Annals of R.S.C.B., ISSN:1583-6258, Vol. 24, Issue 2, 2020, Pages. 1854 - 1863
Received 20 October 2020; Accepted 04 December 2020.

Computer Vision Project Report

Dr. K.Veena,

Associate Professor, Department of Information Technology, Dhanalakshmi Srinivasan College of Engineering and Technology

Dr. E. Gajendiran,

Asso. Professor, Dhanalakshmi Srinivasan College of Engineering and Technology

ABSTRACT

A person with visual impairment is working on his/her computer an attacker can shoulder surf the person and possibly acquire sensitive information. Inability to judge the environment can also raise safety concerns. When a visually impaired person is in a unsafe neighborhood he/she cannot tell if someone is following them . Several researchers has already worked on the face detection problems. The most famous work in this area was conducted by Viola and Jones . They used Haar features and Adaboost algorithm to detect faces and their algorithm gives pretty good results on face detection. We used dlib libraries in this project which uses HOG features.

1. INTRODUCTION

Nowadays, visually impaired people can live their life more independently with the help of several assistive devices and technology. Even though, computing devices like OrCam1 , Victor Reader Stream, CTC Scanner, SignatureGuide and other accessible devices addressed many of the accessibility and mobility concerns of people with visual impairment in the physical world, the physical privacy and security concerns remain largely unaddressed. In a recent work which is conducted by our group[1], reported the privacy and security needs of visually impaired people and discussed several privacy concerns of visually impaired people in the physical world including eavesdropping and shoulder surfing. When you person with visual impairment is using a computing device in a public place or having a personal conversation with someone, the person might not have any idea about their surroundings and can fall as a victim of shoulder surfing. Fig

Received 20 October 2020; Accepted 04 December 2020.

1a depicts one such problem of visually impaired people, whenever a person with visual impairment is working on his/her computer an attacker can shoulder surf the person and possibly acquire sensitive information. Inability to judge the environment can also raise safety concerns

In both of the previous studies, we presented some high level ideas that can potentially address the privacy and safety concerns of people with visual impairments using wearable cameras and computer vision approaches. In the second study, we got feedback from visually impaired population and reported the ways a camera based technologies can help [2]. Our participants expressed the necessity of a technology that can inform the number of nearby people and peoples' proximity would be the most useful information to manage their privacy and safety. In this project, we decided to explore this problem by using computer vision approaches.

2. MATERIAL AND METHODOLOGY

In this paper, we used well established face detection algorithms to count the number of people in the image, which works really well in practice. As the detection problem is already a solved problem, we did not investigated this problem in depth; we investigated couple of libraries and mostly used this information to answer the distance problem. On the other hand, the distance problem is a really difficult problem and we were surprised to see that there has been a nominal work in this area and requires major investigation. They used image from 53 individuals in seven distances and tried to estimate their distance using facial landmarks. They used a linear regression for this problem and found 75% correlation between the ground-truth distance and their estimated distance. However, their goal was to estimate the feasibility as the acknowledge the data set is pretty small. Moreover, their data set is extremely limited and may not work well with other images. In this project, we used their data set and trained a model using Support Vector Machine. Our approach seemed work well as it was able to achieve 42 % accuracy on the classification task. Our work also suggests the feasibility of the study, however, one major limitation with this data set is the size.

Estimating the distance of a person from camera is a really difficult problem to solve. Therefore, we don't expect complete solution, our goal is to explore one variant of this problem. We were surprised to see that there is no good data set which considered this problem, most of the researchers worked on a relatively small data set. We found a moderate data set as a starting point and the limited data set is one of the biggest limitation of our approach. The details of the data set is described in Section 3.1. Figure 2 depicts our complete architecture. Our approach has three important parts: 1) Data Set, 2) Face Detection and Feature Extraction, and 3) Classification. The details of each parts is described below:

2.1 CMDP Dataset

After exploring several data sets, we decided to select the data set which is previously explored by Burgos-Artizzu . They created Caltech Multi-Distance Portraits (CMDP) data set for the distance estimation problem, which consists of the frontal portraits of 53 individuals against a blue background imaged from seven distances spanning the typical range of distances between camera and subject: 2, 3, 4, 6, 8, 12, 16 ft (60, 90, 120, 180, 240, 360, 480 cm)

