CHAPTER 1

INTRODUCTION

With the growing popularity of smart phones and other mobile devices, high quality cameras are becoming increasingly ubiquitous and pervasive, as a result, capturing high quality images has become one part of our daily activities and image sharing has now become very popular in social platforms like Facebook, Instagram etc. By default, the shared images can be seen by anyone in social networks. Since images can intuitively tell when and where a special moment took place, who participated and what were their relationships, the shared images can reveal much of users personal and social environments and their private lives. Thus privacy protection is a critical issue to be addressed during social image sharing. Unfortunately, many people especially young users of social networks often share private images about themselves ,their friends and classmates without being aware of the potential impact on their future lives caused by unwanted disclosure and privacy violations.

To ensure privacy, most social image sharing sites allow users to manually specify coarse-grained privacy settings (preferences): whether an image is public, private or visible to their family members or friends. However, due to the lack of privacy knowledge, it would not be easy for common users to correctly configure privacy settings to achieve their desired levels of privacy protection; also, given the large number of images being shared and the tedious steps needed for fine-grained privacy settings, some users may not be willing to spend extra time on providing such fine-grained privacy settings. Based on these observations, in this work, a new approach called iPrivacy (image Privacy) is developed to automate such privacy setting process for social image sharing. Unlike many previous works which typically recommend privacy settings based on similarity of users' profiles or image tags, our idea is to automatically detect the privacy-sensitive objects from the images being shared, recognize their classes, and identify their privacy settings, so that the image owners can be warned what objects in the images need to be protected before sharing.

Nowadays, the social medias like Facebook and Instagram are very popular in this world. Uploading images on these social medias has become a daily routine. This may leads to many privacy problems. The chance of misusing the images is very high in social medias. This may spoil the life of many people. So, to avoid this problem here introducing a new technique called iPrivacy. This technique prevents the misuse of human images from social medias. Privacy-Aware Image Classification and Search which proposed the technique to automatically detect private images, and to enable privacy-oriented image search. Modern content sharing environments such as Flicker or Youtube contains a large amount of private resources. These resources can be of a highly sensitive nature, disclosing many details of the user's private sphere.

For this purpose learning privacy classifiers trained on a large set of manually accessed Flicker photos combining textual metadata of images with a variety of visual features. This technique is to build classification models on selected visual features and textual metadata, and apply this models to estimate adequate privacy settings for newly uploaded images and search results.

Collective Privacy Management in Social Networks" which proposed to model the problem of collaborative enforcement of privacy policies on shared data by using game theory. Social networking is one of the major technological phenomena of the Web 2.0, with hundreds of millions of people participating. Social networks enable a form of self expression for users, and help them to socialize and share content with other users. In spite of the fact that content sharing represents one of the prominent features of existing Social network sites, Social networks yet do not support any mechanism for collaborative management of privacy settings for shared content. In particular, it proposed a solution that offers automated ways to share images based on an extended notion of content ownership. Building upon the Clarke-Tax mechanism, it describe a simple mechanism that promotes truthfulness, and that rewards users who promote co-ownership. Integrating our design with inference techniques that free the users from the burden of manually selecting privacy preferences for each picture. To the best of the knowledge, the first time such a protection mechanism for Social Networking has been proposed. The paper also show a proof-of-concept application, implemented in the context of Facebook, one of today's most popular social networks. Showing that supporting these type of solutions is not also feasible, but can be implemented through a minimal increase in overhead to end-users.

CHAPTER 2

IMAGE SHARING

Sharing images via mobile phones has become popular. Several networks and applications have sprung up offering capabilities to share captured photos directly from mobile phones to social networks. The most prominent of these is Instagram, which has quickly become the dominant image sharing-centric social network with over 500 million members. Other applications and networks offering similar service and growing in popularity include Streamzoo, Path, PicsArt and Starmatic.

Image sharing, or photo sharing, is the publishing or transfer of a user's digital photos online. Image sharing websites offer services such as uploading, hosting, managing and sharing of photos (publicly or privately).This function is provided through both websites and applications that facilitate the upload and display of images. The term can also be loosely applied to the use of online photo galleries that are set up and managed by individual users, including photo blogs. Sharing means that other users can view but not necessarily download images, and users can select different copyright options for their images.

The emergence of social networks, image sharing has now become a common online activity. Facebook stated in 2015 that there were approximately two billion images uploaded to its service daily. In terms of image sharing, Facebook is the largest social networking service. On Facebook, people can upload and share their photo albums individually, and collaboratively with shared albums. This feature allows multiple users to upload pictures to the same album, and the album's creator has the ability to add or delete contributors. Twitter collaborated with Photo bucket in developing a new photo sharing service so users can attach a picture to a tweet without depending on another application such as Twit Pic or Y frog. More than 500 million monthly active Instagram users.

CHAPTER 3

PRIVACY SENSITIVE OBJECT CLASSES

The critical challenge to be conquered here is how to identify all the privacy-sensitive object classes efficiently and learn the object-privacy relatedness precisely from massive social images. Determine which object classes should be detected from individual images being shared and leverage object detection results to recommend the best-matching privacy settings for image sharing. Considering 1.82 billions active users of social networks and trillions of shared images, there may exist a large set of privacy-sensitive object classes. Such privacy-sensitive object classes can further be partitioned into two categories: (a) user-independent classes such as humans, locations and discrimination texts in images; and (b) user-dependent classes such as home shrines and visual attributes for personal hobbies.

