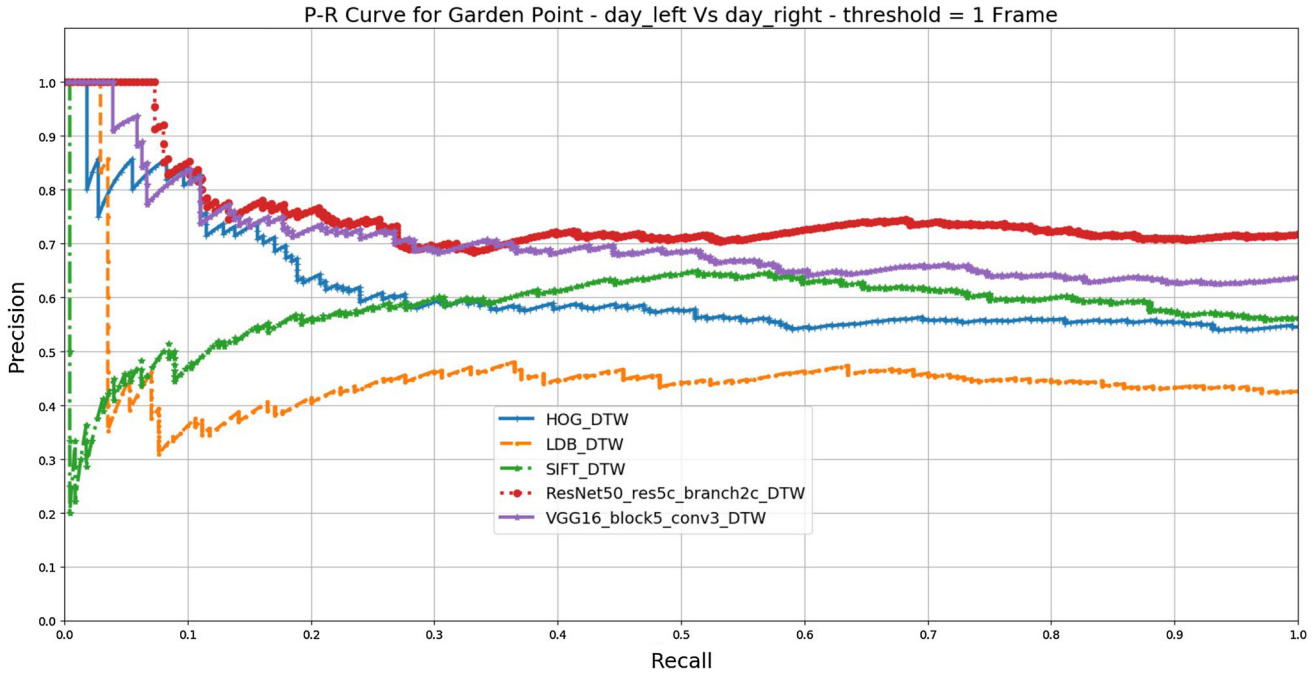

Fig. 8 Precision–recall curves comparing the performance of the features extracted from VGG-16_block5_conv3, the ResNet50_res5c_branch2c and the classical handcrafted features, both using the DTW for place recognition

4.3 What is the Layer with the Best Presentation?

In this section, we explore the VGG-16 and ResNet50 architectures looking for the layer that achieves the best performance according to the P–R curve. This experiment has been formulated to find out the layer among VGG-16 architecture that achieves the best performance when integrated with DTW. In more detail, the output of each layer was injected into DTW to get the best path between the test and reference images. Layers from blocks 3, 4 and 5 were selected. In addition, the "Garden Point" dataset has been used in this experiment where *"day_left"* was the reference sequence and *"night_right"* was the test series. According to this experiment, it could be said that the layers from blocks four and five achieved comparable results. It is worth mentioning that we have repeated this experiment using "day right" as a test sequence and the same results were obtained. Hence, when using the *"day_left"* as reference sequence and the "day right" as a test sequence makes the "Garden Point" dataset has the viewpoint challenging. In contrast, using "night left" make the dataset challenging in terms of illumination and viewpoint which makes the combination of these sequences more difficult than the previous one. The P–R curves are shown in Fig. 6.

