IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 November 2022, accepted 2 December 2022, date of publication 12 December 2022,
date of current version 23 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3228263

RESEARCH ARTICLE

Gabor CNN Based Intelligent System for Visual Sentiment Analysis of Social Media Data on Cloud Environment

SIDDHALING UROLAGIN[1], JAGADISH NAYAK[1], (Senior Member, IEEE), AND U. RAJENDRA ACHARYA[2,3,4]

[1]Birla Institute of Technology and Science, Pilani, Dubai Campus, Dubai, United Arab Emirates
[2]Department of Electronics and Computer Engineering, Ngee Ann Polytechnic, Asia University, Singapore 599489
[3]Department of Bioinformatics and Medical Engineering, Asia University, Taichung 41354, Taiwan
[4]School of Management and Enterprise, University of Southern Queensland, Springfield, QLD 4300, Australia

Corresponding author: Siddhaling Urolagin (siddhaling@dubai.bits-pilani.ac.in)

ABSTRACT Social media contains a plethora of information in the form of text, images, videos, and other data. Users across the globe are increasingly sharing their data on various social media platforms. Sentiment analysis of data, such as text, images, and videos are widely used to understand the feelings of users. In recent years, the convolutional neural network (CNN) has been extensively applied for various applications. The cloud computing environment is a popular service due to its reliability, availability, and easy software integration. However, CNN models are deep neural networks that have a high computational cost. There is a need for CNN models which utilize lesser computational resources especially when these models are deployed in a cloud environment due to the remote physicality of servers, resource optimization, and infrastructure cost reduction. In this research, Gabor filters are integrated with CNN models to improve image sentiment analysis in a cloud environment, with advantages such as the reduction in computation energy and time, the elimination of the need for pre-trained models, and a perceived accuracy improvement. *Two* variants of Gabor-CNN (G-CNN) models with a different number of pooling and normalization layers are developed. The proposed G-CNN is trained and tested using *five* standard databases as SentiBank, Twitter, MVSO, MultiView_I, and MultiView_II. Maximum classification accuracies of 91.71%, 92.52%, 97.39%, 90.88%, and 91.31% are obtained on SentiBank, Twitter, MVSO, MultiView_I, and MultiView_II databases respectively using the developed models. The proposed G-CNN model has provided an accuracy of 92.76% on average.

INDEX TERMS Gabor convolutional neural network, sentiment image analysis, deep learning, cloud environment.

I. INTRODUCTION

Cloud computing has become more popular due to its accessibility, flexibility, and reduced time to bring a model to the market [1], [2]. The sentiment analysis of data collected from social media is a significant step in social data analysis [3], [4], [5] and it provides an understanding of people's opinions and behaviors [6]. Social media data analysis using cloud computing is most vital to bring the analysis into real-world applications. The analysis of the sentiment has widespread applications in marketing, service sectors, opinion polls during elections, investments, and product analysis [7], [8], [9], [10]. The sentiment analysis has been applied to uncover hidden customer behavior from social media data [11]. Along with textual data, social media has a huge collection of both images and videos. Visual sentiment analysis is related to understanding the sentiment of images and videos. In Borth et al. [12] visual sentiment analysis was carried out and the Visual Sentiment Ontology (VSO) was developed with the Adjective Noun Pairs (ANP). Image sentiment classification

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

was performed in [13] using deep networks. Liang et al. [14] have developed a cross-domain semi-supervised deep metric learning for image sentiment analysis. Preethi G. et al., [15] have developed recursive neural networks (RNN) along with deep learning (DL) to perform sentiment analysis of reviews on a cloud platform. In Arulmurugan et al. [16], an ML-based intelligent system was developed and segmentation ranking followed by sentiment classification was performed on the cloud platform. Moreover, issue related to working with secured data and data sharing in the cloud has been addressed in [17] and [18]. A survey on applications of DL for image sentiment analysis can be found in [19].

Cloud computing services provide high reliability and mobility software integration, also their services require effective utilization of resources. Even though CNN-based models have been effective for various image and video based applications, they demand high computational cost [20], [21]. The computational cost is critical when a CNN model is deployed in the cloud computing environment due to the remote physicality of servers [22], resource optimizatoin [23], infrastructure cost reduction [24]. Hence there is a requirement for CNN models that are more effective in terms of computational cost while improving performance. In recent works, Gabor filter integration with CNN models has shown many advantages. Chang et al. [25] initialized CNN's first layer with Gabor filters and fine-tuned it during training. In [26] Gabor filters were introduced in *two* CNN layers. Both these studies showed a decrease in computation energy and time consumption during the training of CNN. Chen et al. [27] have used Gabor based CNN model for hyperspectral image classification and showed that Gabor filters can mitigate the overfitting problem in DL networks. Meng et al. [28] have developed Gabor based CNN models and shown a reduction in computational energy and time of 17–19%. The steerable Gabor filters have the property of extracting low-level features from images thus utilizing them eliminates the need for transfer learning. Moreover, experiments conducted in [29] emphasize that the use of Gabor filters helps in fast convergence.

In this research, we integrate Gabor filters with CNN and developed *two* new Gabor-CNN (G-CNN) models. Our research aim is to take the benefits of Gabor integrated CNN in the cloud environment and develop high-performing Gabor integrated CNN models. In this study, G-CNN models are developed for sentiment image analysis in the cloud environment. Moreover, there is almost rare to none cloud platform-based intelligent system for image sentiment prediction. The objectives of the work are,

(i) Modified 3D information diagram based method to design the Gabor filters.

(ii) Integration of Gabor filters with CNN for visual sentiment recognition.

(iii) To develop *two* new G-CNN models on the cloud platform.

In our research, *two* new G-CNN architectures are developed. The first network is G-CNN with 6 max-pooling (G-CNN-6M) and this network uses max-pooling layers after every convolutional layer. The second G-CNN uses *six* max pooling and normalization layers (G-CNN-6MN). In this network max-pooling and normalization are applied after each convolutional layer. *Five* benchmark databases namely: SentiBank [12], Twitter [13], MVSO [12], MultiView_I [30], and MultiView_II [30] are used. The cross-validation (CV) results of the proposed models are compared with models without Gabor filters (CNN-6MN). Section 2 covers the literature review and the databases used in this study is given in section 3, the Gabor filters design using 3D information is outlined in section 4. The *two* proposed G-CNN architectures are presented in section 5. The experiment setup and results are given in sections 6 and 7 respectively. Followed by them the discussion in section 8. Finally, the paper concludes in section 9 by providing a summary of the main findings of the work.

Algorithm 1 Algorithm for Dataset Preparation

```
Input: $X = \{x_1, x_2, x_3 \ldots, x_n\}$ images obtained from Twitter or Flickr,
    $T = \{t_1, t_2, t_3 \ldots, t_{n'}\}$ text messages obtained from the Twitter,
    $AMT = \{A_1, A_2, A_3 \ldots, A_{n''}\}$ AMT workers for the task of assigning labels.
Output: Refined dataset of images
    Select the images $D = \{x_i, t_i | \, x_i \in X \; and \; t_i \in T \;\; selected \; images \; or \; text \; samples\}$
    Annotate the images $D' = \{x_i, t_i, y_i, z_i, A_j \;\, | (x_i, t_i) \in D, \; y_i = l\,(x_i)\,, \;\; z_i = l\,(t_i) \; and \; A_j \in \; AMT\}$
     $y_i$ the label for the image and
     $z_i$ for text assigned by an AMT worker $A_j$
    Refine the images: $D'' = \{x_i, y_i \;\; |(x_i, y_j, A_k) \in D' \, and \, ((\text{Case:a}) \; or \; (\text{Case:b}) \; or \; (\text{Case:c}) \; or \; (\text{Case:d}))\}$
    Case a: SentiBank
    $\forall A_k(\, y_i = y_j), \;\; 1 \leq k \leq 3 \; and \; |AMT| = 3$
    Case b: Twitter
    $\forall A_k(\, y_i = y_j), \;\; 1 \leq k \leq 5 \; and \; |AMT| = 5$
    Case c: MultiView_I and MultiView_II
     $\forall A_k(y_i = y_j \; and \; y_j = t_j)$
    Case d: MVSO
     $\exists A_k(y_i = y_j), \; 1 \leq k \leq 2 \; and \; |AMT| = 3$
```

II. LITERATURE SURVEY

Machine learning (ML) and natural language processing (NLP) were used to analyze the text shared by users to understand the emotional contents [10], [31], [32] and enhanced sentiment analysis using DL in [33] and [34]. In Roy et al. [35], a support vector machine (SVM) classifier was used to perform the sentiment classification of the tweets. An aspect-level sentiment classification using mutual attention neural network was developed by Jiang et al. in [36]. Aspect-based sentiment identification is a complex task as it is more implicit. Ishaq et al. have tuned a CNN model using GA in [37] on the semantic and word2vec transformed features. A hybrid DL model was developed in [38] using various word embedding techniques. Based on the features extracted the sentiment classification was performed.

CNN has been a widely used DL model for computer vision, image recognition, and pattern recognition. It has the advantage that it can learn both feature representation and classification without human intervention. Therefore, DL architecture such as CNN has been extensively utilized for complex data such as healthcare systems [39], object detection [40], and time series analysis [41]. Szegedy et al. have developed a CNN architecture 'Inception' in [42] for the classification of images from ImageNet. A detailed discussion on various practical measurements of DL models such as parameters, power consumption, utilization of memory, inference time, etc. was presented in [43]. In Biswas et al. [44] a dilated CNN model was utilized for the segmentation of retinal fundus images. An incremental dilation CNN model was proposed in [45] which effectively performs the classification of MRI images. In Samui et al. [46] have discussed many applications related to image analysis, speech, NLP, and risk analysis. A review was conducted by Al-Saffar et al. on CNN for image classification tasks in [47]. CNN was used to classify encrypted images of vehicles on the cloud [48]. CNN along with SVM has been used by Hossain et al. in [49] for facial emotion recognition on cloud computing. The work also focused on fine-tuning the architecture and provided an analysis of pattern visualization learned by CNN. Md Zahangir Alom et al. [50] have presented various advancements in DL along with state-of-the-art architectures.

Siersdorfer et al. [51] have utilized the SentiWordNet thesaurus to gather sentiment values from the metadata of photos shared on Flickr. Then they developed ML-based techniques to perform sentiment prediction of images using visual features. They reported 70.00% accuracy on the Flickr dataset. The visual sentiment analysis was presented by Borth et al in [12] on the social images. They developed an image library SentiBank for recognizing adjective-noun pairs. Their contribution was the development of ANPs and VSO to represent the sentiment of images. Using linear SVM the classification accuracy of 70.00% was achieved on SentiBank. The classical feature-based sentiment analysis [52], [53], [54] was continued until ML-based analysis has been widely applied. The mid-level features of an image have been utilized by Yuan et al. [55] for sentiment prediction. They also used eigenface-based facial expression detection for the additional mid-level feature. They observed 68.00% accuracy using SVM and logistic regression (LR). In Borth et al. [56], the deep CNN was trained based on the Caffe framework. In their experiments, an improvement in accuracy of 62.3% with respect to the binary SVM classifier for Sentibank 2.0 was observed. Wang et al. [57] proposed an Unsupervised SEntiment Analysis (USEA) framework for sentiment analysis of social media images. Using USEA they achieved an accuracy of 56.18% on the Flickr dataset. The Flickr and Twitter datasets of sentiment images were prepared by You et al. in [13]. A progressively trained CNN and transfer learning were developed for image sentiment analysis on these datasets. Several manually label Twitter images were incorporated and an accuracy of 78.60% was noted on the Twitter dataset. Baecchi et al. [58] have used multimodal features such as text and images and developed a sentiment analysis system for social network data. Neural network based models were tested on the Twitter dataset and achieved 80.00% accuracy. Kumar et al. [59] have developed the sentiment analysis for Flickr and Twitter images in *two* phases. In the first phase ANP was generated using the CNN model and in the second phase sentiment prediction was carried out using the SVM classifier. In their experiments, prediction results were given specifically to ANP as sentiment analysis was carried out in *two* phases. Various CNN-based models were designed [60] for automated visual content analysis. Several variations such as CaffeNet and Oversampling on the CNN models were performed and a maximum accuracy of 84.40% was obtained. Kumar et al. in [61] designed a multimodal-based sentiment analysis model. The region CNN was used for image sentiment classification which gave 76.04% accuracy on SentiBank. Most of the works in recent years have focused on using CNN models or ML models for sentiment prediction of images. A lower accuracy of 56.18% was obtained in [57] using the unsupervised method. The highest accuracy of 84.40% was observed in [60] utilizing CNN CaffeNet. A summary of recent works on sentiment image classification is presented in Table 1.

