# Learning to Detect Phone-related Pedestrian Distracted Behaviors with Synthetic Data

Emre Hatay[1], Jin Ma[1], Huiming Sun[1], Jianwu Fang[2], Zhiqiang Gao[1], Hongkai Yu[1]*
[1]Cleveland State University, Cleveland, OH
[2]Chang'an University, Xi'an, China

## Abstract

*Due to the popularity and mobility of smart phones, phone-related pedestrian distracted behaviors, e.g., Texting, Game Playing, and Phone calls, have caused many traffic fatalities and accidents. As an advanced driver-assistance or autonomous-driving system, computer vision could be used to automatically detect distractions from cameras installed on the vehicle for useful safety intervention. The state-of-the-art method models this problem as a standard supervised learning method with a two-branch Convolutional Neural Network (CNN) followed by a voting on all image frames. In contrast, this paper proposes a new synthetic dataset named SYN-PPDB (448 synchronized video pairs of 53,760 computer game images) for this research problem and models it as a transfer learning problem from synthetic data to real data. A new deep learning model embedded with spatial-temporal feature learning and pose-aware transfer learning is proposed. Experimental results show that we could improve the state-of-the-art overall recognition accuracy from 84.27% to 96.67%.*

## 1. Introduction

Pedestrian fatalities and injuries have increased in the past decade. In the United States, the total number of pedestrian fatalities increased from 4,302 in 2010 to 6,283 in 2018 [1]. Phone-related distracted behaviors are one obvious reason for pedestrian-related collisions [19, 20]. Accident probability is increased when pedestrians are distracted and interacting with their mobile phones [23] and engaging in activities such as texting, watching videos, viewing maps, playing games, making phone calls, and so on. As described in [19, 20], the number of injuries to pedestrians engaged with their mobile phones has more than doubled since 2005.

As an advanced driver-assistance or autonomous-driving system, computer vision could be used to automatically detect pedestrian distractions from cameras installed on the vehicle. The pioneering work in phone-related pedestrian distracted behavior detection using computer vision is proposed by [19, 20], which designs a traditional supervised machine learning method including the components of phone location, pose estimation and pattern recognition. With multiple cues, the proposed method in [19, 20] is not an end-to-end learning system and it utilizes a single image as input. The state-of-the-art method by [23] models this problem as a supervised deep learning method with a two-branch Convolutional Neural Network (CNN) with two synchronized cameras installed on the vehicle, where a benchmark dataset named *PPDB* of 448 synchronized video pairs from a vehicle is collected for this research problem. However, this method takes a synchronized image pair as the input and is extended to video recognition by a simple voting through the image sequence, ignoring spatial-temporal feature learning, which has been proven to be important for video recognition [33, 31].

Instead of the standard supervised learning, this paper treats this problem as a transfer learning problem with a proposed deep learning model named PPDBNet from the synthetic data to real data as shown in Fig. 1. Previous work [30, 22, 6] has shown that synthetic data, e.g., computer game data, could be very helpful for computer vision tasks in the real world. In this paper, we make several efforts to improve the research of detecting phone-related pedestrian distracted behaviors in the real world. First, we create a new computer game-based synthetic dataset named *SYN-PPDB* to simulate phone-related pedestrian normal and distracted behaviors. Second, we implement spatial-temporal feature learning by CNN-based spatial feature extraction and Long Short-Term Memory (LSTM) based temporal feature learning. Finally, we transfer the knowledge gained from the synthetic data to the real data using pose-aware transfer learning.

In contrast with many transfer learning methods that learn a latent subspace for feature alignment [11, 27] or minimize the data distribution or style difference [34,

*Corresponding author: Hongkai Yu (h.yu19@csuohio.edu)

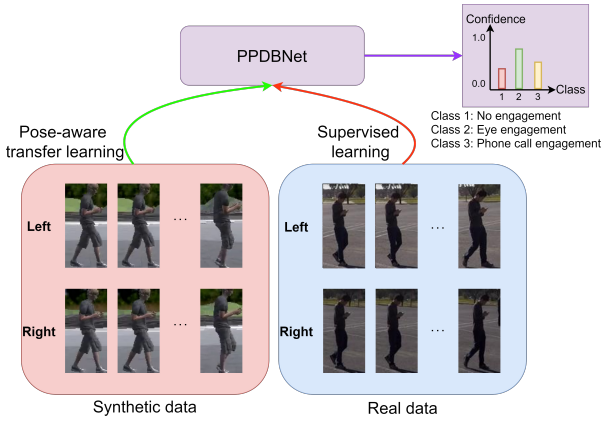


**Figure 1:** Illustration of the transfer learning with the proposed PPDBNet from the synthetic data to real data to detect the phone-related pedestrian distracted behaviors. With one synchronized video pair of the same person from left and right cameras as the input, the goal is to recognize it into three classes as defined in [23]: Class 1: No Engagement, Class 2: Eye Engagement (e.g., Texting, Game Playing), Class 3: Phone Call Engagement (e.g., Phone Calling).

12] or learn the domain invariant features by Generative Adversarial Networks (GANs) [26, 18], our pose-aware transfer learning is accomplished using two strategies: fine-tuning and 2D human pose feature embedding. In this paper, we find that 2D human pose is a relatively stable feature between synthetic data and real data. Therefore, we use the 2D human pose feature as the domain invariant feature for transfer learning.