Also, this experiment is repeated using the ResNet50 architecture, where some layers have been chosen from the third, fourth and fifth blocks to evaluate their performance when they are injected into DTW algorithm. The corresponding P–R curves are shown in Fig. 7. It is clear that the layer from block 5 "res5c_branch2" and the layer from block 4 "res4f_branch2c" outperformed all other layers. In addition, the layers of this model, i.e., ResNet50, have achieved a better result as compared with the best layers of the VGG-16 network.

4.4 Handcrafted vs. Deep Features

In this experiment, the deep features extracted from the ResNet's "res5c_branch2c" layer and the VGG-16's "block5 _conv3" layer used in the previous experiment, i.e., two of the layers that have obtained the highest performance in the previous section have been evaluated against the HOG, SIFT and LDB handcrafted features. It is obvious that deep features are more capable to handle the visually varying conditions. In addition, as shown in Fig. 8, the deep features have outperformed all other features.

4.5 How Good is the DTW for Place Recognition?

In this experiment, the performance of the DTW for place recognition is compared with SVM and Softmax when used as classifiers. In detail, the output of the ResNet50 "res5c_branch2c" layer was extracted to investigate the performance of the mentioned classifiers. All "Berlin_A100," "Garden Point" and "Nordland" datasets are used, and resulted P–R curves are shown in Figs. 9, 10 and 11, respectively. The DTW clearly outperforms both SVM and Softmax. Also, these experiments show that DTW performs

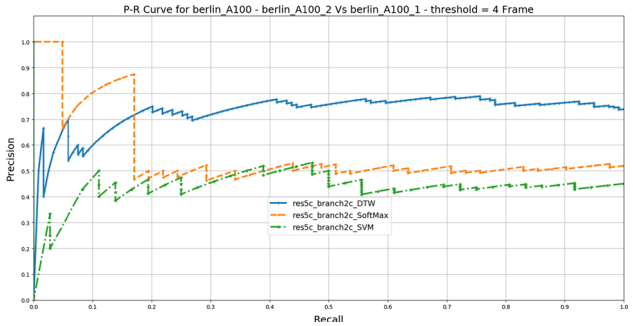

Fig. 9 Precision–recall curves comparing the performance of the DTW, SVM and Softmax algorithms using features extracted from the "res5c_branch2c" layer using "Berlin_A100" datasets, where berlin_A100_1 and berlin_A100_2 refer to the test and reference sequences, respectively

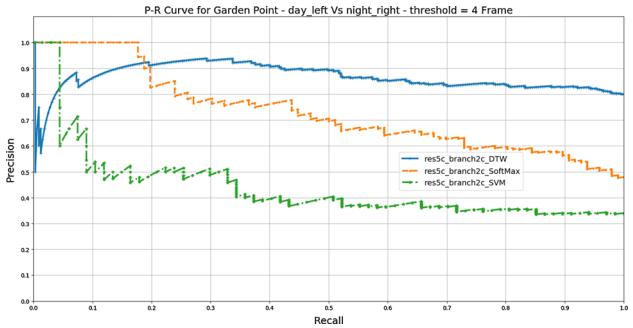

Fig. 10 Precision–recall curves comparing the performance of the DTW, SVM and Softmax algorithms using features extracted from the "res5c_branch2c" layer using "Garden Point" dataset

