Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.9, No.8, 2018
www.iiste.org

A Tool for Privacy-Aware Online Personal Photo Sharing Using Deep Learning Technique

Ghadah Suliman Alghunaimi
College of Computer and Information Sciences, Al-Imam Mohammad Ibn Saud Islamic University
E-mail: Ghadah.suliman202@gmail.com

Abstract

In recent years, online social networking has been considered as a sharing information platform and has occupied an essential part in many individual's lives and business growths. Consequently, there has been a marked increase in the collection and illegal exploitation of photos amassed online without owner consent, thus violating individual privacy rights such as contravention of online published photo laws which contribute to public social anxiety. To address these concerns, we propose a face recognition tool based on the Deep Learning Convolutional Neural Network (CNN) technique, which may be utilized within a social networking website as a gateway control for posting images. The goal of this paper is to preserve user privacy by preventing their images from being posted on social networking sites without prior consent.

This tool will extract features from an input photo posted on a social network site and compare those attributes against the facial characteristics of photos in a prohibited dataset, which is comprised of users unwilling to share their photos. Depending on the result, the CNN-based tool could either allow sharing of the photo or prevent and alert the user attempting to post or share a given photo about his/her potential violation of end-user privacy provided the image belongs to a person on the banned list. Additionally, the CNN tool will provide an option for a user to add his/her photo to the banned list.

The proposed tool includes two main elements which have been developed in Python with Jupyter Notebook. The first component is a deep learning model which is trained on LFW images dataset capable of achieving 91.89% matching accuracy. The second is the GUI of the tool which allows the user to input photos and use the trained model to predict whether this photo belongs to the person in banned list, thus preventing illicit sharing downstream. The integration between two elements has been tested and achieved 85% accuracy.

Keywords: Deep Learning, Face Recognition (FR), Convolution Neural Networks (CNNs), Online Social Networks (OSNs), Online Photo Sharing.

1. Introduction

In the last decade, the world witnessed a huge evolution in technology and computing potentials, which had a very good effect on capital and labor from an economic view. Alongside the benefits and effects on the economy, this evolution impacted the social aspect as well. Technology has highly influenced the sociality of humans, many people became addicted to instantly sharing their daily activities, photos, information, feelings and so on, globally or with their relatives and friends.

Photo Sharing Online Social Networks (OSNs) perform one of the main areas of online social network usage, where the OSNs offer photo sharing as an addition to the messages or posts written by users. The photo is classified under a multi-media content class and considered to be a part of sharing content that is covered by the intellectual property right (Guest, 2016). Recently, there have been massive collections of images over the Web and online networks; this availability increases the risk and probability of illegal exploiting of the images, without the consent of the real owners. Such exploitation is considered as a privacy violation against individuals, thus individuals need to ensure that they share their photos securely. Moreover, there has been an advent of studies reporting that social network users are worried about privacy and want to prevent sharing photos or any other private information publicly. But even with this, they still find it tedious to go back and delete most of their old uploads (p. llia, 2015). Overall, the main contributions of this paper are:

- This paper will add a new viewpoint for utilizing the facial recognition technology to reshape the privacy issue of photo publication and sharing in OSNs.
- As a result of using this tool in social networking websites, it will serve the community by protecting the privacy of individuals regarding controlling their photos sharing.

2. Related Work

Authors on (p. llia, 2015) defined several scenarios of privacy leakage in the online photo sharing, scrutinized in:

-*The Malicious Tagger:* which is the state of publishing an individual photo without their consciousness, showing them in impropriate states, and tagging them after publishing. In such scenarios, the photo publisher embarrasses the users in the photo and neglects any delete request sent by users in a photo.

-*The Silent Uploader*: it is the scenario identical to the malicious tagger scenario, but in this scenario, users in the photo were not tagged when the photo was published. In these, users in the photo are unaware of the existence of the embarrassing photo of them.

-*The Group Photographer:* it is the scenario where users are taking a shared photo with other friends or colleagues, and this photo was published without any concerns about privacy amends of users in the photo.

-*The Accidental Over-sharer:* it was a scenario when a user appeared in the other's shared photo accidentally and without any intended purpose by the photo publisher (uploader), ending up as a photo shared with a much larger audience than the user in the photo desired.

-*The Friendly Stranger*: is a scenario when users in a photo have a common privacy setting and privacy restrictions bounded by friends only, so both will feel safe when any of them shared a photo. However, this sense is most near to be false, because there is a part of friends in one of them that does not have mutual friends with the others' friends, where such friends are considered strangers and by that means users in a photo can still be viewed by strangers.

