



Cyberbullying detection using deep transfer learning

Pradeep Kumar Roy¹ · Fenish Umeshbhai Mali¹

Received: 13 August 2021 / Accepted: 1 May 2022 / Published online: 25 May 2022
© The Author(s) 2022

Abstract

Social networking platforms like Facebook, Twitter, and others have numerous advantages, but they have many dark sides also. One of the issues on these social platforms is cyberbullying. The impact of cyberbullying is immeasurable on the life of victims as it's very subjective to how the person would tackle this. The message may be a bully for victims, but it may be normal for others. The ambiguities in cyberbullying messages create a big challenge to find the bully content. Some research has been reported to address this issue with textual posts. However, image-based cyberbullying detection is received less attention. This research aims to develop a model that helps to prevent image-based cyberbullying issues on social platforms. The deep learning-based convolutional neural network is initially used for model development. Later, transfer learning models are utilized in this research. The experimental outcomes of various settings of the hyper-parameters confirmed that the transfer learning-based model is the better choice for this problem. The proposed model achieved a satisfactory accuracy of 89% for the best case, indicating that the system detects most cyberbullying posts.

Keywords Cyberbullying · Deep learning · CNN · Dataset · Transfer learning

Introduction

These days, where there is extensive use of social media by both adults and teens, the exposure of bullying does not remain just to the traditional way of physical bullying. Still, this medium extends it to the new subpart of bullying, i.e., cyberbullying [1,2]. Cyberbullying has many forms like texts, images, videos, etc. Cyberbullying has become the dark side of the well-connected social life of the internet [3,4]. Information can be sent to millions of people in seconds in the current time. Hence, there must be a filter available on the social platforms to monitor the information's health and needed to assure it will not be harmful to the receiver like cyberbullying. Such messages may create an issue with the victim's mental health or sometimes even on their personality [3,5].

According to UNICEF, “cyberbullying is bullying with the use of digital technologies. It can take place on social media, messaging platforms, gaming platforms, and mobile

phones. It is repeated behavior, aimed at scaring, angering or shaming those who are targeted.” It is called bullying when the victim is less than 18 years of age. When an adult is a victim, it is termed as Harassment [3,6]. There can be multiple ways someone can be cyberbullied. You can know if you are cyberbullied if you have to face any one of these: (i) Received a threatening/mean message, (ii) Have been trolled online for an opinion, (iii) Have been intentionally excluded from the group, (iv) If your personal information has been leaked, (v) Received an obscene image without your consent, etc.

The effects of cyberbullying can be severe. It does not only affect the person's body, but it leaves scars in the personality of the victim [1,7]. Cyberbullying causes severe mental health issues, which leads to lower self-esteem and an anxious personality, which in turn completely changes the person's ability to have a peaceful life [3]. In the severe case of cyberbullying, the victim can have suicidal thoughts as well, which can lead to taking their life without the bully even knowing it [8,9]. The importance of this filter demands a powerful tool to detect the bullying posts and filter them out, thus providing a user with a peaceful and bully-free environment. The reported cases of cyberbullying are rising day by day worldwide because of deeper penetration of the internet and more teens coming on social media. The seri-

✉ Pradeep Kumar Roy
pkronitp@gmail.com

Fenish Umeshbhai Mali
ui17co40@iiitsurat.ac.in

¹ Department of Computer Science and Engineering, Indian Institute of Information Technology, Surat, Gujarat, India

ousness of the problem can be seen by the alarming statistics that are provided by these articles by ceoworld.biz.¹ India is the most affected country by cyberbullying in teens, followed by Brazil and United States in the year 2018. It is not just about the number of cases but also about the unreported cases. According to CyberBAPP (a Mumbai-based anti-cyberbullying organisation), one out of three users has been threatened online. In contrast, almost half of the users have been bullied once or more, in which only half of the cases are reported, which makes cyberbullying detection hard. As only about 50% cases are reported, the impact of this filter would help a greater number of people than the thought of.

These statistics indicate that the issue of cyberbullying needs immediate attention. However, the unavailability of the labelled dataset is one of the biggest challenges of this research. To fill this gap, we have developed an image dataset for this research. Recently, much research has been reported with different machine learning (ML) and deep learning (DL) approaches to address cyberbullying for the textual dataset [4,9,10]. However, cyberbullying with the image received comparatively less attention. This leads to a major issue in the current time, where most posts consist of images and textual content. Such bullied posts remain untraced by the system. Hence, this research focused on detecting image-based social cyberbullying posts. To process the image and get the required features from it, DL-based convolutional neural network (CNN) and transfer learning models have been widely used by the research community in the recent past. Many pieces of research apart from cyberbullying detection have been reported recently using the CNN network [3] such as spam detection [11], question answering [12,13], text quality prediction [14], healthcare [15–19]. By following the performance of CNN models on different research on images and text-domain, this research also used the 2DCNN model. Apart from 2DCNN, the transfer learning approach is also used for the same. The main contributions of our research are as follows:

- We proposed a transfer learning-based automated model to detect image-based cyberbully posts from the social platform. The transfer learning models are capable of extracting hidden contextual features from cyberbullying posts.
- Created two sets of datasets (i) having 1000 images and (ii) 3000 images consisting of cyberbullying and non cyberbullying images. The datasets can be useful for future researchers to extend the research.

- Finding the best-suited model to detect the bully images is a challenging task, hence experimented with both DL and transfer learning models to find the best model.
- The experimental outcomes confirmed that the transfer learning models are the better choice for predicting image-based cyberbullying posts.

The rest of the paper is organised as follows: the next section discusses the existing works. the third section highlights the working of 2DCNN models used in the proposed framework. Fourth section discusses the experimental outcomes of different models. Fifth section discusses the findings and highlight the uses of the proposed framework. Finally, the last section concludes the work with limitation and future scope.

Literature review

Recently, cyberbullying attracted huge attention from the research community. This section discusses the relevant research contribution on cyberbullying detection [3,20–24]. Hosseinmardi et al. [24] developed a model for cyberbullying detection on Instagram by extracting captions, comments and image content. The dataset was developed using Instagram API and the public profile of the users. Further, the collected dataset was annotated using the CrowdFlower platform, where annotators needed to pass a quiz to label the data. The dataset was divided into three sets:(i) Set40+ had 49% as non-cyber bullying and rest bullying; (ii) Set 0+ had 15% not bullying rest bullying and (iii) Set 0 where there were no bullying instances. The ratio of 80:20 was followed for training and testing to the model. The features were extracted from the data as followers, following, early caption, Image content etc. Logistic regression was used to predict the bullying post on the set40+ dataset-features like early comments, caption, post time, user properties, and image content are used. The unigram and bigram combined features produced the F1-score of 0.84.

AlAjlan et al. [25] used a DL model for cyberbullying detection. They used feature selection and feature engineering techniques to extract the features from input. They used a Twitter dataset consisting of 39,000 tweets from which the duplicates were removed during cleaning. The model was trained with 9000 bully, 21,000 non-bully tweets and tested with 2700 bully 6300 non-bully tweets. Their model performed far better than the SVM, with 95% accuracy. Banerjee et al. [26] used a dataset consisting of 69,874 tweets. They converted the word into vectors using Glove word embedding. Removal of stop word accentuation marks was done, and then converting to lowercase was performed during data preprocessing. On processed data, they used CNN based

¹ <https://ceoworld.biz/2018/10/29/countries-where-cyber-bullying-was-reported-the-most-in-2018/>.

DL model to detect the bully posts and achieved 93.97% of accuracy value. Cigdem et al. [20] developed a model for automatic detection of the cyberbullying instance in a social network using text mining methods. They experiment with different types of classifiers with feature selection algorithms to find the best results. The dataset was acquired from three different social networks: (i) Formspring.me, (ii) Myspace, and (iii) YouTube. The dataset was divided into two classes: (i) cyberbullying positive and (ii) cyberbullying negative. The Formspring.me dataset was in an XML file containing 13,158 messages, from which 892 were cyberbullying positive, and the rest 12,266 messages were cyberbullying Negative. The Myspace dataset consists of 1753 messages, out of which 357 are positive, and the rest 1396 are negatively labelled. The YouTube dataset had 3464 messages from different users and out of which 417 were positive, the remaining 3047 were negative. Two classifiers, SGD and MLP, achieved the f-measure value of more than 0.90 for all datasets.

Kumari et al. [3] used a unified representation of text and image, which would eventually make cyberbullying free social media. For research purposes, 2100 images were manually gathered from Facebook, Instagram, Twitter, Google, etc. The CNN based system was used to classify each image and comment into bullying and not-bullying and achieved a weighted F1-score of 0.68. Hate speech almost similar context of cyberbullying was detected by [21]. Two types of approaches are used: unimodal and multimodal. In unimodal, they used InceptionV3 architecture with 2048 dimensional feature vector and then 150 dimension vector for both image text read from OCR. The multimodal dataset consists of 150,000 tweets with both image and text. Tweet text comes from LSTM architecture. The models were run by giving inputs like tweet text, image text, and image. The LSTM model with only text data achieved the F1 value is 0.703 and an accuracy value of 68.30%. On combined input features, i.e., tweet text, image text and images, the model achieved an F1-score of 0.701 and 68.2% of accuracy, similar to the LSTM model with text data only.