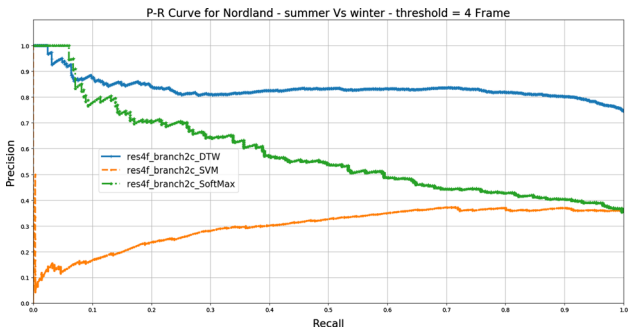

Fig. 11 Precision–recall curves comparing the performance of the DTW, SVM and Softmax algorithms using features extracted from the "res5c_branch2c" layer using "Nordland" dataset

well independently from the power of deep features. Furthermore, as depicted in Table 1, the developed approach has significantly decreased the error mean.

Table 1 Error mean for DTW, the SVM and Softmax algorithms

Dataset	Error mean		
	SVM	Softmax	DTW
Berlin_A100	13.284	9.235	**3.479**
Day_left Vs Night_right	35.075	26.045	**2.930**
Nordland	191.516	197.325	**7.440**

Bold value indicates that the best-obtained results

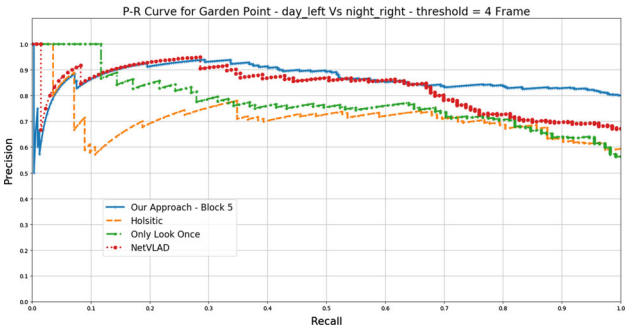

Fig. 12 Precision–recall curves comparing the performance of our approach vs. *Holistic*, *Only look once* and *NetVLAD* approaches using the Garden dataset

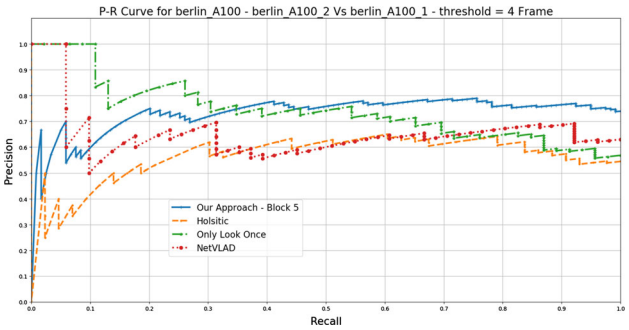

Fig. 13 Precision–recall curves comparing the performance of our approach vs. *Holistic*, *Only look once* and *NetVLAD* approaches using the Berlin_A100 dataset

4.6 Comparison with state-of-the-art CNN-based approaches

In this experiment, the performance of our approach is compared with *Holistic*, *Only look once* and *NetVLAD* approaches.

To ensure the robustness and accuracy of this experiment, all "Garden Point," "Berlin_A100" and Nordland datasets were used. Overall, the results of this experiment can be summarized as follows. (1) Based on the precision–recall curve, as depicted in Figs. 12, 13 and 14 our approach clearly outperforms all, i.e., *Holistic*, *Only look once* and *NetVLAD* when both the "Berlin_A100" and the "Nordland" datasets are used. In addition, when "Garden Point" dataset is used, our approach and *NetVLAD* obtained simi-

