Applied Soft Computing Journal

Road Pothole Detection based on Pattern Recognition Framework integrating SIFT Feature Extraction and Bag of Visual Words

--Manuscript Draft--

Manuscript Number:	ASOC-D-23-07049
Article Type:	Full Length Article
Keywords:	Bag of Visual Words (BOVW), FeedForward Neural Network (FFNN), Local Binary Pattern (LBP), Pothole Detection, Scale Invariant Feature Transform (SIFT)
Corresponding Author:	Harsh S Dhiman Symbiosis International University Symbiosis Institute of Technology INDIA
First Author:	Dev Bhanushali
Order of Authors:	Dev Bhanushali
	Harsh Kotadiya
	Harsh S Dhiman
Abstract:	Road potholes pose a significant threat to transportation safety, causing accidents and damage to vehicles. In this paper, we propose two methods to detect road potholes based on the captured image. The first approach is based on the Bag of Visual Words (BOVW) using Scale Invariant Feature Transform (SIFT) where Count Vectors as features using K-Means clustering are extracted and re-weighted using Term Frequency-Inverse Document Frequency (TFIDF) Transformation. The second approach utilizes Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP) features independently and in conjunction. Further, we compare the performance of K-Nearest Neighbor (KNN), Support Vector Machines (SVM), and Random Forests (RF) in classifying potholes and non-pothole images. We also look at how feedforward Neural Networks can be utilized for pothole detection using previously extracted SIFT features. The experimental results on images featuring elements other than potholes, such as cars, pedestrians, and debris with varying orientations, are promising.

Road Pothole Detection based on Pattern Recognition Framework integrating SIFT Feature Extraction and Bag of Visual Words

Dev Bhanushali[a], Harsh Kotadiya[a], Harsh S. Dhiman*[a]

[a]*Department of Artificial Intelligence and Machine Learning, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, India*

Abstract

Road potholes pose a significant threat to transportation safety, causing accidents and damage to vehicles. In this paper, we propose two methods to detect road potholes based on the captured image. The first approach is based on the Bag of Visual Words (BOVW) using Scale Invariant Feature Transform (SIFT) where Count Vectors as features using K-Means clustering are extracted and re-weighted using Term Frequency-Inverse Document Frequency (TFIDF) Transformation. The second approach utilizes Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP) features independently and in conjunction. Further, we compare the performance of K-Nearest Neighbor (KNN), Support Vector Machines (SVM), and Random Forests (RF) in classifying potholes and non-pothole images. We also look at how feedforward Neural Networks can be utilized for pothole detection using previously extracted SIFT features. The experimental results on images featuring elements other than potholes, such as cars, pedestrians, and debris with varying orientations, are promising.

Keywords: Bag of Visual Words (BOVW), FeedForward Neural Network (FFNN), Local Binary Pattern (LBP), Pothole Detection, Scale Invariant Feature

*Corresponding author

Email addresses: dev.bhanushali.btech2022@sitpune.edu.in (Dev Bhanushali), harsh.kotadiya.btech2022@sitpune.edu.in (Harsh Kotadiya), harsh.dhiman@sitpune.edu.in (Harsh S. Dhiman*)

Transform (SIFT).

1. Introduction

Roads are the most widely used transportation medium. Changes in weather, constant traffic, and other environmental factors contribute to significant degradation of roads over time. This leads to safety hazards for commuters and poses a significant financial burden to various service-providing companies that use automotive transport daily for repair and maintenance. A large amount of time is spent identifying and addressing formed potholes, which can be reduced if the detection of such defects can be automated.

1.1. Related works

The past couple of decades have seen great progress in the formation of different approaches to solve this problem. A fair amount of point-cloud model generation-based methods are developed. Stereo-vision-based systems have been tried [1, 2] where road manifolds are modeled, and potholes are detected by identifying areas classified as 'below road level.' Taking this idea further, deep learning algorithms have been applied to classify frames of data containing potholes [3]. Zhang and Elaksher [4] talked about sparse 3D road geometry reconstruction using SfM (Structure from Motion) and refinement using bundle adjustment. Road potholes are then found using distinguishable features. Du et al. [5] were able to incorporate surface normal information into the road modeling process. K-means clustering and region growth algorithms were used to detect road potholes. In [6], a Light Detection and Ranging (LiDAR) based road pothole detection system was introduced where the 3D road points are classified as damaged and undamaged by distances of these points from the best fitting planar road surface. Mobile sensing systems equipped with accelerometers have been tried and tested [7], which allows pothole detection by learning of signal irregularities.

Some image-based classification methods are as follows. Image-based systems applying threshold and morphological operations [8] have produced masks

representing pothole regions from road images. Possible road pothole contours are extracted based on geometric properties, confirmed by an ordered histogram intersection method. Koch and Brilakis [9] proposed a method in which a road image is segmented into damaged and undamaged regions using histogram-based thresholding methods. The damaged areas are further processed with elliptic regression, and potholes are detected by comparing road textures inside and outside the ellipse. Lin and Liu [10] have talked about extracting the average gray level, contrast, consistency, entropy, and third-order moments from grayscale-converted road pothole images and training NL – SVM models on these features. Hadjidemetriou et al. [11] proposed a method where road pavement images are divided into square blocks, and the SVM classifier is trained on extracted feature vectors. These feature vectors consist of the histogram and two texture descriptors using the Gray-Level Co-Occurrence Matrix. Pan et al. [12] proposed a method that extracted spectral, geometric, and textural features. Using these features, ANN, RF, and SVM models are trained for road image classification. Jakštys et al. [13] have proposed a system that uses triangle and adaptive thresholding methods to segment road images and extract pothole contours using a heuristic edge detection approach. In [14], Otsu's thresholding method is used to segment road images, which are then processed with morphological filters before performing the distance transform. This is later used in the watershed algorithm to detect potholes. In [15], Moazzam et al. analyzed road depth distribution for different azimuth and elevation angles, and the approximate volume of each road pothole was calculated using the trapezoidal rule.

A fair number of object detection-based methods have also been developed. Suong et al. [16] were able to employ two object detection networks, namely F2-Anchor and Den-F2-Anchor, which are based on YOLOv2 for detecting potholes from color images. Dharneeshkar et al. [17] could use YOLOv2, YOLOv3, and YOLOv3 Tiny on color images of road potholes and achieve high mAP, precision, and recall. Ukhwah et al. [18] used YOLOv3, YOLOv3 Tiny, and YOLOv3 SPP for pothole detection on grayscale road images, out of which TOLOv3 SPP was

able to achieve the best overall performance. Kortmann et al. [19] first trained a classifier to distinguish road potholes by their country and then trained FASTER R-CNN for each country for road potholes. Gupta et al. [20] trained two Single Shot detectors (with ResNet-34 and ResNet-50 as backbone networks) on thermal road images, between which ResNet-50 performed significantly better. In [21], Maeda et al. trained two deep-convolution neural networks (Inception V2 and MobileNet as the backbone networks) on color road images captured through smartphones.

1.2. Contributions of this work

The major contributions of this work are summarized as follows.

(i) A novel technique utilizing Scale invariant feature transform (SIFT) feature extraction and bag of visual words is introduced for road pothole detection.

(ii) SIFT feature extraction is compared with baseline Histogram of oriented gradients (HOG) and Local binary pattern (LBP) algorithm.

(iii) Robustness of SIFT algorithm in terms of scale and rotational invariance is tested using three machine learning techniques, namely, KNN, SVM, and Neural network.

The paper is organized as follows. Section 2 discusses the research methodology in brief and describes the working of different feature extraction techniques used. Section 3 looks at the results and how machine learning models are employed for the two methods. In addition, we also address the performance of feedforward neural networks when incorporating previously extracted features using the SIFT features and the bag-of-visual-words algorithm. Section 4 brings the paper to a conclusion and discusses further possibilities of experimentation.

