WASVI : Walking Assistance System for the Visually Impaired in real-time

Jungwook Han[1], **Hogeon Yu**[1], **Inkwon Lee**[1]
[1]Carnegie Mellon University
{jungwoo2, hogeony, inkwonl}@andrew.cmu.edu

Abstract

Introduction

It is said that 4% of the world's population are visually impaired. The visually impaired are almost impossible to walk and live alone because they cannot see. Therefore, they go out with the help of an assistant or a guide dog. However, the fact that it is difficult to maintain them financially significantly reduces the life radius of the visually impaired. There is a lack of financial support, both nationally and internationally. Also, the equipment for the visually impaired installed on the street is not properly maintained. A lot of wear and tear are seen there. Recently, as artificial intelligence technology develops in the vision field and audio field, they are supporting the visually impaired. However, there are still many challenges.

First, the vision models that can replace the eyes of the visually impaired are heavy. Therefore, the inference time is long, so it is difficult to apply them to the real world to support the visually impaired in real time. In order to be reflected in real-time, the model should be lighter, but this leads to a trade-off in which the accuracy is lowered. Second, the current systems do not properly reflect the surrounding environment. For example, the shape and color of a traffic light signal varies by country and region. Current object detection models judged it only as a traffic light and cannot detect the signal of the traffic light. It is trivial because the shape of signal is various over the country.

We propose a Waking Assistant System for the Visually Impared (WASVI) based on factors that may occur when the visually impaired crosses the street or walks the sidewalk. WASVI detects static objects such as traffic lights and crosswalks, and detects dangerous situations for dynamic objects such as movement of surrounding pedestrians, so that the visually impaired can recognize their surroundings.

Additionally, in order to operate through a mobile phone, the size of the model was reduced, and the application was made light and easy to access as well. To make the system light, we use YOLO-r (Wang, Yeh, and Liao 2021) object detection model which is state-of-the-art real-time object detection model. Also we made our own MOT(multi-object tracking) algorithm because MOT based on deep learning(Milan et al. 2016; Wang et al. 2019) takes long time to track each object. For traffic light signal detection, we focused on the signal in USA and detect the signal by image processing. For the pedestrian intention detection, we refer the multi-object pedestrian intention prediction model (Bouhsain, Saadatnejad, and Alahi 2020) and fit to our purpose. Because of its simple architecture, the performance of the bounding box prediction is comparable to the ones yielded by much more complex architectures around more than 2 times faster.

We have three major contributions.

- First, we induced the application to the real time system by greatly reducing the processing time per frame.
- Second, we made it possible to understand the signal of a traffic light through image processing. In other words, it can be applied in any area as long as we know the signal shape of the traffic light.
- Last, we can predict the surrounding pedestrian movement intention with light-weight MOT algorithm and intention detection model.

As a result our system detects the traffic light signal whose figure is various over the country. Also it can predict the surrounding pedestrian movement intention so that it can alert the visually impaired to watch out. The total operating time is under 40ms per frame in average which is reasonable to be real-time system.

In section II, we introduce the background and related work. In section III, we are going to talk about the dataset that we used. In section IV, the propose system architecture is discussed. In section V, we will show the result and evaluation. In section VI, we conclude our work and explain the ways to apply in real world. Last, in section VII, the future work will be discussed.

Related work

Object Detection

There are several sensors(ultrasonic, rider, radar) for performing object detection. However, such sensors for the object detection have a problem of having to pay a high cost for the visually impaired to carry them. With the recent development of computer vision, several studies have been conducted to detect the objects through standard cameras using

deep learning. Object detection is to determine the exact location of a object by identifying all objects in each frame of camera.

In the past research, to find the object, the model was designed to detect the feature of object. Scale Invariant Feature Transform(SIFT), Speeded-Up Robust Features(SURF) (Bay, Tuytelaars, and Van Gool 2006), Haar, Histogram of Oriented Gradients(HOG) (Surasak et al. 2018) are typical studies. In addition, Deformable Part-based Model(DPM) (Li et al. 2018) designed features by dividing objects into several parts, and performed object detection through machine learning such as Support vector machine.

