CSL 462: Computer Vision
Distracted Pedestrian Detection

Pratham Gupta
2015csb1024@iitrpr.ac.in
Department of Computer Science,
IIT Ropar
Rajat Sharma
2015CSB1026@iitrpr.ac.in
Department of Computer Science,
IIT Ropar

Abstract

There are many studies that focus on driver distraction and accidents due to it. These studies try to detect distracted drivers and prevent the road accidents due to this. Here the focus is on the responsibility of the driver on road. In this study we tend to focus on the responsibilities of the pedestrians while on road. We try to detect pedestrians being distracted by their cellphones on road, and prevent accidents caused by the same. For this objective, we first detect pedestrians in an image of the road taken using a CCTV. Then we apply a pipeline incorporating localizing humans, estimating the articulated human pose, extracting the images of both the hands and detecting the presence of phone in these cropped images, and weighing these factors to compute a final score which can then be thresholded. We use YOLO (You Only Look Once) [1] Object Detection algorithm for the problem of localizing and detecting the humans in the original image and phone in the cropped images. For detecting the joints and estimating the human pose, we use the Resnet 101 architecture[3] for feature extraction and fully connected layer based neural networks for predicting the positions of the joints[4, 5].

1 Introduction

This Project works to prevent road accidents due to cell phone distraction. Researchers at The Ohio State University say an estimated 1,500 pedestrians were treated in emergency rooms in 2010 for injuries related to using their cell phones while walking. That was double the number of such incidents reported in 2005, even though pedestrian injuries overall have decreased. Deaths due to cell phone distraction while walking accounted for 15% of the total traffic fatalities in 2014 and the number still continues to increase. The proposed model detects people using cell phones while crossing the road using a CCTV camera installed at the traffic lights. On detecting a distracted pedestrian, an alarm can be rung to warn the person using the phone to pay attention to the road. The proposed model tries to estimate the pose of people crossing the road and use this data to detect pedestrians not paying attention to the road.
This might act as a deterrent against Cell Phone Distraction on road to some extent.

2 Related Work

There has been immense work in pedestrian detection over the past few years. Such works generally employ the use of HOG features as extracted from the training set images and then use either ConvNets or Linear SVM for training a classifier to distinguish between humans and non-humans as proposed by Navneet Dalal and Bill Triggs[12]. The main advantage of such work is that HOG features fast to compute for a local region and provide efficient numerical encoding for interest points. From there on, SVMs or ConvNets can be used to analyze patterns and distinguish among human and non-human interest points. This approach, however, is computationally expensive because the HOG features need to be computed for every window and many different scales.
With the advent of Region Proposal Networks based on Convolution Neural Networks, this problem has been significantly reduced. Convolutional Neural Network techniques such as YOLO (You Only Look Once) [1] speeds up this computation for feature extraction because of the use of several convolutional layers. So, the entire image can be convolved with all the learnt filters just once and regions can be extracted in the convolved version of the image. This provides for very fast object classification. YOLO also integrates the Region Proposal Network as part of a single network. This provides a massive speed-up for realtime systems at the expense of a slight decrease in accuracy compared to other deep learning based localization methods such as faster RCNNs.

Human pose detection involves the use of ConvNets as proposed by A. Newell, K. Yang, and J. Deng [13]. The paper presents a stacked hourglass model for estimating the pose in the image, i.e. multiple hourglasses work in tandem for detecting the pose. An hourglass is a unit of ConvNets which has multiple Convolutional and Max Pooling layers. Since, pose estimation requires both local image analysis and image analysis at a larger scale, the hourglass model successively downsizes the image for analyzing the large scale features to as small as 4x4 pixels and then scales up for analyzing the relation to local features. This process is repeated multiple times to minimize error since each hourglass is capable of adjusting its own parameters and so, in case of wrong estimates by an hourglass, the other hourglasses would likely rectify the output.
Other works[4, 5] rely on the use of holes or deconvolution on the output of the final layer of Resnet 101 CNN architecture for feature extraction so as to correctly position the joints detected in the original image.