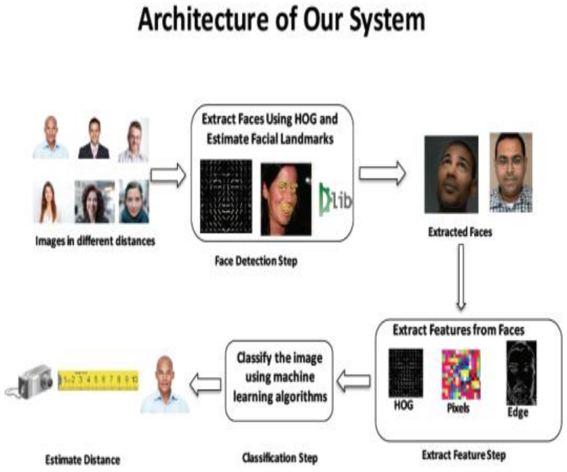

Architecture of Our System
Images in different distances
Extract Faces Using HOG and Estimate Facial Landmarks
dlib
Face Detection Step
Extracted Faces
Extract Features from Faces
HOG
Pixels
Edge
Extract Feature Step
Classify the image using machine learning algorithms
Classification Step
Estimate Distance

Figure 2: Architecture of our system

2.2 Our Data set

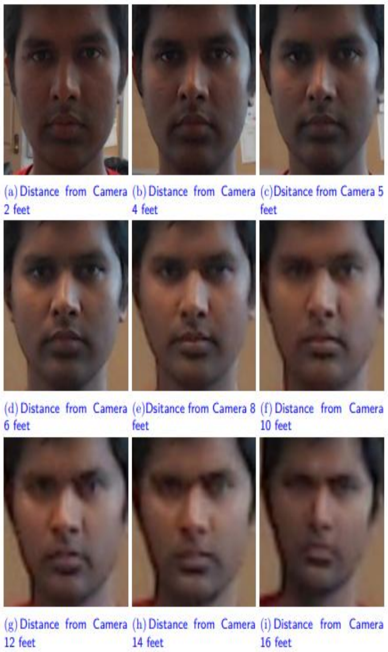
For the purpose of testing, we created our own data set. We knew from the beginning that we will not be able to collect much data, therefore, we decided to collect our data set for the testing purpose. We collected images from seven individuals in 10 possible distances (Two feet to 16 feet with 2 feet apart. Also we have collected images in five feet and 10 feet). We intentionally collected in more distances, so that we can understand how it works with unlearned

Received 20 October 2020; Accepted 04 December 2020.

distances. Figure 2.2 shows some examples from our data set. We collected our data set by using two approaches:

1) We manually labeled different distances and captured the image on those distances

2) We also used Kinect to get the depth map and we estimated the distance from the depth map. However, the kinect data set did not help us as the resolution of the images were really small (640 × 480).



(a) Distance from Camera 2 feet

(b) Distance from Camera 4 feet

(c)Dsitance from Camera 5 feet

(d) Distance from Camera 6 feet

(e)Dsitance from Camera 8 feet

(f) Distance from Camera 10 feet

(g) Distance from Camera 12 feet

(h) Distance from Camera 14 feet

(i) Distance from Camera 16 feet

Fig.2.2.Our data set

2.3 People Detection And Distance Estimation Steps

Detecting peoples in an image and then estimating their distances includes several steps. As a first step we detect all the faces, which is equivalent to finding how many people are nearby. Then we separate the face chips and extract several types of features from them. Below we briefly describe each step.

A.Face Detection

The first step involves finding all the faces in a captured image. For this we are using HOG features in a sliding window scheme, which is implemented in dlib Note that this implementation can only find frontal and profile faces. Due to the unavailability of any existing algorithm to find out faces of people who are looking at an angle more than ninety degree from the camera, as well as any existing dataset of such faces, our system in its current state cannot detect all nearby people.

B. Face Extraction

We extract all the detected faces as separate image, and re-size them to a standard size because faces appearing nearby look bigger than distant faces. This step is necessary because the classifier we use to detect pose expects same number of features for each image. The exact resizing scale depends on the feature we use for classification.

C. Detect Facial Landmarks

Next step is to use the standardized face images to estimate the facial landmarks. For this we are using the algorithm proposed by Kazem [14] which is already implemented in dlib. This algorithm can detect sixty eight facial landmark points, such as location of nose, eyes etc.