Another critical issue for automating the privacy setting process is the time limitation, e.g., users may expect to get their privacy setting recommendations quickly. Because there could have large numbers of privacy-sensitive object classes, detecting the privacy-sensitive objects from the images being shared and recognizing their classes could be very expensive, thus recommending the best-matching privacy settings for image sharing could be an extremely time consuming process. Specifically, if a flat approach is employed, the computational cost will grow linearly with the total number of privacy-sensitive object classes (to be detected and recognized) and hence it is not scalable; if a hierarchical approach is adopted, the object detection process could be speed up dramatically but it would seriously suffer from the so-called inter-level error propagation problem, i.e., the mistakes made at the parent nodes will propagate to their child nodes and such mistakes cannot be recovered.

CHAPTER 4

IMAGE PRIVACY PROTECTION

A. Privacy Protection for Social Image Sharing

Several recent works have studied how to automate the privacy setting process for image sharing . Bonneauet al. proposed the concept of privacy suites which recommend users a suite of privacy settings that “expert” users or other trusted friends have already set, so that normal users can either directly choose a setting or only need to do minor modification. Ravichandran et al. studied how to predict a user’s privacy preferences for location-based data (i.e., share her location or not) based on location and time of day. Fang et al. proposed a privacy wizard to help users grant privileges to their friends. The wizard asks users to first assign privacy labels to the selected friends, and then uses this as the input to construct a classifier to classify friends based on their profiles and automatically assign privacy labels to the unlabeled friends. More recently, Klemperer et al. studied whether the keywords and captions (which are provided by the users when they tag their photos) can be used to help users create and maintain access-control policies more intuitively, where the social tags created for organizational purposes can be re-purposed to help create reasonable access-control rules. The aforementioned approaches focus on deriving policy settings for only traits, so they mainly consider social context such as one’s friend list. While interesting, they may not be sufficient to address challenges brought by image files for which privacy may vary substantially not just because of social contexts but also due to the actual image content as considered in our work. Zerr’s work and Squicciarini et al. have explored privacy-aware image classification by using a mixed set of features, both content and meta-data. More recently, Tonge and Caragea have first integrated the deep features for image privacy prediction, Spyromitros-Xioufis et al. [50] have recently leveraged user-dependent images and privacy settings to support personalized privacy-aware image classification. Both teams have found that the deep features can yield remarkable improvements on the performance as compared with other handcrafted visual features such as SIFT, GIST and color histograms. On the other hand, our approach provides a finer level of image classification by detecting the privacy-sensitive objects from the images.More importantly, our approach is very efficient in:

(a) identifying a large set of privacy-sensitive object classes from massive social images.
(b) learning the object-privacy relatedness, e.g., the correspondences between the object classes and the privacy settings for image sharing.
(c) supporting fast and accurate detection of the privacy-sensitive objects from the images being shared and leveraging the object-privacy relatedness to recommend the best-matching privacy settings for image sharing.

B. Semantic Image Analysis

Even social tags have been leveraged for automating the privacy setting process and social tagging has been proved to be a valuable tool in image sharing platforms, it is worth noting that image privacy protection is about image content (object classes) rather than social tags. Thus it is very attractive to develop new algorithm to automatically identify the correspondences (relatedness) between the object classes and the privacy settings for image sharing. It is worth noting that the privacy settings are given at the image level rather than at the object level and each social image may contain multiple objects, thus there may have huge uncertainty on such object-privacy relatedness. To reduce the uncertainty, it is very attractive to develop new algorithms to:
(a) achieve semantic segmentation of object regions and some pioneering researches have been done recently on leveraging deep learning to achieve semantic image

segmentation.
(b) accurately align the object classes with the privacy settings for image sharing and we can take the lessons from some pioneering researches on automatic object-tag alignment .

By performing semantic image segmentation and automatic object-privacy alignment, we can learn from massive social images (and their privacy settings) to determine the correspon- dences between the object classes and their privacy settings, such object-privacy correspondences may further allow us to: (1) identify a large set of privacy-sensitive object classes; and (2) recommend the best-matching privacy settings for the images being shared by detecting their underlying privacy-sensitive objects, recognizing their classes, and identifying their privacy settings automatically.

C. Deep Multi-Task Learning for Object Detection

Achieving automatic detection of privacy-sensitive objects from images may play an important role in image privacy protection. Deep learning has demonstrated its outstanding abilities on learning high-level features and significantly boosting the accuracy rates for large-scale object detection (i.e., detecting and recognizing large numbers of object classes), but they still have room to improve. For example, soft max is used to flatly map the high-level features into large numbers of object classes ,where the inter-task correlations (inter-class visual similarities) are completely ignored. As a result, the process for learning the deep CNNs may be pushed away from the global optimum because the gradients of the objective function are not uniform for all the object classes and such learning process may distract on discerning the object classes that are hard to be discriminated . In our work, a tree structure is seamlessly integrated with deep network to identify the inter-related learning tasks and avoid such distraction effectively. By considering multiple inter-related learning tasks jointly, multi-task learning has demonstrated its strong ability on learning more discriminative classifiers for large-scale object detection. Even multi-task learning has demonstrated many advantages in theory, there are at least two obstacles for applying multi-task learning to support large-scale object detection. The first obstacle is how to identify the inter related learning tasks automatically, and traditional multi-task learning algorithms usually assume that all the tasks are equally related. However, such assumption may not hold in the scenario of large-scale object detection because it is unnecessary for each object class to be related with all the others. The second obstacle is how to leverage inter-task relationships for enhancing multi-task learning. In order to share information appropriately, multi-task learning usually needs to assume how the tasks are correlated.

