Pedestrian Mobile Phone Usage Detection (PMPUD)

Sudalaiandi Raja Sudalaimuthu, Jayaraman Revathi

May 2020

1 Introduction

According to a new study published in the Journal on Injury Prevention BMJ (British Medical Journal) on 3rd Feb 2020 on “Smartphone texting linked to compromised pedestrian safety”, worldwide around 270,000 pedestrians die annually, making up a fifth of all traffic fatalities. Pedestrian distraction has become a recognised safety issue as more and more people use their smartphones or hand held devices while walking on pavements and crossing roads.

Pedestrians are a particularly vulnerable group of road users as they face an increased risk of fatal or severe injury in the event of an accident. With the safety of pedestrians in mind, Singapore Police Force has amended the highway code to curb the use of mobile communication devices while crossing roads as of December 2019. Pedestrians should exercise caution at all times and not risk their own safety and the safety of other road users. When crossing roads, they should avoid using their devices. They should always use pedestrian crossings where available and obey all traffic signs and lights at all times.

In light of this situation, there is an increasing demand to develop a reliable and efficient intelligent system for detecting pedestrians using mobile phone devices while crossing the road that does not rely on a human observer. The purpose of this project is to develop a vision based system to detect mobile phone usage by pedestrians. By doing so, the project aims to identify appropriate measures to ensure the safety of all road users following the Highway code of Conduct.

2 Related work / References

Distracted walking is an ongoing world-wide problem and extensive research is being conducted to address this problem. City authorities in Chongqing have introduced a 30 metre ‘cellphone lane’ for pedestrians and research papers are being published to analyse the behavioural pattern of pedestrians. The objective of this project is to develop a low-cost pedestrian mobile phone usage detection algorithm using many areas of intelligence such as video, image processing, statistics, data analytics, computer vision techniques such as pose estimation and artificial intelligence.

The challenge is to identify the usage of phone in hands or that phone is being carried in hands while walking by pedestrians. The project initially applied the object detection algorithm, YOLOV3, with transfer learning to classify the ”pedestrian using phone” and ”pedestrian not using phone” for classification. Using labellmsg, around 500 pedestrian images were annotated for each classifications and the transfer learning of YOLOV3 was performed. Due to different ways of carrying the mobile phone in the hands, the performance metric of mean Average Precision(mAP) was only less than 0.5. One more class of ”phone in palm” was then introduced and hence it can be combined to derive the rule ”pedestrian with phone” and ”phone in palm” for detection. However, even if the mAP score was slightly improved, it was still less than 0.5.

To further improve the efficiency of the detection, pose estimation techniques are being used to derive the key features of body pose points and the angle between the hand and the phone was calculated using the pose points. The extracted key point features from the pose techniques and the derived features are fed into Machine learning algorithms such as Support Vector Machine and deep learning algorithms to classify the pedestrians using phone and pedestrians not using phone. .

Below are the URLs that has referenced for this project.

- https://www.bmj.com/company/newsroom/smartphone-texting-linked-to-compromised-pedestrian-safety/
- https://www.theguardian.com/world/shortcuts/2014/sep/15/china-mobile-phone-lane-distracted-walking-pedestrians
- https://pylessons.com/YOLOv3-custom-training/
- https://www.mvig.org/research/alpha pose.html
- https://arxiv.org/abs/1612.00137
- https://github.com/tzutalin/labelImg
- https://github.com/MVIG-SJTU/AlphaPose
- https://www.police.gov.sg/Media-Room/News/20191115_OTHERS_AMENDMENTS_TO_HIGHWAY_COD

3 Environment Setup

Installation steps to setup project execution environment are described below

1. Create a conda virtual environment.

conda config –append channels conda-forge

conda create -n trackPhoneUsage python=3.6 pandas scikit-learn matplotlib opencv spyder notebook

conda activate trackPhoneUsage

2. Install PyTorch conda install pytorch==1.1.0 torchvision==0.3.0 -c pytorch

3. Get Track*Phone*Usage project repo

git clone https://github.com/aivoyagers/Track*Phone*Usage.git

cd Track*Phone*Usage

4. Install

export PATH=/usr/local/cuda/bin/:$PATH‘

export LD*LIBRARY*PATH=/usr/local/cuda/lib64/:$LD*LIBRARY*PATH

pip install cython

sudo apt-get install libyaml-dev

pip install pyyaml

python setup.py build develop

5. For Windows user, if you meet error with PyYaml, you can download and install it manually from here: https://pyyaml.org/wiki/PyYAML. If your OS platform is Windows, make sure that Windows C++ build tool like visual studio 15+ or visual c++ 2015+ is installed for training.