2. Methodology

In this section, we discuss the proposed approach used to classify road pothole images as illustrated in the Figure. 1. Images of plain and pothole images

are acquired from an online dataset [22], which contains raw images of varying sizes and formats. Following data preprocessing, the methodology branches into two paths, each corresponding to different methods. The first method revolves around the extraction of SIFT features for a given image and the generation of vocabulary using the BOVW method. With SIFT feature extraction, unique descriptors per given image are computed, and their occurrences in the entire image data set are calculated. Further, the descriptor clusters are re-weighted using the term-frequency inverse document frequency (TFIDF) transformation. The second method involves extracting HOG and LBP features from images. Machine learning algorithms are trained using features extracted from both approaches, and their performances are compared quantitatively in accuracy. A detailed explanation of the feature extraction algorithms such as SIFT, BOVW, HOG, and LBP is discussed in the subsequent sections.

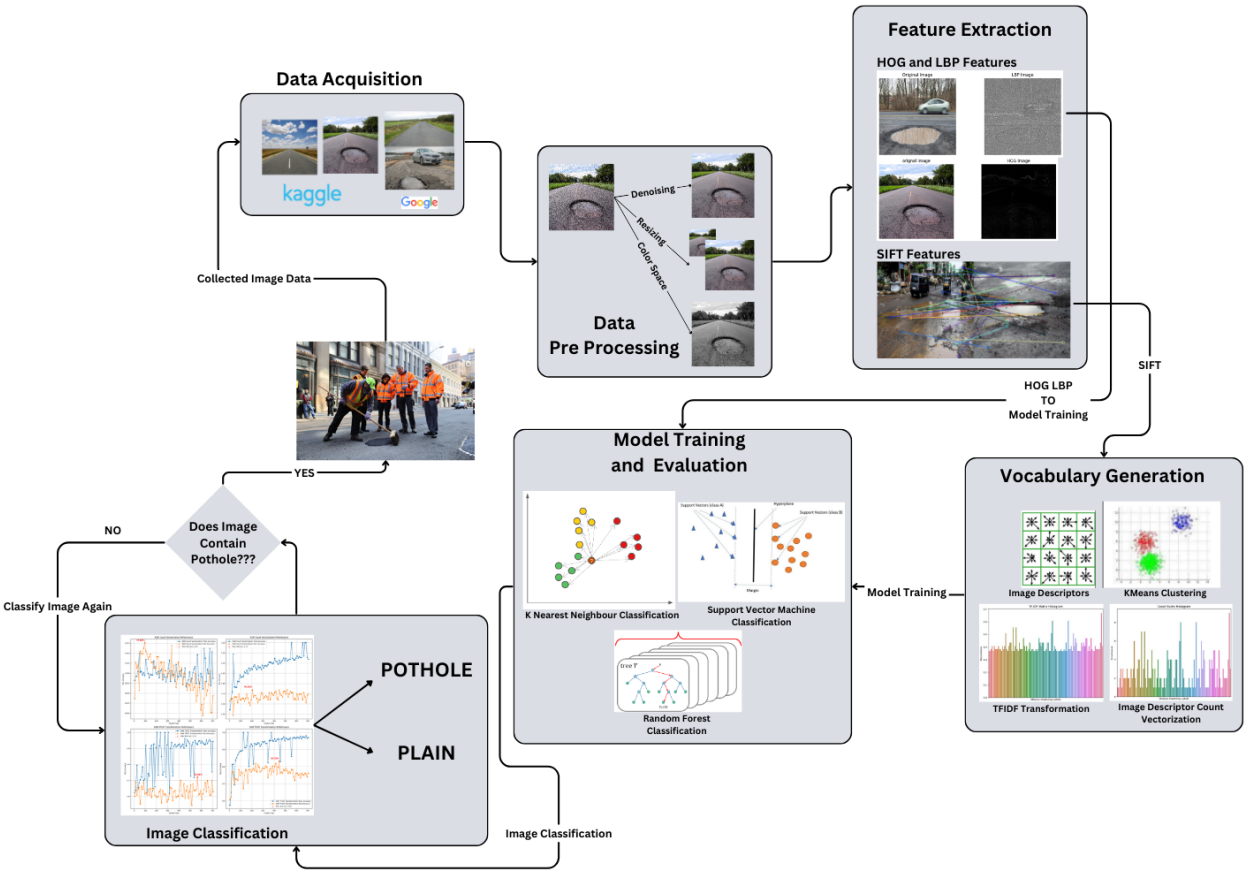

Figure 1: Systematic flowchart for road pothole detection

2.1. Feature Extraction Techniques

In 1999, David Lowe proposed an algorithm to detect and match local image features [23]. The significant steps of this algorithm are described below. Given image $\mathbf{I}(x, y)$, a scale space of it is constructed using different intensities of Gaussian blur $\mathbf{G}(x, y, \sigma_i)$ convoluted with the original image described in eq. (1). We then subtract nearby scales to find the difference of Gaussian $\mathbf{D_i}(x, y)$ and stack them as described in eq. (2).

$$I'(x, y) = G(x, y, \sigma_i) * I(x, y) \tag{1}$$

$$D_i(x, y) = I'_i(x, y) - I'_{i+1}(x, y) \tag{2}$$

This process is repeated for each octave, where the images are sampled down across the octaves by a factor of 2, as illustrated in Figure. 2.

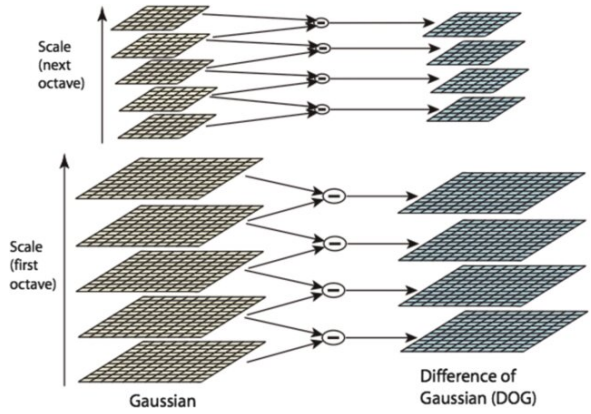

Figure 2: Difference of Gaussian produced by subtracting consecutive pairs of image copies with varying Gaussian blur [24]

If a given point in a scale is an extremum compared to the eight neighbors in the same scale and nine neighbors one scale above and below, as seen in Figure. 3, it becomes a potentially keypoint.

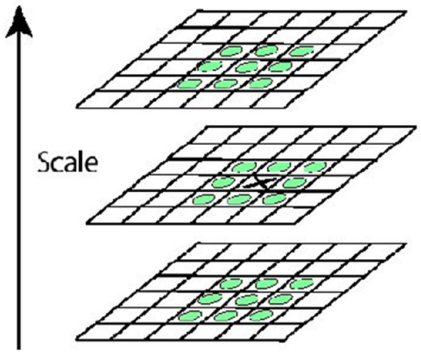

Figure 3: Representation of 26 neighbors to a pixel in one scale [24]

Keypoints that lie too close to or on edges or arise due to noise in the image are filtered out. Finally, keypoints that are stable in the image scale space. This process of selecting keypoints across octaves brings the property of scale invariance.