Regarding such potential risks and privacy leakage, online social networks tried to introduce some control and privacy mechanisms, for example Facebook introduced a privacy setting where the uploader owns the decision of whom can see their photo, chosen from a list called a smart list where various options appear on it, such as friends, only me, friends of friends, family member, and so forth (N. Vishwamitra, 2017). Such mechanisms and the recently adopted ones in the online social networks have provided a sufficient privacy feature for access control, but it can't hinder the massive scenarios of privacy violations.

Due to the importance of preserving privacy of online photo sharing, various studies suggested and presented different mechanisms for online privacy, such as Ahang and Rathod (Serra, 2017) proposed an efficient facial recognition system to identify each individual presented in the photo and notify the individual of sharing activity in low cost. The notified user will have the right to make a decision on the sharing activity. The proposed system is optional and set as a privacy setting, and once the user sets the photo sharing privacy, the face recognition system trains on the figures and photo upload on the user's profile, and the user will get a notification once any sharing activity occurs that is a part of sharing activity.

Ilia et al. (p. llia, 2015) presented a control access system utilizing photo sharing privacy, where the system will ban the unauthorized individual for identifying specific users in the photo, through blurs it out as shown in the below figure. However, the system will compare the photo uploader and the recognized users in the photo, and if the uploader is not in the friend list of the user presented in the photo, then the depicted user will get notification of the sharing activity. The authors take advantage of the available face recognition currently used through social network platforms. The proposed system evaluated on the Facebook social platform, where the system banned users form identifying tier contacts in 87.35% of the prevented photos.

3. Convolutional Neural Networks (CNNs)

In recent times, research and studies have kept track of the developments in facial recognition over the years, in terms of its potential to be a standard means of biometric authentication that will easily handle and solve existing problems and challenges of legally accessing web content or legally publishing new content (p. llia, 2015). Convolutional Neural Network is the most popular deep learning algorithms using in face recognition problems. It is considered as a type of neural network taking advantages of deep learning which drive the features and classification method automatically rather than hand-crafted approaches, which support and enhance the robustness of learning process (Y. Zhog, 2017). The most successful applications of CNN in the face recognition are DeepFace (97.3%) and FaceNet (99.6%) (Serra, 2017), these two impressively have achieved state of the art performance in recent years, DeepFace is close to human performance (97.5%), while FaceNet has even surpassed it.

The CNN stimulated from multi-layers perceptions, the perception defined as simple computational neurons, which have various inputs with a single output. It transforms the original image from the original pixel values to the final class scores. The main parameters of perception are Weight and Bias, where weight is a vector weight of input, and bias is the distinguished weight of input, the perceptions benefit to put a set of parameters as a basis for classification training set (Charniak, 2017). The CNN has various constructive layers of artificial neuron clusters where output is combined with a convolution operation. The CNN utilized the spatial correlations via local linked form among nearby neurons layers, that means the input for the layer is the subset of prior layers, as shown in Figure below (lab, 2017).

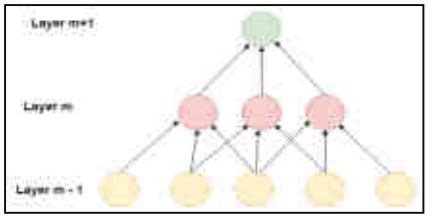

Figure 1: Spatial Local Correlation in CNNs (lab, 2017)

As shown that, CNN layers concerned with a small part of the inputs and disposed of the rest of the parts which reduces the edges, considering the adjacent pixels. Thus, each neuron layer will count same features, but in different locations. Each layer will have replicated features or units with the same parametrization which is weight vectors and bias, those above explained. A group of parametrizations created features on a map as shown in the figure below (lab, 2017).

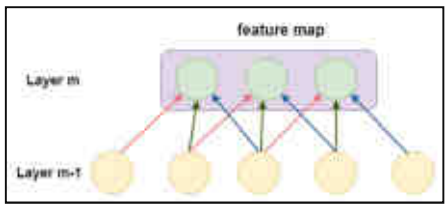

Figure 2: Feature Map on the CNNs in Term of Perceptions (lab, 2017)

Each neuron layer will identify same features; the shared of weight vector and bias concepts. The shared feature is only the set of a feature present in the layer; the receptive fields are a set of the input space that connects neurons. In the convolutional layer, each receptive field will handle one weight value only, meaning there is one weight defined in all neurons that enhance the ability to train large networks (Charniak, 2017), (lab, 2017).