The current work for detecting the usage of phones in pedestrians from images [11] employ the usage of clustering to cluster these pose estimates using clustering algorithms such as Gaussian Mixture Models (GMM) or k-means clustering. This way, the pose of a pedestrian in a test image can be compared with the clusters and the closest cluster can be used to give a score on the probability of cellphone usage. Another information stream is the detection of hands and extracting HOG features and training a linear SVM to detect the presence of cellphones in the cropped images. These sources of data are then combined to give a confidence score to cellphone usage which is later thresholded.

3 Datasets Used

As our detection of pedestrians using phones is done in several steps, different types of databases were used for each step.
Pedestrian detection part required a dataset of images to detect pedestrians on the road. For this, the pre-trained YOLO based object detection model was originally trained on the following datasets -

1. **COCO (Common Objects in Context) Detection 2015**[6]
   There are 200,000 images for 80 object categories.

2. **ILSVRC 2014**[7]
   Training set contains a hierarchical classification of objects with on an average of 500 images per node.

After detecting pedestrians in the test image, their pose estimation was done using the pre-trained model based on Resnet 101 using the images of the dataset:

1. **MPII Human Pose Dataset**[8]
   This dataset includes around 25K images containing over 40K people with annotated body joints.

After detecting the full body pose of the pedestrian, it is divided into clusters, out of which clusters with pedestrians prone to higher chances of being engaged in cellphone activity are chosen. For clustering, we use the following dataset:

1. **Stanford 40 Actions Dataset[10]**
   There are 9532 images in total with 452 images of people using phones.

From these pedestrians we crop the images of their hands and try to detect cellphones in them. We use the same pre-trained YOLO network for cellphone detection.

Due to limited amount of data, all the data was used for training the model. For testing, new images were obtained from Google Images.

4 Proposed Framework

The model we propose relies on deep CNNs for predicting a probability based measure for persons using phone in the given input image. Figure 1 shows a block diagram of the overall pipeline used. Each module is discussed in further detail in the subsequent sections.

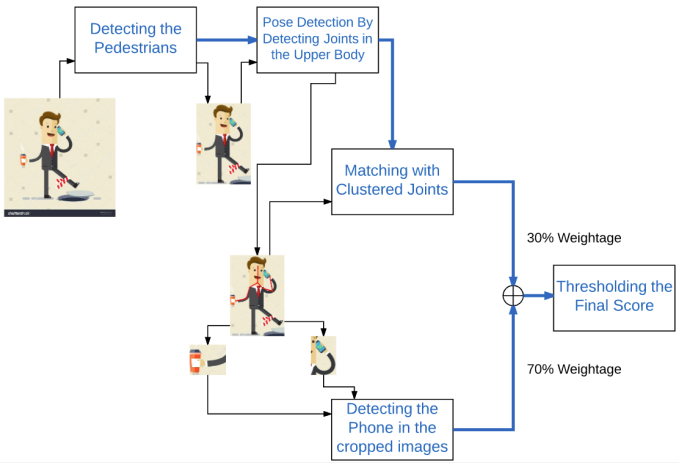

Figure 1: Block Diagram of the Pipeline

4.1 Detecting the Pedestrians

The first step in the proposed pipeline is to detect and localize all the pedestrians present in the input image so that, they can be further analyzed in the following steps. Since the system must work in realtime, this step needs to made as computationally fast and inexpensive as possible. For this purpose, we use a pre-trained YOLO (You Only Look Once) [2] model which has been trained on the ImageNet and the COCO datasets containing images from over a total of 9000 categories.

Figure 2: The Original Image

The model combines the region proposal network and the object classification network into a single deep neural network which allows for blazingly fast object localizations with a high enough accuracy. The model uses a sequence of blocks (convolution, leaky relu and maxpool layers) in succession to form a deep neural network. The output of the convolutional layers is passed into fully connected layers which can both detect and localize objects in one shot. So, all the pedestrians in an image are detected and localized. They are then cropped from the original image and resized to 320 x 140 so as to obtain a standard range scale for the joints obtained in the consequent steps.