To make a keypoint rotation invariant, the magnitude of the gradient is ignored, and only the orientation is considered. Given a keypoint from an image $\mathbf{I}(x, y)$, the orientation θ for a point located at coordinates (x, y) is computed as follows.

$$\Theta\left(x, y\right) = tan^{-1}\left(g_x / g_y\right) \tag{3}$$

where gradient in the Y direction g_y is defined as

$$g_y = I\left(x, y+1\right) - I\left(x, y-1\right) \tag{4}$$

and gradient in the X direction g_x is defined as

$$g_x = I\left(x+1, y\right) - I\left(x-1, y\right) \tag{5}$$

A histogram of orientations θ with 36 bins covering a 360-degree range is constructed, where the bin corresponding to the highest peak is associated with the keypoint as its general orientation.

Every keypoint has an associated descriptor that describes information relevant to its surrounding region. A patch of 16x16 pixels with a keypoint as the center is considered, which is broken down into uniform cells of 4x4 pixels totaling 16 cells. A histogram of orientations is calculated for each pixel with eight bins covering a 360-degree range for each cell, yielding sixteen 1x8 dimensional vectors. These vectors are concatenated and form 1x128 dimensional descriptors.

2.2. The Bag of Visual Words Algorithm

For every image in the dataset, SIFT descriptors $\mathbf{d_i}$ are computed. Because every image can have a different number of keypoints, the descriptor count of each image varies. Finally, we can generalize that a descriptor matrix $\mathbf{A_j}$ of size

$\mathbf{n} \times 128$ can be computed from a given image $\mathbf{I_j}\,(x, y)$ described as

$$A_j = \begin{bmatrix} d_{1j} \\ d_{2j} \\ d_{3j} \\ \vdots \\ d_{nj} \end{bmatrix} \tag{6}$$

Here, $\mathbf{d_{ij}}$ is a row vector of shape $\mathbf{1} \times 128$ and the entire descriptor matrix $\mathbf{A_j}$ contains n such row vectors. Descriptors from all images are then concatenated into a large database of descriptors, which results in a descriptor matrix $\mathbf{A}$ of $\mathbf{m} \times 128$. To cut down computation and the likelihood of overfitting later on, we reduce the given matrix using Principal component analysis (PCA) [25]. Here, $\mathbf{m}$ is the number of descriptor row vectors aggregated across all images from the dataset.

Moving forward, a covariance matrix $\mathbf{C}_{128\times128}$ of descriptor matrix $\mathbf{A}_{m\times128}$ is constructed after column standardization.

$$C_{128\times128} = \frac{1}{m-1} \times \left(A^T \cdot A\right) \tag{7}$$

Further, we perform eigenvalue decomposition to get the eigenvalue matrix $\mathbf{\Lambda}$ and eigenvector matrix ∇ from the following relation.

$$C = \Lambda \cdot \nabla \cdot \Lambda^T \tag{8}$$

Eigenvectors are then sorted by their eigenvalues and the top $\mathbf{K}$ number of eigenvectors $\mathbf{v}_{128\times1}$ are picked into a matrix $\mathbf{W}_{128\times K}$ having the most significant explained variance ratio and whose cumulative explained variance ratio sums up to 0.95 or retains 95% of total variance as shown graphically in Figure. 4.

$$W_{128\times K} = [v_1,\, v_2,\, v_3,\, \cdots\, v_k] \tag{9}$$

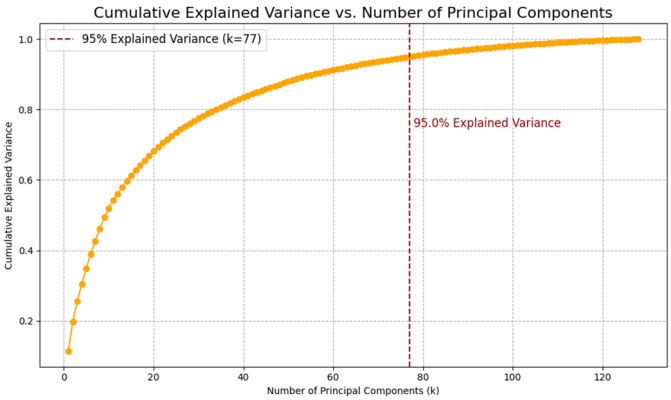

Figure 4: Visualization of the Cumulative Explained Variance Ratio across principal components.

Finally, the reduced descriptor matrix $\mathbf{A}_{reduced}$ is constructed as

$$A_{reduced} = A_{m\times 128} \cdot W_{128\times K} \tag{10}$$

Descriptors are grouped into clusters, which will be used to construct a histogram of descriptor clusters for each image. We start by taking Inertia or WCSS (Within Cluster Sum of Squared Distances) as the performance metric and perform K-Means clustering [26] on the reduced descriptor matrix for a varying number of clusters $\mathbf{K}$.

Due to the absence of any obvious inflection point depicted in Figure. 5, further computation is performed for all values of $\mathbf{K}$.

For a given image $\mathbf{I_i}\,(x, y)$ a count vector $\mathbf{H_{1\times K}}$ representing occurrences of descriptor classes can be constructed using the trained K-Means model. Each extracted descriptor from an image is given a class label, and their frequencies are recorded. Let cluster label be denoted by $\mathbf{c_i}$ which represents the label for the $\mathbf{i^{th}}$ cluster from a label array $\mathbf{c}$ of size $\mathbf{1} \times K$.

$$H\left(I\left(x,y\right),c\right)_{1\times K} = \left[freq\left(c_1\right), freq\left(c_2\right), \ \cdots \ freq\left(c_K\right)\right] \tag{11}$$

It is essential to address other elements in the dataset, such as cars and pedes-

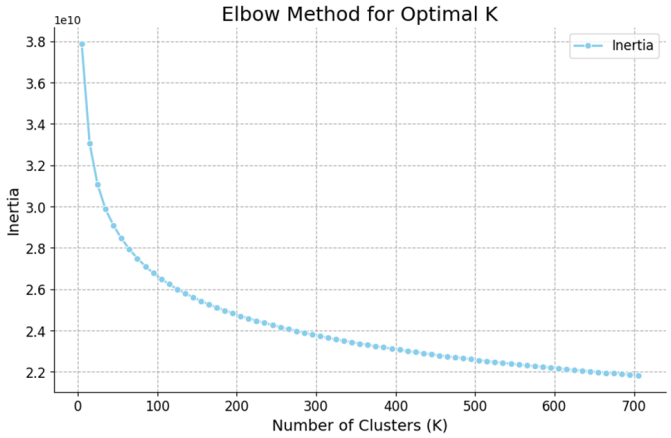

Figure 5: Inertia or Within Cluster Sum of Squared Distances metric calculated for K-Means cluster sizes ranging from 5 to 705 in intervals of 10.

trians, whose influence on model predictions should be minimized. This can be done using TFIDF [27] re-weighting, where descriptor classes that occur more frequently across images are given less weight using the following relation.

$$TFIDF\left(c\right)_{1\times K} = TF\left(c_i\right) \times log\left(\frac{N}{DF\left(c_i\right)}\right) \tag{12}$$

Here, $\mathbf{TF}\left(\mathbf{c_i}\right)$ is the term frequency of the $\mathbf{i}^{th}$ cluster or an array containing previously calculated Count Vectors $\mathbf{H}\left(\mathbf{I}\left(\mathbf{x},\mathbf{y}\right),\mathbf{c}\right)_{\mathbf{1\times K}}$ for all images, $\mathbf{N}$ is the number of images and $\mathbf{DF}\left(\mathbf{c_i}\right)$ is an array containing document frequencies or the count of images contain occurrences of the $\mathbf{i}^{th}$ descriptor cluster. This leaves us with two sets of feature vectors for each image: a count vector and a TFIDF-transformed vector.