A detailed survey on the visual sentiment analysis methods has been carried out by Ji et al. in [3] and a review on the use of DL is provided by Ain et al. in [4]. Luan et al. [62] have used Gabor filters in deep CNN for object recognition with rotation and scale variations. In their experiments, Gabor-based CNN outperformed regular CNN. A handwritten recognition using Gabor filters based on CNN was presented in [63] and a speech recognition model was presented in [25]. Liu et al. have incorporated a hybrid Gabor filter binarization technique to improve the memory efficiency of CNN in [64]. Sarwar et al. [26] developed the CNN model with *two* Gabor-based convolutional layers. Gabor feature identifier and CNN were utilized to detect the parts of face images and extract the intrinsic features from the face images in [65]. In our research work, Gabor filters are integrated, and *two* new CNN models are developed on a cloud environment for visual sentiment recognition. CNN integration with Gabor filters has two main benefits. Firstly, decrease in computation energy [26], [28] and time [28], [29]. Second benefit is an improvement in performance [27], [29]. We also developed a modified 3D information diagram method to design Gabor filters that are suitable for integration with CNN.

III. DATABASES FOR SENTIMENT IMAGES

Five databases for sentiment images mentioned in [19] are used in this work. Table 2 shows the database, total number of images, and number of positive and negative images. SentiBank [12] is a benchmark database used for visual

TABLE 1. Summary of works done on sentiment analysis.

Authors	Model used	Database	Accuracy (%)
Siersdorfer S et al.2010 [51]	SVM	Flickr photos	70.00
Borth et al. 2013 [12]	Linear SVM	SentiBank	70.00
J. Yuan et al. [55] 2013	SVM / Logistic	UBRUG-EgoSenti	68.00
Borth et al. 2014 [56]	Deep CNN Caffe	Flickr	62.30
Wang Y L et al. 2015 [57]	USEA	Flickr	56.18
You et al. 2015 [13]	CNN	Twitter	78.60
Baecchi et al. 2016 [58]	CNN	Twitter	80.00
Kumar et al. 2017 [59]	CNN+SVM	Flickr and Twitter	–
Campos et al. 2017 [60]	CNN/ CaffeNet/ Oversampling	Twitter	84.40
A. Kumar et al. [61] 2019	R-CNN	SentiBank	76.04

TABLE 2. Details of image databases used for sentiment analysis.

Database	# Positive	# Negative	Total
SentiBank [12]	470	133	603
Twitter [13]	581	301	882
MultiView_I [30]	351	351	702
MultiView_II [30]	351	351	702
MVSO [12]	4504	407	4911

sentiment analysis which is prepared using Twitter images. This database includes 603 tweets on 21 various topics, built using 21 hashtags on various topics such as human, social, event, location, technology, and people. The ground truth for 2000 tweet images is prepared from Amazon Mechanical Turk (AMT) annotation. *Three* independent turkers have performed the task of labeling on image only, text only, and both image and text into positive, negative, and neutral classes. This database consists of 603 images which are unanimously agreed upon by all the *three* turkers. The Twitter database present in [13] is gathered from Twitter messages which have images in them. This database consists of 1269 images then all the images are labeled by *five* AMTs. The images considered for this database are such that all the *five* AMTs agree, thus there are 581 positive and 301 negative sentiment images in the Twitter database. MultiView_I and MultiView_II databases are prepared from [30] and collected from the tweets containing 406 emotional words. These emotional words cover most human feelings. The text and images are labeled by annotators separately in positive, negative, and neutral. Both MultiView_I and MultiView_II databases include images for which all annotators have assigned the same label as that of text. Total number of images received same label as text are 4109. We prepared *two* databases MultiView_I and MultiView_II with a total number of images in both databases of 702. There are 351 negative images after the annotator assigned the labels, and they were included in MultiView_I and MultiView_II. The remaining 351 positive images were randomly selected from 4109 images to be included in MultiView_I and MultiView_II. Thus, we have an equal number of positive and negative images in MultiView_I and MultiView_II.

Multilingual Visual Sentiment Ontology (MVSO) database images are collected from Flickr [12]. Around 3316 ANPs are prepared based on the various human emotions. For each ANP, 1000 images are gathered from Flickr, and overall there are more than one million images. ANP-associated Flickr images are given to *three* AMTs who have assigned values between -2 to 2 independently. The turker evaluated database consisting of 11733 images and in their work images for which at least *two* turker assigned the same polarity value are selected. In our research, the database consists of 4911 images with 4504 positive and 407 negative images. In SentiBank, Twitter, and MVSO more positive samples than negatives are available in datasets. For MultiView_I and MultiView_II there is an equal number of positive and negative samples available for training and testing. Image augmentation [66] is one of the techniques to address the class imbalance however our main focus of this work is to develop Gabor filters integrated CNN model and show performance on the sentiment image datasets. Except for MultiView_I and MultiView_II, other databases have a different number of positive and negative samples. The databases SentiBank, Twitter, and MVSO have class imbalance as shown in Table 2. Also, the existing state-of-the-art methods were developed on a similar sentiment image dataset. In our experiments, the databases with class imbalance are used to show the comparison with existing methods. The database preparation is presented in Algorithm 1 and details of the *five* databases are given in Table 2.

IV. A MODIFIED METHOD TO DESIGN GABOR FILTER SUITABLE FOR CNN

In this section, a modified method based on the information diagram [67] is presented for the Gabor filter design. In our approach, the 3D information diagrams are generated from the Gabor responses, which are then utilized to select suitable parameters. A Gabor filter is a sinusoidal plane wave with a specific frequency modulated by a Gaussian envelope. Gabor filters have properties such as optimal localization in spatial, tunable to specific frequency, and orientation sensitiveness. A Gabor filter is represented using equation (1). The Gabor filter has parameters such as σ is the standard deviation of the Gaussian function, θ is the orientation of the normal to the parallel stripes of the Gabor function, λ is the wavelength of the sinusoidal factor, γ is the spatial aspect ratio, ψ is the phase offset and the size of the kernel is κ.

$$g_{\theta,\sigma,\gamma,\psi,\lambda} = exp(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2})exp(i(2\pi\frac{x'}{\lambda}+\psi)) \quad (1)$$

Gabor filters have to be designed by selecting the parameters to utilize them in the convolutional layers of CNN. Many researchers have fixed the Gabor parameters [68] and few authors have selected parameters manually [69]. However, selecting parameters manually has the disadvantage that Gabor responses may not capture features from input space. The effective parameter must be selected such that important features are represented during the convolution operation in G-CNN. In [67], the Gabor parameter selection is discussed. They introduced the concept of an information diagram, which represents the magnitude of Gabor responses. The magnitude of the Gabor response is obtained at a particular point on the image after the convolution of filters with the image. In [67] authors have proposed an extended information diagram, in which several parameters are varied to construct the Gabor filters. Then the magnitude of the Gabor responses for various parameters are obtained to form the information diagram. In this study, the 3D information diagrams are prepared by varying standard deviations of the Gaussian function σ, spatial aspect ratio γ, and the phase offset ψ. The sinusoidal wavelength λ is considered relative to the size of the filters $\lambda = \kappa - 2$. In our method, n is the number of Gabor filters to be designed, then the orientation of Gabor filters are chosen using (2),

$$\theta_i = \{0, \frac{2\pi}{n}, \frac{4\pi}{n}, \frac{6\pi}{n} \ldots, \frac{2(n-1)\pi}{n}\} \tag{2}$$

For each of θ_i a 3D information diagram is generated with standard deviation varying between $\kappa/2 \leq \sigma_i \leq 3\kappa/2$ in steps of $(3\kappa/2 - \kappa/2)/n$. The other parameter spatial ratio is between $0 \leq \gamma_i \leq 150$ in steps of $150/n$ and phase offset is between $0 \leq \psi_i \leq 2\pi$ in steps of $2\pi/n$ and obtain the Gabor response for an image ξ. The Gabor response r_ξ is produced using (3) convolution operation as

$$r_\xi(x, y) = g_{\theta_i,\sigma_i,\gamma_i,\psi_i,\lambda}(x, y) * \xi(x, y) \tag{3}$$

The 3D information diagram for θ_i is formed as given in (4),

$$ID^{\theta_i}(\sigma_i, \gamma_i, \psi_i) = \underset{x,y}{max}(r_\xi) \tag{4}$$

Thus, a 3D information diagram is generated for each θ_i, $0 \leq \theta_i \leq 2(n-1)\pi/n$ and n the number of 3D information diagrams obtained as (5).

$$Information\ Diagram = \{ID^{\theta_1}, ID^{\theta_2}, \ldots, ID^{\theta_n}\} \tag{5}$$

For each, θ_i the parameters such as standard deviation σ, spatial aspect ratio γ, and the phase offset ψ are selected $(\widehat{\sigma}, \widehat{\gamma}, \widehat{\psi})$ using information diagram (5) such that ID^{θ_i} has the maximum values using (6).

$$(\widehat{\sigma_i}, \widehat{\gamma_i}, \widehat{\psi_i}) = \underset{\sigma,\gamma,\psi}{argmax}(ID^{\theta_i}) \tag{6}$$

V. NEW G-CNN ARCHITECTURES AND CLOUD ENVIRONMENT

After designing suitable Gabor filters as discussed in section 4, they are integrated with CNN. *Two* new Gabor integrated G-CNN architectures are developed for this study. These *two* architectures are developed based on CNN discussed in [70], [71], [72], and [73]. The linear stack of the layered architecture of CNN [29], [74] is considered while developing G-CNN models. Both the proposed G-CNN architecture have max-pooling and normalization layers in different configurations. The max-pooling layers perform the downsampling of the input image and thus help to reduce the number of parameters to learn during the training of G-CNN. The use of max-pooling layers can also reduce overfitting [75]. The normalization layers are used to reduce the internal covariate shift [76]. The first model consists of *six* max-pooling layers after every convolution layer as shown in Figure 1. This model is denoted as G-CNN-6M. Figure 2 shows the second architecture which employs both max pooling and normalization layers in between the convolution layers and this model is represented as G-CNN-6MN.

The CNN is a DL model which has been successfully applied to spatial data. CNN is essentially a multilayer neural network inspired by the visual perception of animals. One of the earlier successes of CNN models is AlexNet [77] thereafter many different CNN models are developed to apply for real-world problems [78]. Usually, a CNN model has multiple layers such as a convolutional layer, pooling layer, normalization layer, and at the end a few fully connected layers. Typically, in a CNN architecture, the convolutional layer and pooling are arranged one after another to extract feature representation from the images. Different kernels or filters are present in a convolutional layer whose primary task is to compute the feature maps [50]. Normalization standardizes the input before the layer, which is useful to accelerate the learning process [76]. The flattening and fully connected layers are the last layers of CNN. The fully connected layers consist of multiple layers of neurons connected in succession. In the flattening layer, a single-dimensional feature vector is created, and the final classification task is carried out by the fully connected layer. On the input sentiment image, the feature representation at layer l is computed using convolution operation as the following equation (7).

$$C^l = \sum_{n=1}^{N} \sum_{c=1}^{Ch} C_c{}^{l-1} \otimes Q^l(n) \tag{7}$$

where C^l feature representation is computed for layer l using convolution operation $\otimes$ on C^{l-1} input feature map and N number of filters Q^l and number Ch is the number of channels. A typical convolution layer can be defined using equation (8) similar to that given in [27].

$$c_i^l = f\left(\sum_{j=1}^{n} c_j^{l-1} \times q_{ji}^l + b_i^l\right) \tag{8}$$

The i^{th} feature representation matrix for the convolutional layer l is c_i^l computed using (8) with n number of input feature representations, c_j^{l-1} is j^{th} feature representation of $l-1$ layer, q_{ji}^l filters, b_i^l initialized to zero, $f(\cdot)$ nonlinear function. In our G-CNN, the filters are of the form $Q^l = [q^l{}_{ji}]$ initialized by Gabor filters for the first layer and Glorot for the remaining layers. During the CNN learning phase, the weights of filters Q^l are updated using the backpropagation process by the partial derivatives δ on the loss function and learning rate η

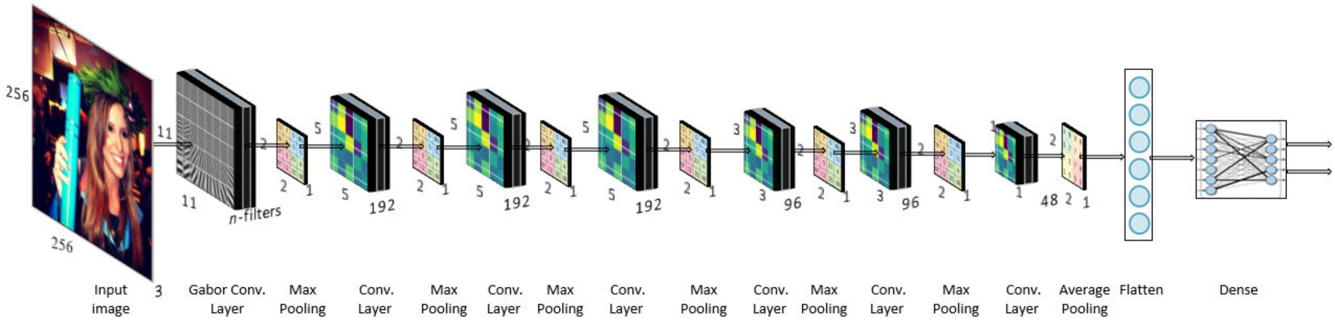


FIGURE 1. Architecture of G-CNN with 6 max-pooling layers (G-CNN-6M).