Recently, CNN has emerged in the image classification field, overwhelmingly exceeding the results of existing studies, and many studies are also being conducted to apply CNN in the object detection field. R-CNN generates the Region Proposal and learns the CNN based on it to locate the object in the image. Fast R-CNN (Girshick 2015), Faster R-CNN (Ren et al. 2015) solved the drawbacks of R-CNN and R-FCN feed the input image to the CNN to build a faster object detection. These two stage detectors provide an adequate accuracy, but have the disadvantage of taking a long time to calculate. In addition, in order to apply object detection in the real life, a processing speed close to real time is required. To solve this problem, YOLO v1 (Redmon et al. 2015), which consist of one deep learning network, has emerged. In recent follow-up studies, YOLO v2 (Redmon and Farhadi 2016), YOLO v3 (Redmon and Farhadi 2018), YOLO-LITE (Pedoeem and Huang 2018), YOLO v3-Tiny (Adarsh, Rathi, and Kumar 2020), and Efficientdet, which show high object detection performance and fast detection speed enough to operate in mobile, have been studied. For real-time object detection model, YOLO-r (Wang, Yeh, and Liao 2021) is the state-of-the-art real-time object detection model. YOLO-r is what we used for our system to detect the object.

Multi Object Tracking

Dynamic objects are another factor that obstructs the walking of the visually impaired. Unlike static objects, their location changes every moment, and it is hard to predict the risk factors. There are many researches to solve these problems in the field of the computer vision and the autonomous driving. They solve it through an algorithm based on the pedestrian head orientation (Rasoul, Kotseruba, and Tsotsos 2017) and pose (Fang and López 2018).

On another research way, the trajectory-based algorithm predicts the future location by observing the pedestrian's past motion history. However, in the real-world camera setting, it is difficult to observe the accurate 3D depth (Keller, Hermes, and Gavrila 2011) of the pedestrian, so it is hard to predict the trajectory of the pedestrian.

The state-of-the-art model (Bhattacharyya, Fritz, and Schiele 2017) proposes two-stream encoder-decoder scheme. They treat the pedestrian as a bounding box and predict the future bounding box location. However, it does not use the visual features of pedestrians.

We figure out that the Multi Object Tracking based on deep learning has a disadvantage with respect to the long inference time.

Dataset

- COCO dataset

  The COCO dataset (Lin et al. 2014) is a dataset created for the purpose of computer vision tasks such as object detection, segmentation, and keypoint detection. In fact, if you read a paper on object detection, you can see COCO 2017 among the datasets that are often used for performance evaluation in the paper. In addition, many object detection libraries provide pre-trained models with these COCO datasets. The training dataset consists of 118,000 images, the validation dataset consists of 5,000 images, and finally the test dataset consists of 41,000 images.

  In this work, we also use this dataset to train the YOLO-r model.

- JAAD dataset

  JAAD (Kotseruba, Rasouli, and Tsotsos 2016) is a dataset for studying joint attention in the context of autonomous driving. The focus is on pedestrian and driver behaviors at the point of crossing and factors that influence them. To this end, JAAD dataset provides a richly annotated collection of 346 short video clips (5-10 sec long) extracted from over 240 hours of driving footage. These videos filmed in several locations in North America and Eastern Europe represent scenes typical for everyday urban driving in various weather conditions.

  Bounding boxes with occlusion tags are provided for all pedestrians making this dataset suitable for pedestrian detection. Behavior annotations specify behaviors for pedestrians that interact with or require attention of the driver.

  We use this dataset to predict the pedestrian's movement intention with light-weight LSTM based model.

- ImVisible dataset

  ImVisible dataset has the image dataset of street intersections, labelled with the color of the corresponding pedestrian traffic light and the position of the zebra crossing in the image. We also use this image dataset to train the YOLO-r model to detect the traffic light.

- Our own dataset

  Finally, we also made our own datasets to predict the movement intention of pedestrian more correctly. The model that we used to predict the pedestrian movement intention is trained using JAAD dataset. But, as mentioned as above, the data clip is focusing on the driver and crossing pedestrian. So it didn't fit in the direction we were going. And the number of JAAD dataset was 347 clips that was not enough for our task. So we made our own dataset clip. The clip was focusing on the pedestrian and corssing pedestrian, so that the position of the pedestrian and speed of the pedestrian is detect more correctly. We use both our dataset and JAAD dataset to train the model to predict the surrounding pedestrian movement intention.

