Hindawi
Journal of Electrical and Computer Engineering
Volume 2022, Article ID 6284158, 17 pages
https://doi.org/10.1155/2022/6284158

Research Article

Learning Condition-Invariant Scene Representations for Place Recognition across the Seasons Using Auto-Encoder and ICA

Tariqul Islam,[1] **Sheikh Rabiul Islam**,[1] **and Mahbubur Rahman**[2]

[1]*Department of Electronics and Communication Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh*
[2]*Department of Civil Engineering, European University of Bangladesh, Dhaka, Bangladesh*

Correspondence should be addressed to Sheikh Rabiul Islam; robi@ece.kuet.ac.bd

Received 8 August 2022; Revised 24 October 2022; Accepted 26 October 2022; Published 30 November 2022

Academic Editor: Gongping Yang

Copyright © 2022 Tariqul Islam et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To localize the position and perfect autonomous navigation, building up a map is essential for mobile robots. The map becomes very important when the weather is not appropriate for the robot. However, the map becomes inconsistent when the robot moves in the environment and detects errors with emotional accuracy. The loop-closure detection is the process through which a robot can acknowledge the location visited previously, which can identify the ultimate solution to the previous problem. The robot faced difficulty identifying its previously visited path when the environment underwent an extreme change. The main motive of our work is to promote a model capable of understanding the scenes that are presented robustly. Moreover, during seasonal changes, this model provides an appropriate loop-closure detection result. Independent component analysis (ICA) and auto-encoder are proposed to complete our research work. ICA is a powerful tool to describe invariant images perfectly. Especially, when the robot moves through a changing environment, ICA provided more accurate outcomes than the other algorithm (baseline algorithm). On the other hand, the auto-encoder can distinguish between two features of scene variant condition and invariant condition. The encoder takes our work's next steps by discovering possible routes. To analyze the performance, this work uses the baseline method with a precision-recall curve and a fraction of correct matches. The proposed algorithm ICA showed a 91.05% accuracy rate, which is better than the baseline algorithms, and the appropriate route-finding rate using an auto-encoder is also acceptable.

1. Introduction

A map is designed for an autonomous robot moving through mysterious surroundings, and the robot uses this map for independent guidance along the route. As the localization and mapping are approved together, the process is known as simultaneous localization and mapping (SLAM), which has been used in differing atmospheres such as indoor and outdoor environments. To obtain complete visualization, the SLAM demands some other fundamental modules like an estimation of maps, processing sensor data, loop-closure detection, and visual odometry, where loop-closure detection is used as the core module. This core module has the power to detect whether the current location has been visited or not previously by the moving robot, and the loop is completed when the robot accurately identifies the previously visited area. Nowadays, loop-closure detection is addressed as a place recognition domain with the tremendous improvement of computer vision and also provides the outstanding benefit of the visual SLAM systems by reducing position errors that accumulate over time. It also assists the SLAM in building up a map that provides a consistent view of the environment.

However, the main challenge in a complex environment where the environmental conditions have undergone numerous changes is detecting loop closure accurately. In different seasons, when the weather undergoes extreme change, the prospect of a place may be looked at otherwise. State-of-the-art is a very popular model that works based on of scene appearance and is a very popular way to produce a

loop-closure detection model. The function of the image matching technique is developed based on the comparison between the current view and the previously visited scene which is stored in the robot memory for the same route or locations. This process is given primarily in two steps: image writing and likeness calculation.

SLAM is a successful algorithm because of its robust scene representation skills; therefore, the condition of perpetual feature learning is important for loop-closure discovery. For instance, Lowry and Milford [1] proposed that the principal component analysis (PCA) may be used to recover robust representations of the scenes by straightforwardly administering PCA to the intensive images where PCA has been chosen as the standard baseline algorithm for this work. The base figures obtained from PCA undergo second-order enumerations, but the most perceptually discernible facts guide the higher-order enumerations. Independent component analysis (ICA) is an individual method that handles the division of order moments into four equal parts [2]. This work uses ICA to get the condition-even likenesses of the scenes across various seasons. Although the hand-crafted features succeed in appearance-based loop closure detection, they are still not suitable in situations, e.g., illumination or seasonal changes. From the benefits of the interconnected network-based feature education, an unsupervised auto-encoder is projected to discover robust facial characteristics across various seasons.

Visual localization acts energetically in rustic atmospheres where the atmosphere does not show extreme and intuitive changes. Vision-located orders struggle to label the same place under various ignition changes, seasons, or weather conditions. Traditional approaches use key point-based descriptors that do not show resistance to different seasons. As a result, the local descriptors, to a degree, SIFT, SURF, etc., do not act reasonably in migratory changes in the surroundings. The focus of this work is to determine the condition-invariant likeness across extreme migratory changes. Instead of utilizing a help-devised appearance, this work aims at finding the condition-invariant scene likeness.

In short, the following are the contributions of this article:

(i) At first, we collected the dataset and preprocessed the data in consideration that the dataset would be appropriate for our projected models. We again split the dataset into two portions named testing (20%) and training (80%) for the preparation of the models.

(ii) Second, we supported a detailed description of independent component analysis (ICA) so that we could acquire a condition-invariant representation of the settings across various seasons.

(iii) Third, we detailed the auto-encoder (FAE and CAE) for finding notable two countenance sight variant condition and sight invariant condition to find the likely route.

(iv) Fourth, we provided an approximate study between the existing baseline algorithm (PCA) and our projected models ICA and auto-encoder used for possible route finding, in consideration that we could indeed and suitably recognize, legitimize, and evaluate our models. In addition, we provide a few challenges, an outlook, and future help to the impending scientists in consideration that along with further research, they can gain support from early research and prevent hazards.

The rest of the paper is organized in the following manner: Section 2 and Section 3 present related work and research methodology individually. The experimental outcomes and run-time analysis are presented in Section 4. Opportunities, challenges, and future guidance are characterized in Section 5. Finally, the conclusion is given in Section 6.

2. Related Work

Object recognition is repeatedly appropriated for examination, registration, and guidance in the engine apparition business. However, new marketing object recognition systems are built almost completely based on correlation-related equal motifs. While template matching is very effective in certain engineered environments with tightly controlled object pose and illumination, it becomes computationally infeasible when object rotation, scale, illumination, and 3D pose are allowed to vary, and even more so when dealing with partial visibility and large model databases. Instead of examining all picture locations for matches, extracting features from the image that is at least somewhat invariant to the image generation process and matching solely to those features is an option. Many likely feature types have been proposed and investigated, including line slices [2], edge groups [3], and domains [4], among many others. While these characteristics have acted efficiently for particular object classes, they are typically not labeled repeatedly or usually enough to determine the foundation for correct labeling. Recent work has drawn attention to the development of significantly ranked groups of ocular features. One example is employing a corner indicator (or, more precisely, an indicator of peaks in local concept difference) to find repeating picture regions for local concept attributes. The Harris corner indicator was proposed by Zhang and others [5]. To locate feature sites for popular adjustment of pictures composed from various views, rather than trying to equate regions from individual pictures against all conceivable domains, one can directly conceive only matching domains concentrated at corner points in each representation that occurred in significant period harvests. Schmid and Mohr [6] used the Harris corner detector to recognize interest sites in the object labeling issue and assembled a local image title at each interest point utilizing an adjustment-even vector of derivative-of-Gaussian figure calculations. These representation descriptors were employed for robust object discovery by probing for great matching descriptors that join object-located adjustment and installation limitations. This study was outstanding in terms of both the speed with which it recognized photographs in a vast database and its ability to handle crowded images.

The corner detectors used in the earlier methods have a crucial imperfection because they only judge a picture at a sole scale. These detectors respond to specific picture spots when the scale shift gets substantial. Furthermore, because the indicator does not offer evidence of the object's magnitude, picture descriptors must be used and an endless number of scales must be tried on counterparts. This study presents a prompt way of finding fixed key positions in a hierarchy. This means that disagreeing picture mountaineering has no influence on the set of key places preferred. Again, handcrafted feature-based approaches accumulate differing facets from face photos to build forceful feature headings that were used earlier to train classifiers to a degree, such as SVM, LDA, and BPNN [7, 8]. These methods examine the textural, picture-value, and motion-located traits of face photographs to equate original and fake face photos. The motion-related properties are demonstrated by the movement of the eyes, face, and head motions in motion-based techniques. Various philosophers have made significant efforts to review motion-connected facets. Similarly, concept feature traits are being examined in light of extracting characteristic-connected facets from pictures for face antispoofing methods [9]. The methods established in textural physiognomy are interpreted in the following paragraphs.

Textural feature facts are composed of face photos in makeup feature-based methods, and these lineaments are reapplied to identify physical face photographs from fake ones. In the brochure, fabric feature descriptors, such as degree local binary patterns (LBP), Hog, LPQ, Gabor wavelets, and SURF, are reported to answer face PAs. Among all, the LBP and its derivatives are the most commonly used descriptors; identification results or algorithm efficiency. Määttä et al. [10] and Chingovska et al. [11] first investigated the multiscale LBP title to find face pictures and duplicate assaults. However, an example of a change started in 2012, with a deep learning model. Alex Net [12] easily achieved the ImageNet findings. Deep learning models have been used to address various challenges in forecasting dreams with natural language processing (NLP), providing hopeful results. Not unexpectedly, biometric identification systems were among the first to be supplanted by deep learning models (with a few years' delay). Models established using deep learning offer end-to-end knowledge which is a whole fit for learning feature likenesses. Models based on deep learning offer an end-to-end learning system capable of learning feature representations while doing classification/regression. This is accomplished by using multilayer neural networks, also known as Deep Neural Networks (DNNs) [13].