By considering the differences of the inter-task relationships among multiple learning tasks, some researchers have proposed joint learning approach to learn the inter-task relationships and the multi-task classifiers simultaneously. For large-scale object detection application, such joint learning approach may seriously suffer from the problem of huge computational cost. Some attempts have recently been made to exploit the tree structures in multi-task learning and deep learning .

CHAPTER 5

AUTOMATIC OBJECT-PRIVACY ALIGNMENT

The first step in our iPrivacy system is to:

- Achieve semantic segmentation of object regions for massive social images.
- Leverage large-scale social images and their privacy settings to identify a large set of privacy-sensitive object classes.
- Assign the privacy settings (which are given at the image level) into the most relevant object regions (object classes).
- Learn the correspondences between the object classes and the privacy settings, e.g., object-privacy relatedness.

In this work, the privacy settings are coarsely partitioned into 3 groups:

(a) public
(b) private and
(c) shared with friends or family.

We have collected massive social images and their privacy settings, where the privacy settings are loosely given at the image level without providing their exact correspondences with the underlying object classes. Based on these observations, each social image is first segmented into a set of semantic object regions.

After the semantic object regions are extracted from images and their object classes are recognized, an automatic object-privacy alignment algorithm is developed to achieve more precise alignment between the object regions (object classes) and the privacy settings for image sharing. By performing semantic image segmentation, each social image is partitioned into a set of semantic object regions and each semantic object region may correspond to one certain type of object classes, e.g., one image may contain multiple objects. The semantics for each social image can be described effectively by all its object classes. For a given social image, by projecting all its object tags (object classes) over the full set of 1000 object classes, we can obtain a 1000-dimensional sparse representation for the given social image, e.g., bag of object classes.

CHAPTER 6

SYSTEM ARCHITECTURE

A system architecture or system’s architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system..

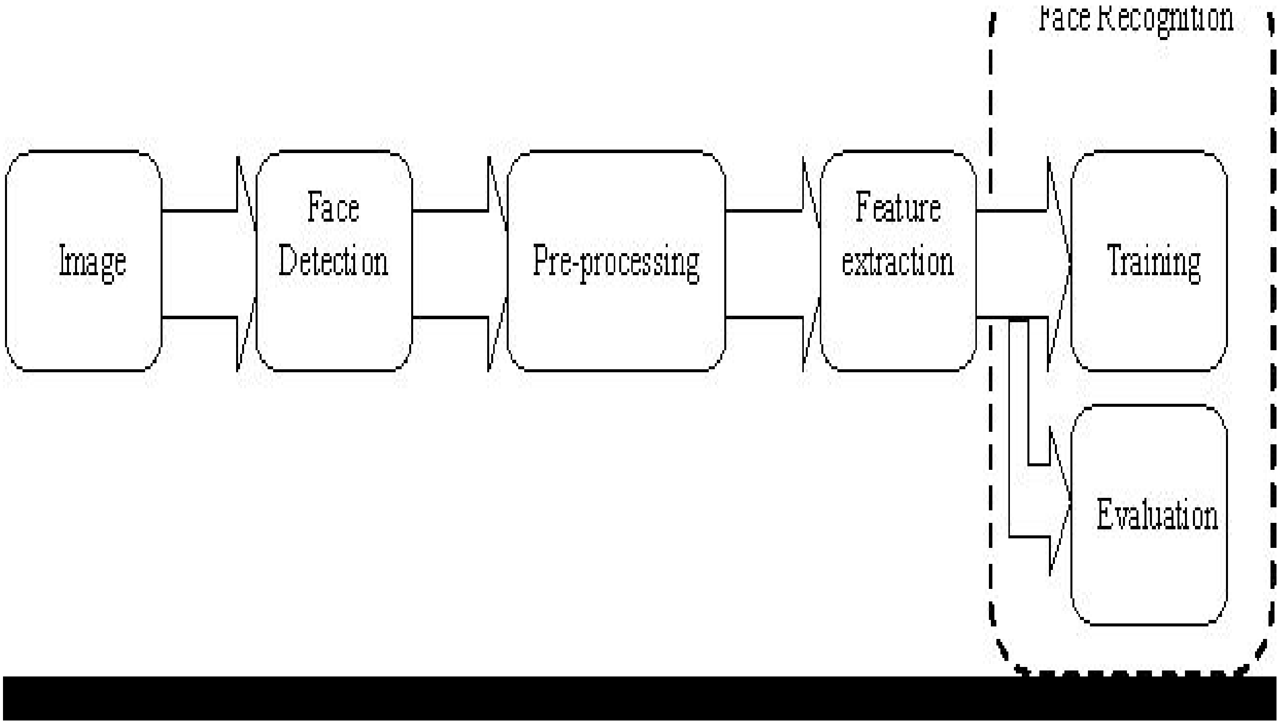

Figure 6.1 :Overall system architecture

Figure1 gives the system architecture of system. Mainly have two phases. The first step is Face detection phase, it is performed to detect human in an image. The second phase is face recognition and blurring the face of human to avoid the misuse .

The system architecture contains the three parts:

1. Image filtering.

2 Canny edge detection.

3. Face recognition and blurring.

1.Image filtering

Gaussian filter is used to smoothening an image to suppress the noise. In signal processing, a Gaussian filter is a filter whose impulse response is a Gaussia function (or an approximation to it).Gaussian filters have the properties of having no overshoot to a step function input while minimizing the rise and fall time. The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump. This kernel has some special properties which are detailed below. The

Gaussian distribution in 1-D has the form:

G(x) = 1/

where is the standard deviation of the distribution. We have also assumed that the distribution has a mean of zero (i.e. it is centered on the line x =0).The distribution is illustrated in Figure 2.