2.3. HOG feature extraction

Histogram of oriented gradients (HOG), initially introduced by Triggs and Dalal [28] is a gradient-based feature extraction method. First, gradient magnitude μ and orientation θ for each pixel $\mathbf{I}\left(x,y\right)$ in the image is calculated using the following formulae.

$$\Theta\left(x,y\right) = tan^{-1}\left(g_x/g_y\right) \tag{13}$$

where the gradient in the Y direction g_y is defined as

$$g_y = I\left(x,y+1\right) - I\left(x,y-1\right) \tag{14}$$

gradient in the X direction g_x is defined as

$$g_x = I\left(x+1,y\right) - I\left(x-1,y\right) \tag{15}$$

and magnitude μ is defined as

$$\mu = \sqrt{g_x^2 + g_y^2} \tag{16}$$

Additionally, the image is divided into 8x8 pixels cells and a histogram is calculated for each cell. Each bin corresponds to gradient orientations ranging from 0 to 180 degrees with a bin size of 20 degrees, resulting in a feature vector of length 1x9. For each pixel in a cell, values are added to bins according to their calculated magnitude and orientation. Suppose the calculated orientation can be described as the center of range of each bin or as odd multiples of 10 degrees. In that case, the value of the bin alone increases with the magnitude associated with that intensity. For instance, if the calculated orientation deviates from the center of range of that bin, the values are distributed proportionately into the current and the neighboring bin, as seen in Figure. 6, using the following formulae.

$$value\ assigned\ to\ n^{th}\ bin = \mu\left(\frac{C_{n+1} - \theta}{\Delta\theta}\right) \tag{17}$$

$$value\ assigned\ to\ n+1^{th}\ bin = \mu\left(\frac{\theta - C_n}{\Delta\theta}\right) \tag{18}$$

Here, $\mathbf{C}_n$ is the center of range of the $\mathbf{n}^{th}$ bin, $\mathbf{\Delta}\theta$ is the bin size, μ is the calculated magnitude, θ is the calculated orientation. In continuation, we now

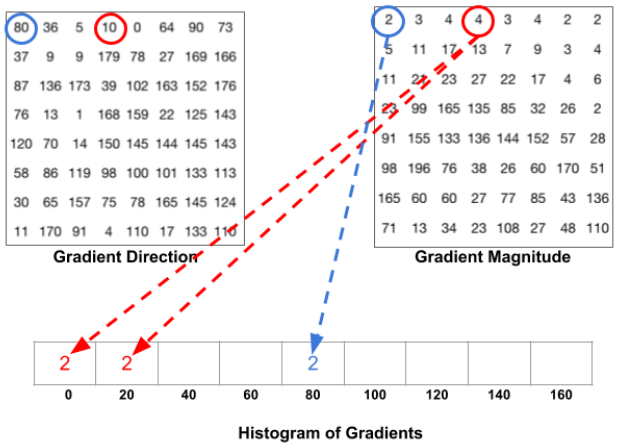

Figure 6: Gradient magnitude of a pixel distributed proportionately to deviation from bin center [29].

consider all cells in groups of 2x2, resulting in larger 16x16 blocks or a concatenated histogram producing a 1x36 feature vector. Block normalization on each block is performed to reduce the effect of lighting on the resulting feature vector. For a given feature vector $V = [b_1, b_2, \cdots b_{36}]$, the normalization factor **K** is calculated as follows.

$$K = \sqrt{b_1^2 + b_2^2 + \cdots b_{36}^2}, \tag{19}$$

where the resultant feature vector is calculated as

$$V_{normalized} = \left[\frac{b_1}{K}, \frac{b_1}{K}, \frac{b_3}{K}, \cdots \frac{b_{36}}{K}\right] \tag{20}$$

Finally, these 1 x 36 feature vectors are concatenated for all 16 x 16 blocks in the image, giving the final HOG feature.

2.4. Local binary pattern feature extraction

Local binary pattern (LBP) is a computationally efficient texture-based feature extraction method introduced by Timo Ojala in 1994 [30]. An LBP feature vector can be calculated by comparing the intensity of the center pixel $(\mathbf{x_c}, \mathbf{y_c})$ with its

neighboring 8 pixels. The number 0 is taken where the pixel's intensity is lower than the center pixel's intensity, and 1 in case its value is greater than or equal to the intensity of the center pixel.

$$S\left(i_n - i_c\right) = \begin{Bmatrix} 1, & i_n - i_c \geq 0 \\ 0, & i_n - i_c < 0 \end{Bmatrix} \tag{21}$$

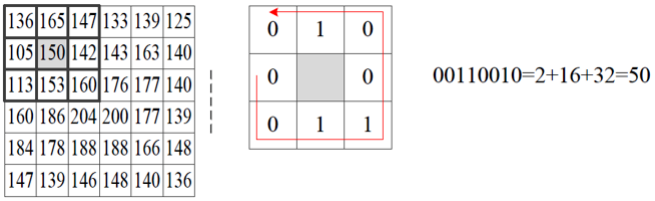

Figure 7: Pixel neighbors are binarized based on relative intensity and are followed counterclockwise to form a binary representation. This is further converted to a decimal number [31].

These numbers are followed around counterclockwise to form an 8-digit binary number, which is converted into a decimal number between the range of 0 to 255, as shown in Figure. 7.

$$LBP(x_c, y_c) = \sum_{n=0}^{7} S(i_n - i_c)2^n \tag{22}$$

This process is followed for every pixel in the image, and a histogram of these decimal numbers is calculated.

2.5. HOG and LBP Feature concatenation

All images from the dataset are converted to dimensions of 645×645 in grayscale color space. HOG and LBP Features are extracted from every image and stored in a database of features. From the extracted HOG $\mathbf{H}$ and LBP $\mathbf{L}$ features, three additional features are constructed. The third and fourth features are HOG and LBP features extracted from the original image $\mathbf{I}\,(x, y)$ after applying Fast Non-Local Means denoising algorithm [32] forming denoised image $\mathbf{I}'\,(x, y)$. Additionally, we perform PCA on the extracted HOG features from both Original

(**H**) and denoised Images ($\mathbf{H}_{denoised}$). The fifth feature **HL** is constructed by combining the denoised HOG feature $\mathbf{H}_{denoised}$ and the extracted LBP feature **L**. This is done by horizontally concatenating both features in the following manner.

$$HL = [H_{denoised},\ L] \tag{23}$$

3. Results and Discussion

The dataset employed for this research has approximately 350 images in each class - Plain and Pothole, as shown in Figure. 9. These images were originally scraped using the google-images-download scraper, naturally resulting in a raw dataset with images having non-uniform resolutions as seen in Figure. 8. It is significant to note that humans and vehicles feature prominently in a substantial portion of the images. Dataset images were resized to dimensions of 256×256 in gray-scale color space to reduce computation time.

Figure 8: Sample images from Plain and Pothole Dataset.

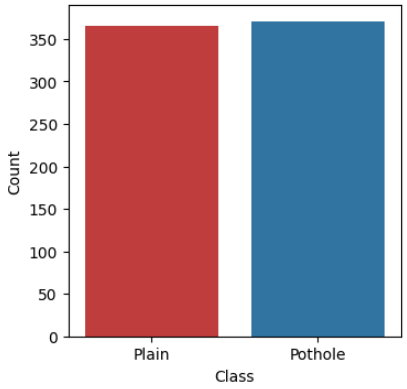

Figure 9: Class distribution of pothole and non-pothole images

The machine learning models along with the different hyperparameters used in the BOVW method are as follows. For model training, the K-Nearest Neighbor [33] and Support Vector Machine [34] classifiers were employed. While testing the KNN Classifier, N-Neighbors ranging upto 75 were used. Both Manhattan and Euclidean Distances were also considered for the evaluation. For SVM Classifier, investigation involved testing regularization parameters ranging from 10^{-2} to 10^{1} in power intervals of 10. Additionally both Linear and RBF kernels were tried to asses model performance. Finally, the train-test ratio for both machine learning models was 4:1. To predict novel images, they must first be converted to the size 256×256 in the gray-scale color space. SIFT descriptors are then extracted into a matrix, which is decomposed using the same PCA model used before, followed by constructing Count Vectors and TFIDF transformed Count Vectors using the same K-Means and TFIDF transformer models. As seen in Figure. 4, the cumulative explained variance ratio of 0.95 was found out to be at the 77^{th} principal component. KNN and SVM models are trained with Count vectors and TFIDF Transformed Count Vectors across K clusters ranging from 5 to 705 in intervals of 10, producing results as seen in Table. 1.