In summary, this paper's main contributions are three folds: 1) We propose a new deep learning model (PPDBNet) to detect phone-related pedestrian distracted behaviors which incorporates spatial-temporal feature learning and pose-aware transfer learning; 2) We propose a new synthetic computer game dataset named *SYN-PPDB* (448 synchronized video pairs of 53,760 images) for this research problem; 3) By modeling this research problem as a transfer learning problem from synthetic data to real data, we improve the state-of-the-art overall recognition accuracy from 84.27% to 96.67%.

## 2. Related Work

**Pedestrian Distracted Behavior Detection:** By detecting nearby pedestrians [4] using cameras on moving vehicles, it is possible to analyze pedestrian behaviors and pedestrian motions for collision avoidance [21], pedestrian trajectory prediction [28] and so on. In this paper, we focus on the detection of phone-related pedestrian distracted behaviors by computer vision methods. Previous works on phone-related pedestrian distracted behaviors detection [19, 20] relied on traditional machine learning techniques for activity classification using one single camera image as input. Meanwhile, some deep learning based works on pedestrian attribute recognition [15] also take phone-related issues into consideration. Unlike these methods that take one single image as input, in order to get better detection result, we use image sequences from videos as input in this work.

The most-related work to this paper is by Humberto et al. [23], which displayed the advantages of using the synchronized video pair from left and right cameras against using one single camera as input, but [23] obtained the video recognition result by a simple voting method from the image recognition result, which ignored the spatial-temporal features hidden in videos. Following the same problem definition in [23], this paper embeds the spatial CNN and temporal LSTM to learn spatial-temporal features and also models this research as a transfer learning problem from synthetic data to real data.

**Learning from Synthetic Data:** Synthetic data is effective to solve the data scarcity problem and patterns learned from synthetic data could also be useful in real data [14, 6, 13, 30, 5, 18]. To solve different computer vision and autonomous driving problems, many previous works collected synthetic data from existing computer game engines [30, 13] or by building their own virtual computer game scenes [10, 29] on Unreal Engine [3] or Unity3D Engine [2]. Since all these excellent works on synthetic data are task-oriented, this paper creates our own synthetic dataset *SYN-PPDB* for detecting phone-related pedestrian distracted behaviors.

We exploit this synthetic data to improve the detection performance in real-world data by modeling it as a transfer learning problem. Transfer learning can be realized in various ways, such as domain adaptation [26, 18], subspace feature alignment [11, 27], image style distribution minimization [34, 12] and so on, while we propose a pose-aware transfer learning method in this paper by incorporating 2D human pose consistency into transfer learning.

**Table 1:** Summary for the proposed synthetic *SYN-PPDB* dataset. This data organization is same as that in the real-world *PPDB* dataset collected by [23]. Each video has 60 frames.

| Set | Class | synchronized video pairs | videos | synchronized image pairs | frames |
|---|---|---|---|---|---|
| Training | 1 | 110 | 220 | 6,600 | 13,200 |
| | 2 | 120 | 240 | 7,200 | 14,400 |
| | 3 | 110 | 220 | 6,600 | 13,200 |
| | **Total** | **340** | **680** | **20,400** | **40,800** |
| Testing | 1 | 38 | 76 | 2,280 | 4,560 |
| | 2 | 40 | 80 | 2,400 | 4,800 |
| | 3 | 30 | 60 | 1,800 | 3,600 |
| | **Total** | **108** | **216** | **6,480** | **12,960** |

## 3. Synthetic Dataset: SYN-PPDB

The first contribution in this work is proposing a SYNthetic dataset of Phone-related Pedestrian Distracted Behaviors, which is named as (*SYN-PPDB*). This dataset is generated using Unity 3D Engine [2], which has been widely used by game developers and researchers from different fields. We created two virtual city environments in Unity, along with ten 3D virtual actors and one virtual vehicle. These actors consist of 5 female and 5 male individuals with different heights, clothes and walking speeds. The vehicle has a fixed velocity of 1 meter per second, and has two cameras attached on its front panel with a distance of 1.8 meters. In the real-world environment, the vehicle speeds may vary, which will be considered as a domain difference as well. These two cameras are used to record synchronized video pairs of the same person from the left and right cameras, same as the setting in the real *PPDB* dataset [23]. In order to prevent the problem of data imbalance, we keep the data size and distribution of *SYN-PPDB* the same as *PPDB*.

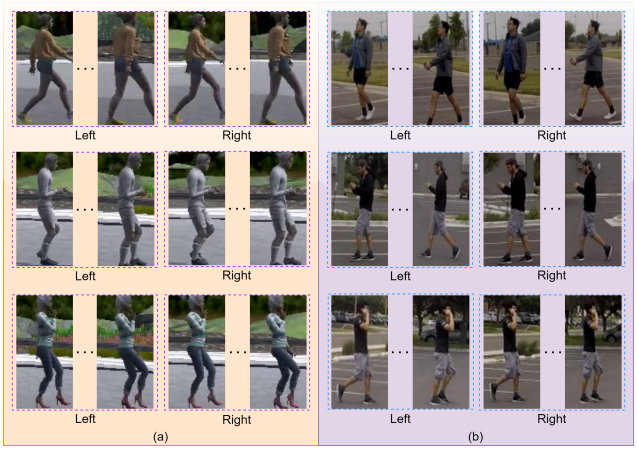


**Figure 2:** Example images of the datasets. (a): proposed synthetic *SYN-PPDB* dataset, (b): real *PPDB* dataset [23]. From top to bottom: synchronized video pairs of the same person for Class 1, Class 2, Class 3 by the left and right cameras.