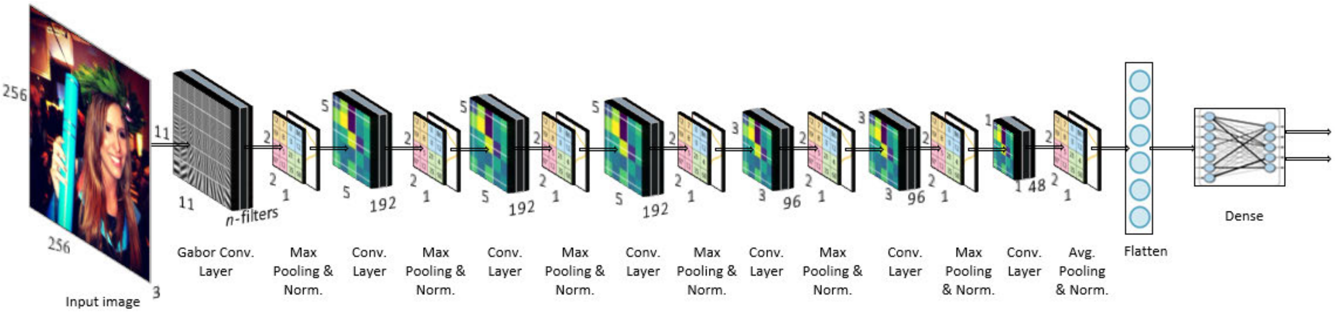


FIGURE 2. Architecture of G-CNN with 6 max-pooling and normalization layers (G-CNN-6MN).

as given in (9),

$$Q^l = Q^l - \eta\delta \tag{9}$$

Cloud computing offers several advantages such as quick deployment, flexibility, reliability, etc [79]. Moreover, deploying CNN architecture on the cloud has the advantage of high availability and usability [79]. However, most CNN architectures have high computation complexity and required high computation resources [80]. The CNN architecture which has lesser utilization of computational resources is highly required. In recent years, Gabor filters integration with CNN has shown many advantages. Firstly, Gabor filters integrated with CNN layers have been found to reduce computation energy and time [25], [26]. In [28] study showed a decrease in computation energy and time of 17-19% when Gabor filters are used with CNN. Secondly, it is shown by authors [77], [81] that, filters of CNN after training with real-life images tend to resemble Gabor-like filters. Hence introducing the selected Gabor filters in the first layer of CNN will make the model learn more effectively and eliminate the need for transfer learning [29]. Lastly in [29], the integration of Gabor with CNN can yield better image classification performance as reported using MNIST [82], CIFAR-10, and CIFAR-100 [83] databases. In this research, Gabor filters are integrated with CNN and *two* architectures G-CNN-6M and G-CNN-6MN are developed. The integration of Gabor with CNN for image sentiment classification in the cloud environment is depicted in Figure 3.

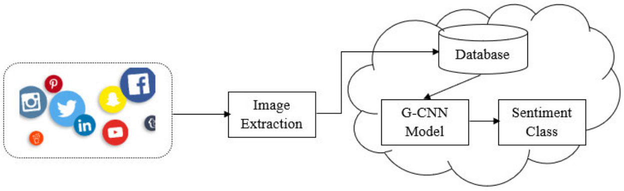


FIGURE 3. Gabor CNN (G-CNN) for image sentiment classification on the cloud environment.

VI. EXPERIMENT SETUP

The experiments are conducted on *five* databases mentioned in Section 3. The proposed G-CNN models are developed using Python and Keras in the Google Colaboratory cloud environment. The stochastic gradient descent (SGD) optimizer with Nesterov momentum is used for the weight optimization of CNN models. The parameters for the SGD optimizer are 0.01 learning rate, 0.0005 decay, and momentum of 0.55. All the CNN models are trained on 25 epochs with a batch size of 10. For architectures G-CNN-6M and G-CNN-6MN 16, 32, 48, and 96 Gabor filters are considered. For $n = 16$, indicates $\theta_i = \{0, \frac{2\pi}{16}, \frac{4\pi}{16}, \frac{6\pi}{16} \ldots, \frac{30\pi}{16}\}$ different angles of orientations. With the known size of the filter (κ), the wavelength is computed $\lambda = \kappa - 2$. The other parameters such as standard deviation (σ), spatial ratio (γ), and phase offset (ψ) are selected to maximize 3D information diagrams at different θ_i. The procedure explained in Section 4 is applied for the filters with $n = 32$, $n = 48$ and $n = 96$ for different

TABLE 3. Details of Gabor parameters chosen for $n = 16$.

Filter	λ	θ	σ	γ	Ψ
1	9	0	7.8	1	2.36
2	9	$\pi/8$	7.45	1	1.97
3	9	$\pi/4$	6.21	1	0.00
4	9	$3\pi/8$	8.16	1	3.54
5	9	$\pi/2$	7.98	1	3.15
6	9	$5\pi/8$	7.98	1	4.33
7	9	$3\pi/4$	6.39	1	5.51
8	9	$7\pi/8$	7.98	1	3.94
9	9	π	7.62	1	3.54
10	9	$9\pi/8$	7.62	1	4.33
11	9	$5\pi/4$	8.16	1	0.00
12	9	$11\pi/8$	7.98	1	2.76
13	9	$3\pi/2$	8.16	1	3.15
14	9	$13\pi/8$	7.98	1	0.79
15	9	$7\pi/4$	6.39	1	0.39
16	9	$15\pi/8$	7.8	1	2.36

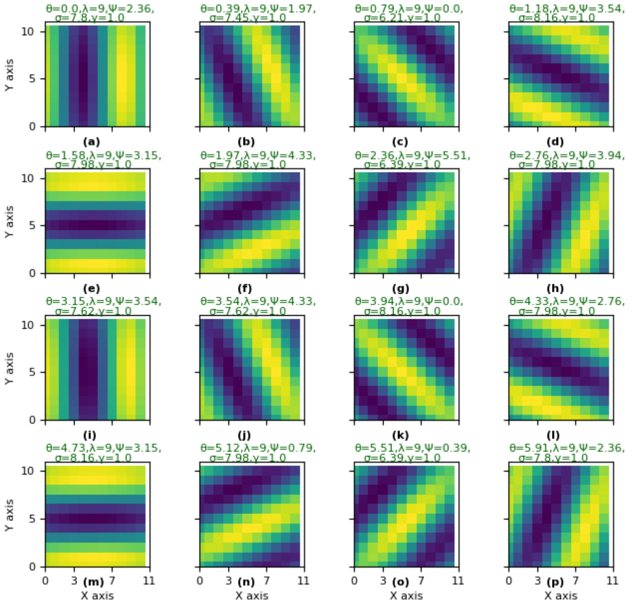


FIGURE 4. Gabor filters for $n = 16$, with filter size κ, $\lambda = \kappa - 2$, and parameters of Table 3.

θ, σ, γ, ψ chosen based on the 3D information diagram. Table 3 shows the parameters selected for $n = 16$ filters.

Figure 4 shows all 16 Gabor filters for the selected parameters. Figure 4(a) is the Gabor filter for λ=9, θ=0, σ=7.8, γ=1, and ψ=2.36. Figure 4(b) to Figure 4(p) show the Gabor filters for parameters of Table 3 from row 2 to row 16. It can be observed from Table 3 and Figure 4 for filter size κ, wavelength $\lambda = \kappa - 2$, and θ between 0 to $15\pi/8$ different values of σ, γ, ψ are chosen based on equation (6). Moreover, the shape of the Gabor filters is varying according to the respective parameters.

An image from SentiBank is convoluted with each filter of Figure 4 and the responses are generated. The Gabor responses are depicted in Figure 5. In this figure, different frequencies and spatial content of the image have appeared in the filtered response. Figure 6 shows the 3D representation of Gabor responses for 16 filters of Table 3. The response of Gabor is shown along the z-axis. It is evident from Figure 6 that, the varying spectral and spatial image content is highlighted in the Gabor responses corresponding to the parameters of Table 3.

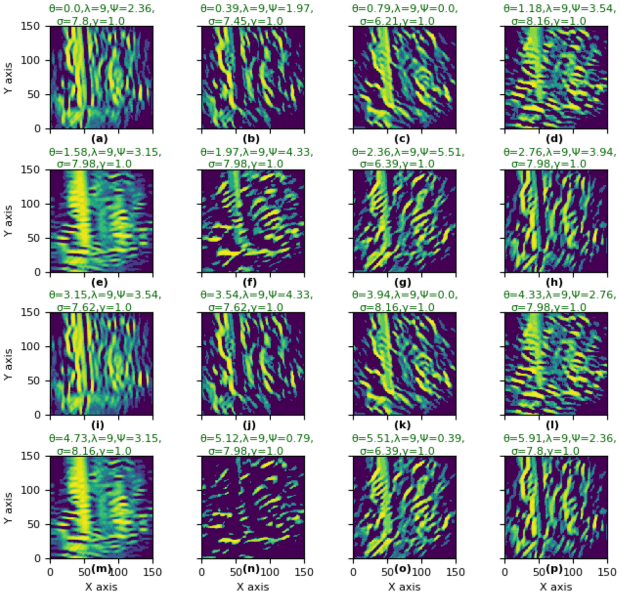


FIGURE 5. Magnitude of Gabor transformation is obtained with $n = 16$ filters provided in Table 3.

The proposed *two* G-CNN models, G-CNN-6M and G-CNN-6MN, are trained and evaluated using *five* databases using 10-fold CV techniques. Before the training process, each image of the database undergoes several pre-processing steps. We prepare the size of each image to resemble the same shape as the input shape of the first convolutional layer. Also, the pixels of the images are normalized between values of 0 to 1. The various performance metrics such as accuracy, sensitivity, precision, specificity, and F-score are obtained using a 10-fold CV.

VII. EXPERIMENT RESULTS

In this section, we elaborate on the experimental results obtained on the *two* proposed G-CNN architectures for sentiment image classification. The results obtained for G-CNN-6M and G-CNN-6MN are compared with the CNN model without Gabor filters (CNN-6MN) that is trained and evaluated on the same databases. Table 4 shows the results of the 10-fold CV obtained for models CNN-6MN, G-CNN-6M, and G-CNN-6MN on the SentiBank database.

