**Abstract:** Deep Learning is known as the most powerful technique for handling almost Computer Vision problems. Many state-of-the-art methods in face recognition focus on recognition from still images. However, tackling still images are challenged by uncontrolled conditions such as head pose, motion blur, occlusion. Nevertheless, in the context of camera surveillance system, faces appear in a series of frames containing more unemployed information. In this paper, we propose a method of combining face tracking and face representation for still images to increase high accuracy rate in the setting of camera surveillance. We evaluated different set comparison techniques on video surveillance dataset (ChokePoint), we observe that our combination method accuracy is 98.87% significantly improvement to only using face representation for still images accuracy is 90.25%. We also collect dataset which is more challenges such as far distance, angle view of camera high and achieve accuracy is 71.96% much higher to only using face representations is 60.55%

**Keywords:** face recognition, surveillance, face tracking, deep learning

**I. INTRODUCTION**

In the rapidly developing of technology, Smart Systems are playing the most crucial role to human beings. It reduces much more workforce, labors, manual tasks and increases high efficiency at work. Nonetheless, there is still being some of the hard stuff that is still needing human to intervene. Hence, researching and developing new high technology are always the challenges and opportunities, developing face recognition system in camera surveillance is one of the most fundamental, therefore, we can gain much more benefits from this system such as checking attendance, security, etc. However, recognizing in surveillance environment challenges by many free-conditional cases occlusion, head pose, motion blur, for instance, that causes recognition accuracy low.

Many methods of face recognition with deep features are used in recent years and make a significant impact result such that SphereFace (CVPR-2017), the state-of-the-art of face recognition, tested on Label Face in the Wild (LFW) and Youtube Faces (YTF). In SphereFace, the network architectures use the residual unit. However, it is quite different from standard ResNets such as the PreLu replaces ReLu and BatchNorm is not used, etc. We use extract deep features from the pre-trained model of SphereFace that trained on CASIA-WebFace and tested on LFW with 20-layer CNN, to apply face recognition on our camera surveillance dataset. After the experiment, the accuracy of face recognition is low.

However, in developing we find out the method to improve and build an end to end face recognition system in camera surveillance. Normally, in camera surveillance, we do face recognition task on each frame, the faces existing in each frame are in various cases, some of them are lacking face details due to head pose or motion blur. Therefore, to improve the loss of face details, we utilize the series frames of a face by using face tracking to augment more face details. By combination face recognition SphereFace and face tracking with Simple online and real-time tracking (SORT) [2] method, it resolves some problematic circumstances motion blur (causing by fast motion), head pose, shortage of facial detail while recognizing, for example. With face tracking, we can gain various face detail of persons in every frame and contain the set for each detected face that improves quality of recognition. Moreover, challenges of motion blur, occlusion can make the loss of information and effect to result of recognition, but with face tracking, we can avoid such bad conditions and loss information. We also propose set comparison applied in recognizing to give out the similarity measurement.

In this work, we experiment on comparison method of non-tracking (only face recognition with SphereFace) and combination with tracking method (face tracking with SORT and face recognition with SphereFace) on ChokePoint [4] dataset and achieving 98.87% significantly higher than non-tracking 90.25%. Otherwise, we collect new surveillance video dataset which is more challenges than ChokePoint such as uncontrolled conditions angle view of camera high causing face not frontal, far-distance 4-5 meters, 30 characters which gallery images of each character are 3-6 images fewer than ChokePoint which is 50-60 images for each character. After the experiment on our dataset, the recognition rate of face tracking and SphereFace is 71.96% higher to only using SphereFace is 60.55%
II. PROPOSED SYSTEM

First of all, we present an overview of our combination system. In figure 1, we have 4 main components face detection to detect facial landmarks and bounding boxes, face tracking to track frame by frame of face and accumulate into the list face frame for each individual, to do face recognition we use the pre-trained model of SphereFace method to extract deep features and pass to similarity measurement component to pick label.