4. Privacy Aware Tool

The paper aims to preserve the people's privacy in social networking by providing a privacy-aware tool for protecting their photos from being shared without their consent through integrating this tool with a social network website. This is accomplished by implementing a face recognition system based on CNN deep learning algorithms to predict whether the upload photo belongs to one of the people who are not willing to share their photos.

4.1 System Design

The experimental work done for this paper included two main parts: the first part is training a deep learning model on a dataset which is considered in this paper as a banned List; photos of people willing to protect their privacy; then extracting the features of these photos and classifying through the utilizing deep learning CNN model. LFW dataset was utilized for training and evaluating the proposed model. The output of these processes is the classified model contains a prediction for each class which then is utilized to recognize the person in the photo to be posted. The second part is a tool that allowed the user to input the photos through GUI, then utilize the trained model to determine whether the inputted photo belongs to the person existing in the banned list or not, then based on the results it will either allow or prevent posting of this photo.

Additionally, the proposed tool provides an option for users to add their photos into the banned list, as well as

archiving the photo in order to allow the system to recognize it in future circumstances and prevent anyone from posting their images without prior permission.

Python is utilized to simulate a face recognition tool for ensuring privacy in online photo sharing besides the Tensorflow framework with the Keras library, which streamlines creating the deep learning algorithm. The Jupyter Notebook environment is employed to develop the proposed model; as it has a high flexibility to run deep learning algorithms. The model utilized in the proposed system is the Convolutional Neural Network (CNN), which is considered the newest face recognition approach and is rapidly becoming the dominant approach in machine learning system and state-of-the-art model architecture for face recognition tasks. Figure 3 shows a high block diagram of the proposed system.

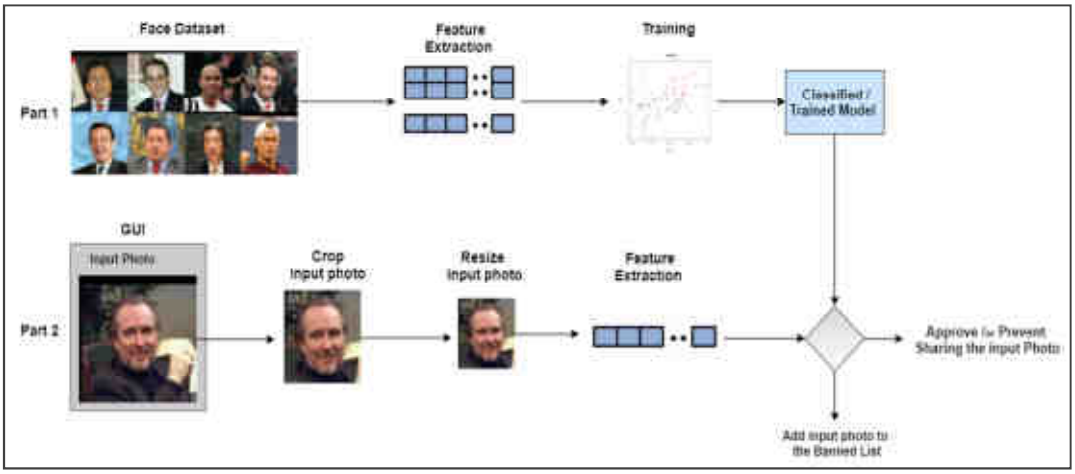

Figure 3: General Block Diagram of Proposed System

4.2 Network Model

Typically, CNN is composed of a stack of convolutional modules that perform features of extraction and classification. The proposed CNN model follows the modern architecture and classification of layers into three functional units: The Input Unit, the Feature-Extracted Unit (including Convolution, ReLU, and Average-pooling layers), and the Classification Unit (including Global Average pooling and SoftMax layers), as shown in Figure 4.