Each G-CNN model is combined with 16, 32, 48, and 96 Gabor filters at the first convolutional layer, and performance is tabulated in Table 4. During both training and testing phases, the G-CNN-6MN model has yielded better accuracies compared to the G-CNN-6M models due to the normalization layer of G-CNN-6MN. A lower accuracy of 84.08% and 77.12% are obtained using training and test datasets respectively with the G-CNN-6M model. In most

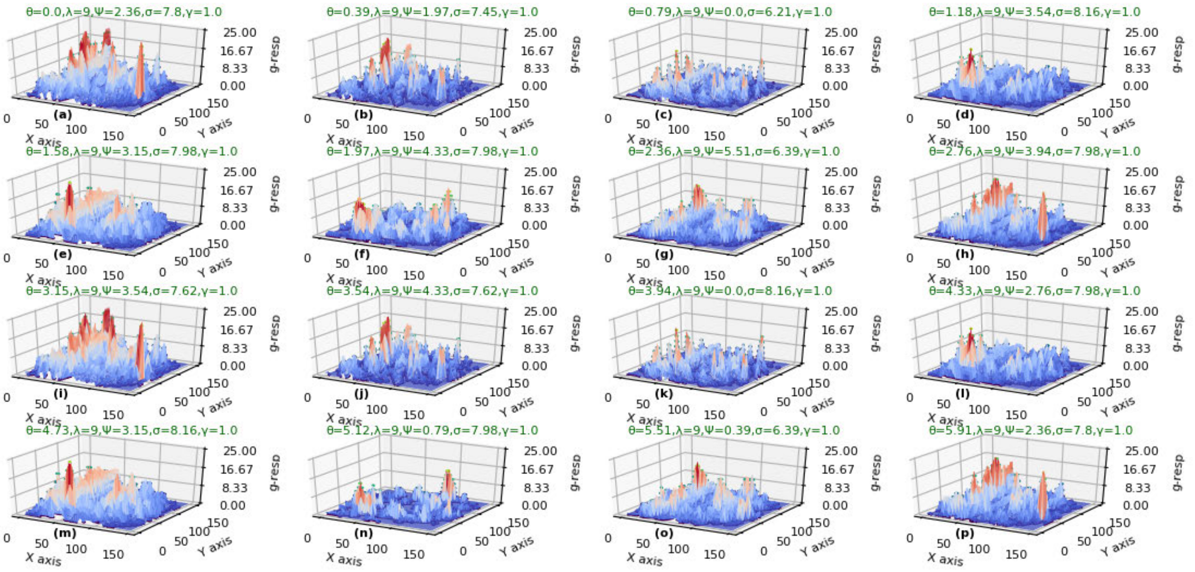


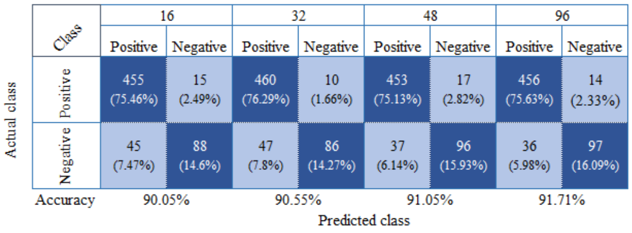
FIGURE 6. Phase response of Gabor transformation obtained with $n = 16$ filters provided in Table 3.

TABLE 4. Results of a10-fold CV are obtained for the developed G-CNN models using the SentiBank database.

Model	# Filters	Train Accuracy in %	Test Accuracy in %	Test Sensitivity in %	Test Precision in %	Test Specificity in %	Test F-score in %
CNN-6MN	16	91.71	80.10	90.64	84.87	42.86	87.66
CNN-6MN	32	94.12	83.42	93.41	86.42	48.13	89.78
CNN-6MN	48	93.74	82.92	90.86	87.68	54.89	89.24
CNN-6MN	96	95.36	84.08	92.77	87.56	53.39	90.09
G-CNN-6M	16	84.08	77.12	96.18	79.03	9.78	86.76
G-CNN-6M	32	87.53	78.61	92.77	82.11	28.58	87.12
G-CNN-6M	48	91.75	81.10	96.81	82.13	25.57	88.87
G-CNN-6M	96	93.33	85.41	98.30	85.24	39.85	91.31
G-CNN-6MN	16	99.22	90.05	96.81	91.00	66.17	93.82
G-CNN-6MN	32	99.38	90.55	97.88	90.73	64.67	94.17
G-CNN-6MN	48	98.89	91.05	96.39	92.45	72.19	94.38
G-CNN-6MN	96	99.76	91.71	97.03	92.69	72.94	94.81

cases, an increase in the number of Gabor filters in the convolutional layer from 16 to 96, increased the accuracy. The G-CNN-6M has given accuracy on the test set of 77.12% for 16 filters while it increased to 85.41% for 96 filters. For the CNN-6MN model without Gabor filters, the training accuracy is between 91.71% to 95.36% and the testing accuracy between 80.10% to 84.08% is obtained. The maximum classification accuracy of 91.71% is observed for the test dataset for the G-CNN-6MN model with 96 Gabor filters.

Figure 7 shows the confusion matrix obtained for the developed G-CNN-6MN model using the SentiBank database. This confusion matrix depicts the actual versus predicted classes using the G-CNN-6MN model with 16, 32, 48, and 96 filters. The number of true positives predicted for a model with 16 filters is 455, and 460 with 32 filters. The maximum number of false positives is 17 for the G-CNN-6MN model with 48 filters. The true negative rate for G-CNN-6MN with 16, 32, 48, and 96 filters are 14.6%, 14.27%, 15.93%, and 16.09% respectively. As the number of filters increased from 16 to 96, which increased the overall performance of the model.



	Class	16		32		48		96	
		Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
Actual class	Positive	455 (75.46%)	15 (2.49%)	460 (76.29%)	10 (1.66%)	453 (75.13%)	17 (2.82%)	456 (75.63%)	14 (2.33%)
	Negative	45 (7.47%)	88 (14.6%)	47 (7.8%)	86 (14.27%)	37 (6.14%)	96 (15.93%)	36 (5.98%)	97 (16.09%)
Accuracy		90.05%		90.55%		91.05%		91.71%	

Predicted class

FIGURE 7. Confusion matrix obtained for the developed G-CNN-6MN model using the SentiBank database.

Figure 8 depicts training and validation losses obtained for the G-CNN model using the SentiBank database during 25 epochs of the training process. These values shown are an average of 5- folds obtained during the training process.

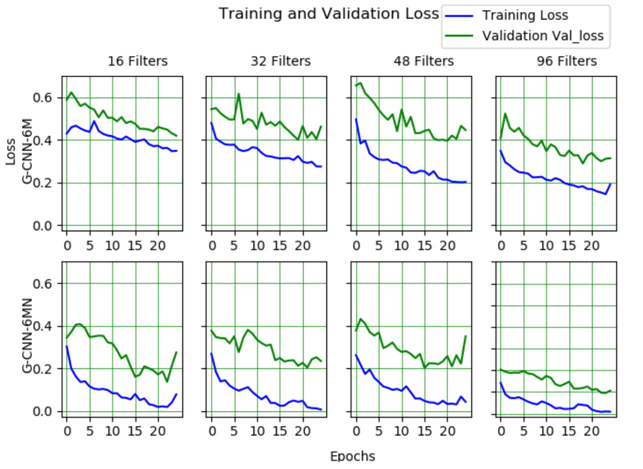
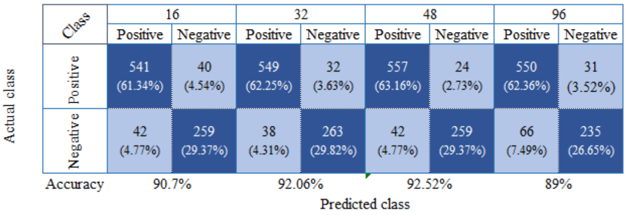


FIGURE 8. Training and validation losses obtained for G-CNN model using SentiBank database.

The training and validation loss corresponding to G-CNN-6M using 16 filters are shown in the first left corner. For this model, the average loss obtained at the first epoch using the training dataset is 0.43026 and is reduced to 0.42106 after the 10th epoch. The training loss after 25 epochs is 0.34924. The average loss for the validation dataset is 0.58912 at the first epoch, it is 0.47594 for the 10th epoch, and 0.41994 for the 25th epoch using the same model. There is a gradual decrease in the training and validation loss observed for G-CNN-6M and G-CNN-6MN models with 16, 32, 48, and 96 filters. As the number of filters is more in G-CNN-6MN with 96 filters it could reduce the error during the learning phase to the lowest value compared to other models.

Table 5 shows the results with a10-fold CV for CNN-6MN, G-CNN-6M, and G-CNN-6MN models using the Twitter database. A minimum accuracy of 66.67% for G-CNN-6M with 32 filters and maximum accuracy of 92.52% for G-CNN-6MN with 48 filters is obtained using the test dataset. Specificity varied from 26.91% for G-CNN-6M with 16 filters to 87.38% for G-CNN-6MN with 32 filters. The increase in the number of true positives and then reduction in false positives and false negatives improve the F-score. For the G-CNN-6MN model using 48 Gabor filters, the highest F-score of 94.41% is obtained. The CNN-6MN model without Gabor filters that have achieved the test accuracies of 83.45%, 82.43%, 82.43%, and 83.22% with 16, 32, 48, and 96 filters respectively which are lower compared to G-CNN-6MN.

Using the Twitter database, the performance of the G-CNN-6MN model with different filters is presented in terms of the confusion matrix in Figure 9. The total number of samples in the database is 882. The G-CNN-6MN model with 16, 32, 48, and 96 filters has obtained 541, 549, 557, and 550 true positives respectively. The true positive rate of 63.16%, false positive rate of 2.73%, false negative rate of 4.77%, and true negative rate of 29.37% are obtained for G-CNN-6MN with 48 filters.



Actual class	Class	16		32		48		96	
		Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
	Positive	541 (61.34%)	40 (4.54%)	549 (62.25%)	32 (3.63%)	557 (63.16%)	24 (2.73%)	550 (62.36%)	31 (3.52%)
	Negative	42 (4.77%)	259 (29.37%)	38 (4.31%)	263 (29.82%)	42 (4.77%)	259 (29.37%)	66 (7.49%)	235 (26.65%)
Accuracy		90.7%		92.06%		92.52%		89%	

Predicted class

FIGURE 9. Confusion matrix obtained for the G-CNN-6MN model using the Twitter database.

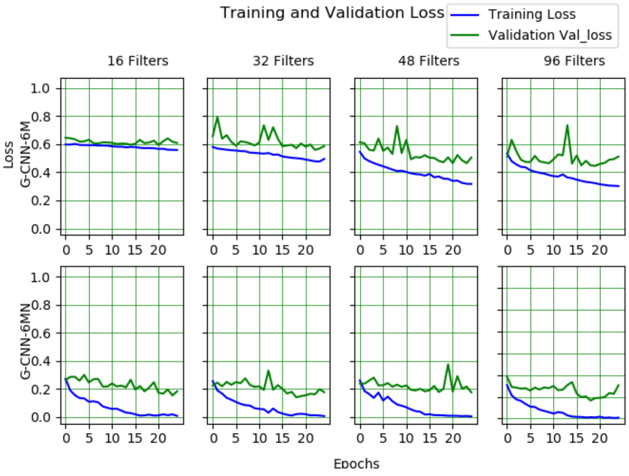


FIGURE 10. Training and validation losses obtained using the G-CNN model with the Twitter database.

The losses observed during the training phase of the G-CNN model using the Twitter database are shown in Figure 10. These loss values corresponding to the training and validation set are obtained after taking the average over 10-fold. The loss for model G-CNN-6M is shown in the first row of Figure 10. The validation loss at the end of the 25th epoch is verified to observe their impact on test accuracies. For the G-CNN-6M model with 16 filters validation loss is 0.60852, for 32 filters is 0.58626, for 48 filters is 0.50460 and for 96 filters is 0.51182. While the validation loss of 0.183187, 0.17712, 0.175905, and 0.305360 are obtained for the G-CNN-6MN model with 16, 32, 48, and 96 filters respectively. Improved accuracies of 89.00% to 92.52% are observed for G-CNN-6MN than G-CNN-6M whose accuracies are between 66.67% to 75.17%.