We then produced a total number of 448 video pairs based on above elements. Each synchronized video pair was captured simultaneously by two cameras on the moving vehicle of the same person, when a randomly selected actor was walking across the street in or near the front of the vehicle while engaging in a phone-related behavior. Each video has a duration of 2 seconds, frame rate of 30 frames per second (fps), resolution of $64 \times 128$ pixels per image, and a corresponding phone-related behavior label from: $c_1$: *no engagement*, $c_2$: *eye engagement* and $c_3$: *phone call engagement*. In the end, all frames were extracted to construct a new dataset with a total number of $448(pairs) \times 2(videos) \times 2(seconds) \times 30(fps) = 53,760$ images. This dataset was split into a training set of $40,800$ images (from 340 video pairs) and a testing set of $12,960$ images (from 108 video pairs). The total number of video pairs and training-testing split of the *SYN-PPDB* dataset are the same as the real *PPDB* dataset [23]. The detailed data distribution and class distribution of the *SYN-PPDB* dataset are shown in Table. 1. Some example images are shown in Fig. 2.

## 4. The Proposed Method

In this section, we will describe the architecture of the proposed network named PPDBNet, along with our strategies for extracting spatial-temporal features and pose-aware transfer learning.

### 4.1. Network Overview

Following the same problem definition in [23], the goal of the proposed PPDBNet is to classify a synchronized video pair $\mathcal{X}$ of the same person from left and right cameras into a behavior class $\mathcal{Y}$. Different from many existing methods only using the appearance color image, we estimate the 2D human pose by applying the OpenPose [7] Body-25 model to the corresponding human appearance image. Each pose image includes 25 key-points and is formatted to be the same size as the input images ($64 \times 128$). Therefore, mapping can be formulated as $h(\mathcal{X}) \rightarrow \mathcal{Y}$, where:

$$\mathcal{X} = \{\mathbf{X_L} = (\mathbf{I_L}, \mathbf{P_L}), \mathbf{X_R} = (\mathbf{I_R}, \mathbf{P_R})\}, \mathcal{Y} \in \{c_1, c_2, c_3\}. \tag{1}$$

As shown in Eq. (1), $\mathcal{X}$ includes two image sequences $\mathbf{X_L}$ and $\mathbf{X_R}$ from a synchronized left-right video pair, and each image sequence $\mathbf{X}$ has an appearance part $\mathbf{I} = (i^1, ..., i^k)$ and a pose part $\mathbf{P} = (p^1, ..., p^k)$, where $i$ and $p$ indicate the appearance image and pose image, $k$ indicates sequence length. $\mathcal{Y}$ is a phone-related behavior label from three classes $\{c_1, c_2, c_3\}$.

The proposed PPDBNet is a siamese-like network, as shown in Fig. 3. It consists of a left branch and right branch, receiving image sequences from the left and right camera video separately, but sharing weights with each other. Each branch has two CNN backbone networks that are used for extracting the spatial feature from appearance image $i$ and pose image $p$ respectively, and two LSTM layers followed by a temporal pooling layer that is used for extracting temporal features. The outputs of these two branches are then concatenated and passed into a fully-connected layer followed by a softmax normalization to get the classification confidences. During the training process, we use the cross-entropy loss function in Eq. (2) to optimize the network:

$$Loss(\mathcal{X}, \boldsymbol{y}) = -\sum_{j=1}^{n} y_j \log v_j, \tag{2}$$

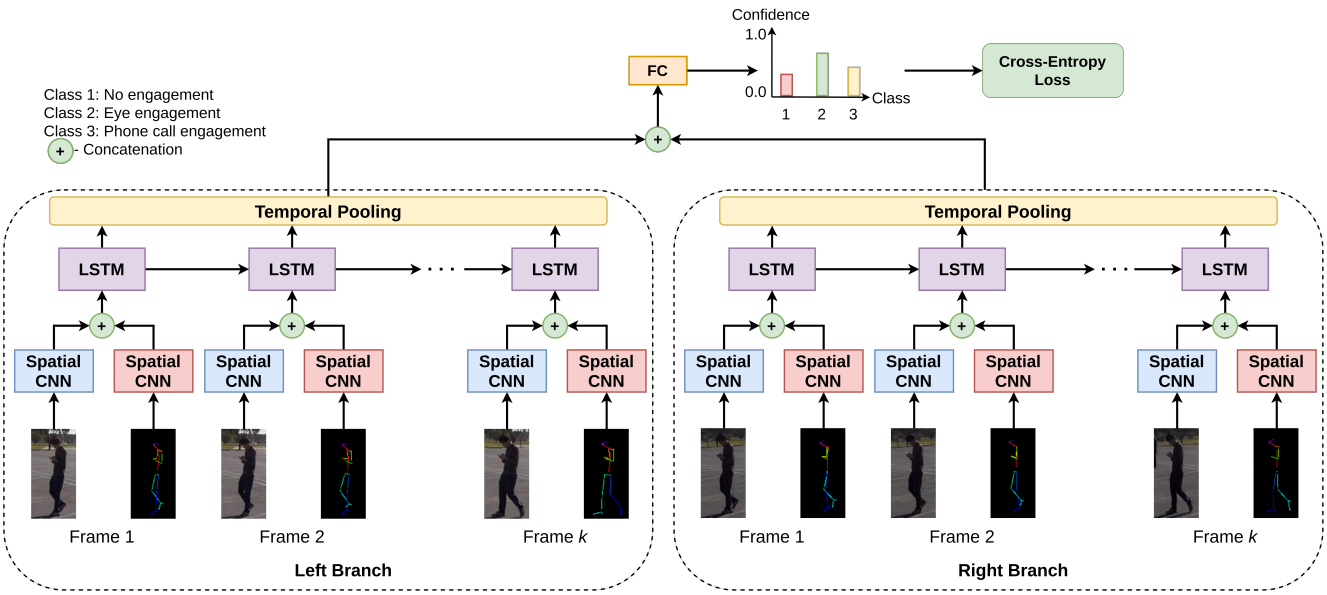


**Figure 3:** The overall architecture of the proposed PPDBNet. Its goal is to classify a synchronized video pair of the same person from left and right cameras into a behavior class, same as the problem definition in [23]. The 2D human pose is estimated by OpenPose [7].

where $n$ is the number of classes, $\boldsymbol{y} = [y_1, ..., y_n]^\top$ is the ground truth one-hot label vector with $y_j = 1$ if the input $\mathcal{X}$ belongs to class $j$, and $\boldsymbol{v} = [v_1, .., v_n]^\top$ is the predicted vector where $v_j$ represents the confidence ($0 \leq v_j \leq 1$) of the input $\mathcal{X}$ belonging to class $j$ after the softmax normalization.

## 4.2. Spatial-temporal Feature Learning

In order to capture spatial-temporal features from input image sequences, we employ both CNN and LSTM in our network. CNN has become a powerful tool for harnessing rich spatial features from a single image in recent years. While RNN (Recurrent Neural Network) has demonstrated its edge over CNN in handling sequential data by exploring the reserved information among different timesteps, and has been applied in a wide range of areas, such as the processing of texts, voices and videos. Here we utilize a classic CNN VGG16 [25] to extract the spatial cues from a single appearance image or pose image, and an improved RNN unit LSTM to extract the temporal cues from image sequences. It is also possible to have two separate LSTMs for the appearance and pose cues respectively, followed by a fusion. In this paper, we just fuse the spatial features of appearance and pose cues first and then feed them into one LSTM for the temporal feature learning.

Since the whole network is a siamese-like network, we only take the left branch as an example to explain the workflow for the sake of simplicity. Given the input image sequence $\mathbf{X_L} = (\mathbf{I_L}, \mathbf{P_L})$ for the left branch, there are two images $(i^t, p^t)$ at timestep $t$, where $i^t$ is the appearance image and $p^t$ is the pose image. These two images are fed into two CNNs accordingly (with different weights) to get a feature vector of size $(4096 \times 1)$. Their feature vectors are then concatenated as a single vector of size $(8192 \times 1)$, and this vector becomes the input data for the LSTM layers at timestep $t$. The LSTM layers take a tensor of size $(8192 \times k)$ as input and produce the output tensor of size $(512 \times k)$, where $k$ is the length of image sequence, 512 is the number of LSTM hidden units. The output tensor is then passed into the temporal pooling layer to get a feature vector of size $(512 \times 1)$ as shown in Eq. (3):

$$\boldsymbol{f} = Pooling(\mathbf{X}) = \frac{\sum_{i=1}^{k} \boldsymbol{x}_i}{k}, \tag{3}$$

where $\mathbf{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_k] \in \mathbb{R}^{512 \times k}$ is the input tensor for temporal pooling layer, $\boldsymbol{x}_t \in \mathbb{R}^{512 \times 1}$ is the vector at time step $t$, and $\boldsymbol{f} \in \mathbb{R}^{512 \times 1}$ is the output vector. In this way, we can construct a robust representation for the input image sequence by extracting its spatial-temporal features within each branch. Finally, the outputs from two branches are concatenated and fed into a fully-connected layer to get the classification result.

## 4.3. Pose-aware Transfer Learning

### 4.3.1 Human pose as domain-invariant feature

In this paper, we treat the 2D human pose as a kind of domain invariant feature. The assumption is based on two observations. First, in the same class, we find that the LPIPS distance [32] between 2D human pose images estimated by [8] is smaller than the corresponding human

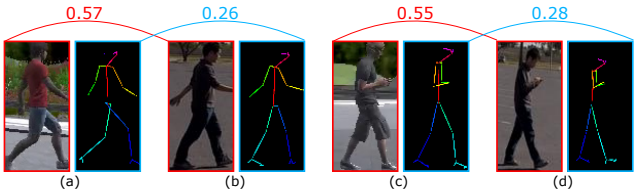


**Figure 4:** Illustration of human pose reducing the domain gap between the synthetic data as source domain (a, c) and real data as target domain (b, d). The LPIPS distance [32] between estimated 2D human pose by [8] is significantly smaller than that between the corresponding human appearance images.