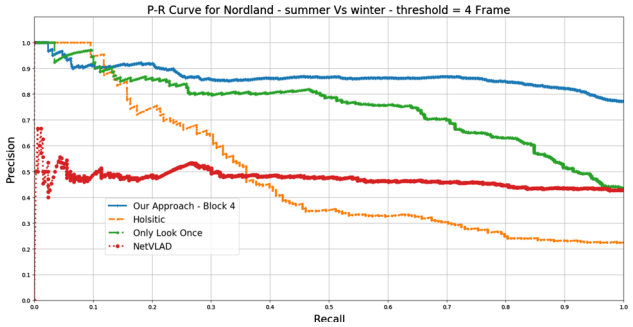

Fig. 14 Precision–recall curves comparing the performance of our approach vs. *Holistic*, *Only look once* and *NetVLAD* approaches using the Nordland dataset

lar precision values when the recall was between 0 and 0.7; however, for the higher recall values our approach was able to outperform all approaches. (2) Based on the F1-score, as depicted in Table 2, our approach clearly outperforms others using "Garden Point," "Berlin_A100" and "Nordland" datasets. (3) Based on the error mean, as depicted in Table 3, the developed approach has significantly decreased the error mean, specifically for the day left vs. night right sub-dataset.

4.7 Comparison with *SeqSLAM* Approach

In this experiment, the performance of our approach is compared with the state-of-the-art handcrafted approach, i.e., *SeqSLAM* approach. Overall, as depicted in Figs. 15, 16 and Table 4, our approach clearly outperforms the SeqSLAM when both "Garden Point" and "Berlin_A100" datasets are used.

Also, Fig. 17 shows a samples of some images that show the performance improvement resulted from the proposed method as compared to *Holistic*, *Only look once*, *NetVLAD* and *SeqSLAM* approaches.

Table 2 F1-score for the proposed approach, *Holistic*, *Only look once* and *NetVLAD* approaches

Dataset	F1-Score			
	Only Look Once	Holistic	NetVLAD	Our approach
Berlin_A100	0.724	0.693	0.773	**0.838**
Day_left Vs night_right	0.714	0.718	0.799	**0.885**
Nordland	0.607	0.364	0.600	**0.850**

Bold value indicates that the best-obtained results

Table 3 Error mean for the proposed approach, *Holistic*, *Only look once* and *NetVLAD* approaches

Dataset	Error mean			
	Only Look Once	Holistic	NetVLAD	Our approach
Berlin_A100	8.160	7.790	6.519	**3.479**
Day_left Vs night_right	21.015	22.745	18.545	**2.930**
Nordland	134.264	183.344	128.940	**7.440**

Bold value indicates that the best-obtained results

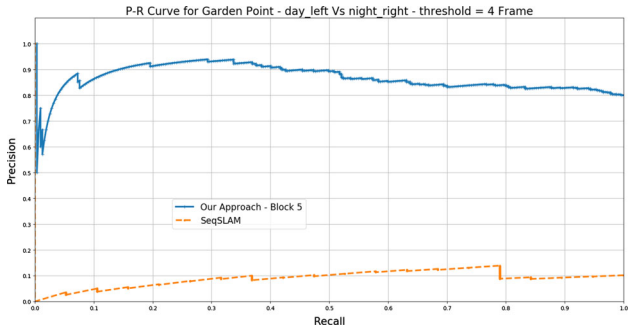

Fig. 15 Precision–recall curves comparing the performance of our approach vs. *SeqSLAM* approach using the Garden dataset

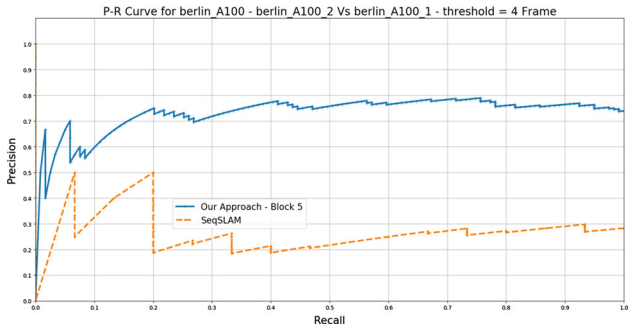

Fig. 16 Precision–recall curves comparing the performance of our approach vs. *SeqSLAM* approach using the Berlin_A100 dataset

5 Conclusions and Future Works

The new visual place recognition method which is presented in this paper employs the dynamic time warping DTW algorithm to match the current frame from a test sequence to a priori annotated reference sequence frame. This algorithm has been used with the features extracted from a deep convolutional neural network (CNN) and can also work with handcrafted features like SIFT, HOG and LDB. The matching is achieved by the construction of a cost function that maximizing the similarity between the frames in both sequences. Then, an optimal path is found using DTW. In addition, multiple layers of the VGG-16 and ResNet50 models were investigated to find the layer that performs better with the DTW algorithm.