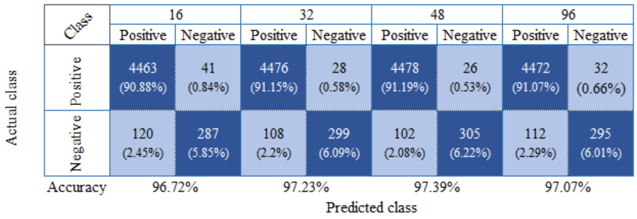
The CV results for *two* G-CNN models using the MVSO database are given in Table 6. The G-CNN-6M with 16 filters has provided an accuracy of 92.04% and 91.61% using training and testing datasets respectively. For the same model sensitivity of 99.67%, specificity of 2.46%, and a F-score of 95.61% are obtained. The G-CNN-6MN models have yielded better results compared to G-CNN-6M. The CNN-6MN has obtained accuracies between 81.43% to 82.37% using the test dataset. Among all G-CNN-6M models, maximum accuracy of 94.26% is obtained using the test dataset for the G-CNN-6M model with 96 filters. Maximum accuracy of 99.96% is

TABLE 5. 10-fold CV results are obtained for the proposed G-CNN models using the Twitter database.

Model	# Filters	Train Accuracy in %	Test Accuracy in %	Test Sensitivity in %	Test Precision in %	Test Specificity in %	Test F-score in %
CNN-6MN	16	95.70	83.45	88.99	86.32	72.76	87.63
CNN-6MN	32	95.89	82.43	89.33	84.81	69.11	87.01
CNN-6MN	48	96.08	82.43	88.65	85.27	70.44	86.92
CNN-6MN	96	96.15	83.22	87.61	87.01	74.76	87.31
G-CNN-6M	16	71.57	67.12	87.95	69.9	26.91	77.90
G-CNN-6M	32	73.47	66.67	84.17	70.77	32.89	76.89
G-CNN-6M	48	82.40	73.47	86.06	76.57	49.17	81.04
G-CNN-6M	96	85.06	75.17	83.30	79.87	59.47	81.55
G-CNN-6MN	16	99.94	90.70	93.12	92.80	86.05	92.96
G-CNN-6MN	32	100.0	92.06	94.49	93.53	87.38	94.01
G-CNN-6MN	48	100.0	92.52	95.87	92.99	86.05	94.41
G-CNN-6MN	96	97.34	89.00	94.66	89.29	78.07	91.90

TABLE 6. 10-fold CV results obtained using G-CNN models with MVSO database.

Model	# Filters	Train Accuracy in %	Test Accuracy in %	Test Sensitivity in %	Test Precision in %	Test Specificity in %	Test F-score in %
CNN-6MN	16	89.68	81.43	85.11	94.09	40.79	89.37
CNN-6MN	32	90.02	81.84	85.20	94.47	44.72	89.59
CNN-6MN	48	90.30	82.23	85.53	94.58	45.71	89.83
CNN-6MN	96	90.32	82.37	85.62	94.65	46.44	89.91
G-CNN-6M	16	92.04	91.61	99.67	91.87	2.460	95.61
G-CNN-6M	32	93.52	91.59	99.38	92.08	5.410	95.59
G-CNN-6M	48	96.41	93.52	98.40	94.74	39.56	96.54
G-CNN-6M	96	97.44	94.26	98.93	95.01	42.51	96.93
G-CNN-6MN	16	99.85	96.72	99.09	97.38	70.52	98.23
G-CNN-6MN	32	99.86	97.23	99.38	97.64	73.46	98.50
G-CNN-6MN	48	99.96	97.39	99.42	97.77	74.94	98.59
G-CNN-6MN	96	99.90	97.07	99.29	97.56	72.48	98.42



Actual class	Class	16		32		48		96	
		Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
	Positive	4463 (90.88%)	41 (0.84%)	4476 (91.15%)	28 (0.58%)	4478 (91.19%)	26 (0.53%)	4472 (91.07%)	32 (0.66%)
	Negative	120 (2.45%)	287 (5.85%)	108 (2.2%)	299 (6.09%)	102 (2.08%)	305 (6.22%)	112 (2.29%)	295 (6.01%)
	Accuracy	96.72%		97.23%		97.39%		97.07%	

Predicted class

FIGURE 11. Confusion matrix obtained for the G-CNN-6MN model using the MVSO database.

obtained with training and 97.39% with the testing dataset for the G-CNN-6MN model with 48 filters.

Figure 11 shows the confusion matrix obtained for the G-CNN-6MN model using the MVSO database. The true positive rate of 91.19%, a false positive of 0.53%, a false negative of 2.08%, and a true negative of 6.22% are obtained for G-CNN-6MN with 48 filters.

Figure 12 shows the training and validation losses obtained for the model during the 25 epochs of the training phase. For the G-CNN-6MN model with 16 filters training loss of 0.01856 and validation loss of 0.091793 is reached at the end of 25 epochs. It can be observed that training loss of 0.014574 and validation loss of 0.081919 for G-CNN-6MN with 96 filters is obtained at the end of 25 epochs.

The G-CNN and CNN models with 16, 32, 48, and 96 filters are trained and tested using a 10-fold CV with the MultiView_I database. These results are tabulated in Table 7. G-CNN-6M with 32 filters obtained an accuracy of 57.41% with the test dataset. As the number of filters increased from 16 to 96, the accuracy of classification gradually increased. The maximum classification accuracy of 76.07% is obtained using CNN-6MN with 96 filters. All the models of G-CNN-6MN have obtained above 99% train accuracies. Maximum accuracy of 90.88% is obtained using the test dataset for G-CNN-6MN with 32 filters. The confusion matrix for the developed G-CNN-6MN model using the MultiView_I dataset is shown in Figure 13.

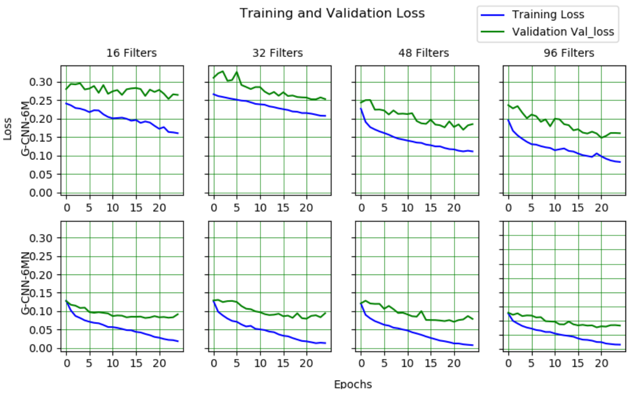
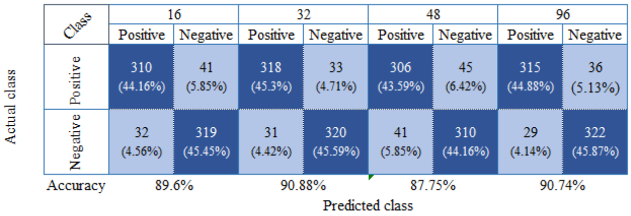


FIGURE 12. Training and validation losses obtained for the developed G-CNN model using the MVSO database.

TABLE 7. 10-fold CV results obtained for the developed G-CNN models using the MultiView_I database.

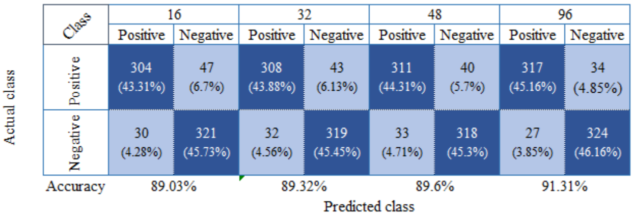
Model	# Filters	Train Accuracy in %	Test Accuracy in %	Test Sensitivity in %	Test Precision in %	Test Specificity in %	Test F-score in %
CNN-6MN	16	89.04	75.50	75.22	75.65	75.79	75.43
CNN-6MN	32	89.07	75.65	73.79	76.63	77.50	75.19
CNN-6MN	48	88.97	73.51	71.80	74.34	75.22	73.05
CNN-6MN	96	89.14	76.07	73.79	77.32	78.35	75.52
G-CNN-6M	16	57.55	57.55	91.74	54.48	23.36	68.37
G-CNN-6M	32	60.90	57.41	62.39	56.74	52.42	59.43
G-CNN-6M	48	75.53	65.24	72.36	63.34	58.12	67.55
G-CNN-6M	96	71.44	64.39	73.79	62.11	54.99	67.45
G-CNN-6MN	16	99.79	89.60	88.32	90.64	90.88	89.47
G-CNN-6MN	32	99.75	90.88	90.60	91.12	91.17	90.86
G-CNN-6MN	48	99.75	87.75	87.18	88.18	88.32	87.68
G-CNN-6MN	96	99.79	90.74	89.74	91.57	91.74	90.65



	Class	16		32		48		96	
		Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
Actual class	Positive	310 (44.16%)	41 (5.85%)	318 (45.3%)	33 (4.71%)	306 (43.59%)	45 (6.42%)	315 (44.88%)	36 (5.13%)
	Negative	32 (4.56%)	319 (45.45%)	31 (4.42%)	320 (45.59%)	41 (5.85%)	310 (44.16%)	29 (4.14%)	322 (45.87%)
Accuracy		89.6%		90.88%		87.75%		90.74%	

Predicted class

FIGURE 13. Confusion matrix obtained for the G-CNN-6MN model using the MultiView_I database.



	Class	16		32		48		96	
		Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
Actual class	Positive	304 (43.31%)	47 (6.7%)	308 (43.88%)	43 (6.13%)	311 (44.31%)	40 (5.7%)	317 (45.16%)	34 (4.85%)
	Negative	30 (4.28%)	321 (45.73%)	32 (4.56%)	319 (45.45%)	33 (4.71%)	318 (45.3%)	27 (3.85%)	324 (46.16%)
Accuracy		89.03%		89.32%		89.6%		91.31%	

Predicted class

FIGURE 15. Confusion matrix obtained for the G-CNN-6MN model with MultiView_II database.

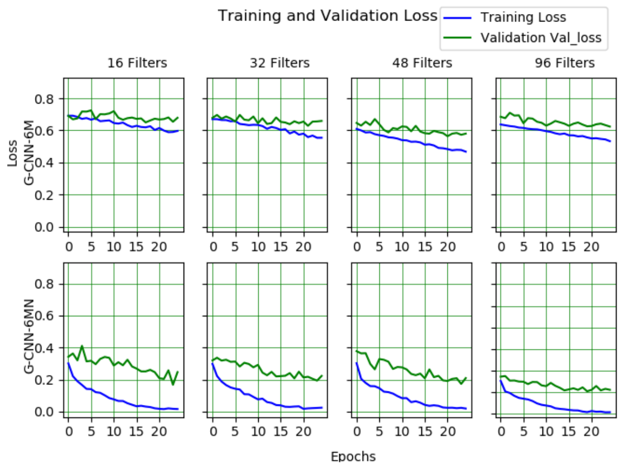


FIGURE 14. 14 Training and validation losses obtained for G-CNN using the MultiView_I database.

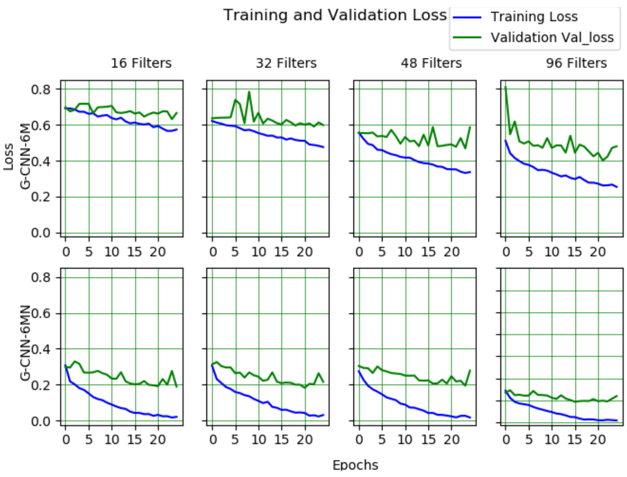


FIGURE 16. Training and validation losses obtained for the developed G-CNN model using the MultiView_II database.

For the MultiView_I database *two* models with a different number of filters are trained and the losses obtained during training and validation are depicted in Figure 14.

Table 8 shows the results obtained using the G-CNN model with a 10-fold CV using the MultiView_II database. The CNN-6MN has given classification accuracy between 75.08% to 75.79% using the test dataset. Compared to G-CNN-6M, the G-CNN-6MN has provided better accuracies for train and test datasets. The model G-CNN-6MN with 96 filters has obtained a maximum accuracy of 91.31% and a F-score of 91.22% using the test dataset with the MultiView_II database. The confusion matrix obtained for the G-CNN-6MN model is shown in Figure 15. The training and validation losses for the G-CNN model with the MultiView_II database are shown in Figure 16.