To determine various preferences of likeness that correspond to differing levels of contemplation that are more adapted to revealing underlying dossier patterns, a multitiered, interconnected system was first projected in the 1960s [14]. However, its feasibility was an issue in and of itself, since the training period would be prohibitively long (because of a lack of powerful computers at the time). Scientists were able to train very deep neural networks much faster thanks to advances in processor technology, particularly the development of General Purpose GPUs (GPGPUs), as well as the development of new techniques (such as Dropout) for training neural networks with a lower chance of over-fitting [15]. The elementary standard behind a neural network is the search of route (inexperienced) recommendation through a network of connected neurons or growth, each of which emulates an uninterrupted or nonlinear function contingent upon feature weights and biases. These weights and biases would change all along by means of backpropagation of the gradients from the output [16], which consistently caused dissimilarities between the expected and accurate current outputs, accompanying the aim of underrating a misfortune function or cost function (distinctness between the virtual and real results under few rhythmical conditions) [17].

Fabric defect detection is one of the very important approaches in the textile manufacturing quality testing system [18]. The grey level co-occurrence matrix (GLCM), autocorrelation analysis, and fractal dimension features were used as spatial distributions as presented by Raheja et al. This approach also constructed a signal graph to calculate interpixel distance and make comparisons between non-defective and text images with the Gabor filter. This approach provided high accuracy with low computational complexity, which is included in the conclusion section [19, 20]. Pourkaramdel et al. presented fabric defect detection by using completed local quartet patterns and a majority decision algorithm whose exact approach is rotation-invariant fabric defect detection. In this approach, they treat local binary patterns as local quartile patterns in order to extract the local texture features of image. In this paper, the benchmark dataset is used to detect their findings consisting of three groups of fabric patterns and 6 defect types. This approach provided high accuracy of detection rate and simple, rotation invariant and grey scale invariant patterns. The quartet pattern descriptor is used as the general texture descriptor which is comprehensively used in different fields in computer vision [21].

Jarallah and Alsaffar introduced the process of isolation and characterization of lytic bacteriophages infecting *Pseudomonas aeruginosa* from sewage water [22]. In this process, the performance of the isolation process depends on sewage water pH value, temperature, bacterial physiological status, phage concentration, presence of certain substances and ions in the media as chemical factors, and their cultural condition. In order to collect data, they used different environment sources and isolation was done in the primary isolation phase. They used Kirby–Bauer disc diffusion methods in order to test antibiotic susceptibility, and Muller–Hinton agar screening methods were used for screening antibiotic susceptibility. When pH remained in the 6–8 range, the isolation activity showed maximum value, and when pH was 9, these activities showed a decreasing pattern. The phase activity remained unchanged during the temperature range at 35–40°C and showed a decreasing pattern when the temperature was 50°C; after 55°C, the reduction rate skyrocketed, lasting for about 8–10 minutes. The methods and findings of this research are appreciable. However, the method is highly sensitive about pH value and temperature, and this method is not an automatic approach. On the other hand, the approach we introduced in our

research work does not depend on temperature or the pH value of water or any other liquid. Moreover, our approach is an automatic approach. The overall EEG channel prediction using various techniques is represented in Table 1.

These studies have proved the ultimate adeptness and correct classification approaches to establish object recognition plans. The efficiency of the ML and DL classifiers accompanying different types of categorization algorithms is very appealing. However, no studies have found that this is the best method to request the automatic district labeling order in object recognition arrangements. This aim of this work is to search an advanced intelligent model to learn the route, though the route is bestowed energetically. This approach can work more all along with the migratory changes (summer to fall, fall to winter, and winter to summer) and is used to aid loop-conclusion discovery accurately.

3. Proposed Methodology

In this section, we will discuss data collection and preprocessing and an in-depth description of the baseline algorithm (PCA), K-means clustering, the BoVW algorithm, and the proposed algorithms (ICA and auto-encoder). The overall arrangement of the research work is shown in Figure 1.

3.1. Data Collection and Procession. In this work, we have used the Nordland dataset that was announced for one Norway TV set party in 2012. This is an open dataset, written as a 728 km long train journey between Trondheim and Bodo in northwestern Norway. The train took 10 hours to complete the trip and has a written dossier in four seasons: summer, fall, spring, and winter throughout the same route. The determination of the dossier is 1920×1080 at 25 frames per second.

To reinforce the value of images and correct results, data augmentation and data enhancement methods are applied. Data are extracted from the original dataset using data augmentation techniques, including horizontal flips, width shifts, height shifts, and rotations. All the limits of improving secondhand work in this place are proved in Table 2. After countenance improves, the dataset is raised. The concept frames elicited one frame per second from the original videos that determine 35,768 countenances per season. Then, the representation is downsampled into 32×64 pixels, and each countenance is convinced for silver scale representation. The dataset is partitioned into two parts. The individual is prepared, and the added individual is the experimenter agreeing to the control. The test dataset consisted 3,569 images in each season which is previously unseen when train going through the tunnels and waiting in the train station. The total dataset details are given in Table 3.

For this work, we have chosen 6 test datasets from the Nordland dataset, and all the selected datasets are shown in Table 4. Each dataset contains the image sequences from the two different seasons. The interpretation of the term "summer–fall" dataset is that the robot has visited the place in the summer and revisited the area in the fall season. This same interpretation applies to all the other datasets.

3.2. Selected Method

3.2.1. Baseline Algorithm. Lowry and Milford [1] showed that the PCA could be used to retrieve robust representations of the scenes by directly applying the PCA to the intensity images. The idea of this algorithm is demonstrated in Figure 2.

Figure 2 shows that the condition-dependent features of the scenes are associated with the first few principal components. So, by discarding the first few principal components and choosing the subsequent principal components, the condition-independent or condition-invariant features can be learned.

(1) Principle Component Analysis (PCA). Principal component analysis (PCA) is a dimensionality reduction algorithm in which a large number of variables are described by a smaller number of variables without a major loss of information; i.e., the low-dimensional data will still be able to explain the original data. For example, an object can be described by numerous properties. PCA tries to summarize the properties of an object by finding a low-dimensional linear subspace onto which the data can be projected. To this end, PCA tries to preserve the maximum possible variance in the data. The first principal component holds the most significant possible variance direction, and the second component contains the second-largest variance direction, and so on.

Consider an n-dimensional dataset $X = \{X^1, X^2, X^3, \ldots\ldots X^P\}$, where P is the number of samples or observations. In each sample x^i, $i = 1, 2, 3, \ldots, P$ is a n-dimensional column vector, called a feature vector. The dataset can be represented in terms of $n \times P$ matrix X. Each column of matrix X refers to the feature vector of a sample. We define a new matrix W $(n \times n)$, which transforms X to Y, as

$$Y = WX. \tag{1}$$

The result of the dot product between W and X is as follows:

$$WX = \begin{bmatrix} W_1X^1 & W_1X^2 & W_1X^3 & \ldots & W_1X^P \\ W_2X^1 & W_2X^2 & W_3X^3 & \ldots & W_2X^P \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_nX^1 & W_nX^2 & W_nX^3 & \ldots & W_nX^P \end{bmatrix} = Y. \tag{2}$$

The data matrix X is projected onto the rows of W. Hence, the rows of W form a new basis for the columns of X, which are denoted as the principal component directions. We look for a transformation matrix W, which keeps as much as the possible variance of the data. Estimation of a covariance matrix of $n \times P$ dimensional data X, assuming X to be zero-mean, can be done as follows:

Table 1: Related work of EEG channel prediction using various techniques.

References	Key purposes	Model
Bellekens et al. [23]	Analyze numerous for place recognition	Cloud coarse registration methods
Marcel et al. [9]	Detect the motion of the objects	SVM, LDA, and BPNN
Chen et al. [24]	Place classification task	AMOSNet and HybridNet
Noh et al. [25]	Landmark recognition task	Global descriptor adaptation
Wang et al. [16]	Place detection	Omnidirectional CNNs
Pourkaramdel et al. [21]	Fabric defect detection	Completed local quartet patterns and majority decision algorithm

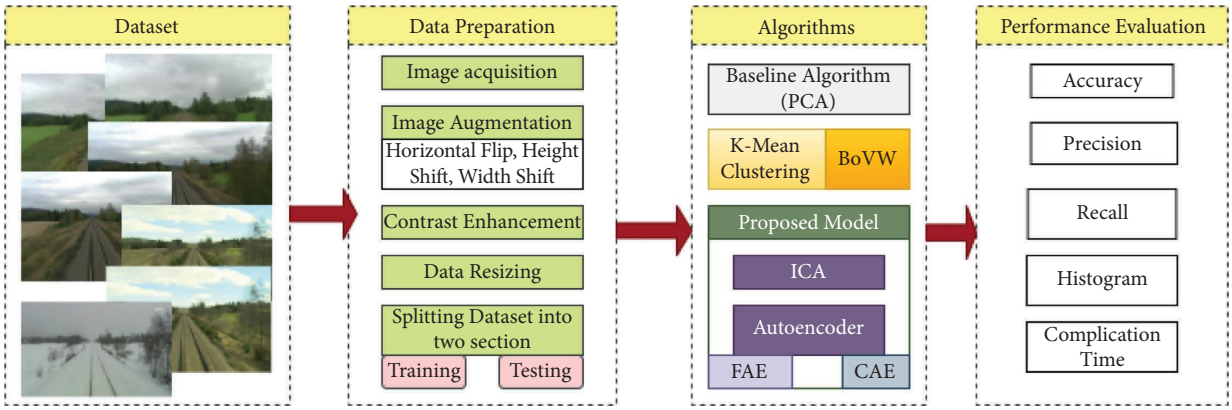

Figure 1: Proposed diagram of the system.

