2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA) | 978-1-6654-6246-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCCMLA56841.2022.9989244

Privacy and Data Control on Social Networks using Deep Learning

A. Parimala[1], Y. Vijayalata[2]
Dept. of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, Telangana, India
parimalaamudalapally@gmail.com

***Abstract*—Online social networks are huge data exchange platforms that help to promote and share a lot of good information about products, news, education, tourism, health care, etc., also there is a great risk involved to individual's privacy and security. Online posted photos can be shared, tagged, and reposted again without having any consent from the people present in the images. In this project, we are proposing a facial recognition-based system that detects every single individual and checks their online account relationship between the individual. The post or image that is reposted, tagged, or copied by known or unknown individuals will be checked. If it is an unknown individual the images convolved through a low-pass filter kernel which removes the high-frequency content like noise and edges to smooth the images that make the image blur. For this mechanism, we proposed to use deep learning based VGG19 architecture for image classification and Haar cascade viola jones algorithm for detection, and computer vision-based blurring techniques used. algorithm performance as the area under the ROC curve (AUC) accuracy of 88% obtained. This proposed model achieved 91.53% classification accuracy and further we evaluated the model using Accuracy, Precision, Recall and F1-Score metrices and achieved 70.83%, 82%, 71% and 0.68%.**

Keywords—Deep Learning, Computer Vision, Viola Jones, VGG19, Privacy-Protection, online social network (OSN)

I. INTRODUCTION

Communication and information transmission through social media are largely facilitated through images. A variety of smart devices are easily able to upload high-resolution photos to a variety of well-known social platforms, such as Facebook, Instagram, Flickr, Tumblr, etc. Users' private information, social relationships, and private lives, however, are often revealed by these photos, risking their personal privacy. The advent of deep neural networks in recent years has brought with it very advanced capabilities that can exploit and use the private data that is contained within images more easily [1]which represents a great threat to the privacy of OSN users [2]. Here, we aim to design an algorithm to protect images from DNN detectors from picking up private information, including human faces, while minimizing the impact on visual quality. Prior to the advent of social media, the methods of protecting social images were mainly divided into two categories. One category of methods protects private information by filtering users who can see images [3]. By filtering clear images according to the social characteristics of the network, users with low intimacy can be identified through the use of private tags and the characteristics of the network's social friends. The disadvantage of this method is that it demands a great deal of prior knowledge to accurately analyse the social network's social friends. The second method of this category involves determining whether images contain private information of any detected individual or not based on images.

Through online media, individuals can easily share data and communicate with one another, which has become an essential part of everyday life. There is a significant amount of data generated by users of online media services through text posts, advanced photographs, and recordings. It is this type of user-generated content that makes online media so unique. Nevertheless, user-produced content often contains sensitive information, so sharing these elements may jeopardize their security. Online communication services like Instagram, Facebook, and Pinterest focus primarily on the sharing of photos. In contrast to literary information, photographs reveal more detailed information to the viewer, which puts the person's identity at risk for identity theft. In addition, a malicious watcher might use the foundation info contained in a picture to deduce information about one. As an added benefit, an individual's information is shielded without harm by anonymization, which is very beneficial for them.

The field of image analysis has been criticized in recent years, but face recognition is considered a promising application. A crucial aspect of face recognition is face detection. It requires a lot of processing power to detect faces in an image. Face recognition is very difficult since different postures, expressions, positions, and directions of the face, along with differences in skin tone, lens, camera gain, and resolution, all make the process very complicated. An algorithm for detecting faces in an image is primarily intended for determining whether there is a face in it. There is an open-source library called OpenCV [4] which contains programming functions primarily for real-time computer vision. Our real-time computer vision applications can be improved using this library, which is a cross-platform library. Images, video recording and video analysis are emphasized, including facial recognition, object recognition, and image processing. People understand faces without having to think about them. Despite the fact that it seems like a very simple task, it is not so simple for the computer, as there is a lot of information to analyse.