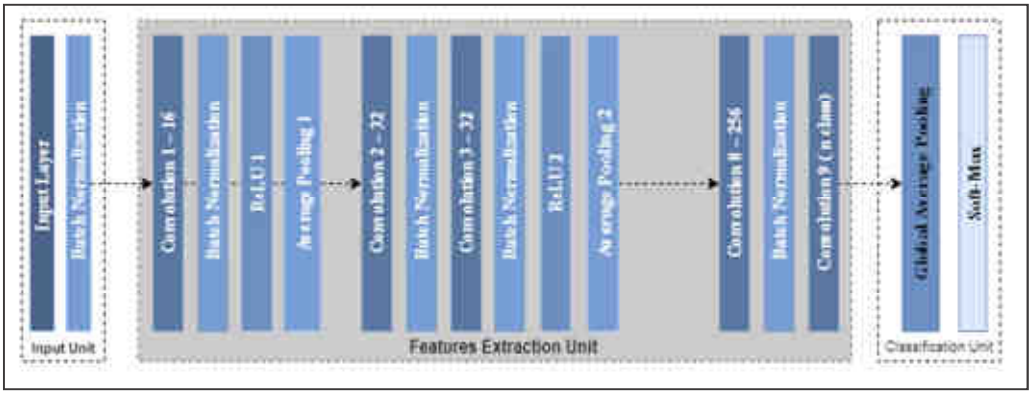

Figure 4: Proposed CNN Model Architecture

The following are a list and descriptions of the layers utilized in the construction of the proposed CNN model:

- ***Input Layer***

Input Layer is where loading the input data of the image for processing in the network. It is including the size of the image (height, width) and the depth which is representing the color channels (typically the number of channels is three for RGB).

- ***Convolutional Layer***

$m_1^{(l-1)}$Convolutional Layer is the core building block of a CNN, and the main function is to extract features from the input image. The first convolutional layer accepts the raw image as input from multiple channels as a matrix

$m_2^{(l-1)} \times m_3^{(l-1)}$

of the pixel value, then slide the filter or 'kernel' through the image and applying the weights for each region to give the output. The output matrix of each convolutional layer is called Convolved Feature, or Feature Map transferred to the next convolutional layer, such the input layer ⌊ consists feature map from the previous layer with the size of (Stutz, 2014). The following figure shows the first convolutional layer input.

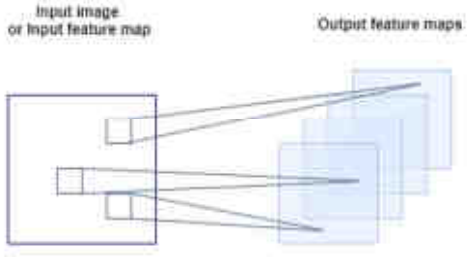

Figure 5: Single Convolutional Layer, Input Layer and Output Maps *(Stutz, 2014)*

The following equation formulated to reflect the convolutional layer function (Stutz, 2014):

$$y_i^{(l)} = B_i^l + \sum_{i=1}^{m_{i-1}} K_{i\ i}^{(l)} \times Y_i^{(l-1)} \qquad (1)$$

$\therefore B_i^l$: Bias matrix

$\therefore K_{i.j}^{(l)}$: Kernel filter function with the size of $2h_1^{(l)} + 1 \times 2h_2^{(l)} + 1$

$\therefore$ J is the j^{th} feature map in layer (l-1) followed kernel filter.

The output feature size map influenced by border effects, thus the map dimensional and size will change as the following equations (Stutz, 2014).

$$m_2^{(l)} = m_2^{l-1} - 2h_1^{(l)} \qquad (2)$$

$$m_3^{(l)} = m_3^{l-1} - 2h_2^{(l)} \qquad (3)$$

For multilayer convolutional, the output will be aggregation output of each layer output, performing in sum operator, and also the size of final output influences of filter border effects.

- ***Batch Normalization Layer***

During the training of deep learning networks, the distribution of each layer's inputs changes which slows down the training, this problem is known as internal covariate shift and can be addressed by normalizing layer inputs. The batch normalization is a recently popularized method for accelerating the training (achieves the same accuracy with 14 times fewer training steps) by reducing the internal covariate shift.

- ***Rectified Linear Unit (ReLU)***

For any neural networks to be powerful, it needs to contain non-linearity. The Rectified Linear Unit is a non-linear activation function computed after the convolution layer; it resulted in more fast training for a large network. The activation is a threshold at zero, in other words. It has output 0 if the input ≤ 0, and the output is equal to the input if the input > 0. It computes the function: *f(x)=max (0, x)*

- ***Average-Pooling Layer***

$m_1^{(u)} = m_1^{(u-1)}$Pooling layer (also call subsampling layer), which is a form of non-linear down-sampling, located between sequential convolution layers to reduce computation complexity and minimize the representation size. The input is the feature map induced by convolutional layers, where the output of Average-Pooling layer (U) combines feature map, which its size is reduced. The figure below simulates pooling layer function concept.