appearance images between the synthetic data (source domain) and real data (target domain), as shown in Fig. 4. The Learned Perceptual Image Patch Similarity (LPIPS) is a popular deep learning-based perceptual metric with a better mimic to human perception, which computes the distance of multi-level activations of a pre-trained deep CNN network. Smaller LPIPS distance indicates more similar perception. Second, we implement the t-Distributed Stochastic Neighbor Embedding (t-SNE) [17] based statistical data analysis. Specifically, we randomly select 100 images and their corresponding 2D human pose images estimated by [8] for each class in the synthetic data and real data. The images are the high-dimensional data and the t-SNE is an algorithm to reduce dimensions to visualize the high-dimensional data. With dimension reduction by t-SNE to 2D space, we extracted the centers for each class in source domain as $C_{S_i}$ and the centers for each class in target domain as $C_{T_i}$, where $i = 1, 2, 3$ for the three defined classes. The $L_2$ distance between $C_{S_1}$ and $C_{T_1}$ is 29.9 using human appearance images while the $L_2$ distance between $C_{S_1}$ and $C_{T_1}$ is much smaller as 6.7 using the estimated 2D human pose images, which also happens for other two classes in our experiment. Based on these two observations, it is obvious that the human pose could reduce the domain gap between the synthetic data and real data. Because the human appearance images might contain different background contexts and many foreground color/texture variances between the synthetic data and real data, the human pose is considered as the relatively domain-invariant feature in this paper.

#### 4.3.2 Transfer learning strategies

In this paper, we propose a pose-aware transfer learning for this task, which mainly includes two strategies: human pose based domain-invariant feature embedding and network fine-tuning. First, we estimate the 2D human pose of each synthetic image and that of each real image by OpenPose [8]. Second, we incorporate the estimated 2D human pose into the proposed network as shown in Fig. 3 and train it on the synthetic data. Third, we fine-tune the pretrained network model using the real data. In this way, the trained network could well inherit the spatial-temporal feature and pose information on the synthetic data and be further adapted to the real data. By embedding the human pose as the domain-invariant feature into the proposed network, the domain gap is reduced, leading to a better transfer learning.

## 5. Experiments

Focusing on the task of phone-related pedestrian distracted behavior detection from videos or image sequences, we study the performance of proposed method in this section. First we introduce the experiment setting and comparison methods, then show the experimental results on the real *PPDB* dataset. Finally, we discuss the performance of the proposed method in different parameter settings.

### 5.1. Experiment setting

We employ two datasets in our experiment: the synthetic *SYN-PPDB* dataset proposed by us and the real-world *PPDB* dataset collected by [23]. It is worth mentioning that the *SYN-PPDB* dataset and the real *PPDB* dataset are with the same data organization (video numbers, training-testing splitting), as shown in Table. 1. During the experiment, we use these two datasets in two different ways. On one hand, we pre-train networks on *SYN-PPDB* and fine-tune them on *PPDB* to observe the effect of transfer learning; on the other hand, we train networks only on *PPDB* for comparison with typical supervised learning. In the end, all experiment results are reported on the testing set of *PPDB*. We utilize the classification accuracy and confusion matrix to show the detailed recognition performance for each behavior class, and use the average classification accuracy of all classes to evaluate the overall recognition performance.

Since each video has 60 frames, it is hard to cover all the 60 frames into the proposed PPDBNet due to the large GPU memory cost. Therefore, we split the 60 frames of one video into $\frac{60}{k}$ sub-sequences (chunks) with sequence length $k$, and the overall classification confidence of the video is averaged over these sub-sequences for the final recognition of the video. We set $k = 10$ in this experiment, and the choice about $k$ will be discussed later in this section.

During our training process, the batch size was set as 1 and Adam method was used for optimizing the network weights with a weight decay of $5 \times 10^{-4}$, and the training epoch was set as 40 and the learning rate was fixed as $1 \times 10^{-7}$. All experiments were conducted on an NVIDIA GeForce RTX 3090 GPU, and the overall GPU memory was about 10GB during training and 2.5GB during testing based on above setting.

### 5.2. Comparison methods

We compare the results of proposed method with several state-of-the-art methods: Dual-branch CNN [23], Graph Convolution Networks based 2s-AGCN [24], and MS-G3D [16]. The last two networks, 2s-AGCN and MS-G3D, take skeletal human pose image sequence as input for action recognition, thus can handle the video recognition task properly. The Dual-branch CNN takes one image pair as input for action recognition and then it applies a voting approach to average the confidence outputs of all video frames to obtain the final video recognition result. All these three networks are trained on the real *PPDB* dataset, note that 2s-AGCN and MS-G3D are trained and tested on skeletal human pose images, while Dual-branch CNN is trained and tested on human appearance images.

In order to investigate the influence of human appearance images and pose images to our proposed PPDBNet, we simplify the input image sequences to get two variants of PPDBNet. The one with only human appearance image sequence as input is denoted as PPDBNet$^*$, the other with only human pose image sequence as input is denoted as PPDBNet$^*_{Pose}$. The PPDBNet$^*$ method is similar to the LRCN method [9] for activity recognition.

Besides, when a network is trained first on the synthetic *SYN-PPDB* dataset and then fine-tuned on the real *PPDB* dataset, we add a "-S2R-FT" suffix to its name, where "S2R" means the transfer from the synthetic data to the real data and "FT" indicates fine-tuning based transfer learning. For example, PPDBNet$^*$-S2R-FT denotes that we train the PPDBNet$^*$ network on the synthetic *SYN-PPDB* dataset first and then fine-tune it on the real *PPDB* dataset, while PPDBNet$^*$ with no suffix denotes that we train it only on the real *PPDB* dataset. In this way, we can study the effect of our transfer learning strategy.