In this paper, we are proposing a facial images privacy protection for OSN users based on their relationship status. For instance, if the relationship status of a user on OSN is a friend or family the user can get an access to watch the photos of the particular individual even though it is tagged by an unknown personal. In case if any anonymous person wants to check out the photos shared by user or any other individual tagged, the photo is blurred. For this experimentation we used VGG19 architecture for facial detection and classification, Haar cascade algorithm to detect the facial detection and OpenCV blur operation to anonymize the user's image. Further the paper is organized as follows: Section-II describes the problem statement. Section-III describes the literature survey on existing research works proposed by different

researchers. Section-IV describes the dataset and algorithms used. Section-V describes the flowchart for the procedure followed and methodology. Section-VI describes the results and analysis. In Section VII describes the conclusion.

II. Problem Statement

In our day-to-day lives, social platforms play an increasingly vital role. OSNs, or online social networks, are a type of online business that allows users, such as Facebook, Google, and other social media platforms, to share personal and open information, and build social connections amongst pals, colleagues, individuals with similar interests, family, and even strangers. In order to be mindful of users' truths, OSNs have changed their heads element motive to that of users' truths. Whenever a photograph or picture is posted, it will finally become part of the evidently eternal record. People may utilize late results for many unexpected purposes, so they may be risky. The mafia may also court any big names in the case of a published or posted article. It doesn't matter if an individual is keen on being a part of the transferred photo/content or now not wishing to be a part of it, they switch the image and tag different people. Different people are described as suffering from a particular condition, so it seems to be more complicated at this point. As the photo is being posted or transferred, the client is completely unaware of the impact his or her image will have on the person in the photo. Unfortunately, such an unavoidable condition cannot be prevented at this point. There is really a need to manage these activities to ensure a limited risk is associated with the labelling or transfer of images. platforms like Facebook and Instagram are encouraging people to become more involved in such issues, rather than imposing barriers or increasing security.

III. Literature Survey

Here in this section, we are giving a brief about the various researchers works on privacy protection and the methodologies used.

Chih-Hsueh Lin et al., with a method to de-identify images of faces using thermal features extracted from thousands of images using CNN and support vector machines [5]. To enhance the fine-tuning of expectation precision and deidentification of crude faces, this study aims to develop a thermographic-based face recognition strategy. The result of the tests, RGB pictures had a precision of 0.834, component pictures had 0.953, and include lattice had 0.967.

Zhongzheng Ren et al., proposed a new principal approach for learning a video face anonymizer [6]. They used an adversarial training scenario in which two competing frameworks battled each other. They used a dataset that consists of videos and photos for training purposes to train face modifiers and simultaneously they trained an action detector to person actions then it is formulated for multi-task learning. Authors used Fast-RCNN, ResNet-101 and MTCNN architectures for classification and regression and spatial action detection purposes. For classification they achieved 95.75% and for modified LFW faces they accuracy achieved 66.35%.

Jun Yu et al., proposed a system called “Iprivacy” using hybrid convolutional Neural network (HD-CNN), hybrid deep multi task learning (HD-MTL) and decision tree classifiers [7]. Using conditional random fields and deep CNN models, they segmented every image into semantic objects. The CNN was trained to enable pixel-level expectations and arrangements and CRF model to determine how to produce semantic article locales based on neighbour pixels for a similar item class. They achieved 92% and 87% accuracy.

Yasuhiro TANAKA et al Proposed a PSM (Price Sensitivity Measurement) system to find a relationship between willingness to share photos and preferred level of photo blurring for privacy protection [8]. the authors experimented with finding the right balance point between revealing private information and ensuring their photographs were blurred to a preferred degree using PSM. Using two types of social media services, with and without access restrictions by PSM, the authors analysed the relationship between the willingness to blur photos and desired level of blurring photo. They used polynomial approximate curves to calculate the intersecting points in order to clarify them. It can be concluded from a comparison of the upper limits of each intersection between the social media with/without access limitations (=0.26) that the difference is smaller than the difference between optimum points (=0.50) and lower limits (=0.5).

Liang Du et al. proposed a simple and effective framework, named GARP-Face, that balances utility preservation in face de-identification [9] which analyses facial features like Gender, Age, and Race attributes of the images and preserve these attributes. Face de-identification is a transformation that is used in the facial recognition process to convert it into blurred image. Also, they proposed (Active Appearance Model) for building an attribute-specific model of faces the proposed approach is evaluated using the MORPH dataset.