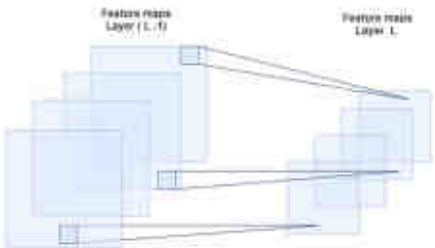

Figure 6: Pooling Layer Function concepts, Layer L is the pooling layer (Stutz, 2014)

- ***Dropout layer***

Dropout layer is a regularization technique to prevent overfitting in training and gives a big improvement in CNN models. Dropout layer used on each of convolutional layer with a small portion in addition to utilized on fully connected layer (N Srivastava, 2014). Dropout layer is applied after each average pooling layer.

- ***Global Average Pooling layer (GAP)***

Global Average Pooling is replaced the traditional fully connected layers in CNN where combines all the features from previous feature extraction layers (convolution and pooling layers) with less consumption of memory resources. It is applied on the top of the last convolutional layer, which has a number of filters equal to the number of classes in the dataset and fed to a SoftMax activate layer (Ahn, 2017), (Shuangfei Zhai, 2016).

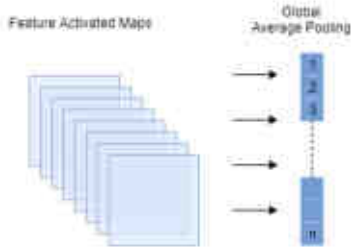

Figure 7: Global Average Pooling, n Represents the Number of Classes

- ***Output Layer***

The last layer of the network is output layer; it consists of the same number of neurons as the number of classes of input data (Morchhale, 2016). The output of neurons is digit probabilities as one for each output class to be specified. The output layer used SoftMax activation function (also call normalized exponential) that works as a classifier which compute the classes scores to lies between zero and one.

4.3 Build and Integrate the Tool

The second part of the proposed system implements a face recognition tool and integrates it with the classified model when serving the required functionality of the proposed system. The paper attempts to design a friendly interface with the Tkinter module in Python which provides all the helper functions in making the GUI. Tkinter provides the support for adding buttons, canvas, messages and error boxes for the user to call a specific object under certain conditions.

Three buttons are added into GUI; each button acts as an object and does a specific task. The first button is for loading the selected photos from a personal device, the second one is for posting this photo on a social network website, while the third button is an option to allow the user to add his photo to the banned list. The canvas is also added to the loaded photo to display, Figure 8 shows the tool interface.

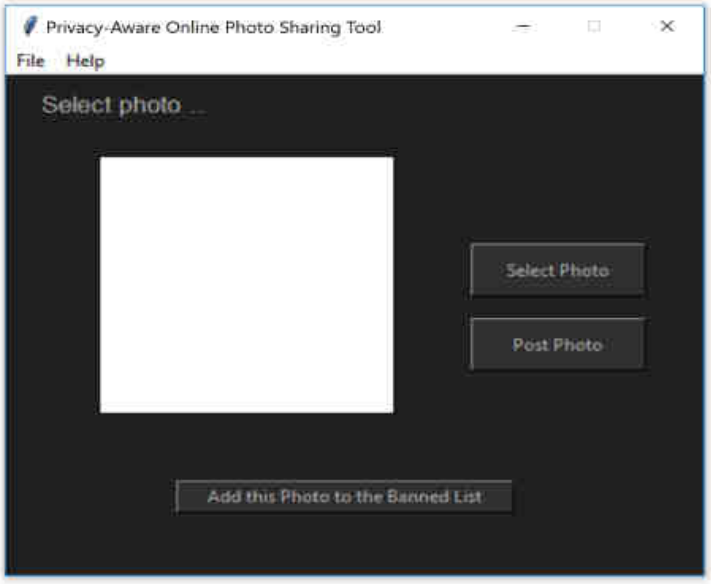

Figure 8: Proposed Tool

Once the user selects a photo from his device, the photo will be shown in the image box and the user has two options, whether he posts this photo on the social media website or he adds it into the banned list. If the user chooses to post a photo, the button callback function takes the inputted image from the box, preprocesses it and then predicts its class using the trained model. The preprocessing and prediction steps are done through a number of tasks. First, resizing the inputted image to 250x250 as a dataset size. Then, it crops it in the middle of the size 128x128, and resizes it to 64x64 afterward, as it did in the model. After that, it reshapes the image to tell the model that it is one image, with 64x64 dimensions and has three channels (RGP). The next step is predicting the class of the inputted image by using the trained model then tell if the person in the input photo exists in the dataset or not. This is done by a sequence of algorithms.

```
FR = model.predict(img)
classes = np.where(FR>0.90)

if(classes[1].size==0):
     showinfo(' ', 'Image Uploaded Successfully')
else:
    showerror(' ', 'This person is in the Banned List, \n You can not Upload his image')
```