By using face tracking, we can employ series frames of different face poses to increase more information in face recognition task.

2.1. Face Detection with Multi-task Cascaded Convolutional Networks (MTCNN)

Convolution neural networks (CNN) are the deep artificial neural networks (ANN) that are used primarily for image classification and face recognition. Some of CNN using for face detection approaches have been famous in recent year. However, the intricate architecture CNN requires more time in practice. The one of best performance that is Multi-task Cascaded Convolution Networks (MTCNN) [1] which using three model of deep convolution networks for three stages and using cascade to increase. They propose the new cascaded CNN is initialized for joint face detection and alignment that make the CNN architecture for the real-time process. The first state, it creates candidate window through a CNN. After that, the other CNN uses for rejecting a vast Number of Non-faces which pass for next CNN to refine the result and return the landmarks position attach by two points of the bounding box.

2.2. Face Tracking with Simple Online and Real-time Tracking (SORT)

The main problem of face recognition in the camera surveillance is how to real-time track the multiple people in the sequence frames which are both results of detection and recognition operate in parallel. In the scene of the camera surveillance. The method of tracking multiple people is very essential to increase the accuracy of system recognition by the statistics results of recognizing the object at sequence face frames to get the final result. Some traditionally multiple object tracking has been solved using Multiple Hypothesis Tracking [7] or the Joint Probabilistic Data Association filters [8], which have the complexity and processing time increase by the number objects in the video. Among them, the current state of the art is Simple Online and Real-time Tracking (SORT) [2] which combines two simple methods Kalman filter [9] and Hungary Algorithm [10] but it is 20x faster than other state-of-the-art trackers, and provide the solution for online tracking multiple objects. Each state of the object has parameters to describe the motion that was used for link with next frames to track the object. The state was defined:
Where: \( u \) and \( v \) are the horizontal and vertical pixel location which is the center of the object.

\[
\begin{align*}
\text{s} & \text{ is the scale just area: } s = w \times h \\
\text{r} & \text{ is the aspect ratio of the target’s bounding box respectively: } r = \frac{w}{h}
\end{align*}
\]  

The detector will be returned two results that are bounding box and landmark of objects. The bounding box \((x_1, y_1, x_2, y_2)\) will be converting \([u, v, s, r]\). The detector is used to detect the object, when the object enters or leave the video, the bounding box is used to update the state of the target where velocity components are solved by Kalman filter [9]. Each the bounding box in the current frame has limited the size and the location of the previous frames. Also, the intersection-over-union (IOU) that be used to compute the distance between bounding boxes of the current frame with all the target of previous frames. If the value distance (IOU is computed by current frames with predicted bounding boxes) less than ‘IOU_min’ (intersection-over-union). The bounding box is. After that Hungary algorithm [10] will be used to optimize.

Computing IOU can, therefore, be determined via:

\[
\text{IOU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}
\]  

When objects leave or enter the frames, it must to destroyed or create new the tracker. For creating trackers, ‘IOU_min’ is material value to detect the untrack object. All untrack objects have the IOU less than ‘IOU_min’ and untrack object would keep track after few frames to make sure that is the new objects.

For the camera surveillance, the accuracy of SORT method depends on the detection and number of frames which is detection can execute per second. This means that not also detection captures the objects successfully but only limited time for detection is very necessary.

2.3. SphereFace

In this section, we present the idea and implementation of SphereFace [3] used in face recognition. We use available SphereFace source code, and we apply the pre-trained model of SphereFace then extract deep feature and then compute cosine similarity to measure the similarity distance.

![Figure 3. Process of training and testing of SphereFace [3]](image)

The pre-trained model train on network architectures using residual units (ResNets) as the building blocks, however, it has different from standard ResNets such as the PreLu replaces the ReLu, BatchNorm is not used, etc. We use 20-layer CNN architecture (proposed in SphereFace[3]). The pre-trained model is trained on CASIA-WebFace and tested on Label Face in the Wild (LFW). In this paper, we use the pre-trained model to extract deep feature of fully connected layer (FC1). The feature vector size is 512.