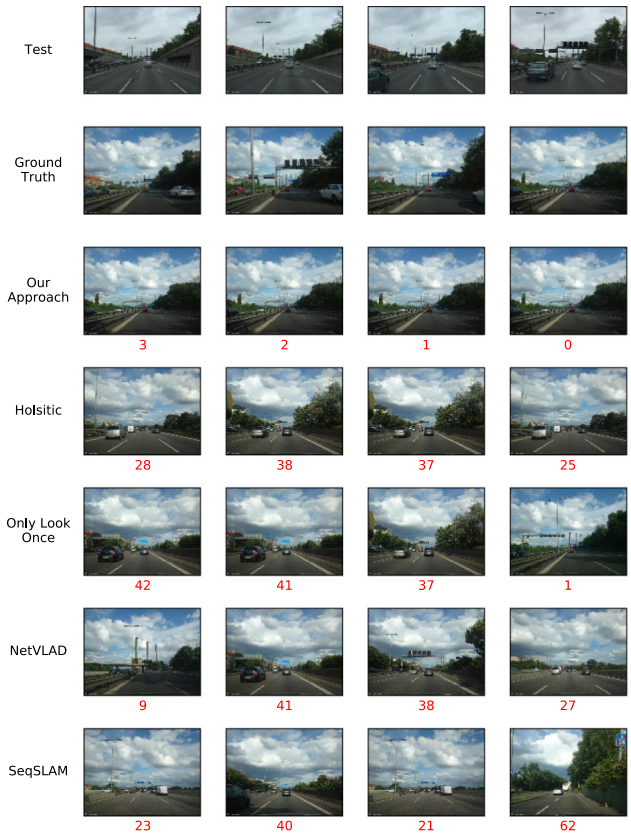

Fig. 17 A sample of four images from the Berlin_A100 test sequence shown in the first row, the ground truth images in the second row and the third till the last are the retrieved images by each VPR method. The numbers in red represent the error in frames between the ground truth and the retrieved image. The samples are selected to show the performance improvement resulted from the proposed method

Table 4 F1-score and error mean for the proposed approach and *SeqSLAM* approaches

Evaluation Matrix	Dataset	Our approach	SeqSLAM
F1-Measure	Berlin_A100	**0.838**	0.353
	Day left vs. night right	**0.885**	0.183
Error mean	Berlin_A100	**3.479**	20.629
	Day left vs. night right	**2.930**	57.0

Bold value indicates that the best-obtained results

Our experiments also compared the performance with other visual place recognition like Holistic, Only look once, NetVLAD and SeqSLAM. The experimental results show superior performance as compared to these state-of-the-art matching algorithms. Also, as shown in the experiments, the approach has the ability to overcome and handle the visually varying conditions such as appearance and viewpoint changes when occurring individually or over simultaneously. Another essential point that must be considered is the ability to work in real time. Hence, the developed approach has this opportunity, i.e., can work in real time, where, for instance using an 800 reference image, the developed approach can process 11 frames per second, where 14 ms is the cost extracting the features of an image, and by assuming that the length of the DTW window is 10 frames, 75 ms is required by the DTW, i.e., starting by calculating the similarity and ending by making a decision on the best-matched reference image.