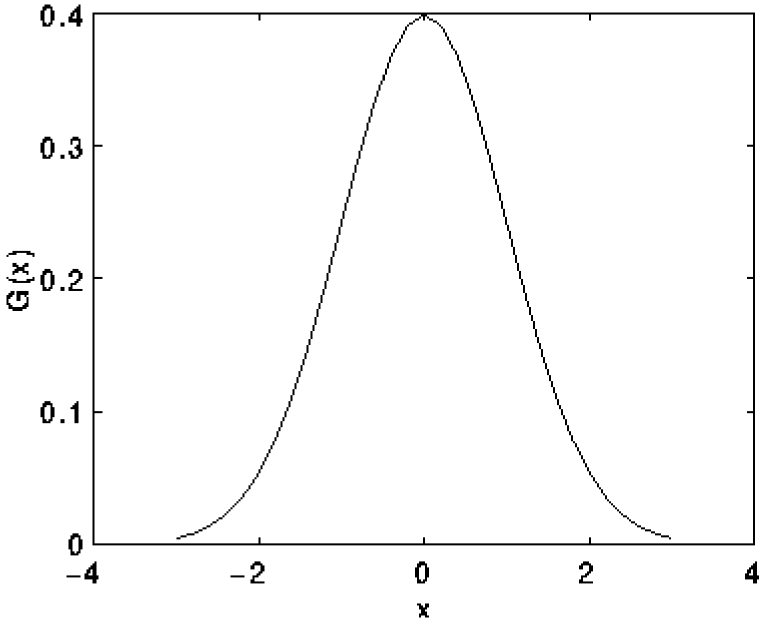

Figure 6.2: 1-D Gaussian distribution with mean 0 and =1

In 2-D, an isotropic (i.e. circularly symmetric) Gaussian has the form:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

This distribution is shown in Figure 3.

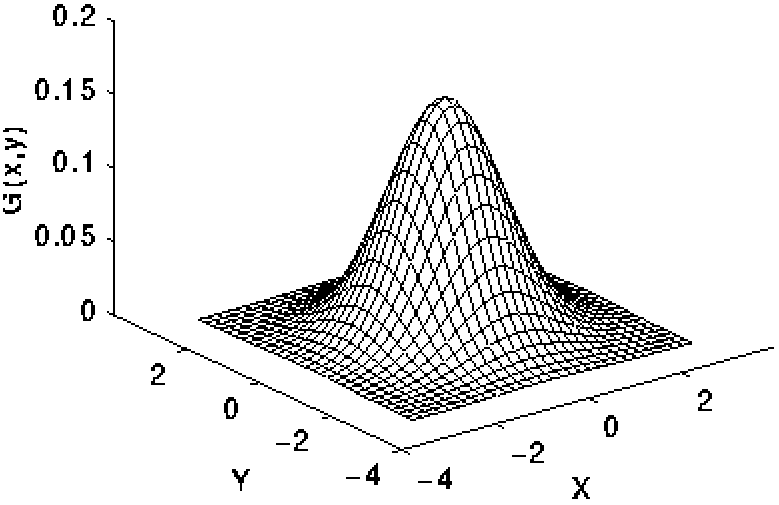

Figure 6.3: 2-D Gaussian distribution with mean (0,0) and =1

The idea of Gaussian smoothing is to use this 2-D distribution as a `point-spread' function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point. Figure shows a suitable integer-valued convolution kernel that approximates a Gaussian with a of 1.0. It is not obvious how to pick the values of the mask to approximate a Gaussian. One could use the value of the Gaussian at the centre of a pixel in the mask, but this is not accurate because the value of the Gaussian varies non-linearly across the pixel. We integrated the value of the Gaussian over the whole pixel (by summing the Gaussian at 0.001 increments). The integrals are not integers: we rescaled the array so that the corners had the value 1. Finally, the 273 is the sum of all the values in the mask.

A further way to compute a Gaussian smoothing with a large standard deviation is to convolve an image several times with a smaller Gaussian. While this is computationally complex, it can have applicability if the processing is carried out using a hardware pipeline. The Gaussian filter not only has utility in engineering applications. It is also attracting attention from computational biologists because it has been attributed with some amount of biological plausibility, e.g. some cells in the visual pathways of the brain often have an approximately Gaussian response.

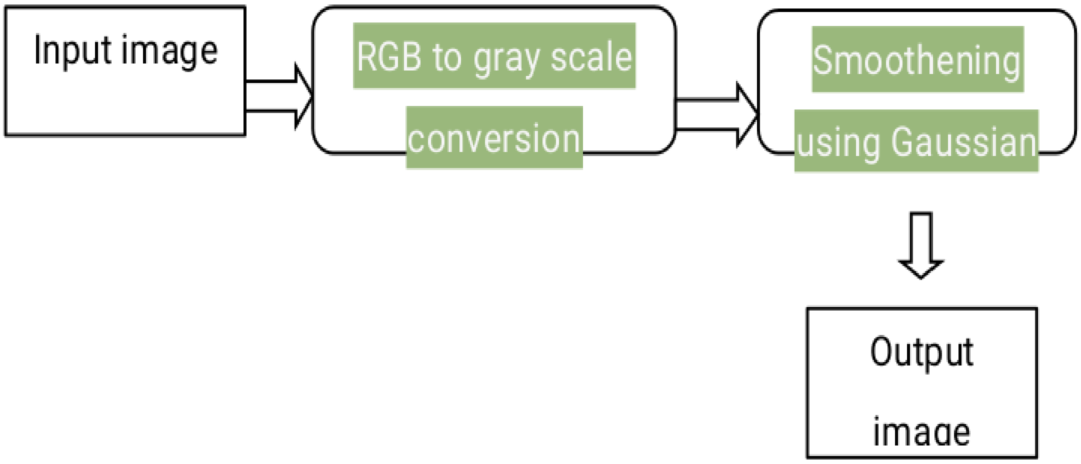

Figure 6.4: Flow diagram of image filtering

2 Canny edge detection

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations.