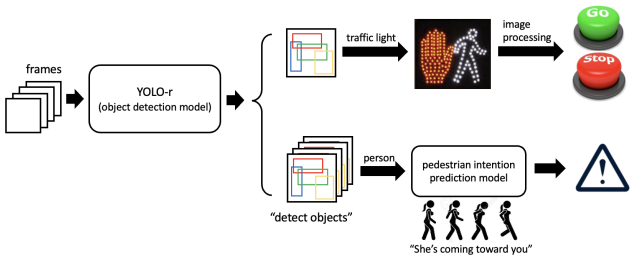

Figure 1: Overall System Architecture of WASVI

Methods

Figure 1 shows the overall architecture of the WASVI system. There are three main subsection in this section. First object detection and MOT is explained. Second how to address the signals from the traffic light is explained. Last, we explain the method to predict the surrounding pedestrian movement intention.

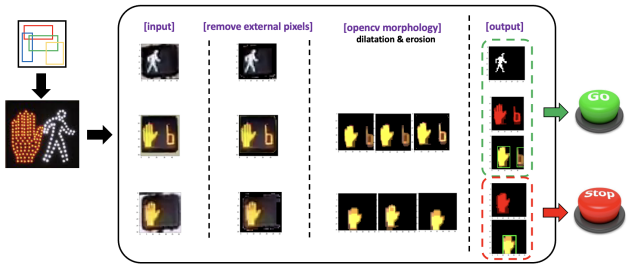

Figure 2: Image processing to detect the signals from the traffic light

Object Detection and Multi Object Tracking

We use YOLO-r (Wang, Yeh, and Liao 2021) object detection model for the real-time system. This model is the state-of-the-art in real time object detection. It can detect a lot of object including person and traffic light.

Although it can detect the traffic light, it can not consider the signal from it. So we need to make it possible to understand which signal is coming from the traffic lights. We use the rule based approach to make the system understand the signals. It will be discussed in detail in next subsection.

We also use our own MOT algorithm to track the history of the bounding box of person object to predict the intention of each person. For that, we needed to track the movement of each bounding box. In this system, the greedy algorithm is used to track the each object in each frame and assign the unique id to them. So that we can track the object by its id without any burden to the overall system.

Image Processing for the traffic light

As mentioned in previous subsection, YOLO-r model can only detect the traffic light itself, not the signals from it. But the signals from the traffic light is very significant to the pedestrian including the visually impaired. We focused on the signals used in USA. The traffic light signal in the figure 2 shows how it looks like. The system understands the signal of the traffic light through the following process (K. Romic 2018).

First, the traffic lights is detected by YOLO-r model and we can find the input image, white human shape or orange palm shape. Second, it remove the external pixel which means it makes the edge pixels black. Then by repeating dilatation and erosion using OpenCV morphology library, make the signals more clear so that the system clearly recognize the shape of white person or the shape of orange palm. Finally, if the image contains the white pixels with person shape, the system consider this sign as "cross". If the orange palm is detected with decimal number on the right side and it started blinking, the system understands that it will soon turn into a stop sign. If the orange palm is detect without any decimal number or blinking, the system considers it as "stop" sign.

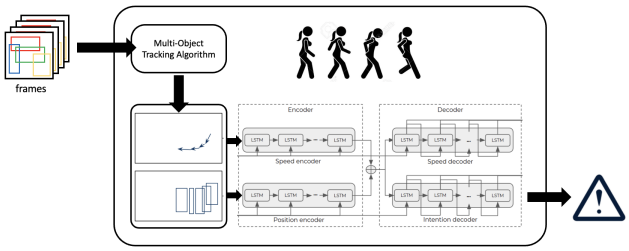

Figure 3: Pedestrian Movement Intention Prediction

Pedestrian Intention Prediction

To predict the surrounding pedestrian movement and to alert the user to watch out, the system uses the pedestrian intention prediction model which is based on LSTM. The figure 3 shows the workflow of the Pedestrian Intention Prediction.

First, the bounding box of each object is tracked by our MOT algorithm and we only track the person object. To predict the future movement of the pedestrian, the system needs to know the past bounding box location histories. We let the system predict the next position from the previous 16 location histories.

Second, if the system has the information about 16 previous location history of person object, it calculates the instantaneous speed and the position in each frame. The model uses these two information as input. The model consists of encoder and decoder. The encoder has the position encoder and the speed encoder. The input about speed goes through speed encoder and the input about position goes through position encoder. The decoder receives the concatenated output from both encoders and calculates the very next speed and predicts the intention. As a result, the system detects the person who is walking to the user and makes the alert to user to watch out.