Chen et al. [27] proposed a text classification model based on CNN for the de facto verbal aggression dataset. They have manually added Tweets, and Facebook comments to the datasets while their emotions and stickers are not considered. Besides the hand labelled comments, they collected social network comment data from ‘sentiment140 corpus’. After the modification, polarities of the tweets are tagged as aggressive or unaggressive. They removed the usernames, which are followed by *at the rate*, *hash* topics with stickers, performed lowercasing during preprocessing. The tf-idf technique did feature extraction. The DL-based CNN model obtained the best results with an accuracy of 0.92 with an AUC value of 0.98.

Kumari et al. [22] automatically extracted features from text and images using DL techniques to classify the image

as cyber aggressive or not. They reduced the features using the binary particle swarm optimization (BPSO) algorithm. A multimodal dataset of 3600 images was manually created, comprising images and comments associated with the image. The images are mainly symbolic images classified into three categories: non-aggressive, medium-aggressive, high-aggressive. The model used here combines the VGG16 network with a 3-layered CNN and BPSO algorithm for optimization. The VGG16 network processes the images. The text features are embedded to BPSO for optimum features selection and then passed to the different classifiers to classify the images into predefined categories. The random forest classifier had the best F1-score of 0.74. Sadiq et al. [23] addressed the challenge of automatic identification of aggression on tweets of the cyber-troll dataset. They used CNN-LSTM and CNN-BiLSTM models. The dataset has 20,001 instances, out of which 7,822 are cyber-aggressive, and 12,179 are non-cyber aggressive instances. The dataset is first preprocessed for improving the result using NLTK. Their model of tf-idf with uni-gram and bi-gram outperformed by achieving an accuracy value of 0.92 and an F1-score value of 0.90.

The existing study on cyberbullying with image [3,21,22,24] is lacking very behind the text based cyberbullying detection [20,23–28] not just in terms of the accuracy and F1-score but also in terms of the number of research. The model developed for textual cyberbullying detection achieved 0.90 F1-score [27], also the accuracy value is greater than 90% [20,23,25–27]. Compared to textual cyberbullying, image-based cyberbullying detection received less attention. However, in the current time, the post is not limited to text but also posted in images and image-text mixed form. Hence, to ensure a cyberbully-free network, it is needed to capture the image-based bullied post soon published on the social platform. This study focused on developing an automated model with a deep transfer learning approach to detect image-based cyberbullying posts on social platforms to fill this research gap.

Methodology

Deep learning based Convolutional Neural Network (CNN) frameworks have shown their effectiveness and precision in various fields of image processing, including healthcare [15,17–19,29], social networks [11,12], agriculture [30–32], and others. We have also utilized the CNN-based models for cyberbullying detection of social platforms by following them. This section discusses the working of a two-dimensional CNN (2DCNN) for cyberbullying detection and also highlight the transfer learning models.

Two dimensional convolutional neural network

As shown in Fig. 1, a 2DCNN works in three phases: (i) extracting the features by convolution operation on input images, (ii) selecting the important features using pooling operation, (iii) pooled features are flattened and passed to a fully connected dense layer present at the end.

Convolution operation: Once the input data is padded, and stride value is defined, the convolution product between the input tensor and filter can be defined. The convolution is a sum of the element-wise product as shown in Fig. 2. Mathematically, an image will be represented in tensor form with the following dimensions (Eq. 1):

$$\dim(\text{image}) = (N_h, N_w, N_c) \quad (1)$$

where; N_h is height, N_w is width and N_c is the channels of the image. For a colourful image (RGB), the value of N_c is 3, which represents three channels—Red, Blue, and Green.

The filter K used for convolution operation is square and has odd size f_d and the same number of channels N_c as the input image. The filter will be applied to each channel to extract the image's pixel information. The dimension of the filter used for convolution operation is (Eq. 2):

$$\dim(\text{filter}) = (f_d, f_d, N_c) \quad (2)$$

The outcome of the convolution operation between the input image and filter K is a 2D matrix. Each value of the 2D matrix was calculated with element-wise multiplication and taking the sum (Fig. 2).

Mathematically, the convolution operation on an image I with a kernel K will be defined as (Eq. 3):

$$\text{conv}(I, K)_{x,y} = \sum_{i=1}^{N_h} \sum_{j=1}^{N_w} \sum_{k=1}^{N_c} K_{i,j,k} I_{x+i-1,y+j-1,k} \quad (3)$$

Mathematically, the output matrix dimension will be:

$$\dim(\text{conv}(I, K)) = \left(\left\lfloor \frac{n_h + 2p - f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_w + 2p - f}{s} + 1 \right\rfloor, n_c \right); s > 0$$

$$(n_h + 2p - f, n_w + 2p - f); s = 0$$

Here s is stride value which is fixed to 1, n_h and n_w are the height and width of the image, p is padding, f is the size of filter. Conclusively, If the input image size having the dimension $n * n$, and the filter size is $f * f$, padding the $p * p$ then the size of the matrix obtained after convolution

operation with image matrix and filter size will be $(n + 2p - f + 1) * (n + 2p - f + 1)$.

Pooling: The features extracted with the convolution operation in Image I and filter K are downsampled in this step. All extracted features might not have equal importance, and hence from each channel, the import features are pooled out. This operation only affects the dimensions of the feature matrix.

$$\dim(\text{pooling}(I)) = \left(\left\lfloor \frac{n_h + 2p - f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_w + 2p - f}{s} + 1 \right\rfloor, n_c \right); s > 0$$

$$(n_h + 2p - f, n_w + 2p - f, n_c); s = 0$$

In general, the CNN works as follows- first, it extracts the features using convolution operation and pooled the important features using the pooling layer. Then the pooled features pass to a fully connected layer at the end of the framework. Suppose the following notations are used to represent the different terminologies at a particular layer i th of 2DCNN. Input of the model is $a^{[i-1]}$ with height $n_h^{[i-1]}$, width $n_w^{[i-1]}$ and channels $n_c^{[i-1]}$, Padding is represented by $p^{[i]}$, the kernel will be moved with a stride size $s^{[i]}$. The filters (F) are used for convolution operation having the dimension of $n * n$. The bias of the network is represented as $b_n^{[i]}$, where n is the convolution number. The processed information pass through an activation function denoted by $\Psi^{[i]}$. The output of the convolution operation having the dimension of with height $n_h^{[i]}$, width $n_w^{[i]}$ and channels $n_c^{[i]}$. Then convolution operation will be represented as follows:

$$\forall n \in [1, 2, \dots, n_c^{[i]}]:$$

$$\text{conv}(a^{[i-1]}) \text{conv}(a^{[i-1]}, F^{(n)})_{xy}$$

$$= \Psi^{[i]} \left(\sum_{i=1}^{n_h^{[i-1]}} \sum_{j=1}^{n_w^{[i-1]}} \sum_{k=1}^{n_c^{[i-1]}} F_{i,j,k}^{(n)} a_{x+i-1,y+j-1,k}^{[i-1]} + b_n^{[i]} \right) \quad (4)$$

$$\dim(\text{conv}(a^{[i-1]}, F^{(n)})) = (n_h^{[i]}, n_w^{[i]})$$

Thus:

$$a^{[i]} = \left[\Psi^{[i]}(\text{conv}(a^{[i-1]})F^{(1)}), \Psi^{[i]}(\text{conv}(a^{[i-1]})F^{(2)}), \right.$$

$$\left. \Psi^{[i]}(\text{conv}(a^{[i-1]})F^{(3)}), \dots, \Psi^{[i]}(\text{conv}(a^{[i-1]})F^{(n_c^{[i]})}) \right]$$