VIII. DISCUSSION

The *two* G-CNN models G-CNN-6M and G-CNN-6MN are developed in this research. The G-CNN architectures are depicted in Fig. 1 and Fig. 2. Visualization of the feature maps helps to understand the internal representation of the convolutional layers in a CNN model [84]. The feature maps of convolutional layers provide insights into the learning and

TABLE 8. Results are obtained using the G-CNN model with a10-fold CV using the MultiView_II database.

Model	#	Train Accuracy	Test Accuracy	Test Sensitivity	Test Precision	Test Specificity	Test F-score
	Filters	in %	in %	in %	in %	in %	in %
CNN-6MN	16	89.11	75.08	72.94	76.20	77.21	74.53
CNN-6MN	32	88.86	75.50	74.36	76.10	76.64	75.22
CNN-6MN	48	89.07	75.79	72.94	77.35	78.64	75.08
CNN-6MN	96	88.93	75.36	73.51	76.34	77.21	74.90
G-CNN-6M	16	59.26	58.12	63.53	57.33	52.71	60.27
G-CNN-6M	32	75.78	65.95	74.93	63.53	56.98	68.76
G-CNN-6M	48	83.62	72.79	79.49	70.10	66.10	74.50
G-CNN-6M	96	80.73	73.93	66.67	78.00	81.20	71.89
G-CNN-6MN	16	99.47	89.03	86.61	91.02	91.45	88.76
G-CNN-6MN	32	99.61	89.32	87.75	90.59	90.88	89.15
G-CNN-6MN	48	99.68	89.60	88.60	90.41	90.60	89.50
G-CNN-6MN	96	99.61	91.31	90.31	92.15	92.31	91.22

functioning of the layer [85]. We carried out the visualization of feature maps of the G-CNN-6MN to explore the functions of the layers. The second model G-CNN-6MN has *seven* convolutional layers in it. In Fig. 17 the feature maps for the G-CNN-6MN are shown. Fig. 17(a) depicts the feature map of the convolutional layer 1, for convolutional layer 4 the feature map is presented in Fig. 17(b). These feature maps are produced at the end of the 25 epoch training of the G-CNN-6MN model on the database SentiBank. Each layer in the G-CNN-6MN receives the input from the previous layer and performs convolution, max-pooling, and normalization operations to detect features. The feature representation at the end of 25 epochs for convolutional layer 1 is shown in Fig. 17(a). Further, the feature representations for intermediate and higher layers from Convolution Layer-4 to Convolution Layer-16 are given in Fig. 17(b) to Fig. 17(f). As observed by these figures, the lower layers represent simple features while the complex features are represented by higher layers [86].

The classification results obtained on the developed *two* Gabor CNN architectures G-CNN-6M and G-CNN-6MN along with CNN-6MN are presented in section 7. The comparison between G-CNN-6M, G-CNN-6MN, and CNN-6MN is made both in terms of computational cost and classification accuracy. For the computational cost, fewer filters involving convolutional operations require less computational cost than more number of filters. The number of filters used in CNN-6MN, G-CNN-6M, and G-CNN-6MN varied from 16 to 96. For SentiBank database results shown in Table 4, CNN-6MN with 96 non-Gabor filters has given a maximum accuracy of 84.08% on the test set. However, G-CNN-6MN has achieved a classification accuracy of more than 84.08% which is 90.05% with 16 filters in the convolutional layer. The only difference between CNN-6MN and G-CNN-6MN is the application of Gabor filters in the convolutional layer while the rest of the architecture of both these CNNs is the same. The use of Gabor filters in G-CNN models has improved the classification accuracy and even with fewer filters and better accuracy is obtained. Moreover, as the number of filters increased from 16, 32, 48, and 96 improvements in the classification accuracy of 90.05%, 90.55%, 91.05%, and 91.71% on the test set are observed for G-CNN-6MN as shown in Table 4 for SentiBank. For the Twitter database in Table 5, the G-CNN-6MN produced classification accuracy of 90.70%, 92.06%, 92.52%, and 89.00% with 16, 32, 48, and 96 filters respectively while CNN-6MN have given 83.45%, 82.43%, 82.43% and 83.22% accuracies on the test set. Similar observations are made on the results obtained for the MVSO database in Table 6. The maximum classification accuracy of 82.37% on the test set is obtained for CNN-6MN with 96 filters while G-CNN-6MN gave 96.72% classification accuracy on the test using 16 filters. The G-CNN-6M has 6 max-pooling layers and CNN-6MN includes both 6 max-pooling and normalization layers in its architecture. It is observed that the classification accuracy on the test set in some instances is better for CNN-6MN than G-CNN-6M. In Table 4 CNN-6MN with 16 filters has given 80.10% accuracy but G-CNN-6M with 16 filters only gave 77.12% accuracy on the test set. A maximum of 85.41% test accuracy is obtained for G-CNN-6M with 96 filters on the SentiBank database. Even the test classification accuracy for G-CNN-6M with filters 16, 32, 48, and 96 have achieved 57.55%, 57.41%, 65.24%, and 64.39% respectively on the test set of the MultiView_I database as presented in Table 7. For the same dataset MultiView_I, G-CNN-6MN has produced classification accuracy on the test set of 89.60%, 90.88%, 87.75%, and 90.74% with the number of filters 16, 32, 48, and 96 respectively. For the MultiView_II database, as shown in Table 8, G-CNN-6MN with 96 filters has reached a maximum classification accuracy of 91.31%, but classification accuracy for CNN-6MN of 75.79% and G-CNN-6M of 73.93% on the test set are obtained. The utilization of the max-pooling layer along with the normalization layer has a higher impact on the accuracy results both in G-CNN-6MN and CNN-6MN. The training and validation losses during the learning process of G-CNN-6M and G-CNN-6MN are depicted in Fig. 8, Fig. 10, Fig. 12, Fig. 14, and Fig. 16 for SentiBank, Twitter, MVSO, MultiView_I, and MultiView_II databases respectively. Higher training and validation losses are observed for G-CNN-6M compared to G-CNN-6MN in these figures. The use of max-pooling and normalization layer is evident for reducing the training losses in G-CNN-6MN.

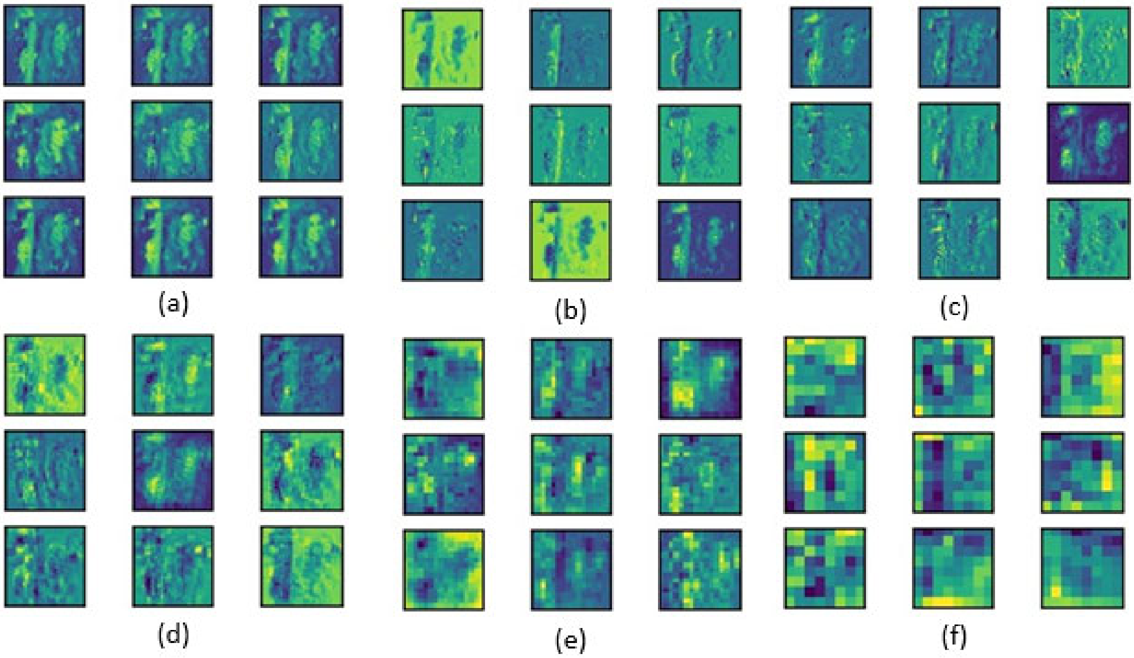


FIGURE 17. Feature maps of (a) Convolution Layer-1. (b) Convolution Layer-4. (c) Convolution Layer-7. (d) Convolution Layer-10. (e) Convolution Layer-13. (f) Convolution Layer-16.

TABLE 9. Results of a 10-fold CV are obtained fromdilated CNN models using the SentiBank database.

Model	# Filters	Train Accuracy in %	Test Accuracy in %	Test Sensitivity in %	Test Precision in %	Test Specificity in %	Test F-score in %
CNN-Dilated_2	16	92.21	81.43	91.49	85.66	45.87	88.48
CNN-Dilated_2	32	95.03	83.75	93.62	86.62	48.88	89.98
CNN-Dilated_2	48	95.65	84.25	91.71	88.51	57.90	90.08
CNN-Dilated_2	96	94.86	82.76	91.92	86.75	50.38	89.26
CNN-Dilated_P	16	93.54	85.08	93.83	87.85	54.14	90.75
CNN-Dilated_P	32	94.03	84.42	94.05	87.01	50.38	90.39
CNN-Dilated_P	48	96.32	83.26	91.07	87.89	55.64	89.45
CNN-Dilated_P	96	95.03	84.08	92.77	87.56	53.39	90.09

The metric such as sensitivity, precision, specificity, and F-score indicate the performance of the G-CNN models. These measurements on the test set are required to gauge and compare the effectiveness of the prediction models. The sensitivity metric for image sentiment classification signifies the performance of the model on the positive sentiment instances. While specificity measures how well a model predicted the negative sentiment instances. The G-CNN-6MN has achieved a better sensitivity rate between 96.39% to 97.88% while the specificity rate is observed between 64.67% to 72.94% on the SentiBank database as shown in Table 4. For the SentiBank database, the model CNN-6MN has a maximum specificity of 54.89% and G-CNN-6M has given a maximum specificity of 39.85%. For the Twitter database as shown in Table 5, the specificity rates are better as compared to the specificity rates of the SentiBank database. A maximum specificity rate of 87.38% is obtained for G-CNN-6MN with 32 filters and a sensitivity rate of 95.87% is given by G-CNN-6MN with 48 filters. The number of samples for the positive sentiment is 470 while the negative samples are 133 for the SentiBank database. For the Twitter database, positive samples are 581 and negative samples are 301. The specificity rate on the Twitter database has improved due to the availability of more negative samples as compared to the SentiBank database. The CNN models have shown better performance to individual classes when there are more samples used for that class during the training. There are 351 positive and negative samples available in both MultiView_I and MultiView_II databases. In Table 7 and Table 8 the sensitivity rate and specificity rates for various CNN models are presented. In Table 7 of MultiView_I, the sensitivity rates in the range of 87.18% to 90.60% and specificity rates in the range of 88.32% to 91.74% are obtained using G-CNN-6MN. Similar results as presented in Table 8, indicate that an equal number of samples in both classes will provide comparable performance for sensitivity and specificity rates. F-score represents the harmonic mean of precision and sensitivity values and it measures the overall accuracy of the models on the database. For the SentiBank database, the model CNN-6MN with 96 filters has given a 90.09% F-score on the test set. F-score has improved with the use of Gabor filters as in G-CNN-6MN with 96 filters and obtained a F-score of 94.81%. Similar improvements in the F-score are observed for G-CNN-6MN with 96 filters on the Twitter database of 91.90%, MVSO database of 98.42%, MultiView_I database of 90.65%, and MultiView_II database of 91.22%.

TABLE 10. Summary of results obtained using the proposed model (G-CNN-6MN) for various databases.

Database	# Filters	Test Accuracy in %
SentiBank	96	91.71
Twitter	48	92.52
MVSO	48	97.39
MultiView_I	32	90.88
MultiView_II	96	91.31
Average	–	92.76

TABLE 11. Synopsis of comparison ofthe developed system with state-of-the-art methods.