IV. Algorithm and Dataset

In this section, we discussed about the dataset, Augmentation process, ImageNet, VGG19 architecture and Haar cascade algorithm.

A. Dataset

In this project we collected the whole dataset from google images. This dataset consists of very few images of well-known celebrities. As the main motive of this experimentation is to provide a privacy protection, the application should take less photos from the user and train. As this is not an ideal case for deep learning, we are using transfer learning process and took ImageNet weights [10]. The dataset for this project comprises two classes of images. Class 1, consists of pictures of the person whose face should be blurred for privacy purposes, and Class 2, includes images of other persons.

B. Data Augmentation

In data augmentation, we try to extrapolate additional images out of the existing set of images available for training and testing [11]. In our use case we apply the following augmentation techniques:

1) Rotation of Image: We rotate the images by an angle of +/- 15º.

2) Shift Image Horizontally: We shift the image randomly in the horizontal axis by 10%.

3) Shift Image Vertically: We shift the image randomly in the vertical axis by 10%.

4) Shear: We tilt the images in both X and Y axes by 10%.

5) Brightness: Manipulate the brightness from 50% to 150%.

6) Flip: Flip the image horizontally and vertically.

Fig. 1. Facial Image dataset

C. ImageNet

Founded in 1998, ImageNet categorizes and labels images by almost 22,000 separate object categories (manually). In addition to the 1.2 million training images, 50,000 images will be used to validate the model and 100,000 images will be used for testing [10].

In recent years, some of the most advanced state-of-the-art convolutional neural networks have been used for ImageNet challenges. Through transfer learning, such as feature extraction and fine-tuning, these models display a strong ability to generalize to images outside of the ImageNet dataset.

D. VGG19 Architecture

The VGG 19 architecture has a total of 19 layers. Refer to Figure 5 for an overview. This model requires an RGB image input of 224*224*3. In what is called VGG19, there are sixteen convolutional layers and three fully connected layers. A one-pixel stride is used with convolution kernels of size 3*3. It consists of five maximum pooling layers with a kernel size of 2*2 and a stride of two pixels in each layer. There are three fully connected layers, in which the first two layers each comprise 4096 channels, and the final layer includes 1000 channels [12]. There is only one layer that has a dedicated architecture, which is known as the SoftMax layer. This architecture is used for the facial recognition (classification) purpose.

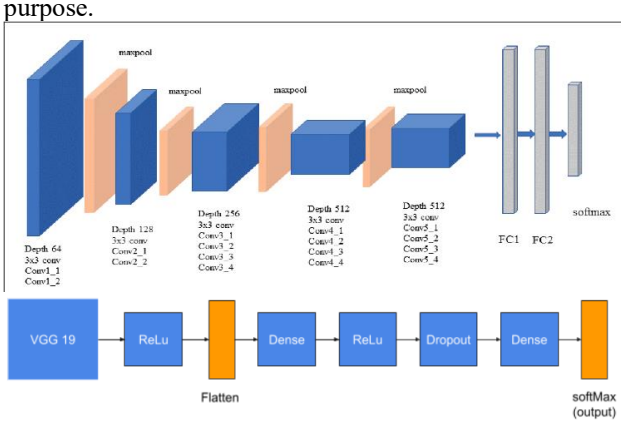

Fig. 2. VGG19 Architecture

E. Viola Jones (Haar Cascade)

Faces in real-time videos or images are identified with Haar-Cascade using Object Detection Algorithms. This research relies on features like edge detection or line detection to identify faces. As a part of the OpenCV face detection function, an algorithm has already been written and trained to find a face in an image using the Cascade Classification Class. Haar features are employed in this function. By combining binary variables that have been calculated by combining a number of functions, the most common Haar characteristics can be represented. The templates depicted in FIG.2 contain the Haar feature templates. It is adopted that each window in the image is placed one by one for the purpose of calculating the features one by one. There are several different features in each image, and each one is represented by a value that is determined by subtracting pixels at the white rectangle location from pixels at the black rectangle location.

