Predicting pedestrian crosswalk behavior using Convolutional Neural Networks

Eric Liang and Mark Stamp
Department of Computer Science, San Jose State University, San Jose, CA, USA

ABSTRACT
Objective: Pedestrian accidents contribute significantly to the high number of annual traffic casualties. It is therefore crucial for pedestrians to use safety measures, such as a crosswalk, and to activate pedestrian signals. However, people often fail to activate the signal or are unable to do so – those who are visually impaired or have occupied hands may be unable to activate the system. Failure to activate the signal can result in an accident. This paper proposes an improvement to crosswalk safety by designing a system that can detect pedestrians and trigger the pedestrian signal automatically when necessary.

Methods: In this study, a dataset of images was collected in order to train a Convolutional Neural Network (CNN) to distinguish between pedestrians (including bicycle riders) when crossing the street. The resulting system can capture and evaluate images in real-time, and the result can be used to automatically activate a system such as a pedestrian signal. A threshold system is also implemented that triggers the crosswalk only when the positive predictions pass the threshold. This system was tested by deploying it at three real-world environments and comparing the results with a recorded video of the camera’s view.

Results: The CNN prediction model is able to correctly predict pedestrian and cyclist intentions with an average accuracy of 84.96% and an absence trigger rate of 0.037%. The prediction accuracy varies based on the location and whether a cyclist or pedestrian is in front of the camera. Pedestrians crossing the street were correctly predicted more accurately than cyclists crossing the street by up to 11.61%, while passing (i.e., non-crossing) cyclists were correctly ignored more than passing pedestrians, by up to 18.75%.

Conclusion: Based on the testing of the system in real-world environments, the authors conclude that it is feasible as a back-up system that can complement existing pedestrian signal buttons, and thereby improve the overall safety of crossing the street. Further improvements to the accuracy can be achieved with a more comprehensive dataset for a specific location where the system is deployed. Implementing different computer vision techniques optimized for tracking objects should also increase the accuracy.

Introduction
Traveling by foot or bicycle are popular transportation methods for people wanting to cover short distances. For such pedestrians, a common danger they face is car accidents when crossing a road. There were more than 7,000 deaths and 104,000 emergency hospital visits for pedestrians in the year 2020 (Center for Disease Control and Prevent [CDC] 2022). Nearly 25% of these fatalities took place at intersections, suggesting the importance of intersection safety (Governor’s Highway Safety Association [GHSA] 2022). There are already several safeguards for pedestrians crossing the road, with the most notable being the crosswalk. Activating a crosswalk increases the safety of pedestrians by over 30% in some scenarios, thereby reducing the number of potential accidents (Chen et al. 2012). However, there are still many cases where pedestrians do not use the crosswalk, whether from carelessness or an inability to do so. People with occupied hands may not be able to put their belongings down, and others who are visually impaired might have difficulty finding the button to activate the pedestrian signal. Furthermore, there are many people who simply ignore the signals.

To reduce the number of accidents from pedestrians crossing the street, a system that is complimentary to current pedestrian signals is considered. Specifically, this study develops and tests a system that can be used to automatically activate pedestrian signals in cases where the button is not pressed. The system uses model images to train a Convolutional Neural Network (CNN) to accomplish this task. CNNs, which loosely model the neurons in the human brain, can accurately classify previously unseen images (Hardesty 2017). CNNs were chosen for this research, since they achieve state of the art accuracy, and they are efficient enough to be used in a real-time, real-world environment.
To recognize pedestrians, a dataset is developed containing images of the three primary objects that are in front of a crosswalk: pedestrians, cyclists, and streets. The authors have also designed an experimental prototype system that incorporates a CNN to automatically activate pedestrian signals. To test the program, it is deployed at three different types of crosswalks to determine its accuracy. The trained CNN achieves an accuracy in excess of 95%, and when the system is deployed in a challenging real-world environment, it can automatically identify more than 85% of pedestrians, with minimal triggers when there are no pedestrians in frame.

Machine learning involving pedestrian detection has been heavily researched in recent years from both vehicle and intersection perspectives. Many established methods of pedestrian detection at intersections utilize surveillance cameras, with CNN being a common model of choice. More detailed predictions can be made through video feed, which allow for the tracking of trajectory and motion (Antonio and Romero 2018). Pedestrian prediction via CNN was first implemented by Sermanet et al. (2013) through an unsupervised deep learning approach tested on the Caltech pedestrian dataset. Tomê et al. (2016) improved upon CNN pedestrian detection in automobiles by using the 8-layer AlexNet model, optimizing the architecture, and employing region proposals. Ng and Kwok (2020) coupled object detection in traffic systems to optimize pedestrian crossing in Hong Kong. With the importance of datasets and models in deep learning, this paper expands the methods used by Tomê et al. specifically for pedestrian signals with a custom dataset and more advanced GoogLeNet model (Albawi et al. 2017), as well allowing for an alternative trigger for cyclists.