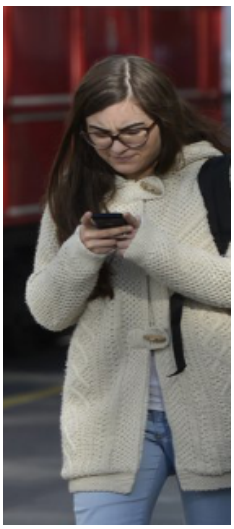

Figure 3: Human detected by the model

4.2 Pose Detection (by detecting joints in the upper body)

The pose of the person inherently encodes upto a certain percentage whether a person is using his/her cellphone or not since cellphone users generally have some particular poses which using their cellular devices. So, the pose of the upper body of the person seems to be a valuable piece of information for detecting the use of cellphone.

For the purpose of detecting pose of the pedestrians, we use a pre-trained model [4, 5] based on the Resnet 101 architecture. The architecture uses Resnet 101 architecture for extracting the features from a given image using subsequent convolutions and max poolings. These features were then passed to fully connected layers which predicted 14 joints in the cropped input image of the pedestrians using regression. The Resnet 101 part of the architecture was fine tuned and the entire network was trained on the MPII Human Pose Dataset consisting of 25,000 images of over 40,000 people, involved in 410 different activities. Due to the substantial size of the dataset, the pre-trained model [4, 5] was used.

Since for our problem, only joints in the upper body are relevant, only 8 joints were considered out of the 14, namely - forehead, chin, both shoulders, both elbows and both wrists. So, for each pedestrian detected in the image, the joints in their upper body was detected. This produced a vector $\bar{\mathbf{j}}$ of 16 dimensions encoding the positions of these 8 joints. $(x_i,\ y_i)$ denote the coordinates of the i^{th} joint. Since, the images were resized to fixed dimensions, i.e. 320 x 140, the vectors across different pedestrians can be compared with each other.

$$\bar{\mathbf{j}} = (x_1, y_1, \ldots, x_8, y_8)$$

where -

$$x_i \in [0,\ 140]$$
$$y_i \in [0,\ 320]$$

4.3 Clustering the Joints

This step occurs only during the training phase. The previous steps were applied onto the Stanford Activity Recognition Dataset [10] to extract pedestrians and their joints. These joints extracted from this dataset was then used for clustering. Since the pedestrians and their joints were extracted experimentally and weren't manually annotated or verified, some natural noise is expected in these joints extracted.

For clustering, K - means clustering was used. All the images of the dataset was used for clustering since this provided a good ground for all the different types of joint positions that might occur. After experimentation, it was observed that for a cluster count of 32, a significantly good enough number of clusters were obtained which were visually verified for representing a good fraction of poses in the upper body for cellphone usage. So, a constant random seed was set and out of these 32 clusters, 21

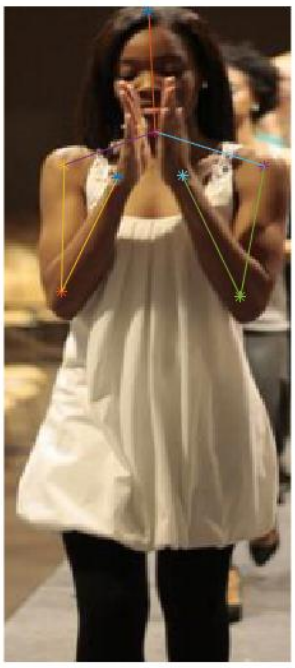

Figure 4: Example of Detected Joints in the Upper Body

clusters were selected which visually appeared to be the most indicative of cellphone usage.