Table 2: Data augmentation parameters.

Augmentation technique	Ranges
Horizontal flip	True
Width shift	0.3
Height shift	0.3
Rotation	False
Vertical flip	False

$$R_{XX} = \frac{1}{N-1} XX^T = \frac{1}{N-1} \begin{bmatrix} X^1X^{1T} & X^1X^{2T} & \dots & X^1X^{nT} \\ X^2X^{1T} & X^2X^{2T} & \dots & X^2X^{nT} \\ \vdots & \vdots & \vdots & \ddots \quad \vdots \\ X^nX^{1T} & X^nX^{2T} & \dots & X^nX^{nT} \end{bmatrix}. \tag{3}$$

The diagonal entities of R_{XX} represent the *variance* of the elements x_i, where $i = 1, 2, 3, \dots, n$, and the off-diagonal elements represent the *cross-covariance* between x_i and x_j (with $i \neq j$). Therefore, one needs to find the W that diagonalizes R_{YY}, where R_{YY} is a covariance matrix of the transformed data. Using (1) expression for Y, R_{YY} can be written as follows:

$$R_{YY} = \frac{1}{N-1} YY^T = \frac{1}{N-1} (WX)(WX)^T = \frac{1}{N-1} WR_{XX}W^T. \tag{4}$$

Note that R_{XX} is a $n \times n$ symmetric square matrix; therefore, it can be orthonormally diagonalized as follows: $R_{XX} = U \Lambda U^T$, where U is a square matrix whose columns are eigenvectors of R_{XX} and Λ is a diagonal matrix with the eigenvalues of R_{XX} as its entries. This diagonalization step is called eigenvalue decomposition (EVD). If we choose rows of W being eigenvectors of R_{XX}, then we can write $W = U^T$.

$$R_{YY} = \frac{1}{N-1} WR_{XX}W^T = \frac{1}{N-1} U^T \wedge U^T U = \frac{1}{N-1} \wedge. \tag{5}$$

The inverse of an orthogonal matrix is its transpose, i.e., $U^{-1} = U^T = U^T U = I$, where I is the identity matrix. As a result, the eigenvectors of R_{XX} are the proper choice for W, as diagonalization of R_{YY} is the goal of PCA, and $W = U^T$ satisfies the condition.

3.2.2. K-Means Clustering. K-means clustering is a popular unsupervised algorithm to find the structure in unlabeled data points. The term clustering refers to the grouping of similar data points from a collection of significant data points. In K-means, clustering aims to find the K number of clusters or groups in a given dataset. Suppose we have an n-dimensional P data sample $X = \{X^1, X^2, X^3, \dots\dots X^P\}$, each sample x^i $(i = 1, 2, 3, \dots, P)$ is an n-dimensional column vector and the goal is to find the K number of clusters. There are four basic steps in the K-means clustering algorithm; the steps are given in Figure 3.

Table 3: Details of the dataset.

Feature	Value
Total number of images	143072
Total number of training images	128796
Total number of test images	14276
Dimension (pixels)	32×64
Color	Grayscale
Total number of images from each season	35768
Total number of training images from each season	32199
Total number of test images from each season	3569

Table 4: Experimental datasets.

Serial number	Datasets	Visit	Revisit
1	Summer-fall	Summer	Fall
2	Summer-spring	Summer	Spring
3	Spring-fall	Spring	Fall
4	Winter-summer	Winter	Summer
5	Winter-fall	Winter	Fall
6	Winter-spring	Winter	Spring

Figure 2: First column: two images from the same locations in two different seasons (Nordland dataset). Second column: images are projected on the first 100 principal components (PCs). Third column: images are projected on the second 100 main components (PCs); dependent features of the scenes are associated with the first few principal components.

Step 1. Random initialization

In this step, K numbers of random n-dimensional vectors μ_k, where $k = 1, 2, 3, \ldots, K$, are chosen in the given data point space. These points are called randomly chosen centroids.

Step 2. Assign points to the nearest centroid

In this step, we assign each data point to the nearest centroid by measuring the distance between the data points and the centroids. We take a single data point and calculate the distance from the point to all the centroids. Then, the point is assigned to the centroid with the minimum distance.

Step 3. Update the centroids

This step updates the centroid by taking an average of all the points that are assigned to that centroid.

$$\mu_{\text{new}} = \frac{1}{r} \sum_{X^i \in S_i} X^i. \tag{6}$$

Here, r is the number of points that belongs to the centroid, X^i is the point that belongs to the centroid, and S_i is a set of the coordinates of the assigned points.

Step 4. Iteration

This step repeats Step 2 and Step 3 until none of the cluster assignments are changed.

3.2.3. Bag-of-Words Model. The first step of the BoVW is to extract features from the training data, as shown in Figure 4. There are various types of local features and global features. Good feature selection plays an important role in different computer vision tasks. It should be invariant to translation, rotation, scale, condition, and so on. After computing the features from the training data, the key task of the BoVW [26, 27] model is to build a dictionary of the visual words [28]. K-means clustering is a convenient way to learn visual words, where the centroids of the clusters represent the visual words. The number of centroids is the number of visual words in the dictionary [29, 30]. The collection of the

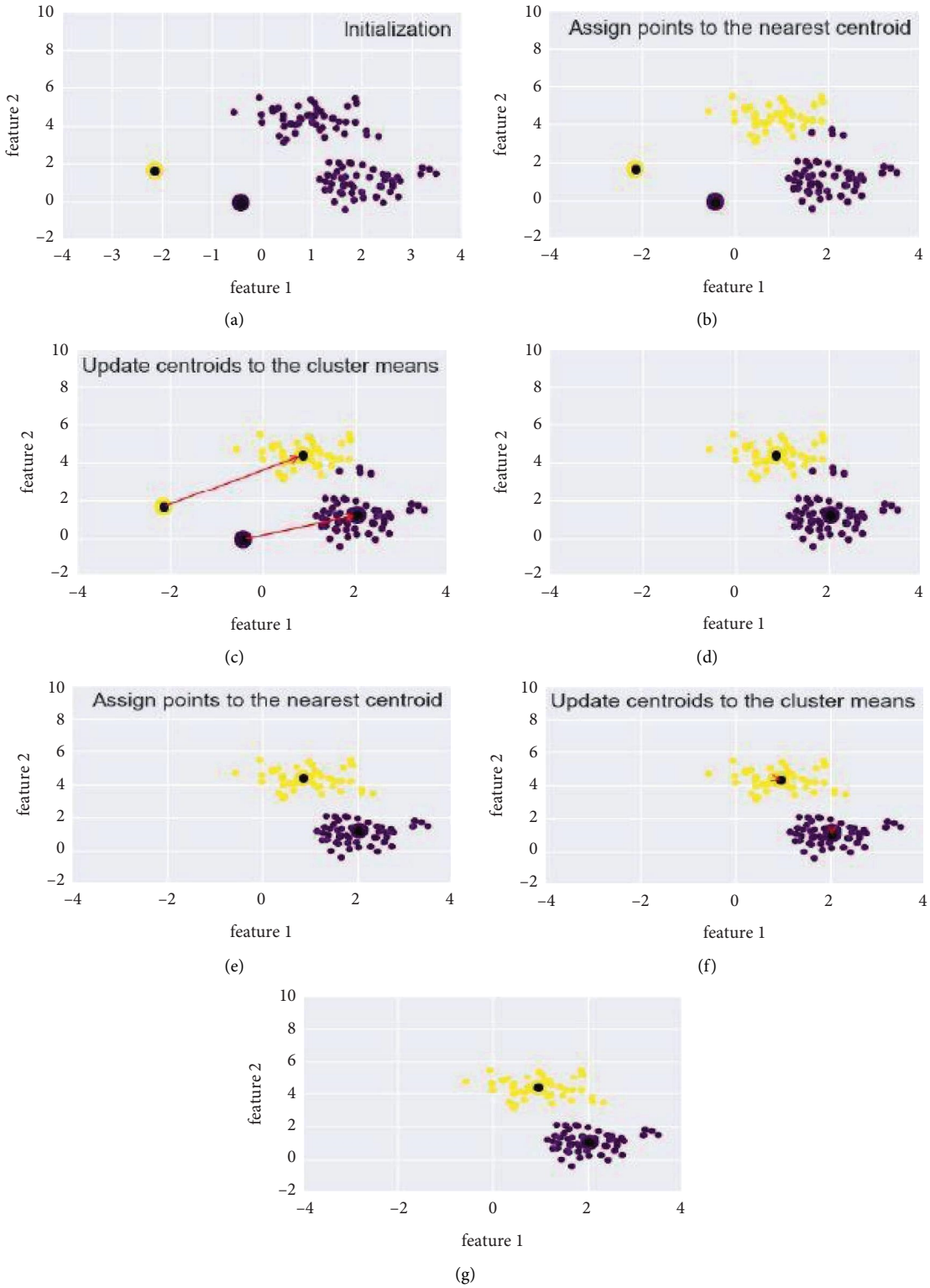

Figure 3: Illustration of K-means clustering.

visual words is shown as the dictionary of words in Figure 4. The training phase of the BoVW model ends with learning the dictionary of visual words. Then, the next task is to extract the features from the test data and calculate the distance from the extracted features to all the visual words in the dictionary. The features are then one by one assigned to the nearest centroids, and we count the number of times the features are assigned to a centroid. As a consequence, this procedure results in a histogram representation of each and every test data [31]. The x-axis values are the visual words,