Table 1: Classification performance using SIFT features

Classification Algorithm	Extracted Feature	Test Accuracy	K-Means Cluster Size
KNN	Count Vectors	$77.62\,\%$	95
	TFIDF Vectors	$72.02\,\%$	565
SVM	Count Vectors	$76.92\,\%$	205
	TFIDF Vectors	$83.21\,\%$	445

Now taking a look at the HOG-LBP method, K-Nearest Neighbor, Support Vector Machine, and Random Forest classifiers were employed. For KNN and SVM classifier, the hyperparmaters tested were the same as that of the BOVW method. A wide variety of Hyperparameters were tested for the Random Forest classifier. The range of experiments included testing the number of trees with values of 50, 100 and 200. Additionally, the maximum depth of these trees were explored with values of None, 10, 20 and 30. The evaluation further considered the minimum samples split with values 2, 5 and 10 along with minimum samples per leaf with values 1, 2 and 4. Moreover, all the three models had a train-test ratio of 4:1. For prediction, novel images are resized to the resolution 645×645 in gray-scale color space, and another copy of this image is made which undergoes Fast Non-Local Means denoising. HOG and LBP features are extracted from both copies, following which we construct a concatenated HOG LBP feature. KNN, SVM, and Random Forest [35] models are trained on these features, producing results as outlined in Table. 2.

Table 2: Classification performance using HOG and LBP features

Classification Algorithm	Extracted Feature	Accuracy
KNN	HOG	$82.85\,\%$
	LBP	$80.00\,\%$
	HOG + LBP	$90.71\,\%$
	HOG Denoised	$90.71\,\%$
	LBP Denoised	$77.85\,\%$
SVM	HOG	$90.71\,\%$
	LBP	$84.28\,\%$
	HOG + LBP	$92.85\,\%$
	HOG Denoised	$92.85\,\%$
	LBP Denoised	$77.14\,\%$
Random Forest	HOG	$87.85\,\%$
	LBP	$85.71\,\%$
	HOG + LBP	$87.85\,\%$
	HOG Denoised	$94.28\,\%$
	LBP Denoised	$77.14\,\%$

3.1. Comparisons

In this manuscript, from the results of the BOVW method as seen in Table. 1, we observe that KNN performs better utilizing Count Vectors as its training feature, resulting in a test accuracy of 77.62% at K-Means cluster size 95 in comparison to TFIDF transformed vectors, while SVM performed better using TFIDF transformed Vectors producing test accuracy of 83.21% at K-Means cluster size 445. Based on results obtained after testing, the optimal hyperparameters for the KNN and SVM classifier are as follows. The N-Neighbors parameter for the KNN classifier is set to 50 adopting the power parameter as Euclidean distance, while the SVM classifier utilizes the RBF kernel and a regularization parameter of 1. Both models are trained on 5 fold cross validation. Overall, SVM with TFIDF transformed Count Vectors performed the best among the four possible combinations. It is speculated that this method can yield higher accuracy with increased

image dimension at the cost of higher computation time during feature preprocessing. Now, we will turn our attention to the alternative approach from Table. 2, we observe that HOG features extracted from Fast Non-Local Means denoised images exhibit the highest performance consistently across all used ML models. Random Forest Classifier paired with the denoised HOG feature showed the highest overall performance with a test accuracy of 94.28%. Based on your our testing outcomes, the most effective hyperparameters for the Random Forest classifier are as follows. The configuration is characterized by n-estimators of value 100, maximum tree depth of value None, minimum sample split of value 5 and minimum samples per leaf of value 2. It is worth noting that models trained on features extracted from lower dimensions of the same images perform significantly worse.

3.2. Baseline comparison with feedforward neural networks

In this subsection, we investigate the efficacy of feedforward neural networks in utilizing previously extracted Count Vectors and TFIDF-transformed vectors to classify unseen data. We evaluate the performance of vectors obtained using the K – Means Cluster Sizes corresponding to the highest accuracies produced by KNN and SVM in combination with Count vectors and TFIDF-transformed vectors, as seen in Table. 1. Starting with TFIDF transformed vectors, after some trial and error, the network architecture that produced relatively satisfactory results is described as follows.

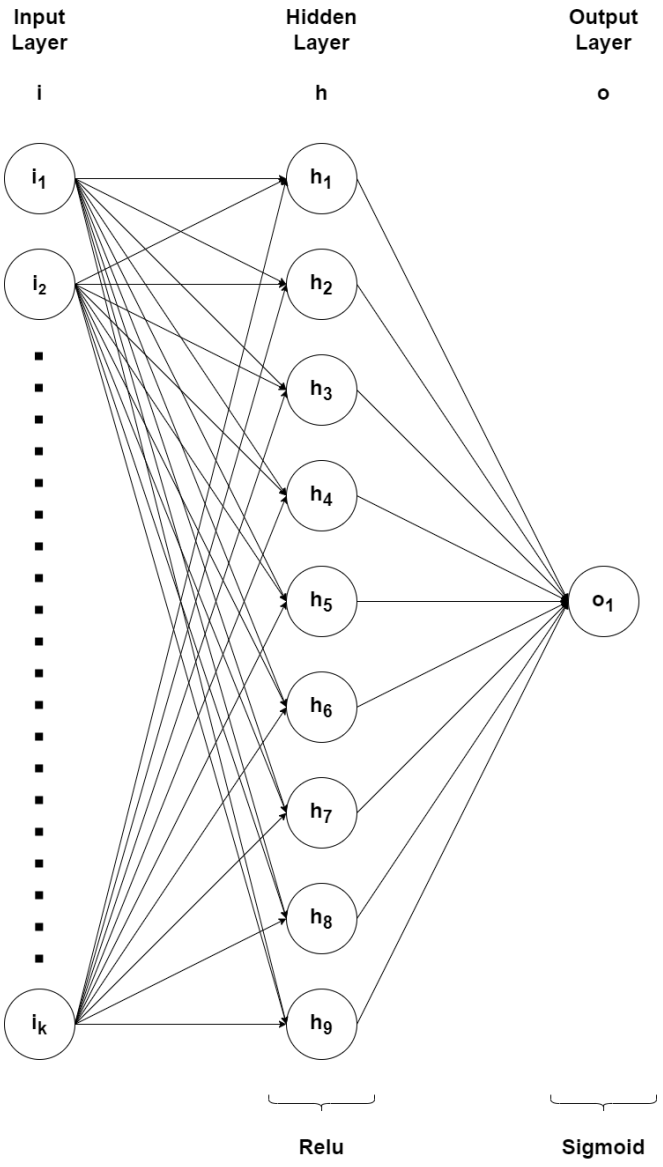

Figure 10: A single-hidden-layer feedforward neural network for training on TFIDF transformed Vectors.

A single layered feedforward neural network is constructed where the input layer has varying sizes depending on the number of clusters **K** chosen as depicted in Figure. 10. The weighted sum of these neurons is calculated and fed to a neuron

in the hidden layer as follows.

$$h_j^{in} = \sum_{i=1}^{K} \left(i_i \, w_1 \left(i, j \right) \right) \tag{24}$$

Where $\mathbf{h}_j^{in}$ is the input passed to the $\mathbf{j}^{th}$ neuron in the hidden layer, $\mathbf{i}_i$ is the value of the $\mathbf{i}^{th}$ neuron in the input layer and $\mathbf{w_1} \left(\mathbf{i}, \mathbf{j} \right)$ is the weight connecting the $\mathbf{i}^{th}$ neuron in the input layer and the $\mathbf{j}^{th}$ neuron in the hidden layer.