The general criteria for edge detection includes:

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible.
2. The edge point detected from the operator should accurately localiz on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection.
4. Apply double threshold to determine potential edges.
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

So far, the strong edge pixels should certainly be involved in the final edge image, as they are extracted from the true edges in the image. However, there will be some debate on the weak edge pixels, as these pixels can either be extracted from the true edge, or the noise/color variations. To achieve an accurate result, the weak edges caused by the latter reasons should be removed. Usually a weak edge pixel caused from true edges will be connected to a strong edge pixel while noise responses are unconnected. To track the edge connection, blob analysis is applied by looking at a weak edge pixel and its 8-connected neighbourhood pixels. As long as there is one strong edge pixel that is involved in the blob, that weak edge point can be identified as one that should be preserved.

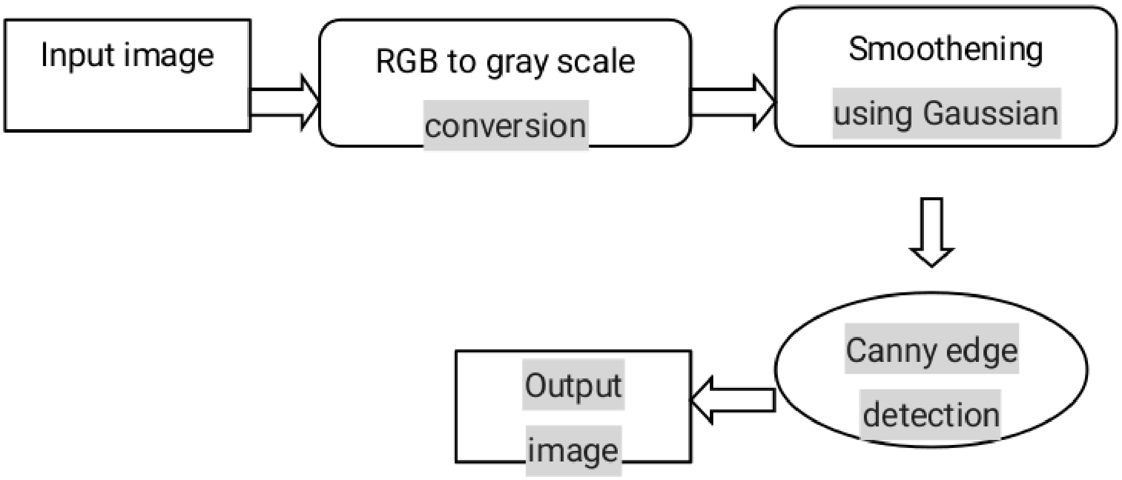

Figure6.5: Flow diagram of edge detection

3. face recognition and blurring

Face detection has been regarded as the most complex and challenging problem in the field of computer vision, due to the large intra-class variations caused by the changes in facial appearance, lighting, and expression. Such variations result in the face distribution to be highly nonlinear and complex in any space which is linear to the original image space. Moreover, in the applications of real life surveillance and biometric, the camera limitations and pose variations make the distribution of human faces in feature space more dispersed and complicated than that of frontal faces. It further complicates the problem of robust face detection. Paul Viola and Michael Jones presented a fast and robust method for face detection . The problem to be solved is detection of faces in an image. A human can do this easily, but a computer needs precise instructions and constraints. To make the task more manageable, Viola–Jones requires full view frontal upright faces. Thus in order to be detected, the entire face must point towards the camera and should not be tilted to either side. While it seems these constraints could diminish the algorithm's utility somewhat, because the detection step is most often followed by a recognition step, in practice these limits on pose are quite acceptable.detector=visionCascadeObjectDetector creates a System object, detector, that detects objects using the Viola-Jones algorithm. The ClassificationModel property controls the type of object to detect. By default, the detector is configured to detect faces.

detector=vision.CascadeObjectDetector(MODEL) creates a System object,detector , configured to detect objects defined by the input character vector, MODEL . TheMODEL input describes the type of object to detect. There are several valid MODEL character vectors, such a ‘FrontalFaccCART', ‘UpperBody’, and ‘.ProfileFace’ See the ClassificationModel property description for a full list of available models.

Creates detector = vision.CascadeObject Detector(XMLFILE) a System object,

Detector detector= vision.CascadeObjectDetector(Name,Value) configures the cascade object detector object properties. You specify these properties as one or more name-value pair arguments. Unspecified properties have default values.

To detect a feature: Define and set up your cascade object detector using the constructor. Call the step method with the input image, I, the cascade object detector object, detector ,points PTS, and any optional properties. See the syntax below for using the step method. Use the step syntax with input image, I, the selected Cascade object detector object, and any optional properties to perform detection. BBox = step(detector,1)returns BBox , an M -by-4 matrix defining M bounding boxes containing the detected objects. This method performs multiscale object detection on the input image, I. Each row of the output matrix, BBox , contains a four-element vector, [x y width height], that specifies in pixels, the upper-left corner and size of a bounding box. The input image I, must be a grayscale or truecolor (RGB) image. Blurring is an optical feature of image.It makes something to be unclear to observer.It can be bothersome to human, as it provides difficulty for human to focus on the features of an image.It can also create certain image effect for a better performance.