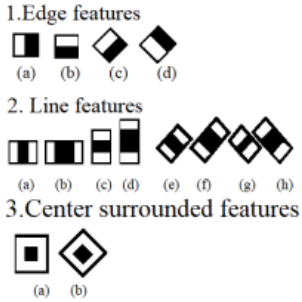

Fig. 3. Figure 1 Features of Haar-Cascade

V. Methodology

In this method, two experiments have been carried out and assessed the performance of our approach. Below Fig.4 describes the flow chart for the experimentation carried out step by step.

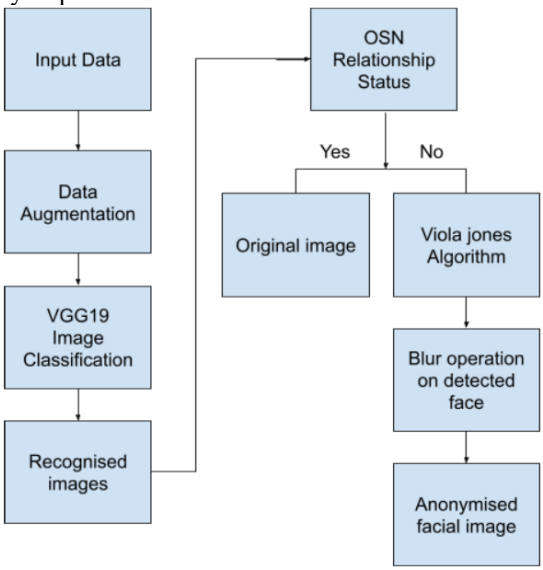

Fig. 4. Flow Chart.

A. Facial Classification

First, we used a data augmentation technique to carried out and to create more samples of our classes and we fed those image data to VGG19 classifier that classifies the person whose photo needs to be blurred from other persons.

In this dataset collection, a number of classes are assessed for robustness of models. For training and testing, different kinds of unique identities were used. Training the networks

was conducted using only a few samples from each identity, and testing was conducted using four to ten images. Fig. 1 depicts an example of a training image from the dataset collection.

Training the VGG19 CNN with ImageNet weights was done after splitting off the training and test data. Our model summary has been augmented with additional layers on top of the model to train the images. This experiment uses a pretrained model so it might overfit or underfit if we were to train it again. As a result, the model's precision is very uncertain. Then, rather than retraining the entire model, we add layers on top of the VGG19 model. We took all the parameters or neurons for the extra layers and put them into a single vector, or may we say, we call it a flatten layer. On top of the dense layer, we have a dropout layer which is fully convoluted after we have converted it into a single vector. We relied on ReLU activation to train the dense, dropout, and final layers, and Softmax functions since it is a two-class classification.

```
Epoch 1/20
1/1 [==============================] - ETA: 0s - loss: 9.1470 - accuracy: 0.3898 - true_positives: 23.0000 - false_negatives:
36.0000 - true_negatives: 23.0000
Epoch 00001: val_accuracy improved from -inf to 0.50000, saving model to ./best_weights.hdf5
1/1 [==============================] - 8s 8s/step - loss: 9.1470 - accuracy: 0.3898 - true_positives: 23.0000 -
false_negatives: 36.0000 - true_negatives: 23.0000 - val_loss: 34.4122 - val_accuracy: 0.5000 - val_true_positives: 12.0000 -
val_false_negatives: 12.0000 - val_true_negatives: 12.0000
Epoch 2/20
1/1 [==============================] - ETA: 0s - loss: 20.8181 - accuracy: 0.6610 - true_positives: 39.0000 - false_negatives
20.0000 - true_negatives: 39.0000
Epoch 00002: val_accuracy did not improve from 0.50000
1/1 [==============================] - 7s 7s/step - loss: 20.8181 - accuracy: 0.6610 - true_positives: 39.0000 -
false_negatives: 20.0000 - true_negatives: 39.0000 - val_loss: 12.1935 - val_accuracy: 0.4583 - val_true_positives: 11.0000 -
val_false_negatives: 13.0000 - val_true_negatives: 11.0000
Epoch 3/20
1/1 [==============================] - ETA: 0s - loss: 6.0747 - accuracy: 0.7119 - true_positives: 42.0000 - false_negatives:
17.0000 - true_negatives: 42.0000
Epoch 00003: val_accuracy improved from 0.50000 to 0.70833, saving model to ./best_weights.hdf5
1/1 [==============================] - 7s 7s/step - loss: 6.0747 - accuracy: 0.7119 - true_positives: 42.0000 -
false_negatives: 17.0000 - true_negatives: 42.0000 - val_loss: 2.8043 - val_accuracy: 0.7083 - val_true_positives: 17.0000 -
val_false_negatives: 7.0000 - val_true_negatives: 17.0000
Epoch 4/20
1/1 [==============================] - ETA: 0s - loss: 5.8303 - accuracy: 0.6441 - true_positives: 38.0000 - false_negatives:
21.0000 - true_negatives: 38.0000
Epoch 00004: val_accuracy improved from 0.70833 to 0.75000, saving model to ./best_weights.hdf5
1/1 [==============================] - 7s 7s/step - loss: 5.8303 - accuracy: 0.6441 - true_positives: 38.0000 -
false_negatives: 21.0000 - true_negatives: 38.0000 - val_loss: 3.0998 - val_accuracy: 0.7500 - val_true_positives: 18.0000 -
val_false_negatives: 6.0000 - val_true_negatives: 18.0000
Epoch 7/20
...
1/1 [==============================] - ETA: 0s - loss: 0.3388 - accuracy: 0.9153 - true_positives: 54.0000 - false_negatives:
5.0000 - true_negatives: 54.0000
Epoch 00015: val_accuracy did not improve from 0.75000
1/1 [==============================] - 7s 7s/step - loss: 0.3388 - accuracy: 0.9153 - true_positives: 54.0000 -
false_negatives: 5.0000 - true_negatives: 54.0000 - val_loss: 0.8222 - val_accuracy: 0.7083 - val_true_positives: 17.0000 -
val_false_negatives: 7.0000 - val_true_negatives: 17.0000
Epoch 00015: early stopping
```