The hidden layer has nine neurons employing the ReLU activation function, and their weighted sum fed to the output layer neuron is calculated as follows.

$$o_k^{in} = H_{ReLU} \left(\sum_{j=1}^{9} \left(h_j \, w_2 \left(j, k \right) \right) \right) \tag{25}$$

where $\mathbf{o}_k^{in}$ is the input passed to the $\mathbf{k}^{th}$ neuron in the output layer, $\mathbf{H}_{ReLU}$ is the ReLU activation function described as $\mathbf{max} \left(0, x \right)$, $\mathbf{h}_j$ is the value of the $\mathbf{j}^{th}$ neuron in the hidden layer and $\mathbf{w_2} \left(\mathbf{j}, \mathbf{k} \right)$ is the weight connecting the $\mathbf{j}^{th}$ neuron in the hidden layer and the $\mathbf{k}^{th}$ neuron in the output layer. Finally, the output layer utilizes one neuron employing the sigmoid activation function described as $\frac{\mathbf{1}}{\mathbf{1+e^{-x}}}$. This setup uses the binary cross-entropy loss function defined as follows.

$$\text{Binary cross-entropy} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \, log \left(p_i \right) + \left(1 - y_i \right) log \left(1 - p_i \right) \right] \tag{26}$$

Where $\mathbf{N}$ represents the number of extracted feature vectors, $\mathbf{y}_i$ is the ground truth label (1 or 0) for the $\mathbf{i}^{th}$ feature vector, $\mathbf{p}_i$ is the probability that the $\mathbf{i}^{th}$ feature vector belongs to class 1. This network is trained for 100 epochs with a batch size of 50 and a learning rate set to 0.0001 utilizing the Adam optimizer, producing results shown in Table. 3. This model trained on Count vectors shows a tendency to overfit quite easily. So, a different architecture for the second model is created, having the following architecture.

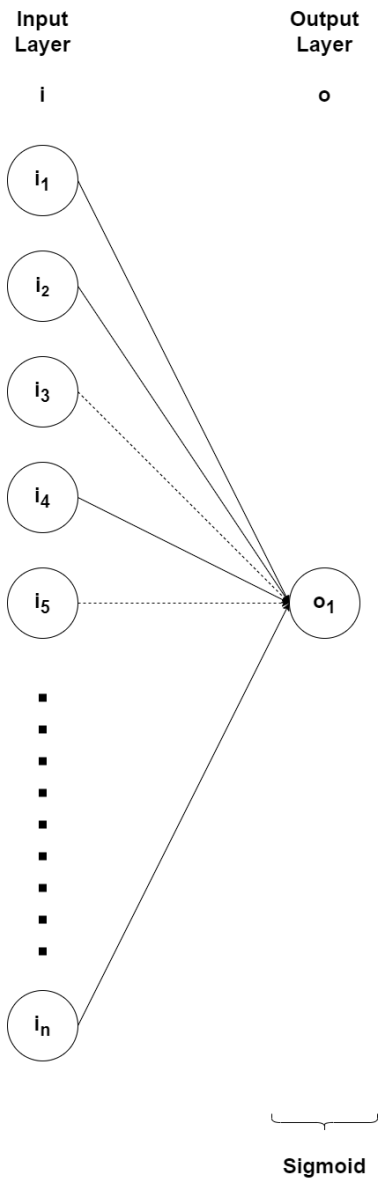

Figure 11: A single-layer feedforward neural network for training on Count Vectors.

A single-layered feedforward neural network is designed with an input layer of varying size and an output layer with a single neuron associated with the sigmoid activation function, as shown in Figure. 11. Additionally, a dropout layer with a

dropout rate of 0.2 is added to minimize the effect of overfitting. The model is trained for 100 epochs with a batch size of 50 and a learning rate set 0.00021. The accuracy and loss values are tabulated in Table. 3.

Table 3: Classification performance using feedforward Neural Networks

Extracted Feature	K-Means Cluster Size	Accuracy	Loss
Count Vectors	95	55.94 %	1.172
	205	55.24 %	0.987
	445	60.84 %	0.749
	565	73.43 %	0.627
TFIDF Vectors	95	67.83 %	0.683
	205	67.13 %	0.674
	445	76.22 %	0.645
	565	81.82 %	0.643

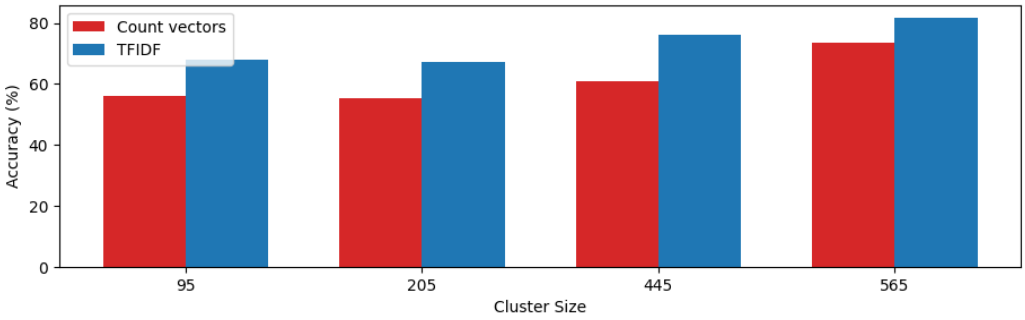

Figure 12: Accuracy comparison with Count Vectors and TFIDF features for the neural network.

From Table 3 and Figure 12, it is observed that for both Count Vectors and TFIDF Vectors, the accuracy of pothole detection increases with an increase in cluster size. When fed with Count Vectors as a feature, for a cluster size of 565, the neural network yields an accuracy of 73.34% while the TFIDF Vectors yield an accuracy of 81.82%.

3.3. Discussion

In this manuscript, road pothole detection using SIFT and a bag of visual words is introduced with a baseline comparison with HOG and LBP feature extraction techniques. SIFT feature extraction has a significant advantage of scale and rotation invariance, where changes to the scale and orientation of the pothole image do not affect the detection accuracy. The SIFT algorithm also produces distinctive features that are highly robust to changes in lighting, noise, and contrast conditions. Since the number of descriptors extracted from images can vary, clusters of descriptors can be defined across images, and classification of images can be performed based on a histogram of these clusters. The more significant number of clusters means more variety of "types" of patches or Visual Words across images. This also means that the final feature vector representing the cluster frequencies of an image will also have a higher dimensionality. From Table 1, it is observed that for KNN, the Count Vectors and TFIDF Vectors achieved the highest accuracy of 77.62% and 72.02% for cluster sizes of 95 and 565, respectively. SVM with a RBF kernel yielded an accuracy score of 76.9% and 83.21% with cluster sizes of 205 and 445, respectively, for Count Vectors and TFIDF Vectors. To validate the feature extraction capabilities of SIFT, image augmentation with varied lighting conditions, contrast, and orientation is carried out. For this, we created a separate dataset of Plain and Pothole images containing 5 copies of each image from the original dataset having varying scale (zoom upto 20%), rotation (upto 350°) and brightness (upto 15%) as illustrated in 13.