Blurring image is a technique involved in image processing.It is used in some conditions where a portion of the image has to be blurred to produce a better result.It is a trending web design technique as it can perfect the features of a website depending on how one can wholly utilize this technique for project. There are some methods that can be used to blur an image. Images have spatial domain and frequency domain where blurring can be performed on. Spatial domain is the domain where the space reveals the real image that can be seen, whereas the frequency domain displays the frequency of each pixel values that are present in the image. There are factors that affect the blurring effect, one of them is the filter kennel.

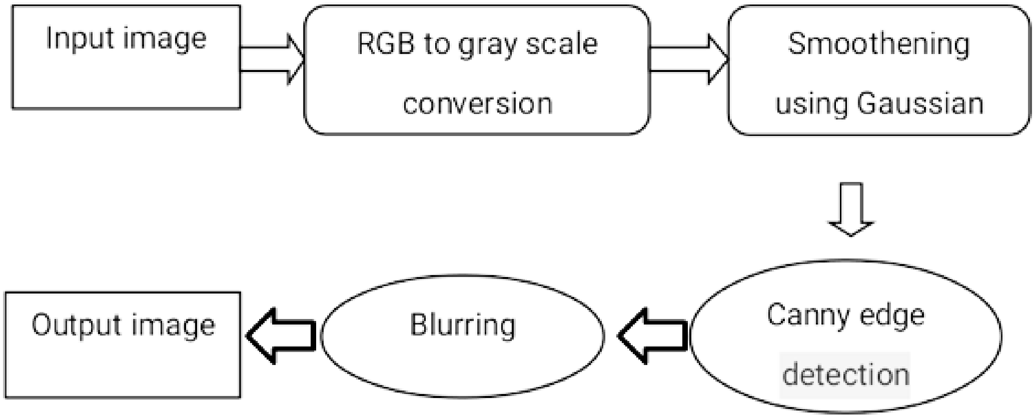

Figure 6.6: Flow diagram of face recognition and blurring.

CHAPTER 7

RESULTS OF IMAGE PRIVACY PROTECTION

1.Filtering an image

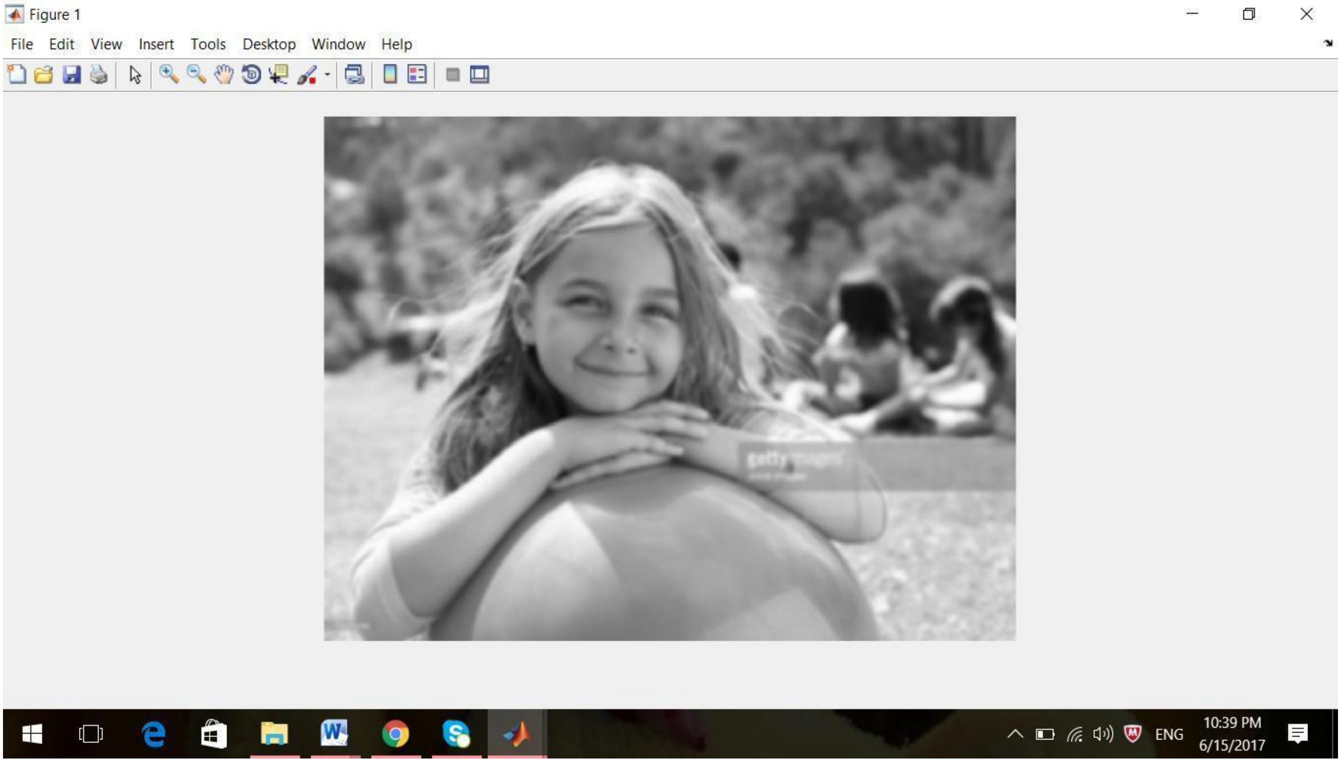

Figure 7.1: Fitering an image

Filtering an image is basically done for smoothing, sharpening, or enhancing an image. The Gaussian filter is a non-uniform low pass filter. This is the common firststep in edge detection.