As a result, if the surrounding pedestrian is walking toward the user, the system calculates the speed and current position of the pedestrian. If the system determines that the pedestrian's speed and location is unsafe, it alerts the user.

The output example figures will be shown in next section in detail.

Experiments and Result

Time Break-down

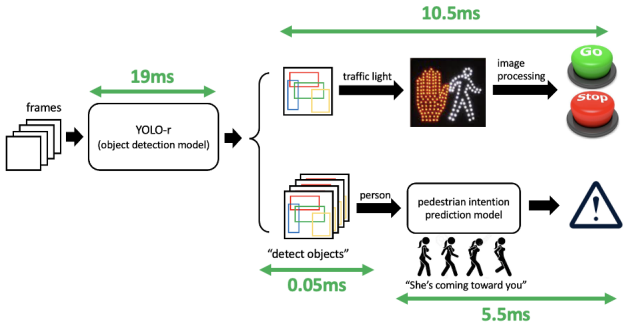

Figure 4: Time break-down for the system

In this section, we show the reusult and discuss the experiment. We use 24 frames per second video, and the goal criterion to satisfy the real time system condition is that the average latency per frame is less than 40 ms. As seen as figure 4, the maximum of the total operation time per frame is 35.05ms. 35.05ms is a number that meets our criteria. Furthermore, the pedestrian movement intention is predicted once every 16 frames, the actual operating time would be less than 35.05ms. The figure 5 below visualizes the operating time comparison through a bar graph.

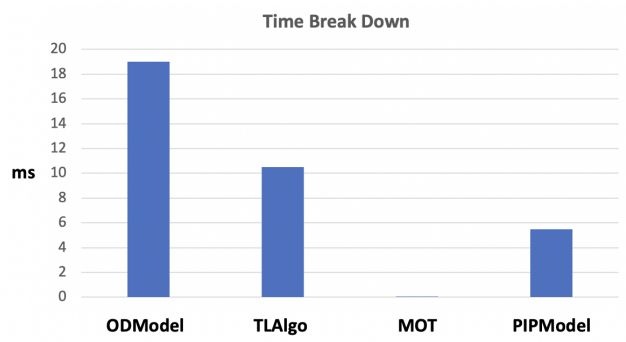

Figure 5: Time break-down for the system in bar graph

- ODModel : Object Detecting Model
- TLAlgo : Traffic Light Signal detecting Algorithm
- MOT : Multi Object Tracking Algorithm
- PIPModel : Pedestrian Intetion Prediction Model

For more detail, figure 5 shows how much time each element takes up. With respect to the object detection model, YOLO-r, the time it occupies is dominant, around 19ms. Fortunately, by changing the deep learning based MOT model to the light-weight rule-based MOT algorithm, the operation time is significantly reduced to less than 0.1ms. Unexpectedly, the image processing time for the traffic light signal is longer than the time taken by the pedestrian intention prediction model. It seems that the burden of image processing to detect the traffic light signals was greater than our expectation. On the other hand, the operating time for the pedestrian intention prediction model was about 4 times shorter than the operating time of the object detection model which is very successful.

Output result

We test our model in real world. We test how well the system can detect the surrounding risk factors and the surrounding information. Both figure 6 and figure7 show the result output of our system. And it will be discussed in detail.

Figure 6: system output example 1

For figure 6, the system can detect the traffic light signal well. If the traffic light shows the cross sign with white human shape, the system detects it properly. If the orange palm is shown in the traffic light, the system interpret differently depending on whether it is blinking or not. If the signal is blinking with the orange palm shape, the system notify the user to hurry up. On the other hand, if the signal is not blinking, the user should stop. And as seen as figure 6, our system can detect the traffic light signal very well.

Figure 7: system output example 2

As seen as the figure 7, the system can also detect the movement intention of the surrounding pedestrian. If the surrounding pedestrian is walking toward the user or is crossing the user, the system detects this dangerous situation with the red box and makes the alert to user to watch out. On the other hand, if the surrounding pedestrian is passing by or does not obstruct the user's path, the system doesn't issue the warning

with the green box. It can predict the pedestrian movement intention very well.