$$\dim(a^{[i]}) = (n_h^{[i]}, n_w^{[i]}, n_c^{[i]})$$

Fig. 1 Convolutional neural network for image classification

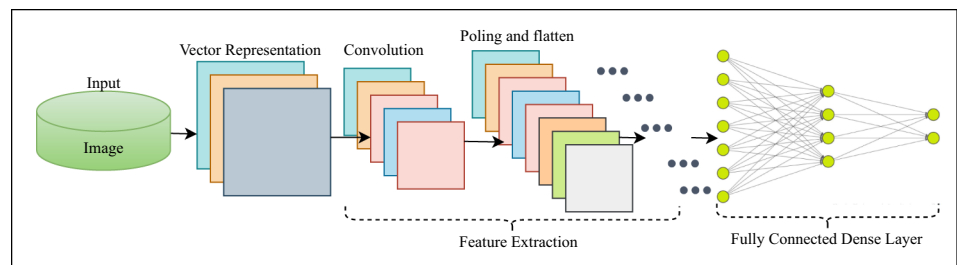
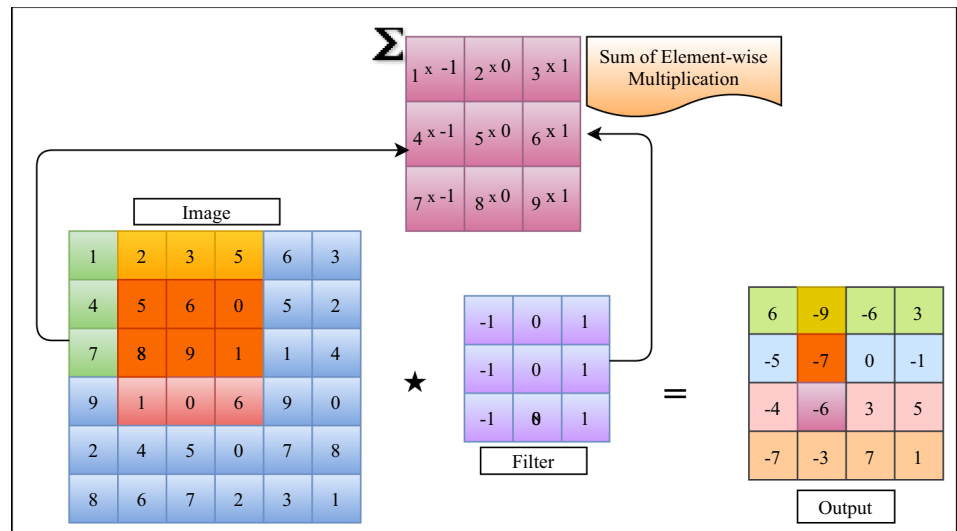


Fig. 2 Output matrix formation with convolution operation



This way, the parameters of the CNN are trained. The convolution operation with the pooling operation helps to detect the filters used and these filters help in identifying the class of the image.

Figure 3 shows the configuration of the convolution layer, and pooling makes up a block; adding more and more blocks increases the computation time and increases the number of features. Thus, with more blocks, more features will be extracted. This research uses three configurations: one block, two blocks and three blocks of 2DCNN models. The extracted features are flattened and pass to the dense layer present at the end. The internal layers of the network use the activation function as ReLU, whereas at the output layer, the softmax activation function is used. The compilation of the model is done with the cross-entropy loss function with two different optimizers: SGD and Adam. The dropout is also used to ensure there is no overfitting. The dropout refers to ignoring units (i.e. neurons) during the training phase of a certain set of neurons chosen at random.

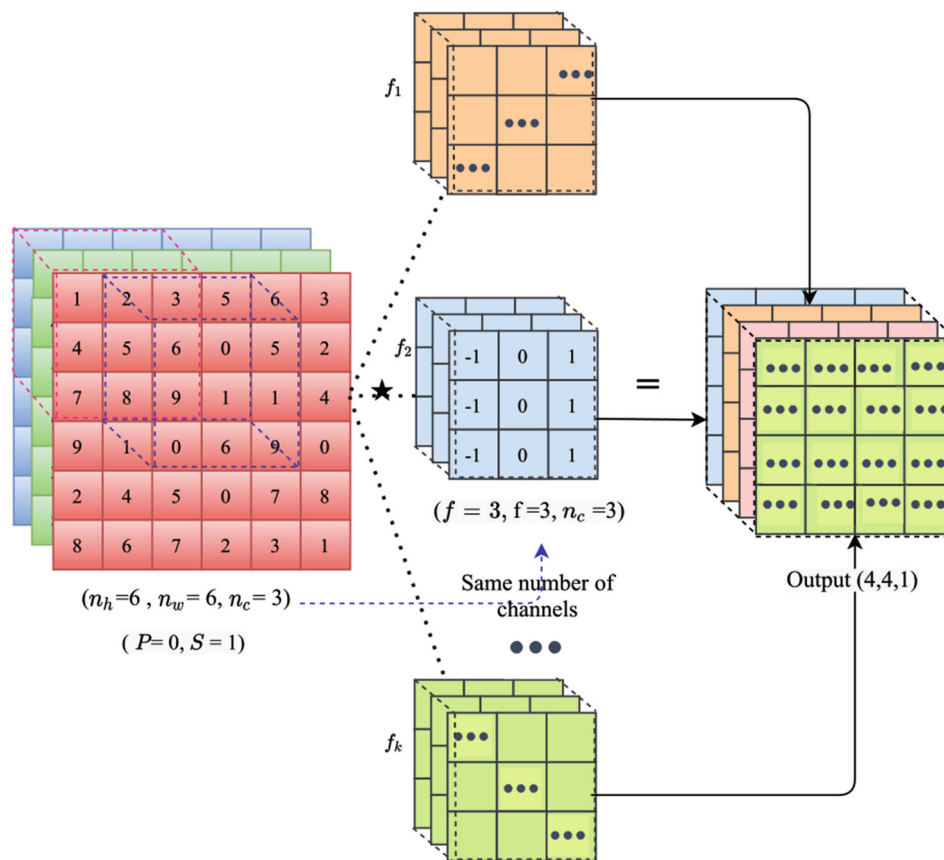
Transfer learning

Transfer learning models have the edge over existing deep learning architecture and are effectively used in multiple domains for the prediction task. Hence, this research also utilizes the benefit of pre-trained transfer learning mod-

els to predict image-based cyberbullying posts. Initially, different transfer learning models like VGG16 [33], Xception [34], VGG19 [33], InceptionResNetV2 [35], ResNet101 [36], InceptionV3 [37] and others available in Keras library² was applied to the selected dataset. Based on experimental outcomes of the different models, it was found VGG16 and InceptionV3 are performing better than the other models. The depth of the VGG16 model is 16, and it is just a simple stack of convolutional and max-pooling layers followed by one another and finally fully connected layers. It was one of the best performing architecture in the ILSVRC challenge in the same year. Hence, we have continued our research with VGG16 [33], and InceptionV3 [37] transfer learning approaches which are CNN based and are used widely for image recognition purposes. InceptionV3 is the successor of InceptionV1 and InceptionV2, which Google develops for ILSVRC. It is comparatively a very light model than the VGG16 and the runner up of image classification in ILSVRC in 2015. It has a depth of 48 and is much more complex than the VGG models, where the concept of inception is used rather than just stacking the convolution and max-pooling layers one after another.

² <https://keras.io/api/applications/>.

Fig. 3 Convolution operation with different kernel sizes



Data preparation

One of the major challenges of this work is data collection and annotations. The image data for cyberbullying is not directly available, and thus images were collected from many sources. The image data were acquired mainly from google images searches by searching for the related terms of cyberbullying. Still, as the images from google search belong to the websites they are hosted on, the sources of the image are all given due here. Some images were also taken from MMHS150K dataset [21]. The image downloaded was converted to .jpg format if they were in any other format to maintain uniformity. Further, with the help of three independent annotators, the dataset was labelled as bully and non-bully. The final dataset consists of 3000 images containing 1458 bullying and 1542 not bullying images. The developed dataset contains four columns, i.e., image name, description, bully or not bully, source. The image names were given in the number series as 1.jpg, 2.jpg,... so on as to which it is easier to acquire them in the models.

The data has been collected to make the model understand the normal case and the cyberbullying cases. For example, if obscene photos with human faces are present, the model may categorise the normal human face as bullying. To avoid this situation, a sufficient number of instances were added in

Table 1 Statistics of the dataset used for model development

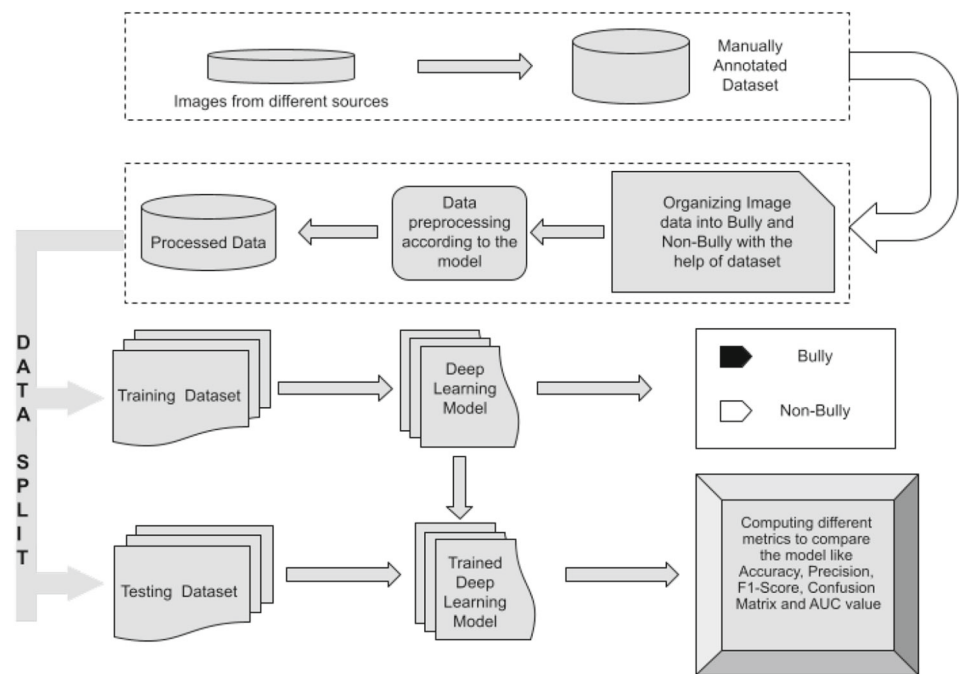
Number of sample	Bully	Not bully	Training	Test
1000	356	644	750	250
3000	1458	1542	2250	750

each dataset category. The statistics of the dataset are shown in Table 1.