![Figure 4. CNN architectures used in SphereFace [3]](image)
In this work, after we extract deep feature from given image, we calculate the cosine distance to measure that which subject is the most similar to the given image. With $f_1$, $f_2$ are feature vectors.

$$cosine\ distance(f_1, f_2) = \frac{f_1 \cdot f_2}{||f_1|| ||f_2||}$$  \hspace{1cm} (3)

2.4. Similarity Measurement with Face Tracking.

After tracking the face of the characters, we collect and extract features from each frame and get a collection of set features. We will compare this set features with gallery features (gallery images of each character extracted) and select the most similar. We select and measure by using two methods: Max-Max, Max-Average.

Max-Max Method

![Figure 5. Similarity Measurement with Max-Max](image)

In the above picture, by using face tracking, we gain a set of face images and extract the whole of them into the feature. Each feature stands for a point in the set. Test Set is the set of test features; Gallery Set is the set of characters existing in the gallery. Here is the formula describe the idea.

$$(i^*, j^*, p^*) = \text{argmax}_{i \in [1 \ldots n]} \cos(T_i, G_{p,j})$$ \hspace{1cm} (4)

Which:  
$T$: features of Test Set  
$G$: features of Gallery Set  
n: number features in Test Set  
m_p: number features in id_p Set  
k: number of person id in Gallery Set

By computing cosine distance between each point in Test Set and Gallery Set. We take the maximum cosine distance value, which is used for picking label of the detected face.

Max-Average Method

The main idea of this section is same as Max-Max Method but instead of finding the max of max, we calculate average feature of both Test Set and Gallery Set depending on set then we find maximum cosine distance of them.

After calculating the average feature for each subject, we have the list of cosine distance, each cosine distance stands for the similarity of the feature in Test Set and character in Gallery Set. We choose the maximum value to be the label of Test Set.

$$(p^*) = \text{argmax}_{p \in [1 \ldots k]} \left( \frac{1}{n} \sum_{i=1}^{n} T_i \cdot \frac{1}{m_p} \sum_{j=1}^{m_p} G_{p,j} \right)$$ \hspace{1cm} (5)

Which:  
$T$: features of Test Set
\[ G: \text{features of Gallery Set} \]
\[ n: \text{number features in Test Set} \]
\[ m_p: \text{number features in } \text{id}_p\text{Set} \]
\[ k: \text{number of persons id in Gallery Set} \]

**Figure 6.** Similarity Measurement with Max-Average

### 2.5. Framework

In this section, we combine all modules and show whole diagram of step by step of face tracking and face recognition.

**Figure 7.** Extract facial features

In Figure 7, we pass the image into MTCNN module to extract facial landmarks. With the facial landmarks and image are passed to face alignment to align face and extract deep feature with the pre-trained model.

In Figure 8, firstly we extract all images of the gallery into features and save into database. With the test image, we also extract feature and calculate the similarity of all gallery features (which saved in the database) to pick label.

**Figure 8.** Process of Face Recognition
III. EXPERIMENT

3.1. ChokePoint Dataset

We evaluate the method on ChokePoint Dataset [4], designed for face identification under the real-world surveillance camera. Three cameras are placed above several portals and record persons who are walking through the portal. The dataset consists of 25 subjects (19 male, 6 female) for portal 1 and 30 characters for portal 2, each image of the character is 96x96 pixels cropped, and alignment face and total images of a character is about 50-60 images. Each image in the dataset is 800x600 pixels. Totally, 48 video sequences and 64,204 face images.

We tested on two portals, in each portal has three cameras placed at three positions with various background.