In addition, we compare our proposed Fine-tuning (FT) based pose-aware transfer learning method with the image style transfer based domain adaptation method CycleGAN [34]. We first train a CycleGAN [34] model to transfer the style of synthetic images to that of real images, and then train the PPDBNet$^*$ network on the transferred fake images by CycleGAN and finally fine-tune the pre-trained PPDBNet$^*$ model on the real *PPDB* dataset, which is denoted as PPDBNet$^*$-S2R-CycleGAN.

### 5.3. Experimental results

The detailed recognition accuracy scores of different methods are reported in Table. 2. The confusion matrices of some representative methods are shown in Fig. 5. From the experimental results, we can see that the proposed PPDBNet-S2R-FT method achieved the best overall average classification accuracy 96.67%, which is a significant improvement from the state-of-the-art performance 84.27% by the Dual-branch CNN [23]. Compared to Dual-branch CNN, other methods include spatial-temporal feature learning, so they obtained better overall performance.

When trained with only human appearance images on the real *PPDB* dataset, the average accuracy of our PPDBNet$^*$ (88.12%) is 3.85% higer than Dual-branch CNN (84.27%), this proves the power of spatial-temporal feature learning in our network architecture. When trained with only human pose images on the real *PPDB* dataset, our PPDBNet$^*_{Pose}$ (92.73%) outperformed MS-G3D (91.91%), but fell behind 2s-AGCN(92.78%) slightly. However, it is worth noting that our PPDBNet$^*_{Pose}$ obtained equal or better accuracy than 2s-AGCN on two more important classes, class 2 Eye Engagement and class 3 Phone Call Engagement.

With the help of transfer learning, PPDBNet$^*$-S2R-FT obtained higher average accuracy (92.28%) than the average accuracy (88.12%) by PPDBNet$^*$, and PPDBNet$^*_{Pose}$-S2R-FT obtained higher average accuracy (93.89%) than the average accuracy by PPDBNet$^*_{Pose}$ (92.73%). This result demonstrates the advantages of the proposed synthetic *SYN-PPDB* dataset, which could improve the recognition accuracy on the real PPDB dataset by transfer learning. This result also shows that the deep learning model performs better in this task if estimated 2D human pose images are embedded by comparing PPDBNet$^*_{Pose}$ and PPDBNet$^*$. PPDBNet$^*$-S2R-CycleGAN [34] did not obtain high overall accuracy (89.96%), which indicates that only using human appearance images for transfer learning might be difficult for this task.

The proposed PPDBNet learns the spatial-temporal features for the recognition ability from both the human appearance and skeletal human pose, leading to the second best performance 95.83%. The PPDBNet is pose-aware and when it is combined with the fine-tuning based transfer learning, the final approach PPDBNet-S2R-FT learns the recognition ability on the synthetic data first and then transfers the learned ability to the real-world data, leading to the best performance 96.67%. This phenomena shows that the Pose-aware Transfer Learning from the synthetic data to real data could really help the recognition on the real data. Compared to other skeletal human pose based activity recognition methods MS-G3D [16] and 2s-AGCN [24], the proposed methods got better performance because it used the comprehensive spatial-temporal features (human appearance and skeletal human pose) and the advanced Pose-aware Transfer Learning method. From the PPDBNet method (Proposed) to the PPDBNet-S2R-FT method (Proposed+), we can see the improvement for Class 2 (Eye engagement) recognition accuracy (from 97.5% to 100%). Because Class 2 (Eye engagement) is the most dangerous class in the phone-related distracted behaviors, we think this improvement is quite valuable. The confusion

**Table 2:** Recognition accuracy (%) on the real *PPDB* dataset [23]. "S2R" means the transfer from the synthetic *SYN-PPDB* dataset to the real *PPDB* dataset. "FT" indicates Fine-tuning based transfer learning. Note that the methods without "S2R" means the pure supervised learning on the real *PPDB* dataset, and the proposed PPDBNet is pose-aware, and Dual-branch CNN [23] applies a voting approach to summarize all the image frames for a video recognition.

| Methods | Class #1 | Class #2 | Class #3 | Average |
|---|---|---|---|---|
| Dual-branch CNN [23] | 71.10 | 95.00 | 86.70 | 84.27 |
| MS-G3D [16] | 97.40 | 95.00 | 83.33 | 91.91 |
| 2s-AGCN [24] | **100.00** | 95.00 | 83.33 | 92.78 |
| PPDBNet* | 86.84 | 87.51 | 90.00 | 88.12 |
| PPDBNet*-S2R-FT | 86.84 | **100.00** | 90.00 | 92.28 |
| PPDBNet*-S2R-CycleGAN [34] | 81.58 | 95.00 | **93.33** | 89.96 |
| PPDBNet$^*_{Pose}$ | 97.37 | 97.50 | 83.33 | 92.73 |
| PPDBNet$^*_{Pose}$-S2R-FT | **100.00** | 95.00 | 86.67 | 93.89 |
| PPDBNet (Proposed) | **100.00** | 97.50 | 90.00 | **95.83** |
| PPDBNet-S2R-FT (Proposed+) | **100.00** | **100.00** | 90.00 | **96.67** |