D. Feature Extraction

We extract and use various types of features so that we can compare how well each one is performing. Figure 5 shows each type of features and below is a brief description of them:

- **HOG:** These features encode the orientation of differences in intensities across overlapping image patches.

- **Pixel:** As a baseline we used pixel data as features as well. First we converted the extracted face chips as gray scale images and resized them 100 × 100. The converted image pixels were used as features for your classifier.

- **Edge:** Edges correspond to sharp changes in image intensity. We used opencv to extract edges using Canny edge detector . Note that we resized the extracted face images as a 100 × 100 pixel image only for edge feature.

- **All pair distance:** We calculated Euclidean distance between every pair of facial landmark points identified in the landmark detection step. This results in a much shorter dimensional feature vector (2278 dimensional) as opposed to 10000 dimensional vector produced for edge features

- **Selected pair distance**: To further reduce the feature vector dimension, we used only 20 pairs of landmark points and created shorter feature vector. This results in dramatic improvement in terms of processing time while loading model file and classifying at run time by the SVM classifier we are using

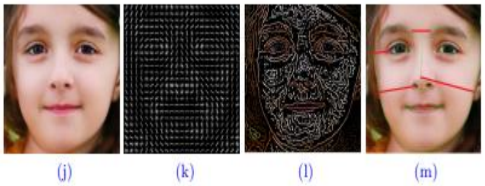


Fig.2.3. Features extracted from an image in our system. (j) shows an extracted face, and (k) , (l), and (m) show the extracted HOG, Edge, and Distance between Landmark Point pairs features respectively from the original image. For (m) the green lines connect facial landmark points, and the red lines denote distances between pair of such points.

2.4 Classification

Using the extracted features we used machine learning methods for estimating distance. For this particular problem, we divided the whole data set in three classes:

Class 1: The distance is less than 5 feet

Class 2: The distance is more than 5 feet but less than 10 feet

Class 3: The distance is more than 10 feet.

We divided the distance in such way for two reasons:

1) It will give us more data in each of the classes

2) Visually impaired people wants to know if someone is really close to them or not, these classes can easily give that answer.

We used Support Vector Machine to train the images. We adopted two testing approaches:

1) We divided the CMDP data set intor training (75% images, 40 people in the training image) and test set (25% images, 13 people in the test images)

2) We trained the SVM using the CMDP data set and tested on our data set.

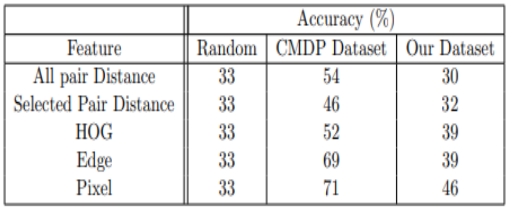
We adopted the second approach to understand how the classfication works in a completely different data set. If this works well, then it proves that the classifier is not overfitting.

3. RESULTS AND DISCUSSIONS

Our experiment yielded unexpected results. We were able to achieve around 70% accuracy for the CMDP data set, where the random accuracy is 33%. When we tested with our data set, the accuracy is 46% which is also better than the random accuracy. this proves that our algorithm can be improved with more data.

3.1 Results on Three Classes

We used several features to train the data set. The accuracy varied for different features. Surprisingly, the pixel worked for this data set even they worked better than Hog and Haar Features



	Accuracy (%)		
Feature	Random	CMDP Dataset	Our Dataset
All pair Distance	33	54	30
Selected Pair Distance	33	46	32
HOG	33	52	39
Edge	33	69	39
Pixel	33	71	46

Table 3.1 Accuracy on test tests

Although, we trained our model on a different data set, still the classifier performs better than

Annals of R.S.C.B., ISSN:1583-6258, Vol. 24, Issue 2, 2020, Pages. 1854 - 1863
Received 20 October 2020; Accepted 04 December 2020.

random accuracy on a completely different data set. Results on all seven distances We investigated further to understand the classification errors. We wanted to understand the reason behind the poor accuracy. Overall, for all seven classes we were able to achieve around 42% accuracy (Random 14%). We tried with different image sizes and saw that for pixel features the accuracy varies depending on the image size. Table 2 shows the accuracy for different image sizes and we can see that there is a positive correlation between the image size and the accuracy. This correlation is understandable as with the increasing number of image size, the number of pixels also increasing which may be giving better features