Data preprocessing

Every image has a different resolution and colour scheme, so we converted images to the same target size of the models. There is a specified image input size for every DL model, and every image does not fit the input size. Thus, we need to pre-process the image before passing it to the model for training or testing purposes. Every model has its pre-processing requirements; therefore, we need different pre-processing for every model. The 2DCNN model does not have a specific input size for the image. However, transfer learning models VGG16 and InceptionV3 needed input in predefined sizes of $224 \times 224 \times 3$ and $299 \times 299 \times 3$, respectively. Hence, all images are reshaped accordingly. Further, the images are converted into three channels, i.e., RGB. Next, images are

Fig. 4 Proposed system design to detect bully posts



converted into array, and applied specific preprocessing for the model using keras³.

System design

Figure 4 shows the system design consisting of three phases: first, data collection, second is, data preprocessing, and the last phase is training and testing of the model. Thus finding the best model with their configuration, we have explained the first two phases in previous sections ‘Data Preparation’ and ‘Data Preprocessing’ briefly. In the third phase, we implemented CNN and two transfer learning models, VGG16 and InceptionV3 and compared them on different variations and configurations by altering the hyperparameter’s value. The outcomes of these models are discussed briefly in fourth section.

Every model is run in Google Colab using Keras and Python. Firstly, we run six different models all based on the CNN methodology using 1000 images dataset and the models are: 2DCNN, VGG16, VGG19 [33], InceptionV3, Inception-ResNetV2 [35] and Xception [34]. We select the best two transfer learning models (i) VGG16 and (ii) InceptionV3 out of the five transfer learning models based on the accuracy and complexity of the model. The other transfer learning models yielded similar accuracy. However, they needed higher resources for execution. Hence, to save the resources and execute the program in less time, the VGG16 and InceptionV3 models are selected for further experiment. Next, the 2DCNN, VGG16 and InceptionV3 models are re-executed

with 3000 samples. For every 2DCNN model, the following variations have experimented:

- Optimizers: SGD with a learning rate of 0.001 and 0.01; Adam with a learning rate of 0.001 and 0.01.
- Dropout layer: There are two variations, one without any dropout and one with 0.2 dropouts.

For VGG16 and InceptionV3 transfer learning model, the following variations have experimented:

- Optimizers: SGD with a learning rate of 0.001 and 0.01; Adam with a learning rate of 0.001 and 0.01.
- Dropout layer: There are three variations, one without any dropout and one with 0.2 and the other with 0.50 dropout.

For every model, the results contain every major variable to analyse the model. The similarities in all the models are that the output layers contain a softmax activation function and two output neurons, which gives the probability of the image class with cross-entropy loss. The weights of pre-trained transfer learning are freezed by making all layers untrainable because it was already trained with a huge corpus. Every model has been run for 50 epochs with an early stopping setting. If the accuracy is not improving continuously for a fixed number of epochs (we used patience value as 10), training will stop, and the corresponding weights are stored.

³ Keras.applications.inception_v3.preprocess_input.

Results

This section discusses the experimental outcomes of different DL and transfer learning models. The total dataset size is 3000, where 1542 samples belong to the non-bully (Class 0) category, and the remaining 1458 samples are of the bully (Class 1). However, the model has experimented with different training and testing sample sizes. The statistics of the dataset used for training-testing is shown in Table 1. The proposed system is a supervised model; hence the model's performance is evaluated with precision, recall, F1-score, confusion matrix, area under ROC curve and an accuracy [38]. Precision measures the number of correctly classified bullying images among all images classified as bullying, while recall measures the number of bullying images among all bullying images in the dataset. The F1-score is the harmonic mean of precision and recall. The confusion matrix is the 2×2 matrix representation; in which the values are arranged as [[TP, FN], [FP, TN]]. Here, TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative. Cyberbullying images that are correctly classified are called TP; that are misclassified are called FN. Non cyberbullying images which are correctly classified are called TN, while misclassified images are called FP. Accuracy is the sum of correctly classified bullying and non-bullying images. AUC is the area under the ROC curve, and it ranges from 0 to 1. AUC is the probability that the model ranks a random positive example more highly than a random negative example.

The experiments were started with a DL-based convolutional neural network (CNN) having a single layer of convolution. Further, the number of convolution layers was increased. Also, the parameter value such as dropout and learning rate is tuned. With these settings, the performance of the model was evaluated on two different optimizers, namely: SGD and Adam. The experimented outcomes of the CNN models are shown in Table 2. The first row of the table shows the outcomes of the CNN model with a convolution layer and having an SGD optimizer with a learning rate of 0.001. The dropout layer is not used in this case. The model achieved 0.64, 1.00, and 0.78 P, R, and F1 values for class 0 (Non-bully), whereas for bully class, all metrics values are 0. The accuracy of the model is 0.64. The confusion matrix [[TP, FN], [FP, TN]] having the value of [[160, 0], [90, 0]] means true positive value is 160 and false-positive value is 90. Here, all class 1 (TN) instance is misclassified and predicted as class 0 by the model. The same model experimented with a dropout layer having the value of 0.2. Still, the outcomes of class 1 remain unchanged, and all instances are misclassified to class 0 by the model (2nd row of Table 1). Next, the same settings have experimented with Adam optimizer. This model achieved better performance with a TP value of 104, TN value of 63, and remaining test sam-

ples misclassified as FN is 56 and FP is 27. Even though the model's performance improved, the model's accuracy is 0.67, indicating that many test samples are misclassified to other classes. Hence, the convolution layer increased to (i) 2 and (ii) 3 and re-experimented the models by fixing the other parameter values that remain the same as the previous model. The outcomes showed that none of the experimented models achieved better performance than a model with a convolution layer and used the learning rate of 0.001 without dropout.

With the CNN model, the prediction of the bully content from the social post is limited to 67% accuracy. Hence, transfer learning-based models (i) VGG16 and (ii) InceptionV3 model have been experimented. The outcomes of the transfer learning models are shown in Table 3. Firstly, the VGG16 model experimented without the dropout layer-the SGD optimizer was used with a learning rate of 0.001. The model achieved better performance than the 2DCNN model with an accuracy value of 0.77. The FN value was 24, and the FP value was 33, indicating that the model misclassified many test samples. Hence, the experiments were repeated with different values of the dropout layer, such as 0.2, 0.3, and 0.5. The optimizer and their learning rate value remain the same as the 2DCNN layer. The best performing VGG16 model achieved with dropout value of 0.5, SGD optimizer with learning rate 0.001 where the accuracy value was 0.81. The FN value was 21, and the FP value was 26 is minimum compared to the previously experimented models.

Next, another transfer learning approach, i.e., InceptionV3, was experimented with similar settings of VGG16. The outcomes of the model are shown in Table 3. In the beginning, the model was experimented without a dropout layer and then repeated with the dropout layer having the value of 0.2 and 0.5. The InceptionV3 model outperformed the VGG16 model in most cases. The best performance of the InceptionV3 model was achieved with the Adam optimizer having a learning rate of 0.001. The FN value was 10, and the FP value was 33. The model's accuracy was 0.83, and the AUC value was 0.79, indicating the best performance and lower misclassification rate than experimented models. It gives almost the same or sometimes even better results than VGG16 with the extra benefit of the size. Also, the InceptionV3 model is very lightweight; it takes on 92MB, whereas VGG16 takes 528 MB.⁴ Thus InceptionV3 is a better choice than the VGG16.

Even though transfer learning approach is used the model accuracy is limited to 83% which is achieved by InceptionV3 model as shown in Table 3. Earlier research confirmed that, the DL framework needed more data to train themselves. However, this research uses 1000 samples only; among them 75% of samples are used for training and remaining 25% samples are used to test the performance. The accuracy of the

⁴ <https://keras.io/api/applications>.