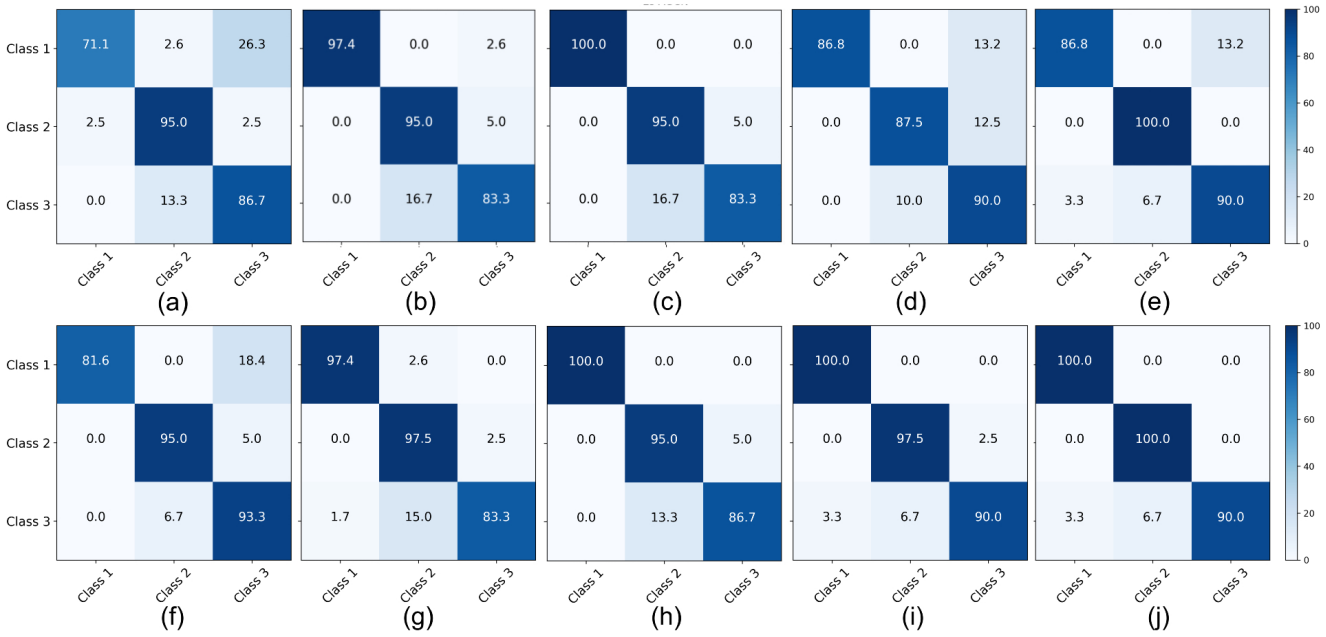


**Figure 5:** Confusion matrices of the three-class recognition accuracy (%) on the real *PPDB* dataset [23] by different methods: (a) Dual-branch CNN [23], (b) MS-G3D [16], (c) 2s-AGCN [24], (d) PPDBNet*, (e) PPDBNet*-S2R-FT [9], (f) PPDBNet*-S2R-CycleGAN [34], (g) PPDBNet$^*_{Pose}$, (h) PPDBNet$^*_{Pose}$-S2R-FT, (i) PPDBNet (Proposed), (j) PPDBNet-S2R-FT (Proposed+).

matrices in Fig. 5 also show that the proposed methods has less misclassifications for the three classes. In the cases of human ambiguity actions, slight variations between the way pedestrians interact with their phones (talking using speaker, *etc.*) can lead to minor misclassifications for the Proposed+ method.

These results demonstrate that the proposed PPDBNet, along with the new synthetic dataset and pose-aware transfer learning method, is able to well classify the phone-related pedestrian distracted behaviors.

### 5.4. Discussion on Sub-Sequence Length

To demonstrate the effect of sub-sequence (chunk) length $k$ on overall classification accuracy, we have experimented on various sub-sequence lengths. When using sub-sequence length of 4 frames, PPDBNet-S2R-FT is able to classify the input image sequence in nearly real-time (43.25 ms) while maintaining a good average classification accuracy of 92.18%. When using sub-sequence length of 10 frames, PPDBNet-S2R-FT obtains the best accuracy of 96.67%, which is used in our experiment. Our results for

**Table 3:** Recognition accuracy (%) and inference time (ms) of the proposed PPDBNet-S2R-FT under various sub-sequence (chunk) lengths.

| Sub-sequence Length | Average Accuracy | Inference Time |
|---|---|---|
| $k$=4 | 92.18 | 43.25 |
| $k$=6 | 94.96 | 64.73 |
| $k$=10 | **96.67** | 107.97 |
| $k$=15 | 94.55 | 161.75 |

**Table 4:** Effect of spatio-temporal learning on the recognition accuracy (%).

| Method | Class 1 | Class 2 | Class 3 | Average |
|---|---|---|---|---|
| PPDBNet-S2R-FT | 100.00 | 100.00 | 90.00 | **96.67** |
| PPDBNet$_{FC}$-S2R-FT | 100.00 | 97.50 | 83.33 | 93.61 |

various sub-sequence lengths are listed on Table. 3. It is worth mentioning that we fix the same length for the non-overlapping frames per sub-sequence (no sliding window approach) to make it simple.