6. Download the below pretrained/offline trained Models, training and validation datasets and other folders from the Google Drive https://drive.google.com/drive/folders/1ZyzRj-G6FolQ-qhJzOx1YkBptYtaWjeS and merge with git cloned folder retaining the structure.

4 Proposed Approach

The proposed approach can be described as a set of clearly defined steps, where each of the step has its own purpose (see Figure 1).

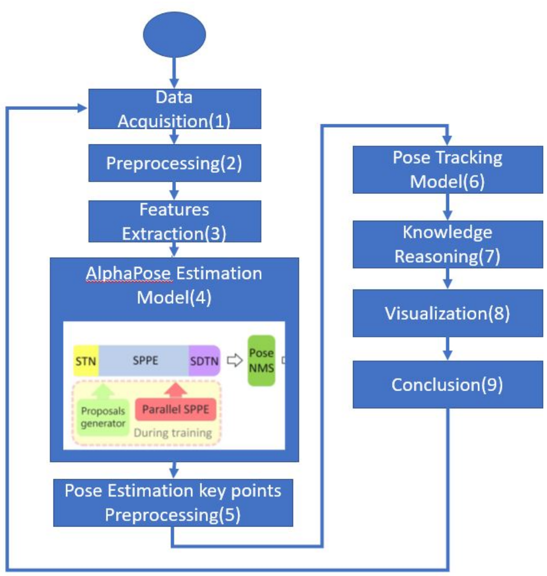

Figure 1: **Steps of the flow chart**

The flow chart shown are described in further detail in the below sections:

4.1 Data Acquisition

The starting point in the process is the data acquisition (1), in which video recordings of pedestrians using the phone and pedestrians not using the phone are recorded as the data is not readily available on the internet. Once the video recordings are captured, the video recordings are clipped for the duration of 15 to 30 secs and annotated with the labels of pedestrians using phone and pedestrians not using phone.

4.2 Preprocessing

The video clips are extracted into frame level of data and feature engineering has been performed(2).

4.3 Feature Extraction

The next step was to compute a set of features for best possible classification (3). Deep learning model of base network ResNet152 is used in place of separate feature extraction. The spatial and temporal features along with edges and shapes are extracted.

4.4 AlphaPose Estimation

AlphaPose Estimation(4) uses the the pipeline of Regional Multi-person Pose Estimation(RMPE). In this, the human bounding boxes obtained by the human detector are fed into the "Symmetric Spatial Transformer Network (SSTN) + single-person pose estimator (SPPE)" module. Symmetric STN consists of STN and

Spactial DeTransformer Network which are attached before and after the SPPE. The STN receives human proposals and the SDTN generates pose proposals. The Parallel SPPE acts as an additional regularizer during the training phase and the pose proposals are generated automatically.The generated pose proposals are refined by parametric Pose Non-Maximum-Suppression to obtain the estimated human poses. During the training, "Parallel SPPE" is introduced in order to avoid local minimums and further leverage the power of SSTN. To augment the existing training samples, a pose-guided proposals generator (PGPG) is designed.

4.5 Pose Estimation KeyPoints Preprocessing

The generated key points are flattened and arm angles are calculated along with confidence score are used as features for pose tracking model. Each of these features are further labeled and sent to the pose tracking model.

4.6 Pose Tracking Model

After Pose Estimation of key points preprocessing, the machine learning algorithms below are identified and created for shallow and deep learning models.

1. Support Vector Machine
2. Convolutional Neural Network

4.7 Knowledge and Reasoning

Once the models are created, the metrics were evaluated for all the algorithms and these are described in the Experimental results section.

4.8 Visualization

In visualization, key points links are drawn in the video output and with corresponding labels such as using and not using phone. The link points are also differentiated in various color notations such as blue for not using and red for using the mobile phones. OpenCV has used to develop the output.