Table 2 Results obtained on different settings with CNN model

CNN layers	Optimizer	Learning rate	Dropout	Class 0			Class 1			Accuracy	Confusion Matrix [[TP, FN], [FP, TN]]	AUC
				P	R	F1	P	R	F1			
1	SGD	0.001	NA	0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	0.99	0.78	0.00	0.00	0.00	0.54	[[159 1], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.79	0.65	0.71	0.53	0.70	0.60	0.67	[[104 56], [27 63]]	0.67
	ADAM	0.001	0.2	0.67	0.98	0.79	0.79	0.12	0.21	0.67	[[157 3], [79 11]]	0.55
				0.68	0.84	0.75	0.52	0.30	0.38	0.65	[[135 25], [63 27]]	0.57
				0.72	0.81	0.76	0.56	0.43	0.49	0.67	[[129 31], [51 39]]	0.62
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
2	SGD	0.001	NA	0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
	ADAM	0.001	0.2	0.71	0.71	0.71	0.48	0.48	0.48	0.63	[[114 46], [47 43]]	0.60
				0.66	0.89	0.76	0.50	0.19	0.27	0.64	[[143 17], [73 17]]	0.54
				0.64	0.99	0.78	0.60	0.03	0.06	0.64	[[158 2], [87 3]]	0.51
				0.64	0.98	0.78	0.50	0.03	0.06	0.64	[[157 3], [87 3]]	0.51
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
3	SGD	0.001	NA	0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
	ADAM	0.001	0.2	0.70	0.71	0.70	0.47	0.46	0.46	0.62	[[113 47], [49 41]]	0.58
				0.63	0.93	0.75	0.15	0.02	0.04	0.60	[[149 11], [88 2]]	0.48
				0.65	0.95	0.77	0.50	0.09	0.15	0.64	[[152 8], [82 8]]	0.52
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50
				0.64	1.00	0.78	0.00	0.00	0.00	0.64	[[160 0], [90 0]]	0.50

Table 3 Results with VGG16 and InceptionV3 transfer Learning model on 1000 samples

Dropout	Optimizer	Learning rate	Class 0			Class 1			Accuracy	Confusion Matrix [[TP, FN], [FP, TN]]	AUC
			P	R	F1	P	R	F1			
VGG16											
NA	SGD	0.001	0.80	0.85	0.83	0.70	0.63	0.67	0.77	[[136 24], [33 57]]	0.74
		0.01	0.81	0.74	0.77	0.60	0.69	0.64	0.72	[[118 42], [28 62]]	0.71
0.2	ADAM	0.001	0.80	0.92	0.85	0.80	0.59	0.68	0.80	[[147 13], [37 53]]	0.75
		0.01	0.79	0.89	0.84	0.74	0.58	0.65	0.78	[[142 18], [38 52]]	0.73
0.5	SGD	0.001	0.82	0.89	0.85	0.76	0.64	0.70	0.80	[[142 18], [32 58]]	0.77
		0.01	0.75	0.93	0.83	0.77	0.44	0.56	0.75	[[148 12], [50 40]]	0.69
0.5	ADAM	0.001	0.81	0.91	0.86	0.80	0.61	0.69	0.80	[[146 14], [35 55]]	0.76
		0.01	0.77	0.95	0.85	0.85	0.49	0.62	0.78	[[152 8], [46 44]]	0.72
0.5	SGD	0.001	0.84	0.87	0.86	0.75	0.71	0.73	0.81	[[139 21], [26 64]]	0.79
		0.01	0.71	0.97	0.82	0.84	0.30	0.44	0.73	[[155 5], [63 27]]	0.63
0.5	ADAM	0.001	0.78	0.93	0.85	0.82	0.54	0.65	0.79	[[149 11], [41 49]]	0.74
		0.01	0.81	0.92	0.86	0.81	0.61	0.70	0.81	[[147 13], [35 55]]	0.77
InceptionV3											
NA	SGD	0.001	0.80	0.94	0.86	0.84	0.59	0.69	0.81	[[150 10], [37 53]]	0.76
		0.01	0.82	0.90	0.86	0.78	0.64	0.71	0.81	[[144 16], [32 58]]	0.77
0.2	ADAM	0.001	0.82	0.94	0.87	0.85	0.63	0.73	0.83	[[150 10], [33 57]]	0.79
		0.01	0.80	0.88	0.84	0.74	0.62	0.67	0.78	[[140 20], [34 56]]	0.75
0.5	SGD	0.001	0.80	0.95	0.87	0.87	0.59	0.70	0.82	[[152 8], [37 53]]	0.77
		0.01	0.80	0.93	0.86	0.81	0.58	0.68	0.80	[[148 12], [38 52]]	0.75
0.5	ADAM	0.001	0.81	0.93	0.87	0.82	0.62	0.71	0.82	[[148 12], [34 56]]	0.77
		0.01	0.79	0.94	0.86	0.85	0.57	0.68	0.81	[[151 9], [39 51]]	0.76
0.5	SGD	0.001	0.80	0.94	0.86	0.84	0.59	0.69	0.81	[[150 10], [37 53]]	0.76
		0.01	0.81	0.92	0.86	0.81	0.61	0.70	0.81	[[147 13], [35 55]]	0.77
0.5	ADAM	0.001	0.81	0.94	0.87	0.85	0.61	0.71	0.82	[[150 10], [35 55]]	0.77
		0.01	0.75	0.94	0.83	0.81	0.43	0.57	0.76	[[151 9], [51 39]]	0.69

model may improve by increasing the total samples. Hence, all models are re-experimented with increased dataset where total number of samples are 3000, with same train-test split ratio. Means, 75% of the total sample used for training purpose, whereas 25% samples used to test the performance of the trained model.

First, the 2DCNN model are re-experimented with same settings and then transfer learning models are experimented. The outcome of the 2DCNN model is presented in Tables 4 and 5 consists the outcomes of transfer learning models. The CNN model with a single convolution layer achieved the accuracy and AUC value of 0.51. Also, the recall value of class 1 is 0.07. The prediction accuracy by the model remains the same. After applying the dropout layer also did not upgrade the performance. The best outcomes of 2DCNN models with 3000 samples are as follows: the precision, recall and F1-score for the bully class is 0.66, 0.72, and 0.69, whereas, for non-bully, it is 0.69, 0.63 and 0.66, respectively. The AUC value of the model is 0.673, and the accuracy is 67%. The AUC value is increased from 0.62 to 0.673, which indicates the improvement of prediction accuracy on larger samples.

Next, the VGG16 transfer learning model received higher performance with increased data samples. The best results of VGG16 for a total of 3000 sample is obtained with a dropout value of 0.5. The optimizer is SGD with a learning rate of 0.001. The F1-score for the non-bully class is 0.87, and for the bully class, the F1-score is 0.86. The accuracy of the model is 86%, and the AUC value is 0.864, which improved compared to the previous model, where the AUC value is 0.79 only. Following the pattern of VGG16, InceptionV3 also yielded better performance with increased samples. The InceptionV3 model trained and tested with 3000 images dataset has the best results with 0.89 F1-score of bully class. The model's accuracy is 89%, and the AUC value is 0.888. The experimental outcomes of 2DCNN and transfer learning models confirmed that if the training samples increase, the model performance will also increase. We have created another train-test sample with a 90:10 ratio to verify this hypothesis. In this case, 90% of samples are used for training, and the remaining 10% samples are used to test the model performance. The outcomes of the models with 90:10 train-test split is shown in Tables 6 and 7.

We expected the results of the 2DCNN would improve with more training data, but it did not happen. The model's best F1-score value is 0.68, the AUC value reduced to 0.696 and accuracy reducing to 0.70 (Table 6). Hence, by following the outcomes of the 2DCNN model concerning different settings of the training samples (Tables 2, 4, 6), it can be said that the 2DCNN model is unable to capture the patterns of the dataset properly.

The VGG16 model with a 90:10 data split has almost the same result as for 75:25 data split. The model's performance

is not increased compared to the 75:25 data split. A similar pattern is identified with another transfer learning model, InceptionV3. With 90:10 data split, InceptionV3 has almost the same result as for 75:25 data split (Table 7). Interestingly, on the increased training dataset, i.e., 90:10 train-test split, the 0.50 dropout has outperformed the 0.2 dropouts, unlike the case in the 75:25 data split case.

We have compared the outcomes of our proposed model with similar works and shown in Table 8. Limited works have been found in literature that consider image-based cyberbullying prediction [3,21,22,24]. Kumari et al. [3,22] proposed two different models, in [3], they used CNN based model whereas in [22] traditional ML model was used. Their model achieved an F1-score of 0.68, and 0.74 using the CNN and RF model in [3], and [22], respectively. The model proposed by [21] was used InceptionV3 and LSTM model and achieved an F1-score value of 0.67. Hossainmardi et al. [24] used traditional ML model and achieved 0.84 F1-score. On the other hand, the proposed system experimented with the DL-based 2DCNN model and transfer learning based on VGG16 and InceptionV3. As shown in Table 8, the 2DCNN model achieved 0.65 F1-score value whereas VGG16 and InceptionV3 achieved 0.86 and 0.87 F1-score value. The outcomes of the proposed transfer learning model with the tuned hyperparameters settings outperformed the existing research.