### 5.5. Discussion on Spatio-Temporal Learning

In the proposed method, spatio-temporal learning in the video plays an important role, which is realized by the LSTM layers. In order to show the effect of spatio-temporal learning, we replace the LSTM layers with a Fully-Connected (FC) layer for the Proposed+ method, which is denoted as PPDBNet$_{FC}$-S2R-FT. The experimental results are available on Table. 4. PPDBNet$_{FC}$-S2R-FT achieves 93.61%, which is lower than 96.67% by PPDBNet-S2R-FT (Proposed+). It means that the LSTM layers for spatio-temporal learning by the Proposed+ method could improve the network's video recognition ability compared to the same network without spatio-temporal learning.

## 6. CONCLUSIONS

In this paper, we proposed a new way for detecting phone-related pedestrian distracted behaviors. First, we proposed a new synthetic dataset named *SYN-PPDB* (448 synchronized video pairs of 53,760 images) for this research. Second, we proposed a new network architecture named PPDBNet with a Pose-aware Transfer Learning method to improve the recognition accuracy on the real-world data by inheriting the information gained from the synthetic data. On the real *PPDB* dataset [23], the proposed method could improve the state-of-the-art overall average recognition accuracy from 84.27% to 96.67%.

## 7. Acknowledgement

This work is supported by Amazon Web Services (AWS) Cloud Credits for Research Award. Dr. Jianwu Fang is supported by NSFC 61806022. We would like to appreciate the suggestions by Dr. Eric Schearer to revise the writing.

## References

[1] Traffic safety facts 2018. *National Highway Traffic Safety Administration, US Department of Transportation*.

[2] Unity real-time development platform. https://www.unity.com.

[3] Unreal engine. https://www.unrealengine.com.

[4] Anelia Angelova, Alex Krizhevsky, and Vincent Vanhoucke. Pedestrian detection with a large-field-of-view deep network. In *IEEE International Conference on Robotics and Automation*, pages 704–711, 2015.

[5] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *IEEE International Conference on Robotics and Automation*, pages 5958–5964, 2019.

[6] Slawomir Bak, Peter Carr, and Jean-Francois Lalonde. Domain adaptation through synthesis for unsupervised person re-identification. In *European Conference on Computer Vision*, pages 189–205, 2018.

[7] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[8] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[9] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[11] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *IEEE International Conference on Computer Vision*, pages 2960–2967, 2013.

[12] Dazhou Guo, Yanting Pei, Kang Zheng, Hongkai Yu, Yuhang Lu, and Song Wang. Degraded image semantic segmentation with dense-gram networks. *IEEE Transactions on Image Processing*, 29:782–795, 2019.

[13] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3105–3115, 2019.

[14] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016.

[15] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *IEEE International Conference on Computer Vision*, pages 350–359, 2017.

[16] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020.

[17] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[18] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020.

[19] Akshay Rangesh, Eshed Ohn-Bar, Kevan Yuen, and Mohan M Trivedi. Pedestrians and their phones-detecting phone-based activities of pedestrians for autonomous vehicles. In *IEEE International Conference on Intelligent Transportation Systems*, pages 1882–1887, 2016.

[20] Akshay Rangesh and Mohan Manubhai Trivedi. When vehicles see pedestrians with phones: A multicue framework for recognizing phone-based activities of pedestrians. *IEEE Transactions on Intelligent Vehicles*, 3(2):218–227, 2018.

[21] Eike Rehder, Florian Wirth, Martin Lauer, and Christoph Stiller. Pedestrian prediction by planning using deep neural networks. In *IEEE International Conference on Robotics and Automation*, pages 1–5, 2018.

[22] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[23] Humberto Saenz, Huiming Sun, Lingtao Wu, Xuesong Zhou, and Hongkai Yu. Detecting phone-related pedestrian distracted behaviours via a two-branch convolutional neural network. *IET Intelligent Transport Systems*, 15(1):147–158, 2021.

[24] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

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

[26] Shaoyue Song, Hongkai Yu, Zhenjiang Miao, Jianwu Fang, Kang Zheng, Cong Ma, and Song Wang. Multi-spectral salient object detection by adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2020.

[27] Shaoyue Song, Hongkai Yu, Zhenjiang Miao, Qiang Zhang, Yuewei Lin, and Song Wang. Domain adaptation for convolutional neural networks-based remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, 16(8):1324–1328, 2019.

[28] Li Sun, Zhi Yan, Sergi Molina Mellado, Marc Hanheide, and Tom Duckett. 3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. In *IEEE International Conference on Robotics and Automation*, pages 1–7, 2018.

[29] Xiaoxiao Sun and Liang Zheng. Dissecting person re-identification from the viewpoint of viewpoint. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 608–617, 2019.

[30] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8198–8207, 2019.

[31] Yang Wu, Jie Qiu, Jun Takamatsu, and Tsukasa Ogasawara. Temporal-enhanced convolutional network for person re-identification. In *AAAI Conference on Artificial Intelligence*, 2018.

[32] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[33] Kang Zheng, Xiaochuan Fan, Yuewei Lin, Hao Guo, Hongkai Yu, Dazhou Guo, and Song Wang. Learning view-invariant features for person identification in temporally synchronized videos taken by wearable cameras. In *IEEE International Conference on Computer Vision*, pages 2858–2866, 2017.

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.