Fig. 5. Number of epochs

We used the ADAM optimizer to compile the model. This optimiser implements gradient descent, which allows us to adjust the weights of the model in an optimized way in order to improve the accuracy. Due to the fact that this is a two-class classification, categorical class entropy has been set as the hyperparameter for loss function and used one hot encoding for classification. With model.fit, the training has been started, and the hyper-parameters are set at 64 batches and 20 iterations in order to pass the entire data set through the algorithm. To evaluate whether the model has been trained accurately, we preferred to use the Keras model in this experiment. This model does a cross validation to determine how accurately it was trained. In figure 16, you can see how accurate the trained model is.

In terms of facial detection, we have achieved a 91.53% accuracy rate. Using removed highlights with deep learning is used for the static method in this work.

B. Facial Detection and Anonymization

Recognition and detection of a face is in any case an undertaking of distinguishing a known or obscure face from an all-around identified item. Humans are capable of performing simple tasks such as this. Our external features (hairline, head shape) or our internal characteristics (eyes, nose, mouth) are utilized for successful face recognition. When someone's face in a group is chosen, it can be removed from the rest of the scene and contrasted with an information base of stored pictures. This algorithm must determine how to distinguish a basic face from the rest of the background so as to work properly. The ability to tell a face apart from another is at the core of face recognition software.

Fig. 6. Viola Jones flow chart

This algorithm consists of four major components: Haar features, the integral image, adaboost, and cascading. Each of the rectangular features is a separate element. The resulting value from each of these features is calculated by subtracting the sum of pixels under the white rectangles from the sum of pixels underneath the black rectangles (Fig. 3). Using a 24 by 24 window as the base size window, the Viola-Jones algorithm evaluates haar features in a given image by using a 24 x 24 window.

Summation of values in rectangular subsets of a grid can be computed by using integral images or summed area tables. Following are the details of how Viola and Jones used integral images to compute Haar-like features rapidly [13]. The integral image is constructed as follows:

$$ii(x,y) = (x+a)^n = \sum_{x' \le x, y' \le y} i(x',y')$$

The integral image is determined by $ii(x,y)$ at the pixel location (x,y). The original image is determined by (x',y') is the original image. This integral image can be constructed by applying the following recurrent formulas to the original image in one pass:

$$s(x,y) = s(x,y-1) + i(x,y) \ldots$$

$$ii(x,y-1) = ii(x-1,y) + s(x,y) \ldots\ldots$$

where s(x, y) is the accumulated pixel values of row x, s(x, y − 1) = 0, ii(x − 1, y)=0.