Discussion

One of the main findings of this research is the requirement of a high amount of annotated data for modelling. If the number of training samples is low, then the DL-based models cannot train properly, and hence, the models fail when unseen data supply for testing. Tables 2 and 4 shows the performance of the CNN model with the different number of convolution layer and hyperparameter values on 250 and 750 samples. The model's outcomes trained with 2250 samples are better compared to the model trained on 750 samples. Another finding of this research is a suitable optimizer to handle the images. Two optimizers with learning rates 0.01 and 0.001 were used in this research and found Adam optimizer is a better option for the 2DCNN model. The outcomes with different training sample sizes on 2DCNN and transfer learning models InceptionV3 and VGG16 confirmed that 2DCNN required more samples for training. The dropout values do not affect more in the predictions.

The experimented transfer learning model performances are comparable in different settings. The outcomes of the VGG16 and InceptionV3 models are shown in Tables 5 and 7. The best outcomes were achieved with transfer learning models trained with 90% of the sample. The performance of the transfer learning model has minimum variation; however, the 2DCNN model's performance changes with a large

Table 4 Results of CNN models on different settings with 3000 samples

CNN layer	Optimizer	Learning rate	Dropout	Class 0			Class 1			Accuracy	Confusion Matrix [[TP, FN], [FP, TN]]	AUC
				P	R	F1	P	R	F1			
1	SGD	0.001	NA	0.50	0.95	0.66	0.58	0.07	0.12	0.51	[[354 18] [353 25]]	0.51
				0.51	0.97	0.66	0.68	0.07	0.13	0.52	[[359 13] [350 28]]	0.52
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.65	0.76	0.70	0.71	0.60	0.65	0.68	[[281 91] [152 226]]	0.677
	ADAM	0.001	0.2	0.69	0.63	0.66	0.66	0.72	0.69	0.67	[[233 139] [106 272]]	0.673
				0.65	0.56	0.60	0.62	0.70	0.66	0.63	[[207 165] [113 265]]	0.629
				0.63	0.68	0.65	0.66	0.61	0.63	0.64	[[253 119] [149 229]]	0.643
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
2	SGD	0.001	0.2	0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
	ADAM	0.001	0.2	0.61	0.63	0.62	0.62	0.61	0.61	0.62	[[234 138] [149 229]]	0.617
				0.60	0.73	0.66	0.66	0.52	0.58	0.62	[[271 101] [183 195]]	0.622
				0.2	0.01	0.02	0.50	0.97	0.66	0.49	[[3 369] [12 366]]	0.488
				0.50	0.98	0.66	0.67	0.04	0.07	0.51	[[365 7] [364 14]]	0.509
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
3	SGD	0.001	0.2	0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.58	0.57	0.58	0.58	0.59	0.59	0.58	[[213 159] [154 224]]	0.583
	ADAM	0.001	0.2	0.66	0.59	0.62	0.63	0.70	0.67	0.65	[[219 153] [112 266]]	0.646
				0.58	0.72	0.64	0.64	0.50	0.56	0.61	[[266 106] [190 188]]	0.606
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50
				0.50	1.00	0.66	0.00	0.00	0.00	0.50	[[372 0] [378 0]]	0.50

Table 5 Performance with VGG16 and InceptionV3 transfer learning with 3000 samples

Dropout	Optimizer	Learning rate	Class 0			Class 1			Accuracy	Confusion Matrix [[TP, FN], [FP, TN]]	AUC		
			P		R		P					R	
			P	R	F1	F1	P	R				F1	F1
VGG16													
NA	SGD	0.001	0.83	0.90	0.86	0.89	0.82	0.85	0.86	[[333 39] [67 311]]	0.859		
		0.01	0.78	0.90	0.84	0.89	0.75	0.81	0.83	[[336 36] [94 284]]	0.827		
	ADAM	0.001	0.82	0.88	0.85	0.87	0.81	0.84	0.85	[[327 45] [70 308]]	0.847		
		0.01	0.83	0.76	0.8	0.78	0.85	0.81	0.81	[[284 88] [58 320]]	0.805		
0.2	SGD	0.001	0.84	0.88	0.86	0.87	0.83	0.85	0.85	[[326 46] [64 314]]	0.854		
		0.01	0.75	0.95	0.84	0.93	0.70	0.8	0.82	[[353 19] [115 263]]	0.822		
	ADAM	0.001	0.86	0.8	0.83	0.81	0.87	0.84	0.83	[[297 75] [50 328]]	0.833		
		0.01	0.67	0.97	0.79	0.95	0.52	0.68	0.75	[[362 10] [180 198]]	0.749		
0.5	SGD	0.001	0.84	0.89	0.87	0.89	0.84	0.86	0.86	[[331 41] [61 317]]	0.864		
		0.01	0.62	0.98	0.76	0.95	0.4	0.57	0.69	[[364 8] [225 153]]	0.692		
	ADAM	0.001	0.82	0.90	0.86	0.89	0.8	0.85	0.85	[[336 36] [75 303]]	0.852		
		0.01	0.78	0.82	0.8	0.82	0.77	0.79	0.8	[[306 66] [86 292]]	0.797		
InceptionV3													
NA	SGD	0.001	0.87	0.89	0.88	0.89	0.87	0.88	0.88	[[332 40] [50 328]]	0.88		
		0.01	0.86	0.91	0.88	0.90	0.85	0.88	0.88	[[337 35] [55 323]]	0.88		
	ADAM	0.001	0.85	0.88	0.86	0.88	0.85	0.86	0.86	[[326 46] [56 322]]	0.864		
		0.01	0.86	0.89	0.88	0.89	0.86	0.87	0.87	[[330 42] [52 326]]	0.875		
0.2	SGD	0.001	0.87	0.91	0.89	0.91	0.86	0.89	0.89	[[340 32] [52 326]]	0.888		
		0.01	0.88	0.87	0.87	0.87	0.88	0.88	0.88	[[324 48] [45 333]]	0.876		
	ADAM	0.001	0.89	0.82	0.85	0.83	0.90	0.86	0.86	[[304 68] [39 339]]	0.857		
		0.01	0.83	0.92	0.87	0.92	0.81	0.86	0.87	[[344 28] [72 306]]	0.867		
0.5	SGD	0.001	0.86	0.91	0.89	0.91	0.85	0.88	0.88	[[339 33] [55 323]]	0.883		
		0.01	0.83	0.93	0.88	0.92	0.81	0.86	0.87	[[347 25] [73 305]]	0.87		
	ADAM	0.001	0.85	0.90	0.87	0.89	0.85	0.87	0.87	[[334 38] [58 320]]	0.872		
		0.01	0.79	0.90	0.84	0.88	0.77	0.82	0.83	[[333 39] [87 291]]	0.833		

Table 6 Results of CNN models with 90:10 train-test split

CNN layer	Optimizer	Learning rate	Dropout	Class 0			Class 1			Accuracy	Confusion Matrix [[TP, FN], [FP, TN]]	AUC
				P	R	F1	P	R	F1			
1	SGD	0.001	NA	0.53	0.95	0.68	0.65	0.09	0.16	0.54	[[148 7] [132 13]]	0.522
				0.53	0.94	0.68	0.62	0.11	0.19	0.54	[[145 10] [129 16]]	0.523
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.67	0.81	0.74	0.74	0.58	0.65	0.70	[[126 29] [61 84]]	0.696
	ADAM	0.001	0.2	0.64	0.77	0.70	0.69	0.54	0.60	0.66	[[120 35] [67 78]]	0.656
				0.64	0.47	0.54	0.56	0.72	0.63	0.59	[[73 82] [41 104]]	0.594
				0.70	0.72	0.71	0.69	0.68	0.68	0.70	[[111 44] [47 98]]	0.696
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
2	SGD	0.001	NA	0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
	ADAM	0.001	0.2	0.67	0.69	0.68	0.66	0.63	0.65	0.66	[[107 48] [53 92]]	0.662
				0.62	0.71	0.66	0.64	0.54	0.59	0.63	[[110 45] [66 79]]	0.627
				0.53	0.98	0.69	0.80	0.08	0.15	0.55	[[152 3] [133 12]]	0.532
				0.52	0.99	0.68	0.75	0.02	0.04	0.52	[[154 1] [142 3]]	0.507
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
3	SGD	0.001	0.2	0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
				0.52	1.00	0.68	0.00	0.00	0.00	0.52	[[155 0] [145 0]]	0.50
	ADAM	0.001	0.2	0.61	0.65	0.63	0.60	0.56	0.58	0.60	[[100 55] [64 81]]	0.602
				0.55	0.92	0.69	0.68	0.18	0.28	0.56	[[143 12] [119 26]]	0.551
				0.55	0.94	0.69	0.72	0.18	0.29	0.57	[[145 10] [119 26]]	0.557
				0.58	0.56	0.57	0.54	0.56	0.55	0.56	[[87 68] [64 81]]	0.56
				0.58	0.56	0.57	0.54	0.56	0.55	0.56	[[87 68] [64 81]]	0.56