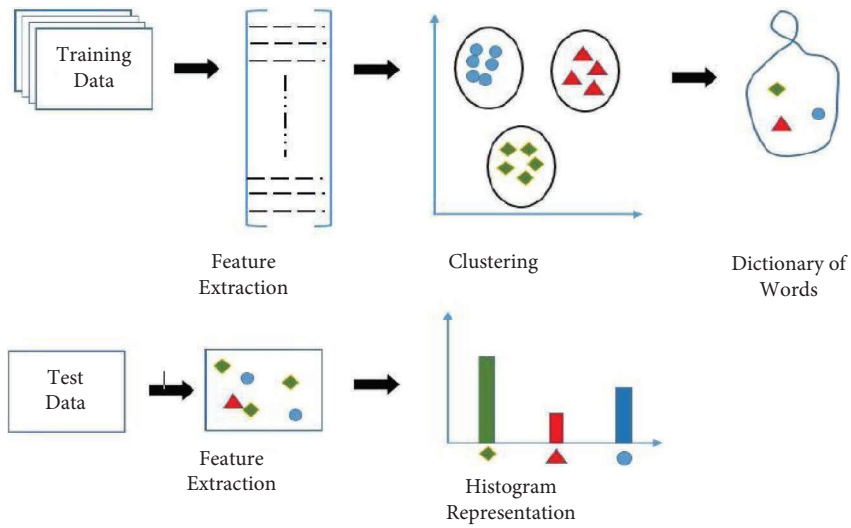

Figure 4: Illustration of the bag-of-words (BoVW) model.

and the y-axis values are the number of times that visual word appears in that test data. It is called the frequency of that visual word. The obtained histogram is considered a feature vector for the comparison between the test data. In the image retrieval or place recognition problem, the images can be retrieved or the places can be recognized by measuring the distance between the histograms using different distance metrics.

3.3. Proposed Models. In this work, two main methods have been adopted to learn the condition-invariant features of the scenes. The first one is ICA, and the ICA will be applied to the complete image or scene as a global feature extractor and to the image patches as a local feature extractor. The second one is an auto-encoder; two variations of an auto-encoder have been proposed in this work. They are the fundamental auto-encoder (FAE) and the convolutional auto-encoder (CAE).

3.3.1. Independent Component Analysis (ICA). ICA finds the directions that maximize the independence of the random variables using the higher-order statistics. This work aims to learn the independent components (ICs) from the training dataset, and the test images will be projected onto the learned IC's. The workflow of finding the independent components is demonstrated in Figure 5. There, the centering and whitening are done as the preprocessing step to accelerate the process of finding the unmixing matrix W. The fastICA algorithm has been used in this work. After finding the unmixing matrix, the columns of the unmixing matrix are called the weight vectors or the directions of the independent components. The test images are projected onto the weight vectors, which give the newly transformed space of the test data. This newly transformed space is considered the condition-invariant representation of the scenes. In matrix form, we can write this relationship as follows:

$$X = As, \tag{7}$$

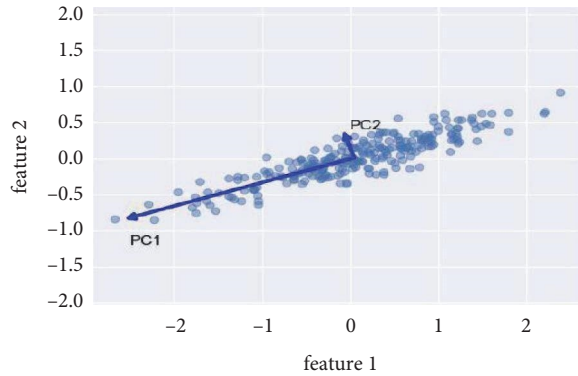

Figure 5: Illustration of the principal component analysis.

where $\mathbf{x}$ is the n-dimensional observed signal vector, $\mathbf{s}$ contains the independent components, and A is called the mixing matrix.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}. \tag{8}$$

For the sake of mathematical convenience, we have assumed that the independent components have *unit variance*. The independent component analysis problem overview is shown in Figure 6.

(1) FastICA Algorithm. The summary of the ICA algorithm is presented in this section. Suppose we have a data matrix $X \in R^{n \times P}$, where each column of the data matrix X represents an n-dimensional sample vector. We want to extract independent components' C numbers, where $C \leq n$. The task is to find the unmixing matrix $W \in R^{n \times C}$ onto which the observed data matrix X is projected to obtain the independent component matrix $S \in \mathrm{R}^{C \times P}$ (Algorithm1).

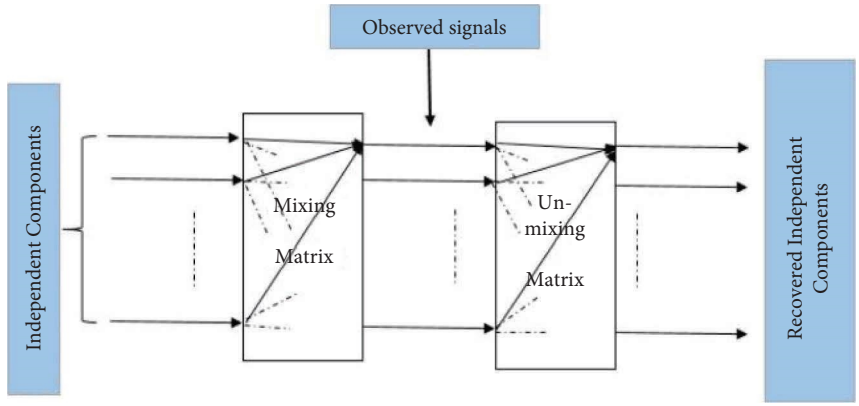

Figure 6: Overview of the ICA problem. The independent components **s** and the mixing matrix **A** both are unknown. The goal is to estimate **A** and **s** from the known x.

(2) ICA on Image Patches. In this procedure, we use the ICA to extract the local descriptors of the scenes. For this purpose, images are divided into patches, and independent components (ICs) are computed from the image patches. To compare the images, the BoVW model is introduced to make the algorithm independent of the order of the image patches. The key task in the BoVW model is to build a dictionary of the visual words. A K-means clustering algorithm is applied to find out the visual words, where each centroid of the clusters represents one visual word. The collection of all the visual words is called the dictionary of visual words. The training image patches are also projected onto the IC's space before applying the clustering algorithm to make sure that the learned visual words and the test image patches are in the same feature space. After learning the dictionary, the next task is to extract the patches from a given test image and project them onto the computed IC's space. Then, the new transformed image patches will be used to represent the image as a histogram, and this workflow is shown in Figure 5. The histogram representation is part of the BoVW model. This histogram representation can be one possible approach to finding the condition-invariant representations of the scenes.

3.3.2. Auto-Encoder. In the auto-encoder, by applying the nonlinear activation function in the encoder and decoder parts, it is possible to extract the nonlinear hidden pattern of the input space as a code vector. The term code vector is defined as the output of the encoder; it is the compressed or low-dimensional representation of the original data space. In this thesis work, two variations of the auto-encoders have been used. They are mentioned as follows:

(i) Fundamental auto-encoder with one hidden layer

(ii) Convolutional auto-encoder with deep convolutional layers

(1) Fundamental Auto-Encoder (FAE). An auto-encoder is an unsupervised learning algorithm that takes unlabeled training data as input and regenerates the input data as an output data. A two-layer auto-encoder is shown in Figure 7. Input data

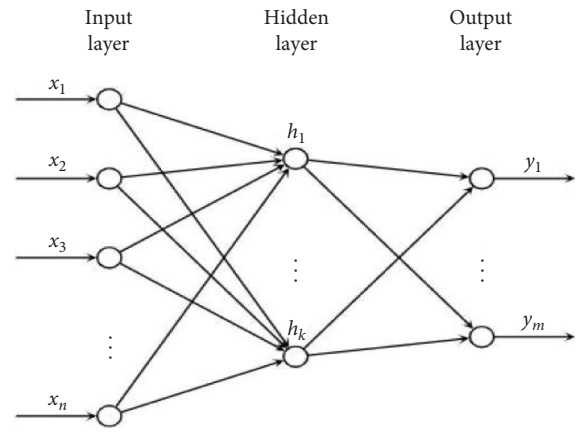

Figure 7: An architecture of a fundamental auto-encoder with a hidden layer and an output layer.

(n-dimensional vector), $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, \ldots, x_n]^T$, are mapped to the output layer with values $\mathbf{y} = [y_1, y_2, y_3, y_4, y_5, \ldots, y_n]^T$ through a hidden layer space representation, where $\mathbf{y} \approx \mathbf{x}$. An auto-encoder has two main parts. One is the encoder, which projects the input data to the internal hidden layer space, i.e., encodes higher-dimensional data into lower dimensions. The second part is the decoder, which tries to reconstruct the original dimension from the encoded dimension. The encoder part gives the inherent information of the data, which can be very useful for various applications. All the hidden layer and output layer neurons have the same working principle. The training process simply minimizes the cost function using the backpropagation algorithm. If the number of hidden neurons is equal to the number of input neurons, then the network fails to extract the useful hidden pattern of the data points [32, 33] by assigning constraints on the number of hidden neurons in the hidden layer, e.g., if the number of hidden neurons is less than the number of inputs. Then, it might be possible to discover the useful hidden patterns of the data [12, 15].

The convolutional auto-encoder (CAE) is very similar to the fundamental auto-encoders, except that an input of the CAE is an image. In CAE, there are three basic types of layers in the architecture:

(i) Convolutional layers
(ii) Max-pooling layers
(iii) Up-pooling layers

So, unlike the fundamental auto-encoder, in CAE, the hidden layers will be replaced by the convolution layers and the pooling layers. It enables a huge reduction in the number of parameters (e.g., weights and bias) to be learned for the network [17], as the convolutional layer's filter dimensions are predefined and independent of the input image size. The architecture of the auto-encoder is shown in Figure 8.