In Fig. 3, shows how an integral image is extremely efficient when computing the sum of any rectangle. Using the example of rectangle region ABCD, we might be able to calculate the sum of pixels as follows:

$$\sum_{(x,y)\ \varepsilon\ ABCD} i(x,y) = ii(D) + ii(A) - ii(B) - ii(C) \ldots\ldots$$

Ad-aboost is a machine learning algorithm that helps find only the best among all 180,000+ features [14]. After these features are found, a weighted combination of all the features is used in evaluating and deciding whether any given window has a face.

Adding top and left pixels is the first step to calculating the new pixel value, then adding the values around the patch to calculate the new pixel value. This is how Ada boost distinguishes between relevant and irrelevant features. In the following step, Adaboost identifies relevant and irrelevant features. Essentially, it combines weak classifiers linearly to construct a strong classification model.

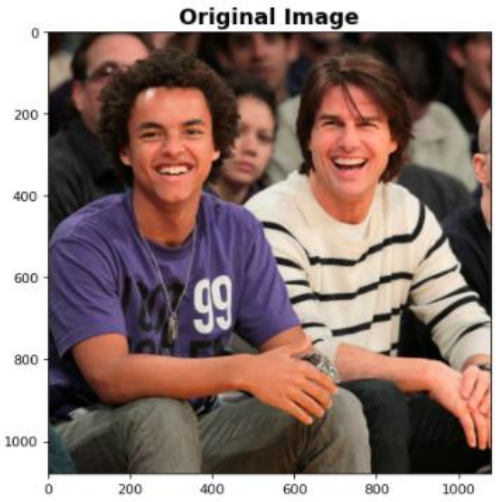

Fig. 7. Original image from classifier.

The calculation of nearly 2500 features is performed. In addition, cascading of calculations can reduce the number of computations. In this case, each set of features is categorized by another set of classifiers and so on. In the above method you will make more rapid decisions regarding whether something is a face or not, and you will be able to reject anything that does not match up with the output you want. A standard resolution of 100x100 is applied to the cropped and resized detected face as shown below in fig.8.

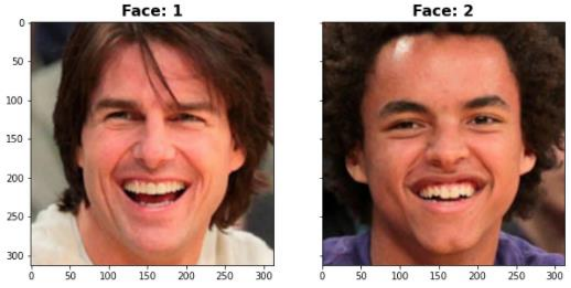

Fig. 8. Highlighted

After the detection, Convolution of an image with a low-pass filter kernel results in blurring. This can be used to remove noise. By doing this, high frequency content from the image is reduced (e.g., noise, edges). A little blurring occurs in edges as a result. By using a normalized box filter, a blurred image can be created. This merely replaces the central element with the average of the pixels below the kernel area. Functions that do this include cv.blur() and cv.boxFilter(). It is important to specify the kernel width and height [15]. Below is an example of the 3x3 normalized box filter:

$$\mathbf{K} = \frac{1}{9}\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

When a person wants to take a fresh photograph with phone and upload it to social media, it is able to recognize the people in the photo and suggests that you tag them. To begin with, it is the ability to identify a person's or a pet's face. Once the algorithm detects the person it checks the profile to confirm the relationship status of the person. If the person is stranger then the image will be blurred as shown below fig.9.

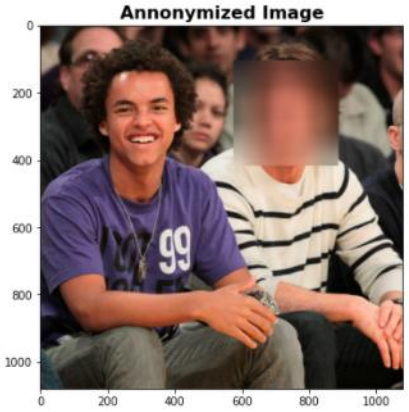

Fig. 9. Original image from classifier.