We conducted several experiments investigating the suitable number of frames (lengths of the test and reference sequences) used in the matching process. We follow the adjustment window scheme presented in Ref. [35]. The time complexity was increasing in a form proportional to the window size, i.e., $O(nm)$. We observed also that there is no improvement in the performance when using window with size 10 or more. It is found that selecting the window size equal to 10 keeps both the time performance and precision performance maximal.

One of the directions that can be done as future work is to investigate the performance of using multiple layers to produce the image's features, and another direction is to investigate the performance of encoding approaches such as bag of words and VLAD.

Acknowledgements The Titan Xp used for this research was donated by the NVIDIA Corporation. This work is partially supported by TUBITAK under project number 117E173.

References

1. Lowry, S.; Sünderhauf, N.; Newman, P.; Leonard, J.J.; Cox, D.; Corke, P.; Milford, M.J.: Visual place recognition: a survey. IEEE Trans. Robot. **32**(1), 1–19 (2015)
2. Abdul Hafez, A.H.; Agarwal, N.; Jawahar, C.: Connecting visual experiences using max-flow network with application to visual localization. arXiv preprint arXiv:1808.00208 (2018)
3. Chancán, M.; Hernandez-Nunez, L.; Narendra, A.; Barron, A.B.; Milford, M.: A hybrid compact neural architecture for visual place recognition. IEEE Robot. Autom. Lett. **5**(2), 993–1000 (2020)
4. Naseer, T.; Burgard, W.; Stachniss, C.: Robust visual localization across seasons. IEEE Trans. Robot. **34**(2), 289–302 (2018)
5. Hausler, S.; Jacobson, A.; Milford, M.: Multi-process fusion: Visual place recognition using multiple image processing methods. IEEE Robot. Autom. Lett. **4**(2), 1924–1931 (2019)
6. Alshehri, M.: A content-based image retrieval method using neural network-based prediction technique. Arab. J. Sci. Eng. 1–17 (2019)
7. Mehmood, Z.; Abbas, F.; Mahmood, T.; Javid, M.A.; Rehman, A.; Nawaz, T.: Content-based image retrieval based on visual words fusion versus features fusion of local and global features. Arab. J. Sci. Eng. **43**(12), 7265–7284 (2018)
8. Abdul Hafez, A.H.; Arora, M.; Krishna, K.M.; Jawahar, C.: Learning multiple experiences useful visual features for active maps localization in crowded environments. Adv. Robot. **30**(1), 50–67 (2016)
9. Fu, R.; Li, B.; Gao, Y.; Wang, P.: Content-based image retrieval based on cnn and svm. In: 2016 2nd IEEE International Conference on Computer and Communications (ICCC), pp. 638–642. IEEE (2016)
10. Du, K.; Cai, K.Y.: Comparison research on IOT oriented image classification algorithms. In: ITM Web of Conferences, vol. 7, p. 02006. EDP Sciences (2016)