The first line is the Keras function **model.predict**, this function gets a prediction from the trained model on the inputted image. The output (FR) is an array of probability with the size of classes in the dataset. The second line sets the threshold test as 90% and defines classes as an array containing only classes from (FR) where the probability is greater than the threshold of 90% confidence. The third line is a function of checks if the classes array is empty or not. If the classes array is empty, that means there is no class which has a probability greater than the threshold of 0.90 for the current image, so the image is considered out of the database. Otherwise, this image is considered inside the database. Based on the previous result, the tool either allows posting the photo or prevents it and shows a message to the user in both cases. In case the user tries to upload a photo of a person, and this person is in the banned list, the system will prevent the user from posting this photo, and the error boxes will appear to explain the reason. As shown in Figure 9, an error message appears after trying to upload a picture of a person on the banned list. Otherwise, the user can upload the photo successfully.

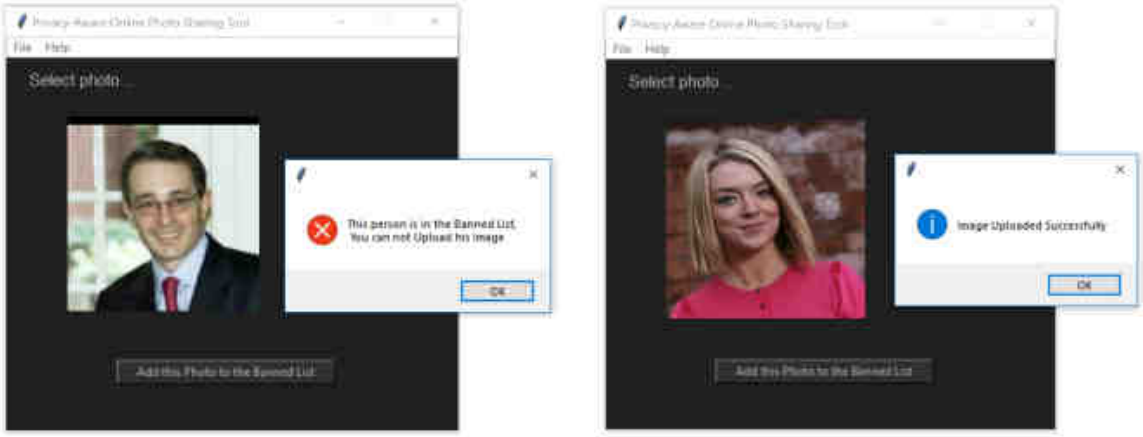

Figure 9: Reject /Approve to Upload the Photo

The code is available on this link: https://github.com/gsa99/Face-Recognition

5. Evaluation

The proposed tool evaluated on Labeled Faces in the Wild (LFW) dataset (LAB, 2012), which has been widely utilized as the benchmark of unconstrained face verification. It is an unconstrained dataset, including variations in background, poses, illumination, and expression. LFW consists of 13,233 images for 5,749 persons (classes), 1,680 classes with two or more images. Image dimension is 250x250 pixels.

Training the CNN model using all classes in LFW would result in a useless model since many of classes have only one image, while the CNN model needs enough valid labeled images in each class to allow the neural network to learn every label.

In order to overcome this problem, set a threshold in the model to select classes to contain more than one image. In this experiment, a threshold of the number of images per class was seated at 30 to train the model on classes containing more than 30 images. Training time required for training is 15 hours with CPU computational power and 1 hour with GPU computational power. Table 1 shows the experimental details of the LFW dataset.

Table 1: Experimental Details of LFW Dataset

Number of Images in the dataset	Number of train Image	Number of Test Image	Number of Classes	Epoch number	Patch size	Training Time	Recognition Rate
13,233	1896	474	34	300	16	1hour GPU	91.98%

5.1 Performance Evaluation

The accuracy of classifications represents the percentage of correct predictions concerning all instances, while the loss represents the wrongly predicted ones concerning all instances. The trained model on LFW dataset achieved a 91.98% accuracy, and a 35% loss rate. Figure 10 shows the loss and accuracy curves, comparing training set performance against the test set. The accuracy keeps going and increases after around epoch 50, while the loss keeps going down.