RESULTS AND ANALYSIS

The purpose of this thesis is to present research findings related to facial expression classification using the face detection method. In order to recognize facial expressions, we provide an architecture based on VGG19. It takes facial images as input and classifies them to determine whether the person is an OSN user or not. We have tried this model and it achieved 91.53% accuracy with a loss of 3%. The comparison graph of accuracy is shown below.

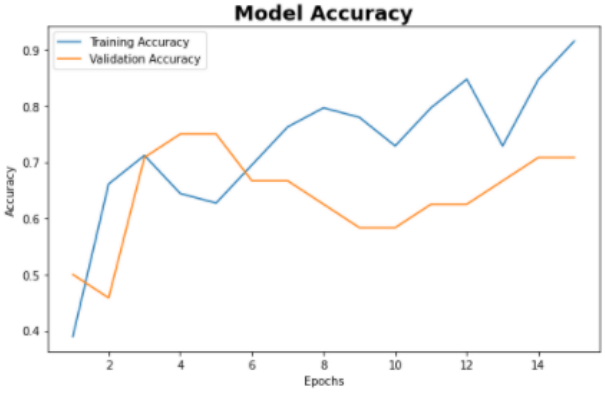

Fig. 10. VGG19 Training and Validation accuracy

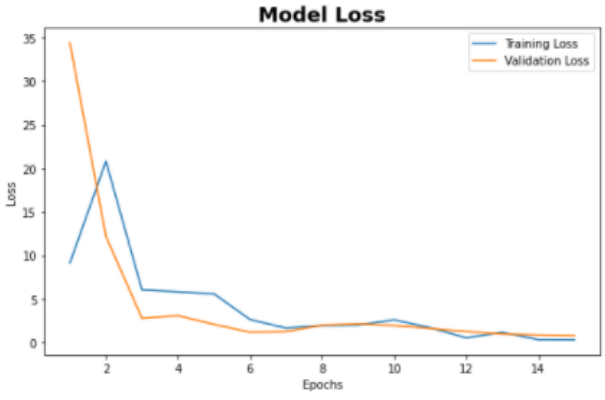

Fig. 11. Training and Validation loss

This section contains the results of various tests conducted on the models used in this study. In the current study, we used the following performance metrics: Accuracy, Sensitivity (Recall), Precision, and F1 [16]. In these metrics, true-positives (TP), false-positives (FP), false-negatives (FN), and true-negatives (TN) are defined. This is an example matrix used to calculate TP, TN, and FN.

TABLE I. PERFORMANCE METRICS

S. No	*Performance Metric*	*Accuracy*
1.	Accuracy	70.83
2.	Precision Macro Micro Weighted	 82% 71% 0.82%
3.	Recall Macro Micro Weighted	 71% 71% 71%
4.	F1 metrics Macro Micro Weighted	 0.68 0.71 0.68

ROC analysis is a method used to illustrate the diagnostic ability of the binary classifier system [14]. It can be interpreted as representing the true positive rate against the false positive rate. Additionally, it can be viewed as representing the number of incorrectly recognized samples.

Table I summarizes the key indicators of face recognition algorithm performance as the area under the ROC curve (AUC) accuracy of 88% obtained. Similar conclusions are drawn in the case of ROC curves.

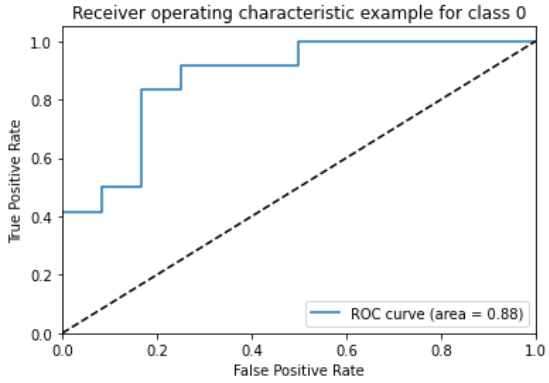

Fig. 12. Original image from classifier.