Authors	Method	Accuracy %
Borth 2013 [12]	Linear SVM	67.00
	LR	70.00
You 2015 [13]	PCNN	77.30
	CNN	78.30
Campos 2017 [60]	CaffeNet-fc9	79.50
	DeepSentiBank	80.60
	PlacesCNN	82.30
	Fine-tuned CaffeNet with oversampling	83.00
	MVSO [EN] With oversampling	84.40
A.Kumar 2019 [61]	SentiBank+RCNN on images	76.04
Proposed Method	**G-CNN-6MN**	**92.76**

Then experiments are conducted to compare the image sentiment classification results with *two* dilated CNN models. The CNN-Dilated models are constructed with a similar structure as that of G-CNN-6M and G-CNN-6MN, but with a few modifications. The max-pooling and normalization layers are removed and then dilations are introduced. For the first dilated CNN model (CNN-Dilated_2) we have introduced a constant dilation rate of 2 to all the convolutional layers. For the second CNN with dilation, a progressively (CNN-Dilated_P) increasing dilation rate of 1, 2, and 4 then decreasing dilation rate 2,1 are followed as in [87] and [45]. To achieve a fair comparison with our proposed model (G-CNN-6MN) the same number of filters are used in both CNN-Dilated models. The results are collected on the SentiBank for 10-fold cross-validation and these results are summarized in Table 9. For CNN-Dilated_2 with 16 filters the training accuracy of 92.21% and test accuracy of 81.43% is obtained. The CNN-Dilated_P with 48 filters has given accuracy of 96.32% for the training set and 83.26% for the test set. A maximum accuracy on the test set of 85.08% is obtained for CNN-Dilated_P with 16 filters, but our proposed G-CNN-6MN has achieved 91.71% for the same database.

The summary of best accuracies obtained using the proposed G-CNN-6MN model on various databases is shown in Table 10. Maximum accuracy of 97.39% is obtained on the MVSO database while a minimum of 90.88% accuracy is observed on MultiView_I. The recent state-of-the-art works for sentiment image recognition using ML and CNN models are presented in Table 11. In [12] the methods used for the classification task are Linear SVM and LR, which have given classification accuracies between 67.0% to 70.0%. The CNN models were employed in [13]. They used *two* models, PCNN produced a classification accuracy of 77.3% and CNN an accuracy of 78.3% obtained. In [60] several deep learning models were developed. A fine-tuning was carried out for the CaffeNet-fc9 model in [60] and this model has produced a classification accuracy of 79.50%. Fine-tuning and oversampling were performed for the model fine-tuned CaffeNet with oversampling, which has achieved a classification accuracy of 83.00%. The oversampled MVSO [EN] with oversampling has given 84.4% accuracy in their work. The image sentiment classification was carried out in [61] using region CNN. They reported 76.04% classification accuracy on the SentiBank database. In our work, we integrated Gabor filters with CNN and developed G-CNN-6M and G-CNN-6MN models. Using our developed G-CNN-6MN model an average of 92.76% accuracy is achieved as given in Table 10.

IX. CONCLUSION

Understanding the user's emotions through sentiment analysis of text, images, and videos from social media has wide applications in marketing research. The sentiment recognition of images is a difficult task because emotional features are abstract in nature. The CNN models have been applied successfully to many image recognition and analysis tasks due to their capability to learn prominent features. The visual data analysis and recognition using CNN models involves deep layers and demands high computational costs [20], [21]. In recent years cloud computing environment has become popular due to its high reliability, mobility, and software integration. While developing CNN applications in the cloud environment, the computational cost is important due to remote server location, the need for resource optimization, and infrastructure cost reduction. Hence CNN models in the cloud environment require efficacious in both performance and computational cost. Integrating Gabor filters with CNN models have improved the performance [27], [29], the decreased time required [28], [29], and computation energy [26], [28].

In this research, we developed *two* new CNN models by integrating Gabor filters with them. The Gabor integrated models G-CNN-6M and G-CNN-6MN are developed for image sentiment analysis in the cloud environment. A modified method to design Gabor filters using 3D information diagrams is devised. This method involved selecting Gabor filters that produced the maximum responses in the 3D information diagram. The selected filters are integrated with the convolutional layers of G-CNN-6M and G-CNN-6MN. A different number of layers of max-pooling and normalization are included in G-CNN architecture to improve their performance. Our developed G-CNN-6M has *six* max-pooling layers and G-CNN-6MN has *six* max-pooling and normalization layers. The image sentiment experiments are carried out on *five* databases: SentiBank, Twitter, MVSO, MultiView_I, and MultiView_II. In order to have a fair comparison, a CNN model (CNN-6MN) with the same number of layers but without Gabor filters is developed. The experiment results of G-CNN-6M and G-CNN-6MN are compared with CNN-6MN. On the SentiBank database, CNN-6MN with 96 filters has given maximum accuracy of 84.08% but using G-CNN-6MN with 16 filters has achieved 90.05% accuracy on the test set. Similar results are obtained in the experiments showing the results of G-CNN-6MN better compared to CNN-6MN even with fewer filters. Thus, reducing the computational cost for the G-CNN-6MN model. The image classification accuracy has been improved in our developed G-CNN models. On SentiBank, maximum classification accuracies on test set of 85.41% and 91.71% are obtained for models G-CNN-6M and G-CNN-6MN respectively. A maximum accuracy on the test set of 97.39% is obtained on the MVSO database for the G-CNN-6MN model with 48 filters. From the developed model G-CNN-6MN an average classification accuracy of 92.76% is obtained, which is better compared to the previous works. As future work, optimization techniques such as genetic algorithms or gravitation search algorithms could be applied to select parameters for Gabor filters. Another possible improvement would be the semantic tagging of images during sentiment image classification.

DATA AVAILABILITY

Previously reported data sets for the sentiment images were used to support this study and these studies are SentiBank [12], Twitter [13], MultiView_I [30], MultiView_II [30], and MVSO [12]. The detailed description of the data set is covered in section 4.

FUNDING STATEMENT

No research grant is received to carry out this research.

Declarations: This paper presents original work not previously published in a similar form and not currently under consideration by another Journal. This research work has been carried out at the above-mentioned institutes.

Conflicts of interest/Competing interests: There is no conflict of interest in this work.

Highlights

- Cloud based intelligent system is developed for image sentiment analysis
- Gabor filters are integrated with the convolutional neural network on Cloud
- 3D information diagrams are used to design Gabor filters for CNN
- High accurate classifications are obtained by the proposed Gabor based intelligent system

ACKNOWLEDGMENT

The authors would like to thank the reviewers for providing important comments on our manuscript. After incorporating suggestions given by reviewers, the quality and readability of the manuscript have improved to many folds.

REFERENCES

[1] P. A. Abdalla and A. Varol, "Advantages to disadvantages of cloud computing for small-sized business," in *Proc. 7th Int. Symp. Digit. Forensics Secur. (ISDFS)*, Jun. 2019, pp. 1–6.

[2] S. D. Müller, S. R. Holm, and J. Søndergaard, "Benefits of cloud computing: Literature review in a maturity model perspective," *Commun. Assoc. Inf. Syst.*, vol. 37, no. 1, p. 42, 2015.

[3] R. Ji, D. Cao, Y. Zhou, and F. Chen, "Survey of visual sentiment prediction for social media analysis," *Frontiers Comput. Sci.*, vol. 10, no. 4, pp. 602–611, Aug. 2016.

[4] Q. T. Ain, M. Ali, A. Riaz, A. Noureen, M. Kamran, B. Hayat, and A. Rehman, "Sentiment analysis using deep learning techniques: A review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 1–10, 2017.

[5] A. M. Abirami and V. Gayathri, "A survey on sentiment analysis methods and approach," in *Proc. 8th Int. Conf. Adv. Comput. (ICoAC)*, Jan. 2017, pp. 72–76.

[6] J. Islam and Y. Zhang, "Visual sentiment analysis for social images using transfer learning approach," in *Proc. IEEE Int. Conf. Big Data Cloud Comput. (BDCloud), Social Comput. Netw. (SocialCom), Sustain. Comput. Commun. (SustainCom) (BDCloud-SocialCom-SustainCom)*, Oct. 2016, pp. 124–130.

[7] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[8] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014.

[9] B. Heredia, T. M. Khoshgoftaar, J. Prusa, and M. Crawford, "Cross-domain sentiment analysis: An empirical investigation," in *Proc. IEEE 17th Int. Conf. Inf. Reuse Integr. (IRI)*, Jul. 2016, pp. 160–165.

[10] P. Ducange, M. Fazzolari, M. Petrocchi, and M. Vecchio, "An effective decision support system for social media listening based on cross-source sentiment analysis models," *Eng. Appl. Artif. Intell.*, vol. 78, pp. 71–85, Feb. 2019.

[11] M. Haenlein and A. M. Kaplan, "An empirical analysis of attitudinal and behavioral reactions toward the abandonment of unprofitable customer relationships," *J. Relationship Marketing*, vol. 9, no. 4, pp. 200–228, Nov. 2010.

[12] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang, "Large-scale visual sentiment ontology and detectors using adjective noun pairs," in *Proc. 21st ACM Int. Conf. Multimedia*, Oct. 2013, pp. 223–232.

[13] Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 381–388.

[14] Y. Liang, K. Maeda, T. Ogawa, and M. Haseyama, "Cross-domain semi-supervised deep metric learning for image sentiment analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4150–4154.

[15] G. Preethi, P. V. Krishna, M. S. Obaidat, V. Saritha, and S. Yenduri, "Application of deep learning to sentiment analysis for recommender system on cloud," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2017, pp. 93–97.

[16] R. Arulmurugan, K. Sabarmathi, and H. Anandakumar, "Classification of sentence level sentiment analysis using cloud machine learning techniques," *Cluster Comput.*, vol. 22, no. 1, pp. 1199–1209, 2019.

[17] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generat. Comput. Syst.* vol. 74, pp. 76–85, Sep. 2017.

[18] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 912–925, Apr. 2018.

[19] E. Ragusa, E. Cambria, R. Zunino, and P. Gastaldo, "A survey on deep learning in image polarity detection: Balancing generalization performances and computational costs," *Electronics*, vol. 8, no. 7, p. 783, Jul. 2019.

[20] N. Mahmoudi, S. M. Ahadi, and M. Rahmati, "Multi-target tracking using CNN-based features: CNNMTT," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 7077–7096, Mar. 2019.

[21] X. Sun, L. Liu, C. Li, J. Yin, J. Zhao, and W. Si, "Classification for remote sensing data with improved CNN-SVM method," *IEEE Access*, vol. 7, pp. 164507–164516, 2019.

[22] G. L. Stavrinides and H. D. Karatza, "Cost-effective utilization of complementary cloud resources for the scheduling of real-time workflow applications in a fog environment," in *Proc. 7th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2019, pp. 1–8.

[23] J. Chen, T. Du, and G. Xiao, "A multi-objective optimization for resource allocation of emergent demands in cloud computing," *J. Cloud Comput.*, vol. 10, no. 1, pp. 1–17, Dec. 2021.

[24] M. Guazzone, C. Anglano, and M. Canonico, "Energy-efficient resource management for cloud computing infrastructures," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci.*, Nov. 2011, pp. 424–431.

[25] S.-Y. Chang and N. Morgan, "Robust CNN-based speech recognition with Gabor filter kernels," in *Proc. Interspeech*, Sep. 2014, pp. 1–5.

[26] S. S. Sarwar, P. Panda, and K. Roy, "Gabor filter assisted energy efficient fast learning convolutional neural networks," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2017, pp. 1–6.

[27] Y. Chen, L. Zhu, P. Ghamisi, X. Jia, G. Li, and L. Tang, "Hyperspectral images classification with Gabor filtering and convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2355–2359, Dec. 2017.

[28] F. Meng, X. Wang, F. Shao, D. Wang, and X. Hua, "Energy-efficient Gabor kernels in neural networks with genetic algorithm training method," *Electronics*, vol. 8, no. 1, p. 105, Jan. 2019.

[29] G. Özbulak and H. K. Ekenel, "Initialization of convolutional neural networks by Gabor filters," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, May 2018, pp. 1–4.

[30] T. Niu, S. Zhu, L. Pang, and A. E. Saddik, "Sentiment analysis on multi-view social data," in *Proc. Int. Conf. Multimedia Modeling*. Cham, Switzerland: Springer, 2016, pp. 15–27.