As a result, the total operation time of each frame was 35.05ms, and it took 841ms to process 24 frames corresponding to 1 second. It is reasonable to be real time system.

Communication with the visually impaired

To support the visually impaired to assist their walk, there are two main parts. First part is to detect and visualize the surrounding risk factors. Second part is to translate the visual information into the sound information so that the user, the visually impaired can understand the surrounding situations.

So far, we have focused on detecting surrounding risk factors through visualization and minimizing the operating time it takes. Now we will discuss about second part, how the system interact with the visually impaired.

Figure 8: How to interact with the visually impaired

Since they are visually impaired, they need the clue of the dangerous factor via sound information, so that they can understand what our system detects as a dangerous factor. So we need some rule to make them understand. As figure 8 above, the system should translate the visual information into the sound information so that the user, especially the visually impaired can recognize the surrounding situation and prepare for the unexpected issues.

First, if the traffic light signal is a white human shape which means the pedestrians can cross the road. While they cross the road, there can be the pedestrian who is walking toward the user. If the system detects the intention of the pedestrian movement and considers it as a risk factor, it sounds a buzzer to notify the user and the people around the user. It can notify the people around user to watch out the user and likewise, it can notify the user to prepare for a bump. The same operation is performed when the user is walking on the sidewalk.

Second, if the traffic light signal is a orange palm shape without blinking, it means the pedestrians should stop and not cross the road. Then it alert to only user, not the people around him to stop. The system is saying "Stop" out loud for the user to hear. So the user can notify the traffic light signal and stop in front of the cross walk.

Last, if the traffic light signal is a orange palm shape with blinking, we consider two cases. When the system detects a signal from a traffic light, if the signal changes to blinking, it notifies the user to hurry up by saying "Hurry up" aloud. Otherwise, if it is already blinking, the system notify the user to stop even if it is possible to cross the road at a fast pace.

So far, we discuss about how the system communicates with the visually impaired with the sound.

Future Work

Using GPS information

Figure 9: Fusion with the GPU information

Using GPS information, the user can input the destination and the system can guide the user to the destination based on the location of the user. Nowadays, GPS based location guidance application is predominant and easy to access such as Google Maps. By applying the API the Google Maps provide, it is possible to make more efficient and helpful application.

Road Tracking Model

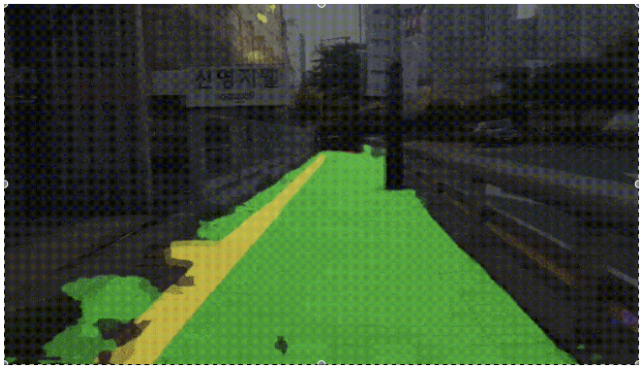

Figure 10: Apply the Road tracking model

In our work, we assume that the user walks well along the road with their canes. So far, it is not enough to be used in the real world. However, if we can develop the application further, we can apply the road tracking function to guide the user to follow the curved road well as seen as figure 10.

Conclusion

In this section, we conclude our work as below. We proposed the "Walking Assistance System for the Visually Impaired" for real-time system.

Our first contribution is to make the system possible to be applied in the real world by reducing the operating time to a minimum. The operating time was up to 35.05ms per frame, and the average operating time was lower than this. Time break down verify that algorithms work well and it is reasonable to be a real time system.

Second contribution is to detect the traffic light signal which is various over the country. We focus on the traffic light shape in USA and do the image processing to detect the signal from it. In addition, it protects the user's safety by sending an appropriate signal to the user according to the traffic light signal.

Last contribution is to make the system possible to predict the surrounding pedestrian movement intention. As same as the case of the traffic light detection, it protects the user's safety by notifying the proper sign to the user when the surrounding pedestrian is considered as a risk factor.

Sometimes, rather than system design through deep learning, it is more beneficial to process by adding a rule based processing as needed from the point of view of a real time system.