**Methods**

**Data collection**

One of the most important aspects of any machine learning model is the dataset used to train the model, as the quality of the model predictions is dependent on the quality of the training data. Consequently, it is important for us to collect data that accurately represents the classes that the model wants to distinguish. The dataset needs to be sufficiently varied to allow the model to generalize the classes so that it can accurately classify images that it has not previously seen. In practice, this means that the data needs to include a wide variety of subjects, as well as different angles, background settings, and so on.

For the signal activation problem, the model should distinguish between the three most common views that can appear at a crosswalk: people walking (pedestrians), people biking (cyclists), and an absence of people (street). Although pedestrians and cyclists are both people who need to use the pedestrian signal, the model distinguishes between the two classes in case users of the prediction system wish to trigger different actions based on a person’s method of travel.

The authors have manually collected pictures for the dataset, as this allows for the greatest flexibility. Since the prediction system is meant to assist the pedestrian signal button, the types of crosswalks used in the dataset are ones with preexisting pedestrian signals installed; these include those found in schools, parks, neighborhoods, and shopping areas. A few key factors are varied throughout the images to create a diverse dataset. For instance, the number of pedestrians/cyclists in the images range from 1 to 5, as clusters of pedestrians may pass by at once. The angles of the images are altered to mirror the installation of the signal button, which are mostly street level. The type of surroundings was also changed, with some locations like the city having more buildings and people in the background and other locations like the suburbs being emptier. Although the images were mostly captured during daytime, they contain a few different weather patterns, including fogginess and cloudiness, and degrees of lighting. At this point, the system is experimental, so there is no need to use automatic cameras to produce the data.

To avoid having the models overfit some classes at the expense of others (Stamp 2022), the dataset is constructed with the ratios between classes similar to what might be expected in practice. The dataset includes a total of 2000 images, of which 207 are cyclists, 668 are pedestrians, and 1125 are streets. The ratio of positive to negative training images is 7:9, which does not over-represent either the negative class (street) or positive classes (pedestrians and cyclists). The ratio between cyclists and pedestrians is roughly 1:3.2, so as to account for the greater number of pedestrians that are expected to be seen crossing the street, as compared to cyclists.

**CNN model**

CNNs are a widely used class of deep learning models that were developed for classifying images. Similar to other neural networks, CNNs contain multiple layers and assign weights to different parts of the input. However, a unique feature of the CNN is its convolutional layers which, in effect, scan the image and model the important features of an image. Through multiple convolutional layers, the CNN is able to achieve higher levels of abstraction, and ultimately classify images via a fully connected output layer (Albawi et al. 2017). Due to their convolutional layers, CNNs provide a high degree of translation invariance, which is crucial when dealing with images. As a result, CNNs avoid the overfitting that tends to occur when fully connected models are used with images, while also vastly reducing the number of weights that must be learned, thereby offering much greater training efficiency.

For this study’s CNN model, transfer learning is employed; that is, a model that has been pretrained on a vast image dataset is used, which only needs to retrain the output layer. In this way, the model can take advantage of the learning that is represented by the hidden layers of the pretrained model, while training the model for the task at hand, namely, distinguishing between images of pedestrians, cyclists, and streets. Specifically, the system uses the GoogLeNet convolutional neural network, which is a pretrained architecture with 22 layers. Using a pretrained model allows the program to save a great deal of training time.
without sacrificing accuracy. The GoogLeNet architecture is a widely used model for computer vision that has proven to be both fast and accurate and is suitable for real-time models (Szegedy et al. 2015). The various layers and interconnections of the GoogLeNet architecture are shown in Figure A.5 in the Online Appendix.

Due to its efficiency and accuracy, an Adam optimizer is used. The Adam optimizer is an efficient first-gradient optimizer that has a limited memory requirement, making it ideal for training the pedestrian prediction system (Kingma and Ba 2014). A cross entropy loss function is also used, as it is useful for classification problems (Paszke et al. 2019).

**Hyperparameter tuning**

Hyperparameters are model parameters that must be set by a user before training a model. Tuning hyperparameters is a vital aspect of training any machine learning model, as the hyperparameters can have a major impact on the resulting accuracy. Since certain hyperparameters are already defined in the GoogLeNet architecture, only the learning rate, number of epochs, and the batch size of the model needs to be optimized. It is important to tune these hyperparameters to optimize the accuracy, efficiency, and to avoid common problems, such as overfitting (Radhakrishnan 2017).