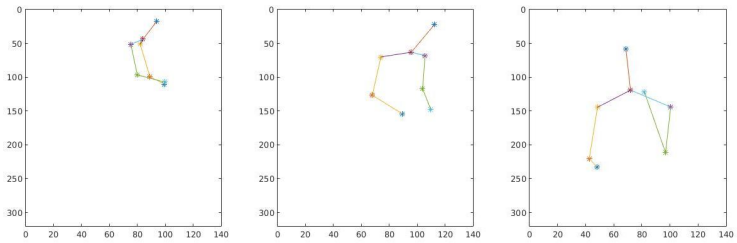

Figure 5: Examples of Positive Cluster Centers, i.e. depicting cellphone usage (either calling or messaging)

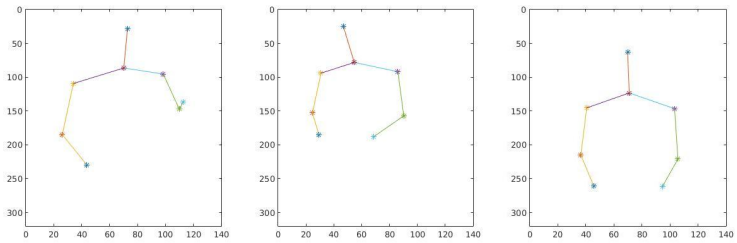

Figure 6: Examples of Negative Cluster Centers, i.e. not depicting cellphone usage

4.4 Matching with Clustered Joints

This step occurs only during the testing phase. The joints of the upper bodies of all the pedestrians detected in the test image are matched with the cluster centers and using the Euclidean distance, the nearest cluster center was selected. The distance served as a score for a probabilistic measure of whether a person is using a cellphone or not. So, the closer the joints of an image are to a Positive Cluster Center, the more is its probabilistic score and similarly, the closer the joints are to a Negative Cluster Center, the lesser is its probabilistic score.

Since, a large number of positive cluster centers were selected, those images which get matched to a nearest negative cluster are outrightly rejected since even their poses don't seem to suggest that they are using their cellphones. For the pedestrians having a positive nearest cluster center, the score of the image is generated for further processing. We notate this score as the *Cluster Score*.

4.5 Detecting Cellphone

The position of joints were used and the images around the ends of both the wrists, including the elbows was cropped. A person using a cellphone would have a high score for cellphone being detected in these cropped images of hands. So, these cropped images provide information regarding the presence of cellphone.

On these cropped images, the YOLO Object Detection was run to detect the presence/absence of cellphone. The YOLO Detector returns a probabilistic score which suggests how strongly does the network believe in the presence of the cellphone. For a particular pedestrian, the score is calculated by taking the max score over both the hands.

Figure 7: Example images of Positive Cellphone Detection

Figure 8: Example images of Negative Cellphone Detection

$$YOLO\ Score = \mathbf{max}(YOLO\ Score_1, YOLO\ Score_2)$$

where -

$$YOLO\ Score_1 = YOLO\ Score\ for\ cellphone\ detection\ in\ left\ hand$$

$$YOLO\ Score_2 = YOLO\ Score\ for\ cellphone\ detection\ in\ right\ hand$$

The score in the end is computed by combining the two scores. Since, the pose does not completely suggest the cellphone usage, it is given lesser weight of 30% while the cellphone detection in hands is given more weight. So, the final score is computed as -

$$Final\ Score = 0.3 * Cluster\ Score + 0.7 * YOLO\ Score$$

This score is then thresholded. A high score leads to the model concluding that the pedestrian is using a cellphone while a low score leads to the conclusion that the pedestrian isn't using a cellphone.

5 Experimental Results

The experimental results obtained suggest that a high enough accuracy, precision and recall were obtained using the proposed model.
The accuracy obtained was **70.89%**. The precision obtained was **77.77%** and the recall obtained was **72.92%**.
The model takes about 22 seconds per image when run on a CPU. So, a very large speedup is expected from GPU usage and so, the system can be deployed for realtime application by sampling images at regular intervals from the video feed.