In the convolutional layer, multiple filters are applied to learn different feature detector filters. In training, it is important to set the filter's value randomly to prevent the filters from learning the same parameters. The number of filters in a convolutional layer represents the number of feature maps in that layer. The encoder and decoder information of the convolutional auto-encoder is shown in Table 5.

4. Results

In this section, first we provide the experimental setup, followed by the evaluation method, experimental outcomes, and runtime analysis in Sections 4.2, 4.3, and 4.4 accordingly.

4.1. Experimental Setup. The test environment is Ubuntu 16.04.4 with 16 GB of RAM, Intel(R) Core (TM) i7-8086K CPU @ 4.00 GHz processor with GeForce GTX 1060 graphics card. The development environments and programming languages that were used for this purpose include MATLAB (R2018a), Python (version 3.6.1) with TensorFlow 1.0, and the Keras 2.2 framework. The programs were run on the system in the GPU2 acceleration mode.

4.2. Evaluation Method. Our algorithms are evaluated using the precision-recall curves and the fraction of correct matches. The accurate matches are the TP, the incorrect matches are the FP, and when an algorithm mistakenly discards a correct match, it is labeled as the false negative (FN) match. Since every scene in this dataset has a ground truth match, there is no true negative (TN) and every negative is a false negative (FN). The precision-recall curves are a very useful evaluation measure in the loop-closure detection model. In a loop-closure detection model, it is essential to avoid false positives because it means two images have been identified as being from the same place, i.e., a loop detection, but in reality, they are from two different locations [34, 35]. This false prediction leads the algorithm to produce an inconsistent map of its surroundings. An ideal error-free model would be the one that reaches 100% precision at 100% recall. The precision-recall curves are generated by varying the number of retrieved or matched images, ranging from 1 to the total number of the test images [1].

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}. \end{aligned} \tag{9}$$

We have used one more evaluation metric, the fraction of correct matches. It is defined as the percentage of the number of times the best match I_s detects the correct loop [36].

$$\text{fraction of correct matches} = \frac{\text{number of correct predicted places}}{\text{number of total evaluated places}} \times 100\%. \tag{10}$$

4.3. Experimental Outcome

4.3.1. Outcome of the Summer-Fall Dataset. Using the summer-fall dataset, the precision-recall curves of our proposed model and the baseline algorithm are shown in Figure 9. From this figure, it is clear that the performance of ICA is better than the baseline algorithm PCA. It shows that the performance of the ICA algorithm shows better results than the baseline algorithm. Though the precision rate is hundred percent for both the algorithm (ICA and PCA) until the recall rate at 67%, for the baseline algorithm, the precision decrement of the precision rate is faster than that of the ICA algorithm. Therefore, when ICA holds a precision rate of 91.05%, the baseline algorithm holds 84% for both cases recall rates at 100%. The fundamental auto-encoder (FAE) and the convolutional auto-encoder (CAE) also follow the same trend as the ICA. The precision dropping rate starts earlier than in the other algorithms, but the decay rate is slower than in the baseline and ICA algorithms. Even for the FAE, the precision rate is higher than for the baseline method at an 82% recall rate, and for the CAE, the precision rate is almost the same as that of the baseline algorithm at a 100% recall rate.

The fraction of correct matches acts as an evaluation matrix, as shown in Figure 10. From Figure 11, it is clear that ICA algorithms provide 91% correct loop-closure detection rate, which is better than of the baseline algorithm, whose accurate match rate is 88%. On the other hand, from Figure 11, the FAE and CAE provide correct match rates of 88% and 83%, respectively, for finding the expected possible root for the moving robot. In general, in the case of the summer-fall dataset, all the algorithms presented in this work have shown good performance. The true perceptual changes in the summer and fall season images are comparatively less. Fewer changes in the appearance of the scenes are the main reason for the success of all the presented algorithms.

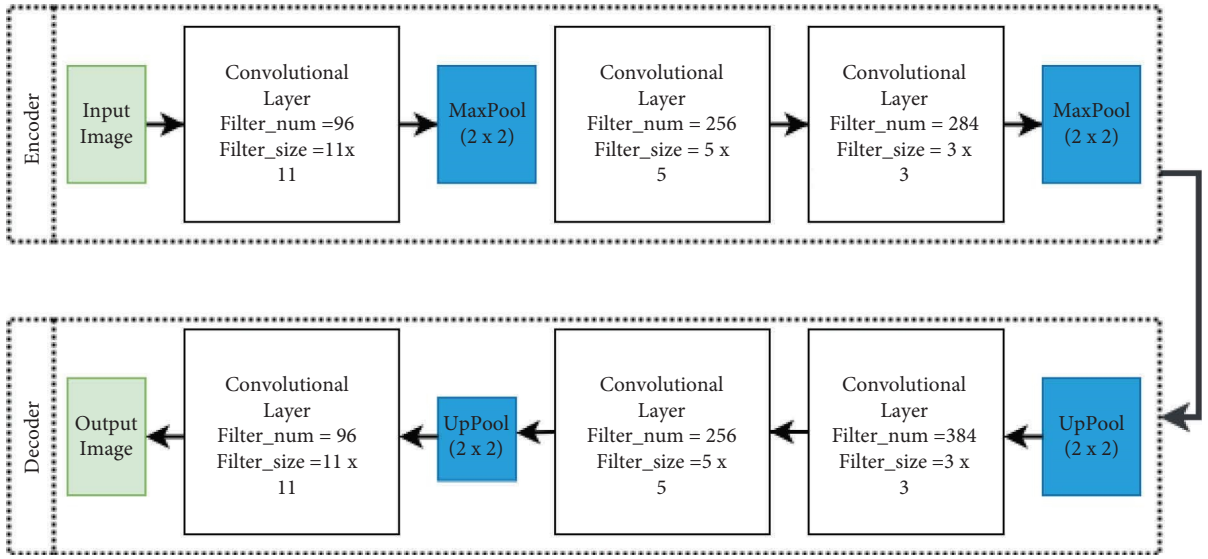

Figure 8: The architecture of the proposed convolutional auto-encoder. Filter number stands for the number of the filters of that associated layer. Filter size refers to the size of the convolutional filter.

(1) **for** $p = 1$ to C **do**
(2) $\mathbf{w}_p \longleftarrow$ *random vector of length* n
(3) **while** $\mathbf{w}_p$ changes **do**
(4) $W^{p+1} \longleftarrow Xg(W_p^T X) - g'(W_p^T X)W_p$.
(5) $W^{p+1} \longleftarrow W_{p+1} - (\sum_{j=1}^{p} W_p^T W_j W_j^T)^T$.
(6) $W^{p+1} \longleftarrow W_{p+1}/\|W_{p+1}\|$.
(7) **end while**
(8) **return** $\mathbf{w}_p$
(9) **end for**
(10) **return** $W = [w_1; w_2; w_3; \ldots; w_c]$

Algorithm 1: FastICA.

(1) ICA Model Using the Image Patches of the Summer-Fall Dataset. We have analyzed the use of the BoVW model on the summer-fall dataset to learn the condition-invariant features. Figure 12 shows the learned filters by applying ICA as a local feature extractor, i.e., applying ICA to the image patches.

This approach fails to recognize the places showing seasonal changes. One downside of the BoVW model is that it fails to consider the spatial relation between the visual words. Hence, it produces a large number of similar visible words.

Figure 13 shows a few visual words from the learned dictionary using K-means clustering. From this figure, it can be observed that many visual words correspond to the railway track (red circles indicate the visual words correspond to the railway track). Due to the similarity between the visual words, the model loses its descriptive capability [34]. This model cannot produce a distinguishable description of the test images. Hence, this model is not applied to any other datasets in this work.

4.3.2. Outcomes of the Spring-Fall Dataset. Using the spring-fall dataset, the precision-recall curves of our proposed model and the baseline algorithm are shown in Figure 14.

Table 5: Summary of the CAE (encoder and decoder) networks.

	Size of filters	Number of filters
Layers (encoder)		
Convolutional layer 1	11×11	96
Max-pooling layer 1	2×2	96
Convolutional layer 2	5×5	256
Convolutional layer 3	3×3	384
Max-pooling layer 2	2×2	384
Layers (decoder)		
Up-pooling layer 1	2×2	384
Convolutional layer 1	3×3	384
Convolutional layer 2	5×5	256
Up-pooling layer 2	2×2	96
Convolutional layer 3	11×11	96

From this figure, it is clear that at a 50% recall rate, all the algorithms fail to hold a 100% precision rate, whereas the baseline algorithm and ICA maintain the same trend in performance. Approximately until 20% recall rate, they hold a hundred percent precision rate, and then the graph

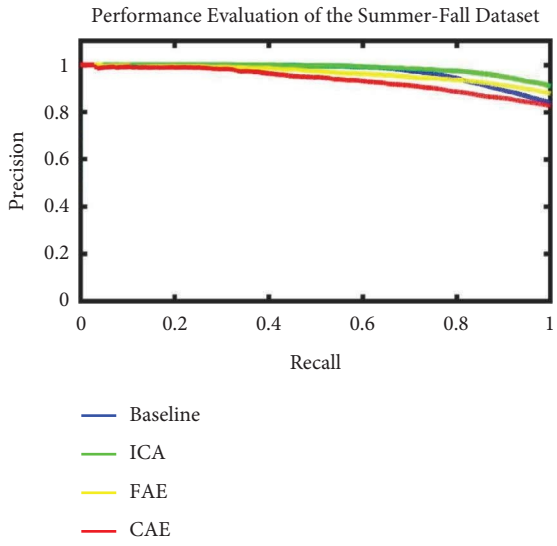

Figure 9: Precision-recall curves of the baseline and proposed algorithms on the summer-fall dataset.