Table 7 Results of transfer learning approach with 90:10 train-test split

Dropout	Optimizer	Learning rate	Class 0			Class 1			Accuracy	Confusion Matrix [[TP, FN], [FP, TN]]	AUC
			P	R	F1	P	R	F1			
VGG16											
NA	SGD	0.001	0.85	0.86	0.85	0.85	0.83	0.84	0.85	[[133 22] [24 121]]	0.846
		0.01	0.81	0.90	0.85	0.88	0.78	0.82	0.84	[[139 16] [32 113]]	0.838
		0.001	0.84	0.87	0.86	0.86	0.83	0.84	0.85	[[135 20] [25 120]]	0.849
0.2	SGD	0.01	0.88	0.80	0.84	0.81	0.88	0.84	0.84	[[124 31] [17 128]]	0.841
		0.001	0.85	0.85	0.85	0.84	0.83	0.84	0.84	[[132 23] [24 121]]	0.843
		0.01	0.77	0.91	0.83	0.88	0.71	0.79	0.81	[[141 14] [42 103]]	0.81
0.5	ADAM	0.001	0.85	0.85	0.85	0.84	0.84	0.84	0.85	[[132 23] [23 122]]	0.847
		0.01	0.92	0.60	0.73	0.69	0.94	0.80	0.77	[[93 62] [8 137]]	0.772
		0.001	0.86	0.85	0.85	0.84	0.85	0.85	0.85	[[132 23] [22 123]]	0.85
0.5	ADAM	0.01	0.92	0.50	0.65	0.64	0.95	0.77	0.72	[[78 77] [7 138]]	0.728
		0.001	0.85	0.88	0.86	0.86	0.83	0.85	0.86	[[136 19] [24 121]]	0.856
		0.01	0.78	0.94	0.85	0.91	0.72	0.80	0.83	[[145 10] [41 104]]	0.826
InceptionV3											
NA	SGD	0.001	0.85	0.88	0.86	0.87	0.83	0.85	0.86	[[137 18] [25 120]]	0.856
		0.01	0.87	0.85	0.86	0.85	0.87	0.86	0.86	[[132 23] [19 126]]	0.86
		0.001	0.82	0.87	0.85	0.85	0.80	0.83	0.84	[[135 20] [29 116]]	0.836
0.2	SGD	0.01	0.92	0.74	0.82	0.77	0.93	0.84	0.83	[[114 41] [10 135]]	0.833
		0.001	0.88	0.86	0.87	0.86	0.88	0.87	0.87	[[134 21] [18 127]]	0.87
		0.01	0.82	0.97	0.89	0.96	0.78	0.86	0.88	[[150 5] [32 113]]	0.874
0.5	ADAM	0.001	0.88	0.80	0.84	0.81	0.88	0.84	0.84	[[124 31] [17 128]]	0.841
		0.01	0.86	0.88	0.87	0.87	0.84	0.86	0.86	[[137 18] [23 122]]	0.863
		0.001	0.88	0.88	0.88	0.87	0.88	0.87	0.88	[[136 19] [18 127]]	0.877
0.5	ADAM	0.01	0.88	0.88	0.88	0.87	0.88	0.87	0.88	[[136 19] [18 127]]	0.877
		0.001	0.87	0.85	0.86	0.85	0.87	0.86	0.86	[[132 23] [19 126]]	0.86
		0.01	0.78	0.96	0.86	0.94	0.71	0.81	0.84	[[149 6] [42 103]]	0.836

Table 8 Performance comparison of the proposed models with existing research

Source	Model	Precision	Recall	F1-score
Kumari et al [3]	2DCNN	0.68	0.68	0.68
Gomez et al. [21]	InceptionV3 and LSTM	–	–	0.67
Kumari et al [22]	Random Forest	0.74	0.75	0.74
Hosseinmardi et al. [24]		0.85	0.83	0.84
Proposed	2DCNN	0.74	0.58	0.65
	VGG16	0.85	0.85	0.86
	InceptionV3	0.87	0.88	0.87

margin when the values of the hyperparameters are tuned. VGG16 results have higher variance as compared with InceptionV3 with the change in hyperparameters. The hypothesis is that the more the training data, the better results do not fit for transfer learning models as the 75:25 data split gives better results than the 90:10. The hypothesis is that the more the data, the better results apply to every model, whether transfer learning or simple 2DCNN model, as an increase in the dataset from 1000 images to 3000 images has shown significant improvement in results.

Theoretically, 2DCNN model do not have any inherent reason to show the variance in the result with the change in optimizers only. But, this type of variance in result makes it important to look forward to the hyperparameters selection. Transfer learning is beneficial when solving complex problems like cyberbullying as it has varied subproblems. These models are trained on a large corpus having many classes. Hence utilizing the learning capabilities to handle the complex problem is easy with transfer learning. On the other hand, a pure CNN model is trained with training samples provided by the users. Hence, their knowledge base is limited to the supplied dataset. If any test sample falls out of the scope of the training sample, then tough for the model to predict their actual class. This may be one reason behind gets better prediction accuracy with transfer learning models compared to the CNN mode. The epoch wise loss is plotted and shown in Fig. 5. The loss value of the 2DCNN model is very high as compared to VGG16 and InceptionV3 in all settings. It means the CNN model needed more epochs to train, and then the loss value may be decreased. On the other hand, the transfer learning models are pre-trained, and hence the loss values are very low at the beginning itself.

Cyberbullying is a major issue that has existed on social platforms to date. Many people, especially teenagers, are affected by this. The textual cyberbullying events detection mechanism suggested by the researchers. However, the proposed model is designed to detect cyberbullying posts having images. The model can be used as an initial scanner of the social post. If any posts are predicted as cyberbully, they will be migrated or generate a notification to the sender and receiver to check and report it. This mechanism can help to reduce the number of bullying posts from social platforms

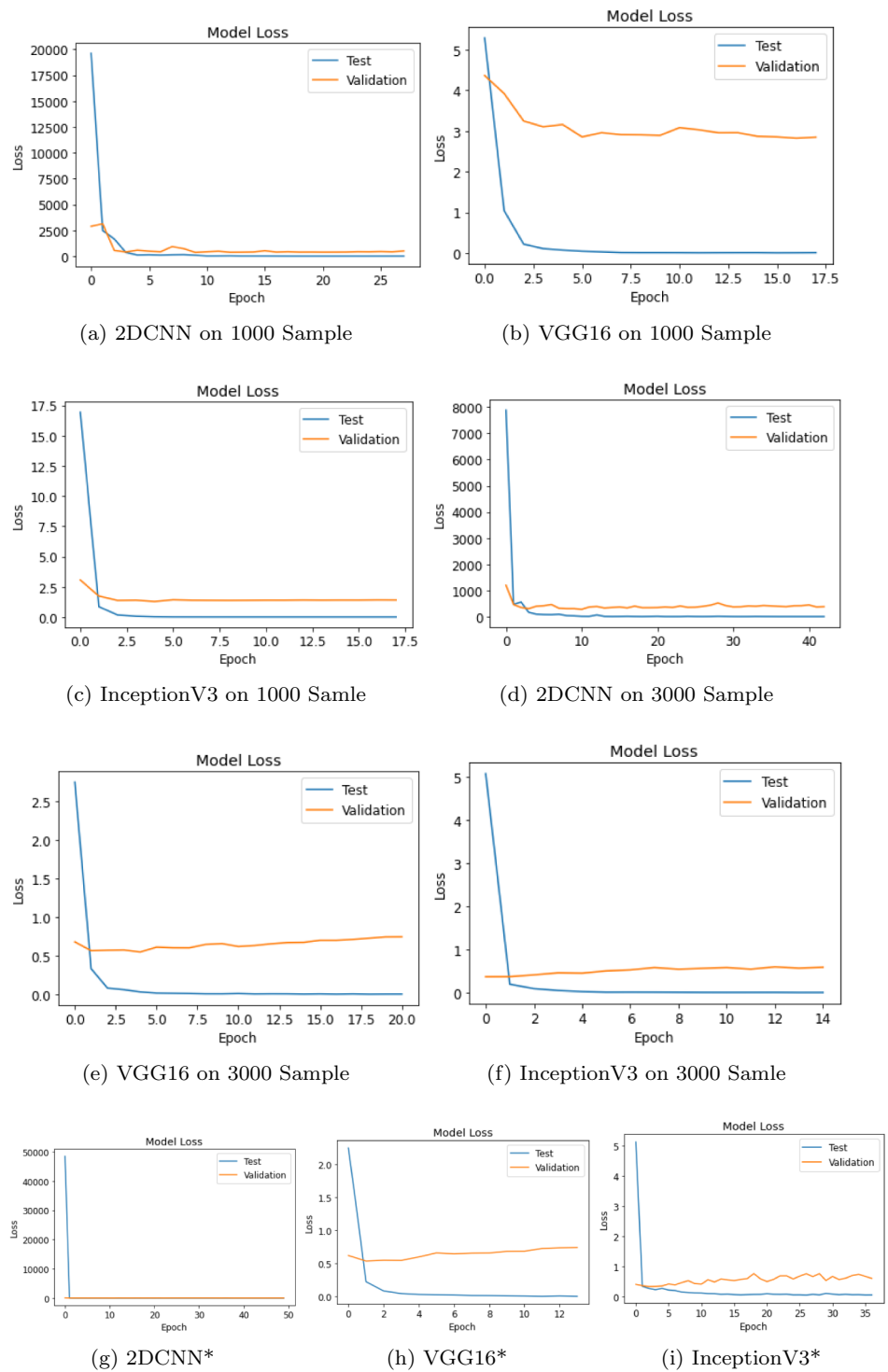
and consequently reduce the incidents happening due to this. Building an automated model to detect image-based cyberbullying is a complex task and hence requires a large number of labelled data for training. Hence, the 2DCNN model is not a better choice; instead, the pre-trained transfer learning models like VGG16 and InceptionV3 performed better and hence can be preferred. These models are available in the Keras library and can be tuned as per the requirement by the researchers.