2.Canny edge detector

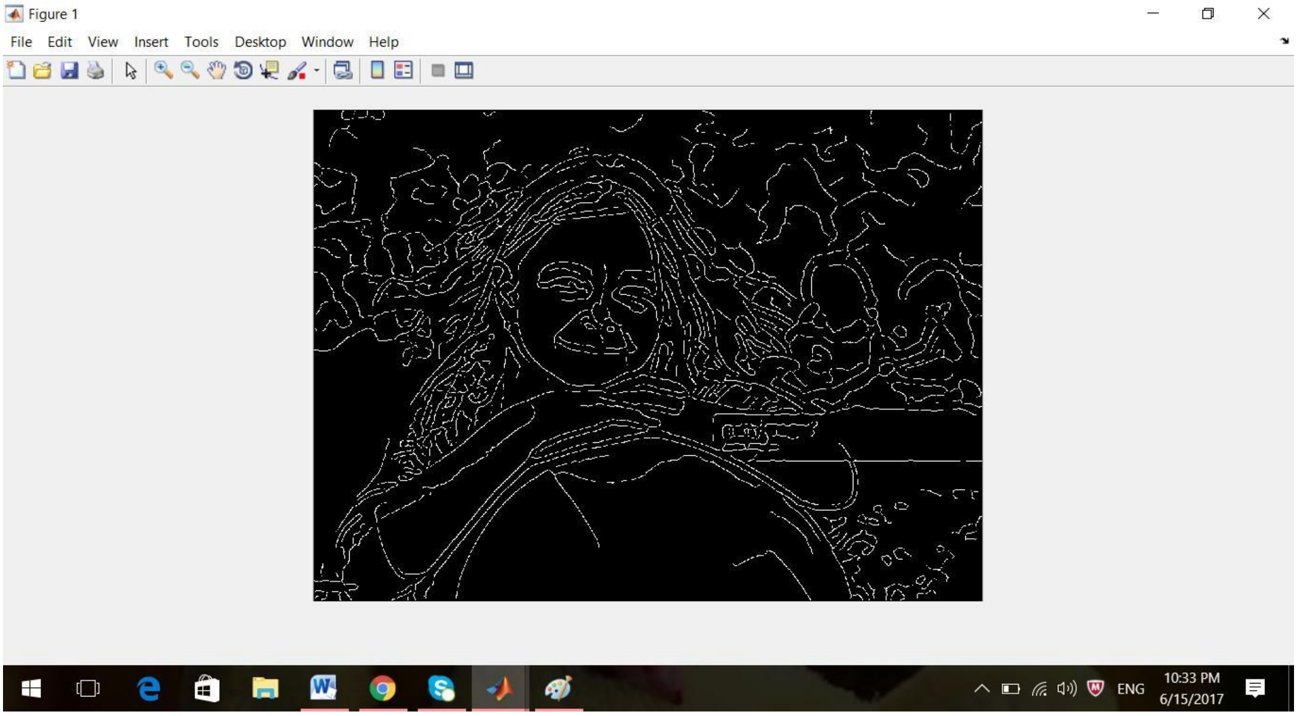

Figure 7.2: edge detecting

The canny edge detector is an edge detection operator that uses a multi stage algorithm to detect a wide range of edges in images. A given edge in the image should only be marked once, and where possible, image noise should not create false edges. Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. All edge detection results are easily affected by image noise, it is essential to filter out the noise to prevent false detection caused by noise.