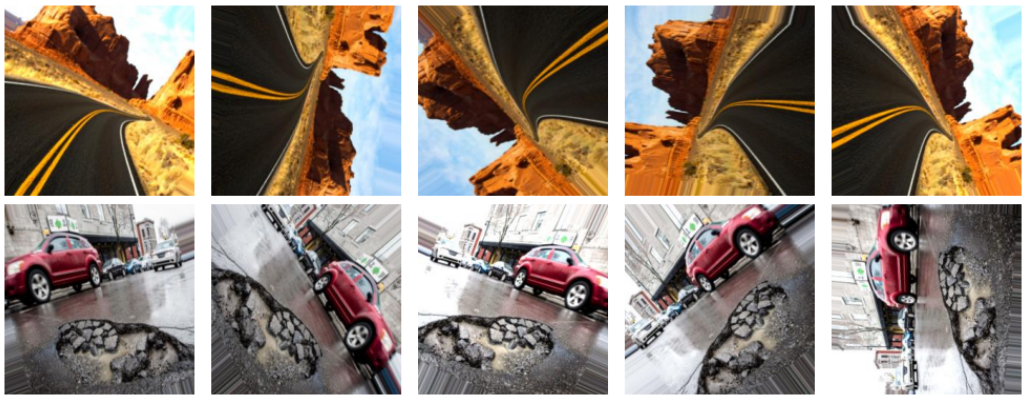

Figure 13: Sample images from plain and pothole dataset.

Table 4: Performance metrics for augmented images

Learning Algorithm	Extracted Feature	K-Means Cluster Size	Accuracy	Cluster Accuracy
KNN	Count Vectors	95	68.51 %	69.04 %
	Count Vectors	205	70.43 %	72.38 %
	Count Vectors	445	70.04 %	74.48 %
	Count Vectors	565	67.7 %	69.04 %
	TFIDF Vectors	95	57.94 %	64.02 %
	TFIDF Vectors	205	57.21 %	60.81 %
	TFIDF Vectors	445	57.82 %	63.04 %
	TFIDF Vectors	565	54.14 %	54.67 %
SVM	Count Vectors	95	71.55 %	74.62 %
	Count Vectors	205	71.05 %	73.92 %
	Count Vectors	445	72.38 %	76.01 %
	Count Vectors	565	69.71 %	72.66 %
	TFIDF Vectors	95	71.13 %	76.71 %
	TFIDF Vectors	205	72.38 %	76.01 %
	TFIDF Vectors	445	72.86 %	76.99 %
	TFIDF Vectors	565	71.77 %	74.76 %
FFNN	Count Vectors	95	51.46 %	51.60 %
	Count Vectors	205	55.90 %	60.11 %
	Count Vectors	445	60.22 %	66.39 %
	Count Vectors	565	59.36 %	66.25 %
	TFIDF Vectors	95	62.62 %	67.22 %
	TFIDF Vectors	205	62.82 %	67.78 %
	TFIDF Vectors	445	68.20 %	73.08 %
	TFIDF Vectors	565	66.47 %	72.38 %

Table. 4 depicts the detection performance of KNN, SVM, and FFNN for augmented images. For KNN, Count Vectors as feature input gives a maximum accuracy of 70.43% and has superior performance compared to TFIDF Vectors. For SVM with RBF kernel, the non-linear capabilities enhance the detection performance with TFIDF features with highest accuracy of 72.86% for a cluster size of 445. Similarly, for feedforward neural network, TFIDF features with cluster size of 445 yields highest accuracy of 68.20%. Here, cluster accuracy is calculated by contrasting the majority classification label of the 5 augmented images to the ground truth label of their source image. The classification performance is found superior for non-linear SVM employing RBF kernel, followed by FFNN and KNN. The robust nature of SIFT algorithm can be leveraged to develop a more sophisticated pothole detection system in future.

4. Conclusion

In this paper, two pothole detection techniques are proposed, and differences in performance are highlighted. We also examined how feedforward neural networks could classify images using SIFT features. Some aspects of feature extraction deserve further exploration. In addition to current methods, dimensions of cells and blocks in HOG feature extraction can be experimented with. Instead of scaling down images to a fixed size, image dimensions can be left as is, and proportionately altering Cell and Block dimensions to produce the exact feature vector dimensions is also worth considering. In LBP features, changing radius sizes and using shape-changed LBP variants, for instance, the Corner Rhombus Shape LBP Variant [36] may also lead to better performances. In both HOG and LBP and Bag of Visual Words approaches, different dimensionality reduction techniques can be experimented with. Results reveal the highest detection accuracy of 90.71% for KNN, 92.85% for SVM, and 87.85% for the random forest when considered with HOG+LBP features, as discussed in the manuscript. Further, SIFT algorithm is tested for its robustness to scale and rotational invariance and results reveal superior performance with SVM when compared to KNN and

FFNN. Changing the dimensions of images in preprocessing steps may also lead to better detection performance.

Funding source

This research did not receive a specific grant from funding agencies in the public, commercial or non-profit sectors.

CRediT authorship contribution statement

Dev Bhanushali: Conceptualization, Methodology, Software, Writing - Original Draft. **Harsh Kotadiya**: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Writing - Original Draft. **Harsh S. Dhiman**: Supervision, Writing - Review & Editing.

References

[1] Y. Li, C. Papachristou, D. Weyer, Road pothole detection system based on stereo vision, in: NAECON 2018 - IEEE National Aerospace and Electronics Conference, 2018, pp. 292–297. doi:10.1109/NAECON.2018.8556809.

[2] R. Fan, U. Ozgunalp, B. Hosking, M. Liu, I. Pitas, Pothole detection based on disparity transformation and road surface modeling, IEEE Transactions on Image Processing 29 (2020) 897–908. doi:10.1109/TIP.2019.2933750.

[3] N. Ma, J. Fan, W. Wang, J. Wu, Y. Jiang, L. Xie, R. Fan, Computer vision for road imaging and pothole detection: a state-of-the-art review of systems and algorithms, Transportation Safety and Environment 4 (4) (2022) tdac026. arXiv:https://academic.oup.com/tse/article-pdf/4/4/tdac026/47169346/tdac026.pdf, doi:10.1093/tse/tdac026.
URL https://doi.org/10.1093/tse/tdac026

[4] C. Zhang, A. Elaksher, An unmanned aerial vehicle-based imaging system for 3d measurement of unpaved road surface distresses, Comp.-Aided Civil and Infrastruct. Engineering 27 (2012) 118–129. doi:10.1111/j.1467-8667.2011.00727.x.

[5] Y. Du, Z. Zhou, Q. Wu, H. Huang, M. Xu, J. Cao, G. Hu, A pothole detection method based on 3D point cloud segmentation, in: X. Jiang, H. Fujita (Eds.), Twelfth International Conference on Digital Image Processing (ICDIP 2020), Vol. 11519, International Society for Optics and Photonics, SPIE, 2020, p. 1151909. doi:10.1117/12.2573124.
URL https://doi.org/10.1117/12.2573124

[6] R. Ravi, D. Bullock, A. Habib, Highway and airport runway pavement inspection using mobile lidar, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B1-2020 (2020) 349–354. doi:10.5194/isprs-archives-XLIII-B1-2020-349-2020.
URL https://isprs-archives.copernicus.org/articles/XLIII-B1-2020/349/2020/

[7] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, L. Selavo, Real time pothole detection using android smartphones with accelerometers, in: 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011, pp. 1–6. doi:10.1109/DCOSS.2011.5982206.

[8] S. Ryu, T. Kim, Y.-R. Kim, Image-based pothole detection system for its service and road management system, Mathematical Problems in Engineering 2015 (2015) 1–10.
URL https://api.semanticscholar.org/CorpusID:59359915

[9] C. Koch, I. Brilakis, Pothole detection in asphalt pavement images, Advanced Engineering Informatics 25 (3) (2011) 507–515, special Section: Engineering informatics in port operations and logistics. doi:https://doi.org/10.1016/j.aei.2011.01.002.
URL https://www.sciencedirect.com/science/article/pii/S1474034611000036

[10] J. Lin, Y. Liu, Potholes detection based on svm in the pavement distress image, in: 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science, 2010, pp. 544–547. doi:10.1109/DCABES.2010.115.