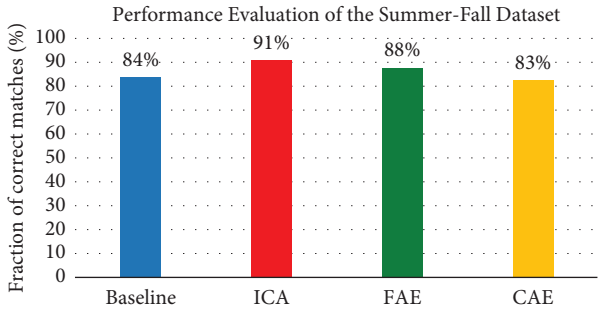

Figure 10: Fraction of correct baseline matches and proposed algorithms on the summer-fall dataset.

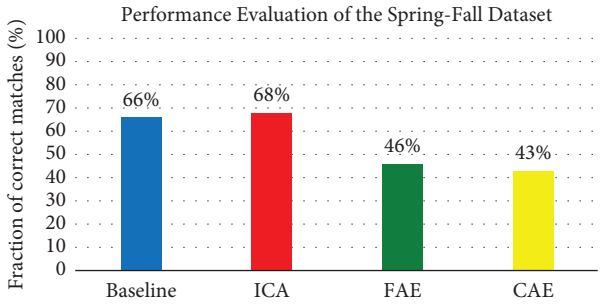

Figure 11: Fraction of correct matches of the baseline and proposed algorithms on the spring-fall dataset.

shows a decreasing pattern in the slow speed of the precision rate for both algorithms. However, the FAE and CAE start to drop the precision rate at a great pace at the very beginning and reach 60% from 100% at only a 32% recall rate. After that, the precision rate is smoothly going down, hence, they have a below 50% precision rate at a 100% recall rate.

The fraction of correct matches for the baseline algorithm and proposed model for the spring-fall dataset is shown in Figure 11. This figure illustrates that our proposed ICA algorithm shows a slight improvement in detecting the proper loops over the baseline algorithm. The baseline model achieves 66% of accurate matches, while the ICA model achieves 68%. FAE and CAE also obtain 46% and 43% of correct matches, respectively. The reason for the rapid decline in precision rate in the spring-fall dataset can be attributed to the perceptual changes between the spring and fall season images. The sky is less cloudy in the spring season compared to the fall season. Moreover, nature is more green in the spring than in the fall.

4.3.3. Outcomes of the Summer-Spring Dataset. Figure 15 shows the precision-recall curves of the baseline and the proposed algorithms on the summer-spring test dataset. In this figure, we can see that our proposed algorithms do not outperform the baseline algorithm. We notice a drop in the precision rate in the ICA algorithm at around 1% recall due to finding false-positive matches. After that, the precision rate gets higher and reaches 100% precision rate. The baseline, ICA, and FAE models hold a 100% precision rate with 40% recall. After that, the precision rate drops for the ICA and FAE compared to the baseline method. The CAE algorithm shows poor precision values compared to the other three algorithms. Its precision rate starts to decrease from the beginning. It achieves a precision rate of 78% and 58% at a recall rate of 50% and 100%, respectively.

Figure 16 shows the fraction of correct matches of the different algorithms on the summer-spring dataset. It shows that the baseline algorithm detects the proper loops at 88%, whereas the proposed ICA, FAE, and CAE methods detect 83%, 77%, and 59%, respectively. The ICA and the FAE algorithms have the second and third-best loop detection rates among the presented algorithms in this work.

4.3.4. Results of Winter versus Other Seasons. In the winter-summer, winter-spring, and winter-fall datasets, our proposed algorithms show poor performance. The obtained results from the proposed algorithms and the baseline algorithm are demonstrated in Figure 17. This is the most challenging dataset among all the chosen test datasets. In the winter, the sky is cloudy and the landscape is covered by snow. Hence, extreme perceptual changes exist between the winter and other seasons. The scenes are poorly illuminated, which gives them a textured, featureless, and low-contrast view. The ICA works as an edge detector filter in natural settings [31, 37]. Hence, ICA fails to detect the robust patterns of the featureless winter season scenes. In every test dataset, it has been seen that the auto-encoders do not perform up to the mark as the neural network algorithms demand a high volume and wide variety of training data. We have the intuition that in the training dataset, the training images' variations are not enough so that the network can learn the generalized parameters. As a result, the auto-encoders fail to learn the condition-invariant representation of the unseen test data across the extreme seasonal changes.

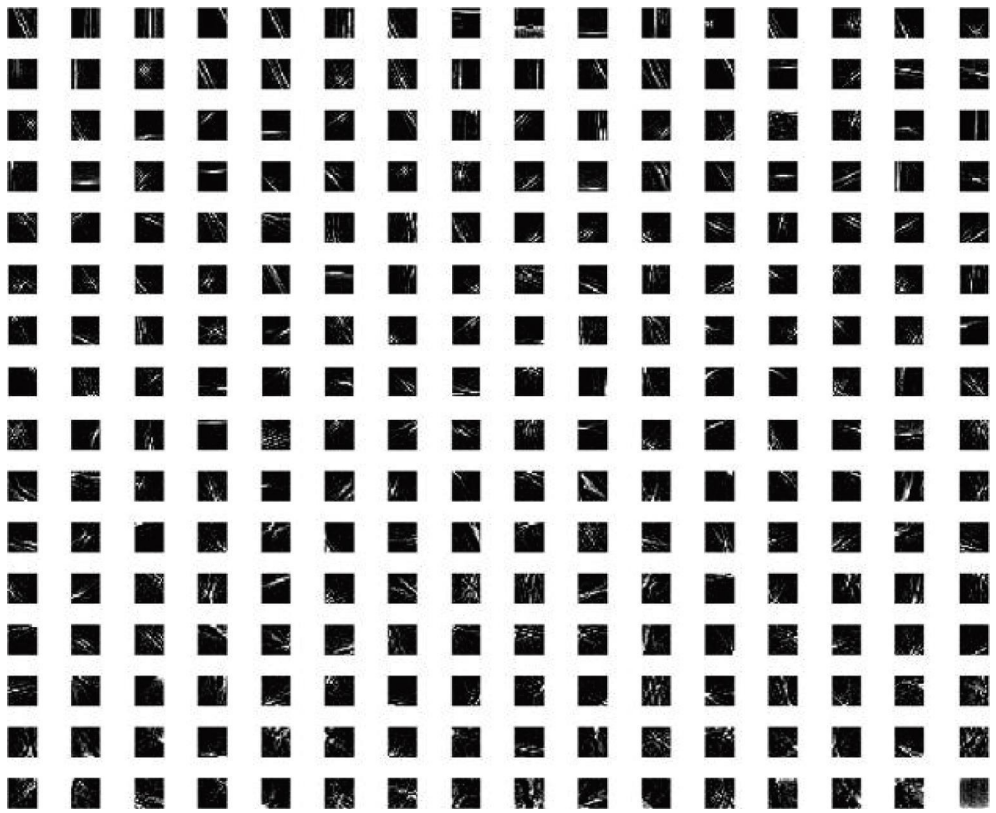

Figure 12: Obtained ICA filters (256) using the image patches of the summer-fall dataset.

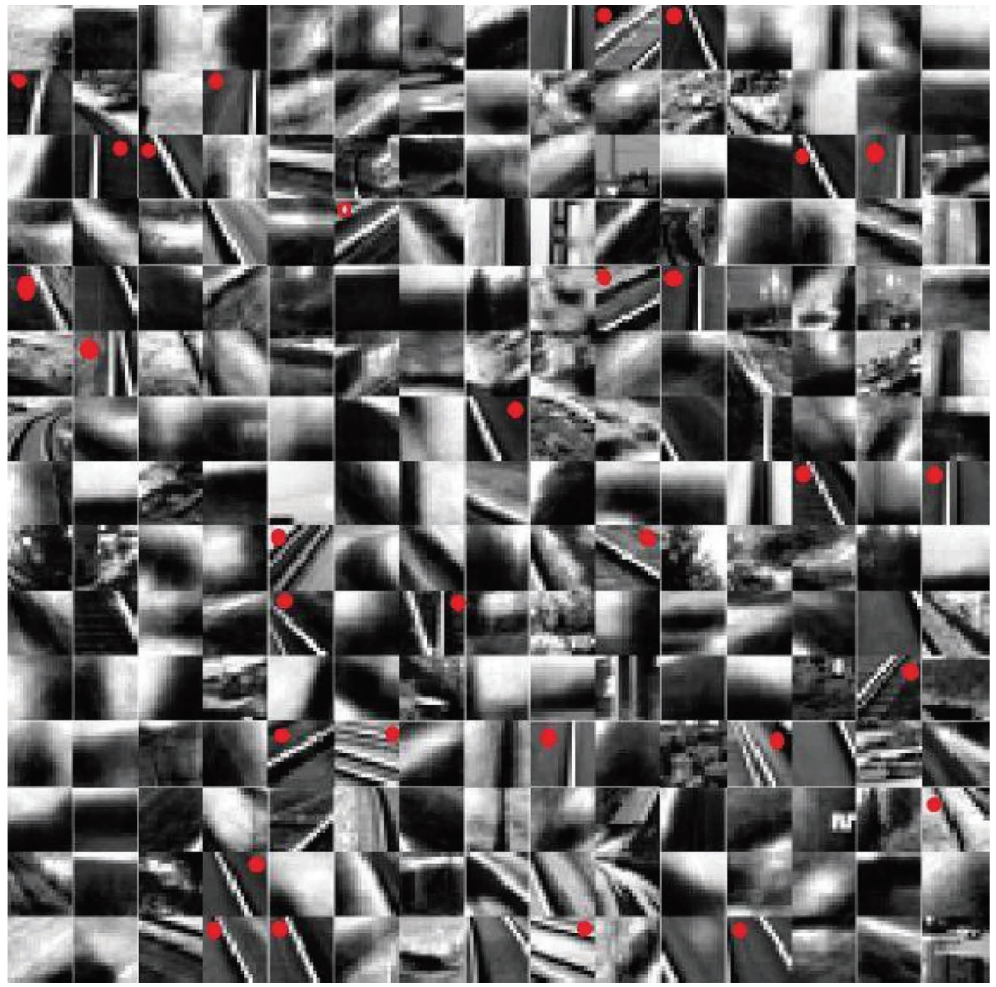

Figure 13: Examples of the visual words in the summer-fall dataset. Red circles indicate the visible words corresponding to the railway track.