4.9 Action/Conclusion

The usage of mobile phones by pedestrian is detected by the above algorithmic models. The detection helps to raise an alarm to traffic police and helps traffic police to initiate the appropriate measures to be taken to minimize the risk of pedestrians using mobile phones.

5 Experimental results

5.1 Dataset

Dataset has been obtained for this problem from the Google drive as it is very huge to upload to LumiNUS :

Download the data folder to the Track_Phone_Usage folder.

https://drive.google.com/drive/folders/1ZyzRj-G6FolQ
-qhJzOx1YkBptYtaWjeS

5.2 List of existing modules

Below table represents the list of existing modules.

Module Name	Explanation
AlphaPose	Open-source pose estimation
YoloV3	Publicly available object detection module

5.3 List of PMPUD modules

Below table represents the list of PMPUD modules.

Module Name	Explanation
Data Preparation	Video recordings and Extraction of images
Feature Engineering	Enhancing features of estimated pose points
Pose Tracking	SVM Classifier
Visualization	Phone usage tracking visualization

5.4 List of existing models

Below table represents the list of existing models.

Model Name	Explanation
fast_421_res152 _256192	Backbone network for vision
yolov3-spp.weights	object detection model

5.5 List of PMPUD models

Below table represents the list of PMPUD models.

Model Name	Explanation
phone$_{v}iew_{d}etection-$ 0.$joblib$	phone view detection model

5.6 Performance metric

Below are the performance metrics of the pose tracking models using Support Vector Classifier with enhanced pose points has achieved accuracy score of 0.9365 with precision and recall score of 0.935.

```
=======================Pose Tracking Model Metrics =============
[[699  43]
 [ 41 540]]
Accuracy Score :  0.9365079365079365
                  precision    recall  f1-score   support

not using phone       0.94      0.94      0.94       742
    using phone       0.93      0.93      0.93       581

       accuracy                           0.94      1323
      macro avg       0.94      0.94      0.94      1323
   weighted avg       0.94      0.94      0.94      1323

weighted f1 score         : 0.9356
weighted precision score  : 0.9354
weighted recall score     : 0.9357

================================================================
```

Figure 2: **PoseTracking Model Metrics using SVM**

Below are the performance metrics of the pose tracking models using Convolutional Neural Network

```
Evaluating network...
Best accuracy (on testing dataset): 47.82%
                 precision    recall  f1-score   support

not using phone     0.5455    0.0111    0.0218       540
    using phone     0.4775    0.9899    0.6442       493

    avg / total     0.5130    0.4782    0.3188      1033

[[  6 534]
 [  5 488]]
```

Figure 3: **PoseTracking Model Metrics using CNN**

5.7 Experimental setup

In order to efficiently train and test the model the sample dataset is split into 2 distinct datasets. It is split in to Train dataset of 80% and Test dataset of 20% of sample.

Training dataset is used for training the model and the testing datase is used in the evaluation of the model performance. Various hyper parameter tuning techniques were applied to fine tune the models and then evaluated using performance metrics obtained using test sample dataset.

5.8 Comparison of performance metrics

For the comparison of the best performing metric, hyper parameter tuned models from each classification are tested with test-dataset (20% of initial sample dataset). SVM Classifier is found to perform better than the Convolutional Neural Network.

6 Conclusion

6.1 Key Results

1. For this kind of action detection problems, the pose estimation techniques is found to give better performance results than the object detection algorithms.

2. Noticed that the openpose is too slow for training the pose estimation key points and hence used the AlphaPose techniques.

6.2 Demo Videos

Demo videos are available in demo folder of google drive specified as below. https://drive.google.com/drive/folders/1ZyzRj-G6FolQ-qhJzOx1YkBptYtaWjeS

6.3 Challenges

1. Dataset is not readily available in the internet. Videos has to be recorded and annotated manually.

2. In light of COVID19 situation, it was challenging to obtain the pedestrian recordings. recordings

6.4 Future Work

1. Fusion of the convolutional vision features along with pose points can be atttempted to achieve better pose tracking.

2. Can be enhanced to differentiate phone usage while crossing the road as it poses higher safety concern.