We compare two methods: Non-Tracking and Tracking:

- **Non-Tracking**: Only use face recognition with SphereFace method.
- **Tracking**: Using face tracking with SORT and face recognition with SphereFace. We track and collect multiple frames starting from appearing face of persons until disappearing.

To evaluate the method, we test on two sub-set which are two portals. The results are listed in below tables.

<table>
<thead>
<tr>
<th>Method</th>
<th>Non-Tracking</th>
<th>Tracking (Max-Max Method)</th>
<th>Tracking (Max-Average Method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portal 1</td>
<td>96.11%</td>
<td><strong>99.10%</strong></td>
<td><strong>99.75%</strong></td>
</tr>
<tr>
<td>Portal 2</td>
<td>84.40%</td>
<td><strong>98.64%</strong></td>
<td><strong>99.86%</strong></td>
</tr>
<tr>
<td>Overall</td>
<td>90.25%</td>
<td><strong>98.87%</strong></td>
<td><strong>99.81%</strong></td>
</tr>
</tbody>
</table>

We calculate average result of both portal 1 and portal 2 and showing Overall result. The result with tracking method outperform to non-tracking.
3.2. Our dataset

In this section, we test on our dataset with more challenges. Our dataset has 30 characters, gallery images of each character are about 3-6 images much fewer than ChokePoint which each character has 50-60 images. Besides, we record on the camera surveillance placing on top of the laboratory, and we record 50 video clips which totals are 16000 frames of face images, angle view of the camera is high that makes a face is not frontal. Below the table is the description of our dataset.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Our dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subjects</td>
<td>30 characters (20 males, 10 females)</td>
</tr>
<tr>
<td>Images of each subjects</td>
<td>3-6 images</td>
</tr>
<tr>
<td>Resolution</td>
<td>1920x1080</td>
</tr>
<tr>
<td>Camera</td>
<td>1.0 Megapixels</td>
</tr>
<tr>
<td>Dataset</td>
<td>16000 frames</td>
</tr>
<tr>
<td>Angle view of camera</td>
<td>50-60 degrees</td>
</tr>
<tr>
<td>Distance</td>
<td>4-5 meters (from camera to front door)</td>
</tr>
</tbody>
</table>

Due to the various pose of the person moving that causes recognition rate reducing more. In below pictures are challenges due to motion blur, non-frontal state, far distance. Our method can resolve face in motion blur and non-frontal face depending on face tracking (collecting more information of face poses). However, face at far distance is still the challenge.

![Face in motion blur.](image1)

![Face in non-frontal state.](image2)

![Faces at far distance.](image3)

**Figure 10.** Faces in various conditions

<table>
<thead>
<tr>
<th>Method</th>
<th>Non-Tracking</th>
<th>Tracking (Max-Max Method)</th>
<th>Tracking (Max-Average Method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>60.55%</td>
<td>71.15%</td>
<td>71.96%</td>
</tr>
</tbody>
</table>

On this dataset, we recognize all of the persons exist in the videos. By the combination of SphereFace and face tracking (SORT) we get the average recognition rate for the whole dataset is 71.96%, but with only SphereFace the recognition rate is 60.55%

Accuracy rate decreases much due to the effect of far-distance causing the images are in low-resolution. The face tracking method helps in augmenting information of various face poses. In our method accuracy rate is significantly improved.
IV. CONCLUSION

In this work, we propose the method by the combination of face tracking (SORT), face recognition (SphereFace) and method of similarity measurement of sets to increase high accuracy. According to the above two experiments, our new method works well on ChokePoint dataset and achieve the significant result. With face tracking giving more information on various poses, view of the face, the system can predict more accurate. However, with the experiment on our dataset accuracy of recognition rate is decreased due to far distance (causing low resolution) on our dataset. In the future, we will try super-resolution face image, other face recognition methods and do more experiment to enhance our result.

V. ACKNOWLEDGEMENT

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number C2018-26-01.

VI. REFERENCES