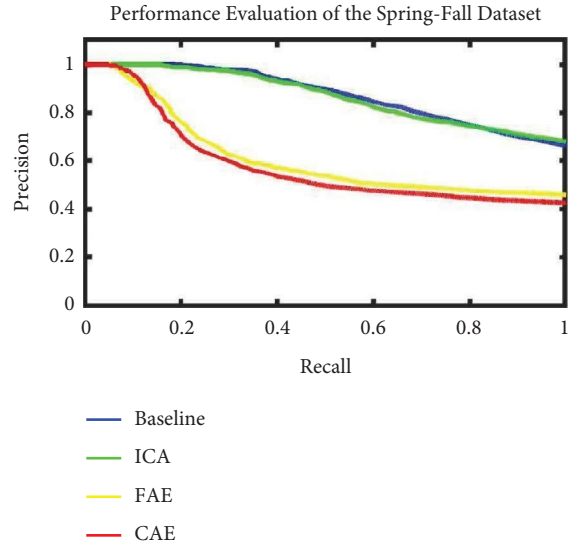

Figure 14: Precision-recall curves of the baseline and proposed algorithms on the spring-fall dataset.

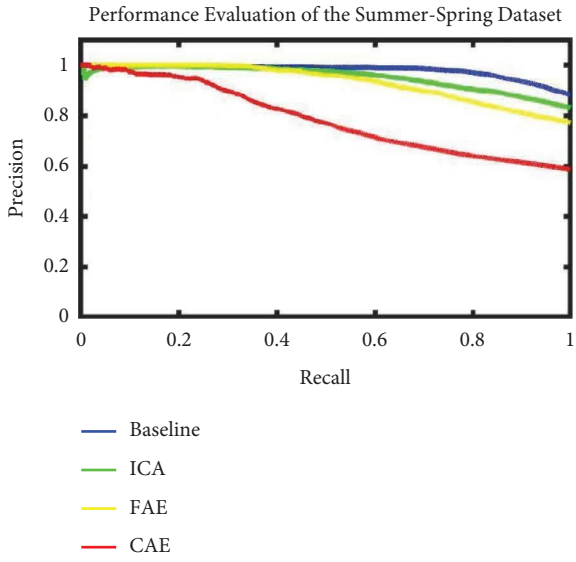

Figure 15: Precision-recall curves of the baseline and proposed algorithms on the summer-spring dataset.

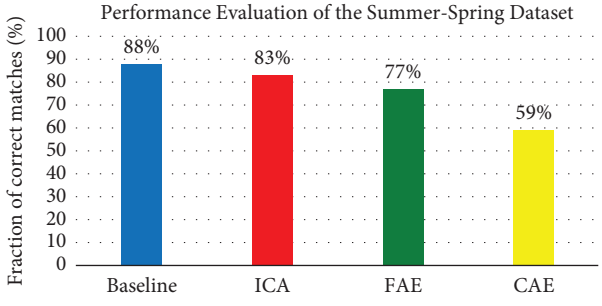

Figure 16: Fraction of correct matches of the baseline and proposed algorithms on the summer-spring dataset.

4.3.5. Effect of the Training Dataset on Robust Learning. Figure 18does not show a noticeable difference in the performance;the ICA model shows stable results, so the ICA model is not affected by the training dataset. In contrast, the baseline algorithm stays almost unchanged until the training dataset is prepared from three different seasons' images (summer, winter, and spring). When the training dataset contains images from all four seasons, i.e., summer, winter, spring, and fall, there is a decline in the percentage of correct matches for the baseline algorithm.

4.3.6. Robust Feature Learning Using Auto-Encoders. Figure 19 shows the results from the three datasets: summer-fall, spring-fall, and winter-summer. It can be seen that the encoded representation and the output of the network are less robust compared to the normalized patch representation. The CAE model also follows the same approach to extract the robust features, evidencing a similar pattern. As a result, the normalized patch representations are used in all the test datasets to find the best match between images. The Output + Patch normalization approach shows better results than the other two methods. The same scenario is observed in the convolutional auto-encoder. Consequently, the normalized patch representation of the output image is used as a condition-invariant feature.

4.4. Runtime Analysis. The required time to compile the models and to extract the condition-invariant features of a test image is shown in Tables 6 and 7, respectively. The feature extraction time can be reduced using powerful GPUs. The CAE model takes a long time to produce the output because of the high number of convolutional filters in the encoder and decoder parts. As in the ICA model, all the ICs have been used without reducing the dimensions; hence, the PCA-based baseline algorithm performs slightly faster than the ICA model.

5. Discussion

This work aims to learn the robust representations of the view. So that, when a scene goes through significant perceptual changes, it will still be possible for an autonomous robot to recognize its previously visited locations. This problem is known as the loop-closure detection problem. Correct loop-closure detection plays a significant role in correcting the errors of the map of an environment in an autonomous navigation model. Two possible methods have been proposed to learn the condition-invariant features of a scene in this work: the Independent Component Analysis (ICA) and the Convolutional autoencoders. The learned condition-invariant features are used to find the image match or detect the loop. This work perceives that the proposed algorithms (especially the ICA algorithm) outperform or perform up to a considerable level in the summer, fall, and spring seasons, i.e., where there are no extreme perceptual changes in the appearance of the scenes.

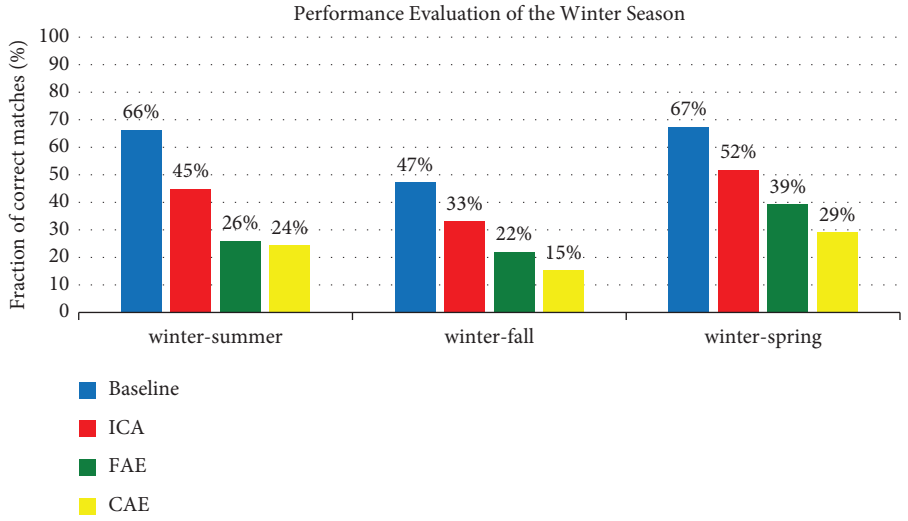

Figure 17: Fraction of correct matches of the baseline and proposed algorithms on the winter season dataset.

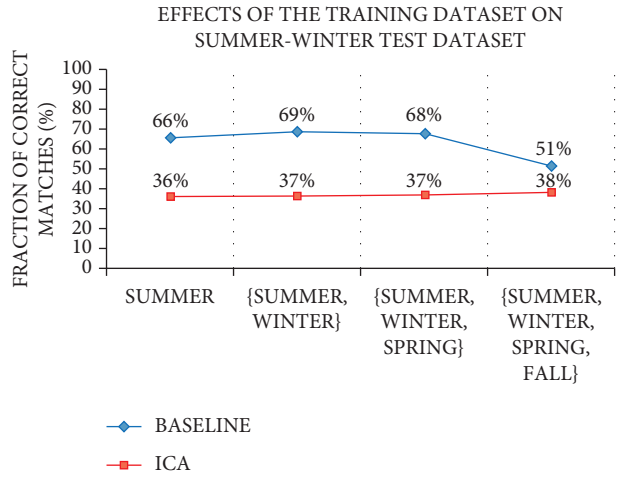

Figure 18: "SUMMER" indicates that the training images are taken from the summer season. "{SUMMER, WINTER}" suggests that the training dataset contains images from the summer and winter seasons. Similar interpretations apply for the other two training datasets.