[11] G. M. Hadjidemetriou, S. E. Christodoulou, P. A. Vela, Automated detection of pavement patches utilizing support vector machine classification, in: 2016 18th Mediterranean Electrotechnical Conference (MELECON), 2016, pp. 1–5. doi:10.1109/MELCON.2016.7495460.

[12] Y. Pan, X. Zhang, G. Cervone, L. Yang, Detection of asphalt pavement potholes and cracks based on the unmanned aerial vehicle multispectral imagery, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 11 (10) (2018) 3701–3712. doi:10.1109/JSTARS.2018.2865528.

[13] V. Jakštys, V. Marcinkevičius, P. Treigys, J. Tichonov, Detection of the road pothole contour in raster images, Information Technology and Control 45 (3) (2016) 300–307.

[14] T. D. Chung, M. K. A. A. Khan, Watershed-based real-time image processing for multi-potholes detection on asphalt road, in: 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 2019, pp. 268–272. doi:10.1109/ICSEngT.2019.8906371.

[15] I. Moazzam, K. Kamal, S. Mathavan, S. Usman, M. Rahman, Metrology and visualization of potholes using the microsoft kinect sensor, in: 16th Interna-

tional IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013, pp. 1284–1291. doi:10.1109/ITSC.2013.6728408.

[16] L. Suong, K. Jangwoo, Detection of potholes using a deep convolutional neural network, Journal of Universal Computer Science 24 (9) (2018) 1244–1257, funding Information: This work was supported by an INHA UNIVERSITY Research Grant. Publisher Copyright: © J.UCS.

[17] D. J, S. D. V, A. S. A, K. R, L. Parameswaran, Deep learning based detection of potholes in indian roads using yolo, 2020 International Conference on Inventive Computation Technologies (ICICT) (2020) 381–385.
URL https://api.semanticscholar.org/CorpusID:219591292

[18] E. N. Ukhwah, E. M. Yuniarno, Y. K. Suprapto, Asphalt pavement pothole detection using deep learning method based on yolo neural network, 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA) (2019) 35–40.
URL https://api.semanticscholar.org/CorpusID:209901063

[19] F. Kortmann, K. Talits, P. Fassmeyer, A. Warnecke, N. Meier, J. Heger, P. Drews, B. Funk, Detecting various road damage types in global countries utilizing faster r-cnn, 2020. doi:10.1109/BigData50022.2020.9378245.

[20] S. Gupta, P. Sharma, D. Sharma, V. Gupta, N. Sambyal, Detection and localization of potholes in thermal images using deep neural networks (Jul. 2020). doi:10.1007/s11042-020-09293-8.
URL http://dx.doi.org/10.1007/s11042-020-09293-8

[21] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata, Road damage detection and classification using deep neural networks with smartphone images, Computer-Aided Civil and Infrastructure Engineering 33 (12) (2018)

1127–1141. doi:10.1111/mice.12387.
URL http://dx.doi.org/10.1111/mice.12387

[22] (2019). [link].
URL https://www.kaggle.com/datasets/virenbr11/pothole-and-plain-rode-images

[23] D. G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the seventh IEEE international conference on computer vision, Vol. 2, Ieee, 1999, pp. 1150–1157.

[24] S. Prasomphan, J. Jung, Mobile application for archaeological site image content retrieval and automated generating image descriptions with neural network, Mobile Networks and Applications 22 (08 2017). doi:10.1007/s11036-016-0805-6.

[25] K. P. F.R.S., Liii. on lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2 (11) (1901) 559–572. doi:10.1080/14786440109462720.

[26] X. Jin, J. Han, K-Means Clustering, Springer US, Boston, MA, 2010, pp. 563–564. doi:10.1007/978-0-387-30164-8_425.
URL https://doi.org/10.1007/978-0-387-30164-8_425

[27] C. Sammut, G. I. Webb (Eds.), TF–IDF, Springer US, Boston, MA, 2010, pp. 986–987. doi:10.1007/978-0-387-30164-8_832.
URL https://doi.org/10.1007/978-0-387-30164-8_832

[28] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 2005, pp. 886–893 vol. 1. doi:10.1109/CVPR.2005.177.

[29] 2016. [link].
URL https://learnopencv.com/histogram-of-oriented-gradients/

[30] T. Ojala, M. Pietikäinen, T. Mäenpää, A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification, in: S. Singh, N. Murshed, W. Kropatsch (Eds.), Advances in Pattern Recognition — ICAPR 2001, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 399–408.

[31] 2018. [link].
URL https://d-in-cloud.tistory.com/23

[32] V. Karnati, M. Uliyar, S. Dey, Fast non-local algorithm for image denoising, in: 2009 16th IEEE International Conference on Image Processing (ICIP), 2009, pp. 3873–3876. doi:10.1109/ICIP.2009.5414044.

[33] A. Mucherino, P. J. Papajorgji, P. M. Pardalos, k-Nearest Neighbor Classification, Springer New York, New York, NY, 2009, pp. 83–106. doi:10.1007/978-0-387-88615-2_4.
URL https://doi.org/10.1007/978-0-387-88615-2_4

[34] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297.

[35] T. K. Ho, Random decision forests, in: Proceedings of 3rd international conference on document analysis and recognition, Vol. 1, IEEE, 1995, pp. 278–282.

[36] I. Al Saidi, M. Rziza, J. Debayle, A new lbp variant: Corner rhombus shape lbp (crslbp), Journal of Imaging 8 (7) (2022). doi:10.3390/jimaging8070200.
URL https://www.mdpi.com/2313-433X/8/7/200

From:
Dr. Harsh S. Dhiman,
Department of AI & ML,
Symbiosis Institute of Technology,
Symbiosis International (Deemed University)
Pune, MH, India

December 15, 2023

To,
Editor-in-chief,
Applied Soft Computing,

Dear sir,

We would like to submit this manuscript by Dev Bhanushali, Harsh Kotadiya, and Dr. Harsh S. Dhiman entitled "Road Pothole Detection based on Pattern Recognition Framework integrating SIFT Feature Extraction and Bag of Visual Words" to Applied Soft Computing. The major contributions of our work can be summarized as follows

i). A novel technique utilizing Scale invariant feature transform (SIFT) feature extraction and bag of visual words is introduced for road pothole detection.
ii). SIFT feature extraction is compared with baseline Histogram of oriented gradients (HOG) and Local binary pattern (LBP) algorithm.
iii). Robustness of SIFT algorithm in terms of scale and rotational invariance is tested using three machine learning techniques, namely, KNN, SVM, and Neural network.

Correspondence related to the paper may please be directed to Dr. Harsh S. Dhiman, at the following address, and e-mail address:

Dr. Harsh S. Dhiman,
Department of AI & ML,
Symbiosis Institute of Technology,
Symbiosis International (Deemed University)
Pune, MH, India, E-Mail: harsh.dhiman@sitpune.edu.in

We affirm that this manuscript is original, has not been published before and is not currently being considered for publication in another journal.

Thank you very much for your attention to our paper.

Sincerely yours,
Dr. Harsh S. Dhiman

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Highlights

i). Road pothole detection utilizing Scale invariant feature transform (SIFT) feature extraction.

ii). Bag of visual words used for identifying key descriptors.

iii). SIFT feature extraction is compared with baseline HOG and LBP algorithm.

iv). Robustness of SIFT algorithm in terms of scale and rotational invariance is tested.