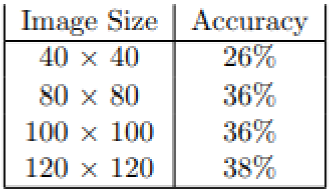


Image Size	Accuracy
40 × 40	26%
80 × 80	36%
100 × 100	36%
120 × 120	38%

Table 3.2 Accuracy depending on the image sizes

In this paper, we think we took the initial step really well. A very constrained data set shows promise. Although, the data set was really noisy the classifier is working surprisingly well on both of the data sets. The most surprising result is that although we are training in a completely different data set, the model is working well in a completely different data set. This gives us confidence that this approach may work well in real life. The benefit of our approach is the system can estimate distance really quickly once we have learnt the model. Therefore, this system can detect faces and estimate their distance in run time. If we can think about an app which can give the number of people nearby and their proximity really quickly that may help them to manage their privacy in a really good way

To estimate the distance, we only have considered faces which may give poor results in the real life scenario. The accuracy can be improved if we consider other body parts such as shoulder, hand, neck or the overall upper parts of the body. This other parts can be classified independently and then can be combined. This will give more confidence and can reduce the False Positive rate. We also found that the data set is noisy. The distance of some people is not consistent or not distinguishable. This can be another reason for which our data is not performing well. We anticipate that with a good data set, the algorithm may perform better.

4. CONCLUSION AND FUTURE WORK

In this work, we have seen that it is feasible to estimate the distance of a person from camera to some extent. So far, most of the researchers are trying to estimate the distance either by using stero or by using facial features. To our knowledge, none of the researchers have tried the data driven approach yet. Our approach shows promise and with more data it might be possible to improve the accuracy which will not only help visually impaired people, but also it has significant impact on the field of robotics, security and forensics.We have already acknowledged that one of the major limitation of our work is we did not have a good data set. If we had enough data, and a robust data set then deep learning might have given a good result. We did not try deep learning as the data set was relatively small. Another limitation of this data set is it only considers frontal images. But, in real life people can shoulder surf in different orientations such as by looking at left or right. In future, we have a plan that we would like to collect data in different orientation and then we will try to estimate their distance.

To estimate the distance, we only have considered faces which may give poor results in the real life scenario. The accuracy can be improved if we consider other body parts such as shouldler, hand, neck or the overall upper parts of the body. This other parts can be classified independently and then can be combined. This will give more confidence and can reduce the False Positive rate. We also found that the data set is noisy. The distance of some people is not consistent or not distinguishable. This can be another reason for which our data is not performing well. We anticipate that with a good data set, the algorithm may perform better.

REFERENCES

[1] Tousif Ahmed, Roberto Hoyle, Kay Connelly, David Crandall, and Apu Kapadia. Privacy concerns and behaviors of people with visual impairments. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI ’15, pages 3523–3532, New York, NY, USA, 2015. ACM.

[2] Tousif Ahmed, Patrick Schaffer, Kay Connelly, David Crandall, and Apu Kapadia. Privacy

concerns and behaviors of people with visual impairments. In Submitted to Symposium on Usable Privacy and Security, SOUPS '16, 2016.

[3] Xavier P. Burgos-Artizzu, Matteo Ruggero Ronchi, and Pietro Perona. Distance estimation of an unknown person from a portrait. In Computer Vision–ECCV 2014, pages 313–327. Springer, 2014.

[4] Robert Templeman, Roberto Hoyle, Apu Kapadia, and David Crandall. Reactive security: Responding to visual stimuli from wearable cameras. In Proceedings of the Workshop on Usable Privacy & Security for wearable and domestic ubIquitous DEvices (UPSIDE '14), pages 1297–1306, September 2014.

[5] Robert Templeman, Mohammed Korayem, David Crandall, and Apu Kapadia. PlaceAvoider: Steering first-person cameras away from sensitive spaces. In Proceedings of The 21st Annual Network and Distributed System Security Symposium (NDSS), February 2014.

[6] Mohammed Korayem, Robert Templeman, Dennis Chen, David Crandall, and Apu Kapadia. Enhancing lifelogging privacy by detecting screens. In ACM CHI Proceedings of the Conference on Human Factors in Computing Systems (CHI), 2016.

[7] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. Vizwiz: Nearly real-time answers to visual questions. In Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology, UIST '10, pages 333–342, New York, NY, USA, 2010. ACM.