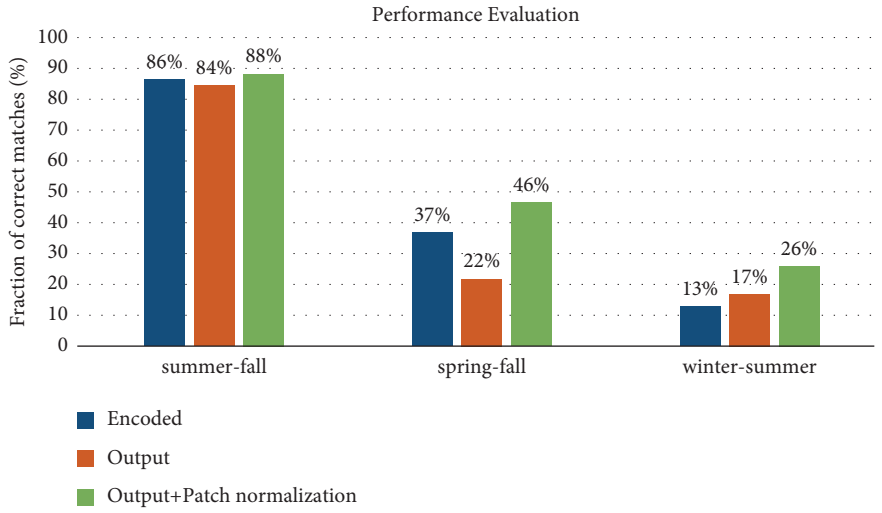

Figure 19: This result is obtained using the fundamental auto-encoder.

Table 6: Compilation times of the models.

Models	Required times
Baseline	1 hour 32 min 28 sec.
ICA	2 hour 56 min 48 sec.
FAE	4 hour 49 min 53 sec.
CAE	6 hour 43 min 21 sec.

Table 7: Required time to extract the condition-invariant features of a test image.

Models	Required times (ms)
Baseline	0.8
ICA	12
FAE	32.87
CAE	56.54

The proposed ICA algorithm performs better than the baseline algorithm in the summer-fall dataset with 91% of correct loop-closure detections.

6. Conclusion

Visual place recognition should be a trendy research area in computer vision, particularly in SLAM. This area has realized important progress. Still, it is a very questionable field due to the change in the presentation of demonstrations in various ways, to varying weather, seasons, light, etc. Auto-encoders may be convenient algorithms to learn the healthy visage of the neighborhoods in challenging atmospheres accompanying the appropriate amount of preparation dossier. The minor variation in the preparation dataset is noticed all at once as the attainable reason for the poor efficiency of the projected auto-encoders. The preparation dataset can be renovated by accumulating enough dossiers covering different atmospheres, including seasons, posture changes, illumination, etc. [38–40].

Data Availability

The data were acquired from raw images related to zip files and are available on an open-access SPM website named Attention to Visual Motion Data Set. The link is https://www.fil.ion.ucl.ac.uk/spm/data/attention/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. M. Lowry and M. J. Milford, "Supervised and unsupervised linear learning techniques for visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 600–613, 2016.

[2] W. E. L. Grimson and T. Lozano-Perez, "Localizing overlapping parts by searching the interpretation tree," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 469–482, 1987.

[3] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[4] R. Basri and D. W. Jacobs, "Recognition using region correspondences," *International Journal of Computer Vision*, vol. 25, no. 2, pp. 145–166, 1997.

[5] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 87–119, 1995.

[6] C. Schmid and R. Mohr, "Local gray value invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–535, 1997.

[7] M. Hannan, M. R. Islam, M. A. Haque, M. S. Hossain, A. Ul-Haq, and J. J. Sawan, "Automated face detection, recognition and gender estimation applied to person identification," *Journal of Computer Science*, vol. 15, no. 3, pp. 395–415, 2019.

[8] J. H. Waterberg, "The Lowry method for protein quantitation," in *The protein protocols Handbook*, pp. 7–10, Springer, Totowa, NJ, 2009.

[9] S. Marcel, M. S. Nixon, and S. Z. Li, *Handbook of Biometric Antispoofing. Volume 1*, Springer, London, 2014.

[10] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *Proceedings of the 2011 international joint conference on Biometrics (IJCB)*, pp. 1–7, IEEE, Washington, DC, USA, December 2011.

[11] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proceedings of the 2012 BIOSIGproceedings of the international conference of biometrics special interest group (BIOSIG)*, pp. 1–7, IEEE, Darmstadt, Germany, September 2012.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[13] K. M. Hasib, M. A. Habib, N. A. Towhid, and M. I. H. Showrov, "A novel deep learning based sentiment analysis of twitter data for us airline service," in *Proceedings of the 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pp. 450–455, IEEE, Dhaka, Bangladesh, February 2021.

[14] A. Ivakhnenko and G. L. Valentin, "Cybernetic predicting devices," CCM Information Corp, New York, 1965.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[16] T. H. Wang, H. J. Huang, J. T. Lin, C. W. Hu, K. H. Zeng, and M. Sun, "Omnidirectional CNN for visual place recognition and navigation," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2341–2348, IEEE, Brisbane, QLD, Australia, May 2018.

[17] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, p. 12, 1991.

[18] M. T. Habib, S. B. Shuvo, and M. S. Uddin, "Automated textile defect classification by bayesian classifier based on statistical features," in *Proceedings of the 2016 International Workshop on Computational Intelligence (IWCI)*, IEEE, Dhaka, Bangladesh, December 2016.

[19] J. L. Raheja, S. Kumar, and A. Chaudhary, "Fabric defect detection based on GLCM and Gabor filter: a comparison," *Optik*, vol. 124, no. 23, pp. 6469–6474, 2013.

[20] R. S. Sabeenian, "Fabric defect detection using discrete curvelet transform," *Procedia Computer Science*, vol. 133, pp. 1056–1065, 2018.

[21] Z. Pourkaramdel, S. Fekri-Ershad, and L. Nanni, "Fabric defect detection based on completed local quartet patterns and majority decision algorithm," *Expert Systems with Applications*, vol. 198, Article ID 116827, 2022.

[22] E. Jarallah and M. Alsaffar, "Isolation and characterization of lytic bacteriophages infecting *Pseudomonas aeruginosa* from sewage water," *Scientific Reports*, vol. 9, pp. 220–230, 2016.

[23] B. Bellekens, V. Spruyt, R. Berkvens, R. Penne, and M. Weyn, "A benchmark survey of rigid 3d point cloud registration algorithms," *Int. J. Adv. Intell. Syst*, vol. 8, pp. 118–127, 2015.

[24] Z. Chen, A. Jacobson, N. Sünderhauf et al., "Deep learning features at scale for visual place recognition," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3223–3230, IEEE, Singapore, May 2017.

[25] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE international conference on computer vision*, pp. 3456–3465, IEEE, Venice, Italy, December 2017.

[26] C. Hentschel, S. Stober, A. Nurnberger, and M. Detyniecki, *Adaptive Multimedial Retrieval: Retrieval, User, and Semantics*, pp. 45–56, Springer-Verlag, Berlin, Heidelberg, 2008.

[27] L. Zhang and J. Ma, "Image Annotation by Incorporating word correlations into multi-class SVM," *Fifth International Conference on Natural Computation Bd*, vol. 1, pp. 516–520, 2009.

[28] H. Yi and T. Wenbin, "Experimental analysis on classi_-cation of unmanned aerial vehicle images using the probabilistic latent semantic analysis," in *Proceedings of the The International Society for Optical Engineering*, SPIE, October 2009.

[29] N. M. Ali, S. W. Jun, M. S. Karis, M. M. Ghazaly, and M. S. M. Aras, "Object classi_cation and recognition using Bag-of-Words (BoW) model," in *Proceedings of the IEEE 12th International Colloquium on Signal Processing Its Applications (CSPA)*, pp. 216–220, IEEE, Melaka, Malaysia, March 2016.

[30] C. Feng and X. Wang, "Image retrieval system based on bag of view words model," in *Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1–4, IEEE, Okayama, Japan, June 2016.

[31] F. Zeng, Z. Huang, and Y. Ji, "Discriminative bag-of-words-based adaptive appearance model for robust visual tracking," *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 907–911, 2017.

[32] M. T. Islam and S. M. R. Islam, "A new image quality index and its application on MRI image," *International Journal of Image, Graphics and Signal Processing*, vol. 13, no. 4, pp. 14–32, 2021.

[33] S. M. R. Islam, M. T. Islam, and X. Huang, "A new approach of image quality index," in *Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, pp. 223–228, IEEE, Dhaka, Bangladesh, September 2017.

[34] K. M. Hasib, N. A. Towhid, and M. R. Islam, "Hsdlm: a hybrid sampling with deep learning method for imbalanced data classification," *International Journal of Cloud Applications and Computing*, vol. 11, no. 4, pp. 1–13, 2021b.

[35] M. R. Islam, I. Razzak, X. Wang, P. Tilocca, and G. Xu, "Ucbvis: understanding customer behavior sequences with visual interactive system," in *Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Shenzhen, China, July 2021.

[36] K. M. Hasib, M. Iqbal, F. M. Shah et al., "A survey of methods for managing the classification and solution of data imbalance problem," 2020, https://arxiv.org/abs/2012.11870.

[37] A. J. Bell and T. J. Sejnowski, "The "independent components" of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[38] M. T. Islam and A. N. Tusher, "Automatic detection of Grape, Potato and Strawberry Leaf Diseases using CNN and image processing," in *Data Engineering for Smart Systems*, P. Nanda, V. K. Verma, S. Srivastava, R. K. Gupta, and A. P. Mazumdar, Eds., Springer, Singapore, 2022.

[39] D. S. Islam, S. M. R. RobaiatMou, and M. T. Islam, "Design and implementation of low cost ECG monitoring system for the patient using smart device." in *Proceedings of the 2017 International Conference on Electrical Computer and Communication Engineering (ECCE)*, Cox's Bazar, Bangladesh, pp. 774–778, IEEE, February 2017.

[40] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of features: spatial Pyramid matching for recognizing natural scene Categories," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) Bd*, vol. 2, pp. 2169–2178, 2006.