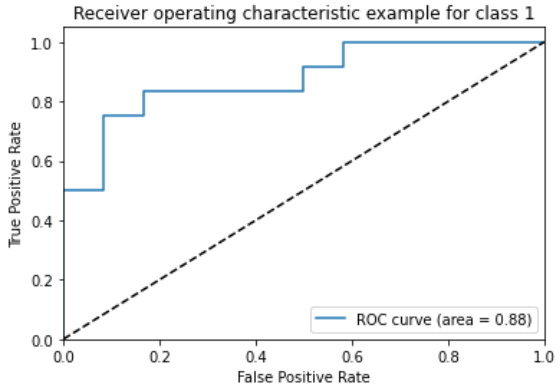

Fig. 13. Original image from classifier.

CONCLUSION

In this paper, our work consists of developing a visual privacy protection system for online social networking services. Our architecture has been designed to protect individual privacy by checking the status of relationships, and if any unknown source is stalking, capturing, or tagging the other's profiles without consent. An analysis of the effects of blur and motion blur on the performance of face recognition is presented in this paper. During the experimentation part of the study, we employed Haar features to recognize faces. We used the VGG19 model using transfer learning for facial detection and achieved an accuracy of 91.53%. Also, we validated the whole model using different matrices as follows: Accuracy, F1-Score, ROC(AUC) curves, Recall, and precision matrices.

REFERENCES

[1] "Tonge, Ashwini Kishore, and Cornelia Caragea. "Image privacy prediction using deep features." Thirtieth AAAI Conference on Artificial Intelligence. 2016.".

[2] "Acquisti, Alessandro, and Christina Fong. "An experiment in hiring discrimination via online social networks." Management Science 66.3 (2020): 1005-1024.".

[3] "Bonneau, Joseph, Jonathan Anderson, and Luke Church. "Privacy suites: shared privacy for social networks." SOUPS. Vol. 9. 2009.".

[4] "Bradski, Gary, and Adrian Kaehler. "OpenCV." Dr. Dobb's journal of software tools 3 (2000): 2.".

[5] "Lin, Chih-Hsueh, Zhi-Hao Wang, and Gwo-Jia Jong. "A de-identification face recognition using extracted thermal features based on deep learning." IEEE Sensors Journal 20.16 (2020): 9510-9517.".

[6] "Ren, Zhongzheng, Yong Jae Lee, and Michael S. Ryoo. "Learning to anonymize faces for privacy preserving action detection." Proceedings of the european conference on computer vision (ECCV). 2018.".

[7] "J. Yu, B. Zhang, Z. Kuang, D. Lin and J. Fan, "iPrivacy: Image Privacy Protection by Identifying Sensitive Objects via Deep Multi-Task Learning," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 5, pp. 1005-1016, May 2017, doi: 10.".

[8] "Tanaka, Yasuhiro, et al. "Relationship between willingness to share photos and preferred level of photo blurring for privacy protection." Proceedings of the ASE BigData & SocialInformatics 2015. 2015. 1-5.".

[9] "L. Du, M. Yi, E. Blasch and H. Ling, "GARP-face: Balancing privacy protection and utility preservation in face de-identification," IEEE International Joint Conference on Biometrics, 2014, pp. 1-8, doi: 10.1109/BTAS.2014.6996249.".

[10] "J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.".

[11] F. Chollet, "The Keras Blog," 05 06 2015. [Online]. Available: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html.

[12] "Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).".

[13] "Wang, Yi-Qing. "An analysis of the Viola-Jones face detection algorithm." Image Processing On Line 4 (2014): 128-148.".

[14] "Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." Journal of computer and system sciences 55.1 (1997): 119-139.," [Online].

[15] "OpenCV," [Online]. Available: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html.

[16] "Chowdary, M. Kalpana, Tu N. Nguyen, and D. Jude Hemanth. "Deep learning-based facial emotion recognition for human–computer interaction applications." Neural Computing and Applications (2021): 1-18.".

[17] "A.Parimala, Y.Vijayalata, Ashlin Deepa R N." Survey on Image Authentication and Privacy in public networks(2022).