References

Adarsh, P.; Rathi, P.; and Kumar, M. 2020. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 687–694. doi:10.1109/ICACCS48705.2020.9074315.

Bay, H.; Tuytelaars, T.; and Van Gool, L. 2006. SURF: Speeded Up Robust Features. In Leonardis, A.; Bischof, H.; and Pinz, A., eds., *Computer Vision – ECCV 2006*, 404–417. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-33833-8.

Bhattacharyya, A.; Fritz, M.; and Schiele, B. 2017. Long-Term On-Board Prediction of People in Traffic Scenes under Uncertainty. *CoRR* abs/1711.09026. URL http://arxiv.org/abs/1711.09026.

Bouhsain, S. A.; Saadatnejad, S.; and Alahi, A. 2020. Pedestrian Intention Prediction: A Multi-task Perspective. doi:10.48550/ARXIV.2010.10270. URL https://arxiv.org/abs/2010.10270.

Fang, Z.; and López, A. M. 2018. Is the Pedestrian going to Cross? Answering by 2D Pose Estimation. *CoRR* abs/1807.10580.

Girshick, R. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

K. Romic, I. Galic, H. L. K. N. 2018. Real-time Multiresolution Crosswalk Detection with Walk Light Recognition for the Blind.

Keller, C. G.; Hermes, C.; and Gavrila, D. 2011. Will the Pedestrian Cross? Probabilistic Path Prediction Based on Learned Motion Features. *DAGM* LNCS 6835: 386–395.

Kotseruba, I.; Rasouli, A.; and Tsotsos, J. K. 2016. Joint Attention in Autonomous Driving (JAAD). doi:10.48550/ARXIV.1609.04741. URL https://arxiv.org/abs/1609.04741.

Li, J.; Wong, H.-C.; Lo, S.-L.; and Xin, Y. 2018. Multiple Object Detection by a Deformable Part-Based Model and an R-CNN. *IEEE Signal Processing Letters* 25(2): 288–292. doi:10.1109/LSP.2017.2789325.

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; and Dollár, P. 2014. Microsoft COCO: Common Objects in Context. doi:10.48550/ARXIV.1405.0312. URL https://arxiv.org/abs/1405.0312.

Milan, A.; Leal-Taixe, L.; Reid, I.; Roth, S.; and Schindler, K. 2016. MOT16: A Benchmark for Multi-Object Tracking. doi:10.48550/ARXIV.1603.00831. URL https://arxiv.org/abs/1603.00831.

Pedoeem, J.; and Huang, R. 2018. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. doi:10.48550/ARXIV.1811.05588. URL https://arxiv.org/abs/1811.05588.

Rasoul, A.; Kotseruba, I.; and Tsotsos, J. K. 2017. Are They Going to Cross? A Benchmark Dataset and Baseline for Pedestrian Crosswalk Behavior. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 206–213. doi:10.1109/ICCVW.2017.33.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2015. You Only Look Once: Unified, Real-Time Object Detection. doi:10.48550/ARXIV.1506.02640. URL https://arxiv.org/abs/1506.02640.

Redmon, J.; and Farhadi, A. 2016. YOLO9000: Better, Faster, Stronger. doi:10.48550/ARXIV.1612.08242. URL https://arxiv.org/abs/1612.08242.

Redmon, J.; and Farhadi, A. 2018. YOLOv3: An Incremental Improvement. doi:10.48550/ARXIV.1804.02767. URL https://arxiv.org/abs/1804.02767.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

Surasak, T.; Takahiro, I.; Cheng, C.-h.; Wang, C.-e.; and Sheng, P.-y. 2018. Histogram of oriented gradients for human detection in video. In *2018 5th International Conference on Business and Industrial Research (ICBIR)*, 172–176. doi:10.1109/ICBIR.2018.8391187.

Wang, C.-Y.; Yeh, I.-H.; and Liao, H.-Y. M. 2021. You Only Learn One Representation: Unified Network for Multiple Tasks. doi:10.48550/ARXIV.2105.04206. URL https://arxiv.org/abs/2105.04206.

Wang, G.; Wang, Y.; Zhang, H.; Gu, R.; and Hwang, J.-N. 2019. Exploit the Connectivity: Multi-Object Tracking with TrackletNet. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, 482–490. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368896. doi:10.1145/3343031.3350853. URL https://doi.org/10.1145/3343031.3350853.