[31] E. Cambria, D. Olsher, and K. Kwok, "Sentic activation: A two-level affective common sense reasoning framework," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 186–192.

[32] I. Perikos and I. Hatzilygeroudis, "Recognizing emotions in text using ensemble of classifiers," *Eng. Appl. Artif. Intell.*, vol. 51, pp. 191–201, May 2016.

[33] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017.

[34] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 4, 2018, Art. no. e1253.

[35] S. S. Roy, M. Biba, R. Kumar, R. Kumar, and P. Samui, "A new SVM method for recognizing polarity of sentiments in Twitter," in *Handbook of Research on Soft Computing and Nature-Inspired Algorithms*. Hershey, PA, USA: IGI Global, 2017, pp. 281–291.

[36] N. Jiang, F. Tian, J. Li, X. Yuan, and J. Zheng, "MAN: Mutual attention neural networks model for aspect-level sentiment classification in SIoT," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2901–2913, Apr. 2020.

[37] A. Ishaq, S. Asghar, and S. A. Gillani, "Aspect-based sentiment analysis using a hybridized approach based on CNN and GA," *IEEE Access*, vol. 8, pp. 135499–135512, 2020.

[38] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, pp. 58080–58093, 2020.

[39] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Med.*, vol. 25, no. 1, pp. 24–29, Jan. 2019.

[40] C. Bhagya and A. Shyna, "An overview of deep learning based object detection techniques," in *Proc. 1st Int. Conf. Innov. Inf. Commun. Technol. (ICIICT)*, Apr. 2019, pp. 1–6.

[41] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Mar. 2019.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[43] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," 2016, *arXiv:1605.07678*.

[44] R. Biswas, A. Vasan, and S. S. Roy, "Dilated deep neural network for segmentation of retinal blood vessels in fundus images," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 44, no. 1, pp. 505–518, Mar. 2020.

[45] S. S. Roy, N. Rodrigues, and Y.-H. Taguchi, "Incremental dilations using CNN for brain tumor classification," *Appl. Sci.*, vol. 10, no. 14, p. 4915, Jul. 2020.

[46] P. Samui, S. S. Roy, and V. E. Balas, *Handbook of Neural Computation*. New York, NY, USA: Academic, 2017.

[47] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, "Review of deep convolution neural network in image classification," in *Proc. Int. Conf. Radar, Antenna, Microw., Electron., Telecommun. (ICRAMET)*, Oct. 2017, pp. 26–31.

[48] V. M. Lidkea, R. Muresan, and A. Al-Dweik, "Convolutional neural network framework for encrypted image classification in cloud-based ITS," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 35–50, 2020.

[49] M. S. Hossain and G. Muhammad, "Emotion recognition using secure edge and cloud computing," *Inf. Sci.*, vol. 504, pp. 589–601, Dec. 2019.

[50] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electron*, vol. 8, no. 3, p. 292, Mar. 2019.

[51] S. Siersdorfer, E. Minack, F. Deng, and J. Hare, "Analyzing and predicting sentiment of images on the social web," in *Proc. Int. Conf. Multimedia (MM)*, 2010, pp. 715–718.

[52] B. Li, S. Feng, W. Xiong, and W. Hu, "Scaring or pleasing: Exploit emotional impact of an image," in *Proc. 20th ACM Int. Conf. Multimedia (MM)*, 2012, pp. 1365–1366.

[53] J. Jia, S. Wu, X. Wang, P. Hu, L. Cai, and J. Tang, "Can we understand van Gogh's mood?: Learning to infer affects from images in social networks," in *Proc. 20th ACM Int. Conf. Multimedia (MM)*, 2012, pp. 857–860.

[54] V. Vonikakis and S. Winkler, "Emotion-based sequence of family photos," in *Proc. 20th ACM Int. Conf. Multimedia (MM)*, 2012, pp. 1371–1372.

[55] J. Yuan, S. Mcdonough, Q. You, and J. Luo, "Sentribute: Image sentiment analysis from a mid-level perspective," in *Proc. 2nd Int. Workshop Issues Sentiment Discovery Opinion Mining (WISDOM)*, 2013, pp. 1–8.

[56] T. Chen, D. Borth, T. Darrell, and S.-F. Chang, "DeepSentiBank: Visual sentiment concept classification with deep convolutional neural networks," 2014, *arXiv:1410.8586*.

[57] Y. Wang and B. Li, "Sentiment analysis for social media images," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 1584–1591.

[58] C. Baecchi, T. Uricchio, M. Bertini, and A. Del Bimbo, "A multimodal feature learning approach for sentiment analysis of social network multimedia," *Multimedia Tools Appl.*, vol. 75, no. 5, pp. 2507–2525, Mar. 2016.

[59] A. Kumar and A. Jaiswal, "Image sentiment analysis using convolutional neural network," in *Proc. Int. Conf. Intell. Syst. Design Appl.* Cham, Switzerland: Springer, 2017, pp. 464–473.

[60] V. Campos, B. Jou, and X. Giró-i-Nieto, "From pixels to sentiment: Fine-tuning CNNs for visual sentiment prediction," *Image Vis. Comput.*, vol. 65, pp. 15–22, Sep. 2017.

[61] A. Kumar and G. Garg, "Sentiment analysis of multimodal Twitter data," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 24103–24119, Sep. 2019.

[62] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4357–4366, Sep. 2018.

[63] A. Calderon, S. Roa, and J. Victorino, "Handwritten digit recognition using convolutional neural networks and Gabor filters," in *Proc. Int. Congr. Comput. Intell.*, 2003, pp. 429–441.

[64] C. Liu, W. Ding, X. Wang, and B. Zhang, "Hybrid Gabor convolutional networks," *Pattern Recognit. Lett.*, vol. 116, pp. 164–169, Dec. 2018.

[65] B. Kwolek, "Face detection using convolutional neural networks and Gabor filters," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2005, pp. 551–556.

[66] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[67] P. Moreno, A. Bernardino, and J. Santos-Victor, "Gabor parameter selection for local feature detection," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.* Cham, Switzerland: Springer, 2005, pp. 11–19.

[68] H. Wu, Y. Yoshida, and T. Shioyama, "Optimal Gabor filters for high speed face identification," in *Proc. Object Recognit. Supported User Interact. Service Robots*, Aug. 2002, pp. 107–110.

[69] F. Smeraldi and J. Bigun, "Retinal vision applied to facial features detection and face authentication," *Pattern Recognit. Lett.*, vol. 23, no. 4, pp. 463–475, Feb. 2002.

[70] J. Nagi, F. Ducatelle, G. A. D. Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, Nov. 2011, pp. 342–347.

[71] A. Giusti, D. C. Ciresan, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Fast image scanning with deep max-pooling convolutional neural networks," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 4034–4038.

[72] Z. Yin, B. Wan, F. Yuan, X. Xia, and J. Shi, "A deep normalization and convolutional neural network for image smoke detection," *IEEE Access*, vol. 5, pp. 18429–18438, 2017.

[73] V. Thakkar, S. Tewary, and C. Chakraborty, "Batch normalization in convolutional neural networks—A comparative study with CIFAR-10 data," in *Proc. 5th Int. Conf. Emerg. Appl. Inf. Technol. (EAIT)*, Jan. 2018, pp. 1–5.

[74] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014, *arXiv:1412.6806*.

[75] Z. Ma, D. Chang, J. Xie, Y. Ding, S. Wen, X. Li, Z. Si, and J. Guo, "Fine-grained vehicle classification with channel max pooling modified CNNs," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3224–3233, Apr. 2019.

[76] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[77] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.

[78] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," in *Proc. 4th Int. Conf. Res. Comput. Intell. Commun. Netw. (ICRCICN)*, Nov. 2018, pp. 122–129.

[79] A. Gajbhiye and K. M. P. Shrivastva, "Cloud computing: Need, enabling technology, architecture, advantages and challenges," in *Proc. 5th Int. Conf. Confluence Next Gener. Inf. Technol. Summit (Confluence)*, Sep. 2014, pp. 1–7.

[80] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, "TSE-CNN: A two-stage end-to-end CNN for human activity recognition," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 1, pp. 292–299, Jan. 2020.

[81] A. Alekseev and A. Bobe, "GaborNet: Gabor filters with learnable parameters in deep convolutional neural network," in *Proc. Int. Conf. Eng. Telecommun. (EnT)*, Nov. 2019, pp. 1–4.

[82] Y. LeCun. (1998). *The Mnist Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[83] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[84] S. Mostafa, D. Mondal, M. Beck, C. Bidinosti, C. Henry, and I. Stavness, "Visualizing feature maps for model selection in convolutional neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1362–1371.

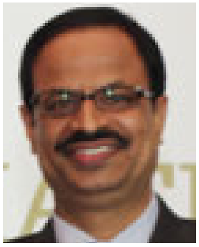
[85] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.

[86] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 609–616.

[87] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 472–480.

SIDDHALING UROLAGIN received the B.E. degree in computer science and engineering from the Gogte Institute of Technology, Belgaum, in 2001, the M.Tech. degree in computer science and engineering from the Manipal Institute of Technology, Manipal, in 2004, and the Ph.D. degree in machine learning and pattern recognition from Manipal University. He worked as a Post-Doctoral Researcher in the application of machine learning to bioinformatics at Lund University, Sweden, from 2012 to 2014. He is currently working as an Assistant Professor with the BITS-Pilani, Dubai, United Arab Emirates. He published around 27 international journal articles, 59 international conference papers, and five book chapters. His research interests include machine learning, data science, deep learning, NLP, text mining, and image recognition. His research has been recognized as the second best research at Manipal University. He has been granted the AICTE fund for an application of document image analysis to translate a scanned document to Braille equivalent during his Ph.D. He has been awarded the Ph.D. degree in computer science and engineering by the President of India Hon'ble Smt. Prathibha Patil, in 2010. He has given invited talks in various forums. Currently, he is a recognized guide in BITS-Pilani and Vishwesharya Technical University, Belgaum, and guiding five Ph.D. students.

JAGADISH NAYAK (Senior Member, IEEE) received the bachelor's degree in electronics and communication engineering from Karnataka University, the Master of Technology degree in digital electronics and communication from the Manipal Academy of Higher Education, Manipal, and the Ph.D. degree from the National Institute of Technology Karnataka Surathkal for his Thesis entitled "Automated Detection of Eye Abnormalities and Patient Data Handling." Earlier to present position, he worked as an Associate Professor at the Manipal Academy of Higher Education and as a Faculty Member at NITK Surathkal. He also worked at a renowned company Bradma of India Ltd., Bengaluru, as a Customer Support Engineer. He is currently working as an Associate Professor with the Birla Institute of Technology and Science-Pilani, Dubai Campus. He has guided one Ph.D. student. He has a total of 27 years of experience both in industry, teaching, and research. He also handled some projects in the field of medical image processing. Recently, he handled couple of projects in the Internet of Things. He has published around 18 research articles in reputed international journals, three book chapters, 20 papers in international conferences, and ten papers in national conferences. His research interests include signal processing and its application, machine learning methods for medical signals and images, microelectronics, VLSI design, embedded systems design, and the Internet of Things. He is a member of several professional bodies. He is also a Reviewer for many international journals, specifically *Journal of Medical Systems*.



U. RAJENDRA ACHARYA received the Ph.D., D.Eng., and D.Sc. degrees. He is currently a Senior Faculty Member with Ngee Ann Polytechnic, Singapore; a Distinguished Professor with the International Research Organization for Advanced Science and Technology, Kumamoto University, Japan; an Adjunct Professor with the University of Malaya, Malaysia; Asia University, Taiwan; and University of Southern Queensland, Australia; and an Associate Faculty with the Singapore University of Social Sciences, Singapore. He has authored over 500 publications, including 345 in refereed international journals, 42 in international conference proceedings, and 17 books. His research interests include biomedical imaging and signal processing, data mining, and visualization, as well as applications of biophysics for better healthcare design and delivery. His funded research has accrued cumulative grants exceeding six million Singapore dollars. He has received over 60,000 citations on Google Scholar, with an H-index of 123. According to the Essential Science Indicators by Thomson, he consistently ranked among the top 1% of Highly Cited Researchers in Computer Science for the last seven years, from 2016 to 2022. He sits on the editorial boards of multiple journals and has served as a guest editor on several AI-related issues.