3. Face recognition and blurring

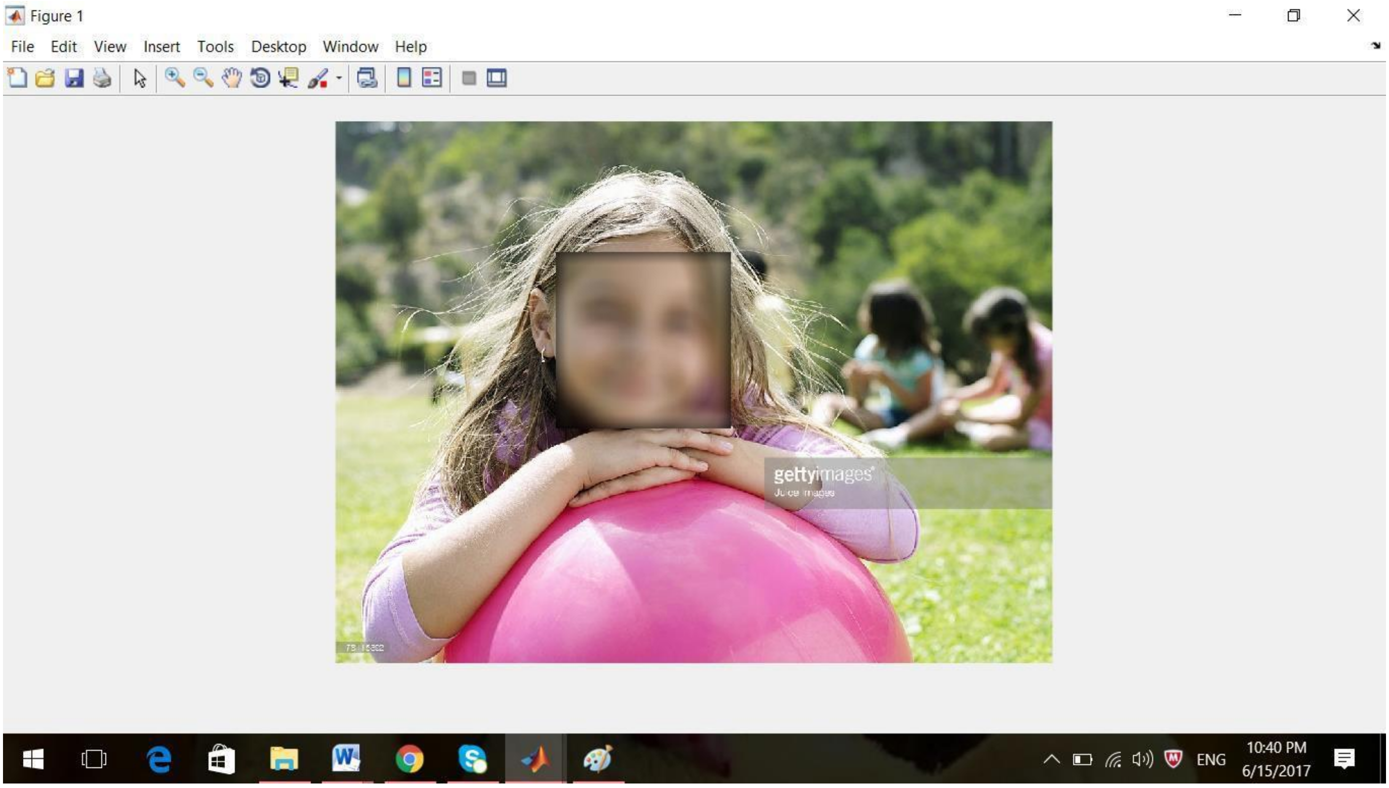

Figure 7.3: Face recognition and blurring

The basic principle of the Viola-Jones algorithm is to scan a sub-window capable of detecting faces across a given input image. Rescale the input image to different sizes and then run the fixed size detector through these images. Detection is the first step in the recognition process.

CHAPTER 8

ADVANTAGES

- It helps to detect objects in an image.
- Automatically detects the human images.
- It Protect human faces while image sharing by blurring it.
- It helps to reduce the misuse of images.
- It automates the privacy setting process in an image.
- Establish more privacy on social image sharing.

CHAPTER 9

APPLICATIONS

- It provide effective and secure image sharing in social medias.
- It provides secure image sharing.
- It helps to provide the privacy to the human images.

CHAPTER 10

LIMITATIONS

- The rising popularity of social networks, it’s little surprise that there have been several high-profile breaches of security on sites as huge as MySpace and Facebook.
- It lead to fraud and online theft -Loss of privacy in the misuse of human images.

CONCLUSION

The system is using filters and edge detectors to detect the edges of image. Image sharing in social medias gained wide popularity. Due to this misuse of images are happening in social medias nowadays. The research in the paper adopted an approach iPrivacy which helps to protect human in an image while sharing. Canny edge detectors and viola jones are used to detect the edges and to blur the image. The basic advantage is to reduce the image, especially humans in image while image sharing . Indeed, experimental results confirmed the effectiveness of the proposed method: as compared with other popular methods.

This technique is used to automatically find the privacy sensitive objects and provide privacy for them. The more privacy sensitive object is a human. Here we detect human in an image through canny edge detection. If a human image is accessed by any other person who has no permission to access the image then the face of human in that image already become blurred. So we can avoid the misuse of human images by this technique.

REFERENCES

1. C. Wang, B. Zhang, K. Ren, J.M. Roveda, “Privacy-assured outsourcing of image reconstruction service in cloud”, IEEE Trans. on Emerging Topics in Computing.
2. A. C. Squicciarini, H. Xu, X. Zhang, “CoPE: Enabling collaborative privacy management in online social networks”, J. of American Society for Information Science and Technology.
3. N. Wang, H. Xu, J. Grossklags, “Third-party Apps on Facebook: Privacy and the illusion of control”.
4. M. Choudhury, H. Sundaram, Y.-R. Lin, A. John, D. Seligmann,“Connecting content to community in social media via image content, user tags and user communication”, IEEE ICME.
5. C. Yeung, L. Kagal, N. Gibbins, N. Shadbolt, “Providing access control to online albums based on tags and linked data”, AAAI Symposium.

6 J. Pesce, D. Casas, G. Rauder, V. Almeida, “Privacy attacks in social media using photo tagging networks: A case study with facebook”, ACM PSOSM.

7. D. Christin, P. Lopez, A. Reinhardt, M. Hollick, M. Kauer, “Sharing with strangers: Privacy bubbles as user-centered privacy control for mobile content sharing applications”, Information Security Technical Report.
8. R. Ravichandran, M. Benisch, P. Kelley, N. Sadeh, “Capturing social networking privacy preferences”.
9. J. Bonneau, J. Anderson, L. Church, “Privacy suites: shared privacy for social networks”.
10. J. Bonneau, J. Anderson, G. Danezis, “Prying data out of a social network”.
11. T. Evgeniou, C.A. Micchelli, M. Pontil, “Learning multiple tasks with kernel methods”, *Journal of Machine Learning Research.*

12. H. Fei, J. Huan, Structured feature selection and task relationship inference for multi-task learning, IEEE ICDM.

13. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks.

14. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A deep convolutional activation feature for generic visual recognition.