Conclusion, limitations and future scope

Complex problems like cyberbullying, which have various problems embedded in, are difficult to trace with the normal system. Especially, image-based social cyberbullying post-detection is a challenging task. This research explored deep learning and transfer learning frameworks to find the best-suited model to predict image-based cyberbullying posts on social platforms. The deep learning-based 2DCNN has initially experimented and, by tuning their hyperparameters, achieved the accuracy value of 69.60%. On the other hand, the transfer learning models VGG16 and InceptionV3 always achieved better prediction accuracy. The VGG16 achieved an accuracy value of 86% whereas, InceptionV3 achieved 89% accuracy. Hence, the transfer learning models VGG16 and InceptionV3 have an accuracy margin of 16.40 and 19.40%, respectively, compared to the best configured 2DCNN model. Therefore, it can be concluded that the proposed system detects most of the image-based cyberbullying posts.

The limitations of the proposed model include the following: (i) it is not considered textual cyberbullying detection, which means a post having only text is not a part of this research, (ii) combining the image with text has been found in cyberbullying posts. However, this study is limited to image-oriented cyberbullying detection. Hence, the future scope of this research is always open to discussion as it has varied subproblems. The accuracy achieved by the proposed system was 89%, which can be improved by increasing the training sample size. Also, the other combinations of the models can opt, and an ensemble system will form to achieve better prediction accuracy. The textual part can be considered along

Fig. 5 Loss Vs Epoch on different settings; **a–c** represents the loss Vs epoch observation with a total of 1000 samples which was splitted 75:25 ratio for training and test. **d–f** represent the loss Vs epoch observation with a total of 3000 samples which was splitted 75:25 ratio for training and test. **g–i** represents the loss Vs epoch observation with a total of 3000 samples which was split 90:10 ratio for training and test



with the image to catch more cyberbullying related posts on social platforms.

Declarations

Conflict of interest There is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Smith PK, Mahdavi J, Carvalho M, Fisher S, Russell S, Tippett N (2008) Cyberbullying: its nature and impact in secondary school pupils. *J Child Psychol Psychiatry* 49(4):376–385
- Ak Şerife, Özdemir Y, Kuzucu Y (2015) Cybervictimization and cyberbullying: the mediating role of anger, don't anger me! *Comput Human Behav* 49:437–443
- Kumari K, Singh JP, Dwivedi YK, Rana NP (2020) Towards cyberbullying-free social media in smart cities: a unified multi-modal approach. *Soft Comput* 24(15):11059–11070
- Balakrishnan V, Khan S, Arabia HR (2020) Improving cyberbullying detection using twitter users' psychological features and machine learning. *Comput Secur* 90:101710
- Cheng L, Li J, Silva YN, Hall DL, Liu H (2019) Xbully: Cyberbullying detection within a multi-modal context. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 339–347
- Bastiaensens S, Vandebosch H, Poels K, Van Cleemput K, DeSmet A, De Bourdeaudhuij I (2014) Cyberbullying on social network sites. an experimental study into bystanders' behavioural intentions to help the victim or reinforce the bully. *Comput Hum Behav* 31:259–271
- López-Vizcaíno MF, Nóvoa FJ, Carneiro V, Cacheda F (2021) Early detection of cyberbullying on social media networks. *Future Gener Comput Syst* 118:219–229
- Singh VK, Ghosh S, Jose C (2017) Toward multimodal cyberbullying detection. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2090–2099
- Singh VK, Huang Q, Atrey PK (2016) Cyberbullying detection using probabilistic socio-textual information fusion. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 884–887, IEEE
- Reynolds K, Kontostathis A, Edwards L (2011) Using machine learning to detect cyberbullying. In *10th International Conference on Machine Learning and Applications and Workshops*, vol. 2, pp. 241–244, IEEE
- Roy PK, Singh JP, Banerjee S (2020) Deep learning to filter sms spam. *Future Gener Comput Syst* 102:524–533
- Roy PK, Singh JP (2019) Predicting closed questions on community question answering sites using convolutional neural network. *Neural Comput Appl* 32:10555–10572
- Roy PK (2021) Deep neural network to predict answer votes on community question answering sites. *Neural Process Lett* 53(2):1633–1646
- Roy PK (2020) Multilayer convolutional neural network to filter low quality content from quora. *Neural Process Lett* 52(1):805–821
- Khan MA, Kadry S, Parwekar P, Damaševičius R, Mehmood A, Khan JA, Naqvi SR (2021) Human gait analysis for osteoarthritis prediction: a framework of deep learning and kernel extreme learning machine. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-020-00244-2>
- Yu X, Yang T, Lu J, Shen Y, Lu W, Zhu W, Bao Y, Li H, Zhou J (2021) Deep transfer learning: a novel glucose prediction framework for new subjects with type 2 diabetes. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-021-00360-7>
- Li S, Liu B, Li S, Zhu X, Yan Y, Zhang D (2021) A deep learning-based computer-aided diagnosis method of x-ray images for bone age assessment. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-021-00376-z>
- Kaur H, Koundal D, Kadyan V, Kaur N, Polat K (2021) Automated multimodal image fusion for brain tumor detection. *J Artif Intell Syst* 3(1):68–82
- Aggarwal S, Gupta S, Alhudhaif A, Koundal D, Gupta R, Polat K (2021) Automated COVID-19 detection in chest x-ray images using fine-tuned deep learning architectures. *Expert Syst* 39:e12749
- Çiğdem A, Çürük E, Eşsiz ES (2019) Automatic detection of cyberbullying in formspring.me, myspace and Youtube social networks. *Turk J Eng* 3(4):168–178
- Gomez R, Gibert J, Gomez L, Karatzas D (2020) Exploring hate speech detection in multimodal publications. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1470–1478
- Kumari K, Singh JP, Dwivedi YK, Rana NP (2021) Multi-modal aggression identification using convolutional neural network and binary particle swarm optimization. *Future Gener Comput Syst* 118:187–197
- Sadiq S, Mehmood A, Ullah S, Ahmad M, Choi GS, On B-W (2021) Aggression detection through deep neural model on Twitter. *Future Gener Comput Syst* 114:120–129
- Hosseinmardi H, Rafiq RI, Han R, Lv Q, Mishra S (2016) Prediction of cyberbullying incidents in a media-based social network. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 186–192, IEEE
- Al-Ajlan MA, Ykhlef M (2018) Deep learning algorithm for cyberbullying detection. *Int J Adv Comput Sci Appl* 9(9):199–205
- Banerjee V, Telavane J, Gaikwad P, Vartak P (2019) Detection of cyberbullying using deep neural network. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pp. 604–607, IEEE
- Chen J, Yan S, Wong K-C (2020) Verbal aggression detection on twitter comments: convolutional neural network for short-text sentiment analysis. *Neural Comput Appl* 32(15):10809–10818
- Ali WNH, Mohd M, Fauzi F (2018) Cyberbullying detection: an overview. In *2018 Cyber Resilience Conference (CRC)*, pp. 1–3, IEEE
- Bhat S, Koundal D (2021) Multi-focus image fusion using neutrosophic based wavelet transform. *Appl Soft Comput* 106:107307
- Kamilaris A, Prenafeta-Boldú FX (2018) A review of the use of convolutional neural networks in agriculture. *J Agric Sci* 156(3):312–322
- Udendhran R, Balamurugan M (2021) Towards secure deep learning architecture for smart farming-based applications. *Complex Intell Syst* 7(2):659–666

32. Xue G, Liu S, Ma Y (2020) A hybrid deep learning-based fruit classification using attention model and convolution autoencoder. *Complex Intell Syst.* <https://doi.org/10.1007/s40747-020-00192-x>
33. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Preprint at [arXiv:1409.1556](https://arxiv.org/abs/1409.1556),
34. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258
35. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*
36. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778
37. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826
38. Roy PK, Ahmad Z, Singh JP, Alryalat MAA, Rana NP, Dwivedi YK (2018) Finding and ranking high-quality answers in community question answering sites. *Global J Flex Syst Manag* 19(1):53–68

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.