11. Kate, R.J.: Using dynamic time warping distances as features for improved time series classification. Data Min. Knowl. Discov. **30**(2), 283–312 (2016)
12. Petitjean, F.; Forestier, G.; Webb, G.I.; Nicholson, A.E.; Chen, Y.; Keogh, E.: Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. Knowl. Inf. Syst. **47**(1), 1–26 (2016)
13. Hafez, A.H.A.; Tello, A.; Alqaraleh, S.: Visual place recognition by dtw-based sequence alignment. In: 2019 27th Signal Processing and Communications Applications Conference (SIU), pp. 1–4 (2019)
14. Lu, F.; Chen, B.; Guo, Z.; Zhou, X.: Visual sequence place recognition with improved dynamic time warping. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1034–1041 (2019)
15. Milford, M.J.; Wyeth, G.F.: Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In: 2012 IEEE International Conference on Robotics and Automation, pp. 1643–1649. IEEE (2012)
16. Cummins, M.; Newman, P.: Fab-map: Probabilistic localization and mapping in the space of appearance. Int. J. Robot. Res. **27**(6), 647–665 (2008)
17. Yue-Hei Ng, J.; Yang, F.; Davis, L.S.: Exploiting local features from deep networks for image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 53–61 (2015)
18. Chandrasekhar, V.; Lin, J.; Liao, Q.; Morere, O.; Veillard, A.; Duan, L.; Poggio, T.: Compression of deep neural networks for image instance retrieval. In: 2017 Data Compression Conference (DCC), pp. 300–309. IEEE (2017)
19. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J.: Netvlad: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5297–5307 (2016)
20. Chen, Z.; Jacobson, A.; Sünderhauf, N.; Upcroft, B.; Liu, L.; Shen, C.; Reid, I.; Milford, M.: Deep learning features at scale for visual place recognition. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3223–3230. IEEE (2017)
21. Hafez, A.A.; Alqaraleh, S.; Tello, A.: Encoded deep features for visual place recognition. In: 2020 28th Signal Processing and Communications Applications Conference (SIU), pp. 1 – 4. IEEE (2020)
22. Khaliq, A.; Ehsan, S.; Chen, Z.; Milford, M.; McDonald-Maier, K.: A holistic visual place recognition approach using lightweight CNNs for significant viewpoint and appearance changes. IEEE Trans. Robot. **36**(2), 561–569 (2020)
23. Chen, Z.; Liu, L.; Sa, I.; Ge, Z.; Chli, M.: Learning context flexible attention model for long-term visual place recognition. IEEE Robot. Autom. Lett. **3**(4), 4015–4022 (2018)
24. Mousavian, A.; Kosecka, J.: Deep convolutional features for image based retrieval and scene categorization. arXiv preprint arXiv:1509.06033 (2015)
25. Wang, T.H.; Huang, H.J.; Lin, J.T.; Hu, C.W.; Zeng, K.H.; Sun, M.: Omnidirectional cnn for visual place recognition and navigation. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2341–2348. IEEE (2018)
26. Li, Z.; Zhou, A.; Wang, M.; Shen, Y.: Deep fusion of multi-layers salient CNN features and similarity network for robust visual place recognition. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 22–29. IEEE (2019)
27. Chen, Z.; Maffra, F.; Sa, I.; Chli, M.: Only look once, mining distinctive landmarks from convnet for visual place recognition. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9–16. IEEE (2017)
28. Sivic, J.; Zisserman, A.: Video google: A text retrieval approach to object matching in videos. p. 1470. IEEE (2003)
29. Zaffar, M.; Ehsan, S.; Milford, M.; McDonald-Maier, K.: Cohog: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments. IEEE Robot. Autom. Lett. **5**(2), 1835–1842 (2020)
30. Li, H.: On-line and dynamic time warping for time series data mining. Int. J. Mach. Learn. Cybern. **6**(1), 145–153 (2015)
31. Zhang, X.; Zou, J.; He, K.; Sun, J.: Accelerating very deep convolutional networks for classification and detection. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 1943–1955 (2015)
32. He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
33. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A.C.; Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
34. Fang, C.: From dynamic time warping (DTW) to hidden markov model (HMM), p. 19. University of Cincinnati, Cincinnati (2009)
35. Sakoe, H.; Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process **26**(1), 43–49 (1978)
36. Alsmadi, M.K.: Content-based image retrieval using color, shape and texture descriptors and features. Arab. J. Sci. Eng. 1–14 (2020)
37. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
38. Dalal, N.; Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893. IEEE (2005)
39. Yang, X.; Cheng, K.T.T.: Local difference binary for ultrafast and distinctive feature description. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 188–194 (2013)
40. Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
41. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
42. Sünderhauf, N.; Shirazi, S.; Dayoub, F.; Upcroft, B.; Milford, M.: On the performance of convnet features for place recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4297–4304. IEEE (2015)
43. Olid, D.; Fácil, J.M.; Civera, J.: Single-view place recognition under seasonal changes. In: PPNIV Workshop at IROS 2018 (2018)