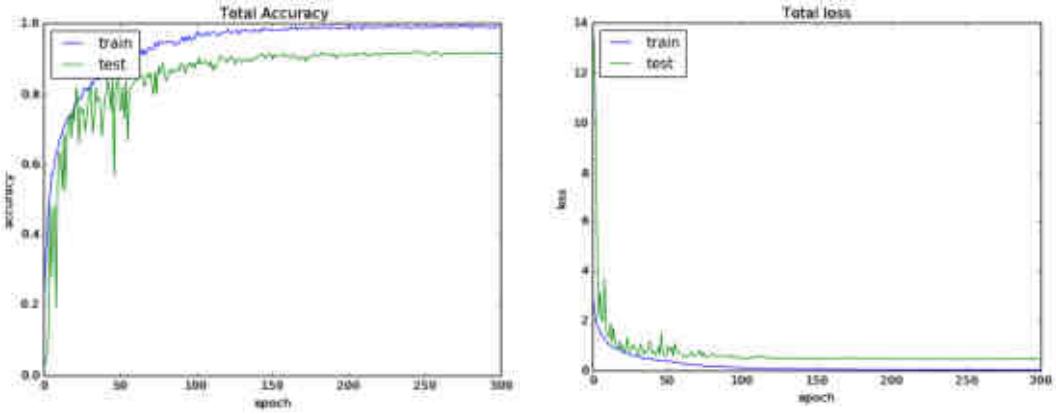

Figure 10: Accuracy and Loss Rate on LFW

5.2 Privacy Evaluation

After evaluating the trained model, the complete and fully integrated proposed system was checked by examining the integration between tool algorithms and the trained model. In order to conduct this test, 100 photos have been selected as a test sample, 50% from the dataset and 50% selected from the Internet. The results of a test on two groups are obtained:

True Positive (TP) = number of cases where the tested photos are correctly classified as an image on the banned list.
True Negative (TN) = number of cases where the tested photos are correctly classified as not in the banned list.
False Positive (FP) = number of cases where the tested photos are incorrectly classified as an image on the banned list.
False Negative (FN) = number of cases where the tested photos are incorrectly classified not on the banned list.

These fractions are represented in a confusion matrix table to measure accuracy in the proposed system. The examination was conducted on LFW, then by recording the results and notes. The test is conducting testing 200 photos, 100 photos from LFW database and 100 photos collected from the Internet with faces in the middle as in the LFW dataset. The result obtained from the test is shown in the following confusion matrix.

Table 2: Confusion Matrix of Integration Test on LFW Dataset

		Actual Value		
		Images in dataset (+)	Images not in dataset (-)	
Predicted Value	Test Image in DS (+)	TP: 89	FP: 19	109
	Test Image not in DS (-)	FN: 11	TN: 81	91
	Total	100	100	200

6. Comparison with Others Work

This section will compare the CNN model utilized in this paper to another faces recognition system utilizing the same method on the LFW dataset. The result of the comparison in Table 3 shows that our method is better than (Yi Sun, 2013), (Guosheng Hu, 2015) and (Serra, 2017), but worse than (Haoqiang Fan, 2014).

Table 3: Comparison with state-of-the-art methods on LFW

Method	Accuracy
proposed method	91.98%
Network fusion +JB (Guosheng Hu, 2015)	88.7%
Method in (Serra, 2017)	89.6%
ConvNet-RBM (Yi Sun, 2013)	91.75%
Pyramid CNN (Haoqiang Fan, 2014)	97.2%

7. Conclusion

Photo sharing through online social networks (OSNs) has been integral in fostering online communication and key feature offered to OSN users. However, in addition to the value it provides OSN users, it may also lead to anxiety and privacy concerns among social networks users. As one of their top priorities, social network administrators and management personnel have expended vast resources to ensure user privacy with a high level of preservation.

This paper aims to protect photo privacy in online photo sharing environment through the use of innovative face recognition algorithms and systems to govern photo sharing.

Face recognition technology is one of the latest advertised information technology approaches based on detecting and recognizing facial parameters. Although face recognition methods and techniques have increased gradually and achieved high efficiencies in face recognition, many challenges still exist in recognizing faces in unconstrained environments.

The proposed tool consists of two main parts that integrate together to provide the required functionality. The first part is Deep Learning CNN model, which is based on a multilayer concept where each layer defines simple computational neurons from the input while abstracting the pixels and taking into account the relation of adjacent pixels. Therefore, each layer parametrizes specific features from the input, and the group of layers creates a feature map. The last layer performs the image classification task and represents the scores for each class. The second part of the tool allows the user to input a photo and then utilize the output of the network model to predict whether this photo belongs to the person in the banned list, and thus allow or prevent sharing of a photo.