6 Future Extensions

We believe that the model we proposed can be improved by using a faster RCNN based model instead of YOLO on a GPU. The estimated time for faster RCNN to run on a GPU is 0.2 seconds. For the purpose of this project, due to lack of GPU, time was optimized was CPU computation and so, YOLO was used.

In addition to this, for clustering, currently K-means clustering was employed. This is a hard clustering algorithm. In the future, softer clustering algorithms such as Gaussian Mixture Models can be used. K-means being a hard clustering algorithm might suppress relevant clusters which are in minority since the dataset used for clustering contains a very small subset of images with people engaged with their cellphones.

The model proposed works only for still images sampled at regular intervals from the video feed. So, in each subsequent image, all the humans are detected completely from scratch being completely oblivious to the detection of humans in the previous frame. So, a tracking based algorithm such as Kalman filters can ever further increase the computational efficiency due to a sharp decrease in redundant computations, enabling the use of more accurate algorithms such as faster RCNNs to run.

Lastly, we were greatly limited by the very small amount of data we had and we couldn't use a single dataset due to the absence of a correctly annotated dataset for this problem. So, a cross-dataset technique had to be used. In the future, more correctly annotated relevant data can be collected specifically for this problem which would significantly improve the accuracy of the proposed model.

7 Conclusion

So, we conclude that the model we propose gives a good enough accuracy on the test images used. Currently, the model was only tested on CPU which took about 22 seconds. When run on a GPU, it would likely be sped up quite a lot. So, the model proposed can easily be deployed for realtime systems.

8 References

[1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi
You Only Look Once: Unified, Real-Time Object Detection, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June, 2016

[2] Joseph Redmon and Ali Farhadi
YOLO9000: Better, Faster, Stronger, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), December, 2016

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun
Deep Residual Learning for Image Recognition, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015

[4] Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres and Bernt Schiele
ArtTrack: Articulated Multi-person Tracking in the Wild, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), May, 2017

[5] Eldar Insafutdinov and Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka and Bernt Schiele
DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model, in ECCV'16

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick and Piotr DollÃąr
Microsoft COCO: Common Objects in Context. arXiv:1405.0312

[7] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei
ImageNet Large Scale Visual Recognition Challenge IJCV, 2015

[8] Mykhaylo Andriluka and Leonid Pishchulin and Peter Gehler and Schiele, Bernt.
2D Human Pose Estimation: New Benchmark and State of the Art Analysis, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June, 2014

[9] Vincenzo Lomonaco and Davide Maltoni.
"CORe50: a new Dataset and Benchmark for Continuous Object Recognition". arXiv preprint arXiv:1705.03550 (2017).

[10] B. Yao, X. Jiang, A. Khosla, A.L. Lin, L.J. Guibas, and L. Fei-Fei. Human Action Recognition by Learning Bases of Action Attributes and Parts. *Internation Conference on Computer Vision (ICCV)*, Barcelona, Spain. November 6-13, 2011.

[11] Akshay Rangesh, Eshed Ohn-Bar, Kevan Yuen and Mohan M. Trivedi.
Pedestrians and their Phones - Detecting Phone-based Activities of Pedestrians for Autonomous Vehicles, IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016

[12] N. Dalal and B. Triggs
Histograms of oriented gradients for human detection in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE, 2005, pp. 886âĂŞ893.

[13] A. Newell, K. Yang, and J. Deng
Stacked hourglass networks for human pose estimation, 2016.

[14] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna and Tomaso Poggio
Pedestrian Detection Using Wavelet Templates Proceedings of Computer Vision and Pattern Recognition, Puerto Rico, June 1997

[15] Cristian Sminchisescu and Alexandru Telea *Human Pose Estimation From Silhouettes A Consistent Approach Using Distance Level Sets* 10th International Conference on Computer Graphics,Visualization and Computer Vision (WSCG '02), Feb 2002