In addition to utilizing the Keras library to build deep learning network, Python and Jupyter Notebook are utilized to implement the proposed system. When combined, Python and Keras together create a powerful deep learning development environment.

The network model preprocessed the images before passing them into the CNN layers in order to reduce the noise in images. The CNN architecture consists of nine convolutional layers each followed by batch normalization, non-linearity and Average-pooling layers. Then flowing by a global average layer, which combines all the features from previous layers. The final layer is of a size representing the number of classes. Then the last layer is an output layer with a SoftMax activation function that works as a classifier, which computes the classes scores. The network model has been trained and evaluated on LFW dataset and has achieved 91.89% accuracy. Additionally, the privacy tool has been tested on an independent set of photos from inside and outside the dataset and achieved 85% accuracy.

In addition to the challenge in choosing a suitable environment, this work has faced some key challenges, including limited computational power and time-consuming initialization periods during training of the model. However, the limitations of time and computational power were overcome by conducting the training on a cloud-based platform.

The proposed tool is a novel approach to achieving privacy protection on social networks. As a result of using this tool in social networking websites, it will serve the community by protecting the privacy of individuals regarding controlling their photos sharing.

8. Future Work

As a result of this paper, the following avenues have been identified as areas of potential improvement for the overall performance of the system:

1. To improve the algorithm in detecting the face within an image
2. To enhance the algorithm in order to better deal with the noise within an image
3. To allow users to add their photos into the banned list
4. To apply the proposed tool on an open source social network website

References

Ahn, B. B., 2017. The Compact 3D Convolutional Neural Network for Medical Images, CA ,USA: Stanford University.

Charniak, E., 2017. Introduction to Deep Learning, Rhode Island , USA: Brown university.

Scikit developers, s.-l., n.d. sklearn.metrics.precision_recall_fscore_support. [Online]
Available at: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html
[Accessed 3 April 2018].

Guest, E., 2016. Photo editing: Enhancing social media images to reflect appearance ideals. Journal of Aesthetic Nursing, 5(6), pp. 444-446.

Guosheng Hu, Y. Y. D. Y. J. K. W. C. S. Z. L. T. H., 2015. When Face Recognition Meets with Deep Learning: an Evaluation of Convolutional Neural Networks for Face Recognition. Santiago, Chile, Computer Vision Foundation CVF.

Haoqiang Fan, Z. C. J. a. Q. M., 2014. Learning Deep Face Representation. arXiv.

L. lab, 2017. Deep Learning Tutorial , 1st release, Montreal , USA: University of Montreal.

LAB, V., 2012. Labeled Faces in the Wild Home (LFW) face image dataset. [Online] Available at: http://vis-www.cs.umass.edu/lfw/ [Accessed 11 Sep 2017].

Morchhale, S., 2016. Deep Convolutional Neural Networks for Classification of Fused Hyperspectral and Lidar Data, North Carolina, USA: North Carolina University.

N Srivastava, G. H. A. K. I. S. a. R. S., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. The Journal of Machine Learning Research (JMLR), 25 Jan, 15(1), pp. 1929-1958.

N. Vishwamitra, Y. L. K. W. H. H. K. C. a. G. A., 2017. Towards PII-based Multiparty Access Control for Photo Sharing in Online Social Networks. New York , USA, s.n.

P. Abhang, a. S. R., 2017. My Privacy My decision: Control of Photo Sharing on Online Social Networks. International Research Journal of engineering and technology (IRJET), July, 4(7), pp. 655-658.

p. llia, L. P. E. A. a. S. L., October 12 - 16, 2015 . Face/off: Preventing privacy leakage from photos in social networks. Colorado, USA, s.n.

Serra, X., 2017. Face recognition using Deep Learning, Catalonia , USA: Polytechnic University of Catalonia.

Shuangfei Zhai, Y. C. ,. W. L. a. Z. (. Z., 2016. Doubly Convolutional Neural Networks. Spain, s.n.

Stutz, D., 2014. Understanding Convolutional Neural Networks, Aachen, German: RWTH Aachen University.

Y. Zhog, J. C. a. B. H., 2017. Towards End-to-End Face Recognition through Alignment Learning. arXiv, Volume 1701.07174v1.

Yi Sun, X. W. ,. X. T., 2013. Hybrid Deep Learning for Face Verification. Computer Vision Foundation CVF, pp. 1490-1493.