This study identifies suitable hyperparameter values by using a grid search over the learning rate, batch size, and number of epochs. This allows it to determine a combination of these three values that is optimal within the search space. When testing the hyperparameters, 1900 images are used for training the model and 100 images are used for testing the model. The most optimal combination of these hyperparameters is a learning rate of 0.0005, a batch size of 32, and 4 epochs (Table A.2, online Appendix).

When plotting the accuracy and loss for both training and validation phases of the GoogLeNet model on a graph, the loss approaches 0 after two epochs for both datasets, whereas the accuracy remains at 0.99 for the training dataset and 0.97 for the validation dataset, with slight increases as the number of epochs increase (Figure A.3, online Appendix). The training and validation accuracy track closely, as do the loss plots, indicating that overfitting is not a significant issue for the models when trained through the 14 epochs considered. It can be observed that near-optimal accuracy and loss are achieved with four epochs, so for the sake of efficiency, all subsequent experiments are trained for four epochs.

Note that using the optimal hyperparameters previously mentioned, the overall accuracy for the three classes classification problem (i.e., pedestrians, cyclists, and streets) is 0.9567. Furthermore, the model is relatively quick to train and validate, requiring roughly 850 s for four epochs.

**Thresholding**

To reduce instances where the pedestrian signal is needlessly activated, the study considers a series of images and a threshold for the number of positive images, where a "positive" image is one that is classified by the model as a pedestrian or cyclist. The *n* most recent images are considered, for each value of *n* ∈ {1, 2, 3, 4, 5, 6, 7}, and all thresholds *t* = 0, 1, …, *n*, where a threshold of *t* = 0 is defined as a pedestrian or cyclist always being identified.

For each number of consecutive images *n*, the study tests 50 instances of a pedestrian (or cyclist) walking past the camera without stopping and 50 instances of a pedestrian (or cyclist) stopping in front of the camera for a specified amount of time. The time it takes for people to walk past the camera or wait in front of the camera when crossing the street is varied. A prediction is correct if the model does not predict "pedestrian" or "cyclist" when a person is passing by, or it predicts "pedestrian" or "cyclist" at least once before a person crosses the street; otherwise, the prediction is incorrect. The results of these experiments are summarized in the graphs in Figure 1.

Note that Figure 1(c) gives the accuracies based on the combined results from Figures 1(a) and 1(b). Also, Figures 1(d) gives the absence trigger rate, where an absence trigger is defined as a pedestrian or cyclist being detected when neither is present. That is, absence triggers occur when images of the street—as opposed to images of someone intending to cross the street or someone passing by—result in a misclassification that would trigger the pedestrian signal. Note also that for a threshold of 0, the model assumes that a classification of either pedestrian or cyclist always results, which implies that the pedestrian signal would always be activated. This arbitrary assumption makes the graphs in Figure 1 clearer.

In terms of accuracy, the authors observe that *n* = 4 images with a threshold of *t* = 3 balances the accuracy for detecting passing and crossing people, while the (*n*, *t*) = (3, 3) case is slightly better at identifying passing people, while the (*n*, *t*) = (5, 3) case is slightly better at identifying crossing people (Table A.3, online Appendix). Thus, the optimal choice of (*n*, *t*) would depend on whether a crosswalk tends to have more people passing by or crossing the street. Also, a smaller *n* implies a shorter wait time, which could be a factor to consider. For all subsequent experiments, the study uses (*n*, *t*) = (5, 3).

**Experimental setup**

The study’s goal here is to deploy this CNN in a real-time experimental pedestrian prediction system. The authors do
so by creating a program that captures images with a short delay of one second and evaluates the images using the threshold discussed in the previous section. This program’s output can then be used to trigger actions (e.g., pedestrian signals) if a pedestrian or cyclist wanting to cross the street is detected. For the experiment, the camera’s view is simultaneously recorded, and the output for a positive prediction is mapped to a beeping noise.

To better analyze the accuracy of the pedestrian prediction system, the model is deployed in various real-world environments. As depicted in Figure 2, a Raspberry Pi running the prediction algorithm is attached to a camera to capture the image data and record footage, and a portable battery to supply battery. The Raspberry Pi positioned so it captures only the crosswalk controlled by the pedestrian signal button so as not be influenced by other crosswalks. While previous tests consisted of test subjects, the experiments reported in this section are from actual users at three distinct crosswalk locations. The authors compare the prediction beeps emitted by the CNN model with the recorded video of the camera’s view, from which an evaluation is made on whether a subject actually crossed the street or simply passed by without crossing the street.

To account for the different settings that a pedestrian prediction system might be used in, this experiment is conducted at the following three different locations:

1. A crosswalk next to a summer school at the end of the school day
2. A busy traffic intersection next to a shopping mall
3. A crosswalk next to a park

The frames recorded by the prediction system for each location can be found in Figure A.7 in the online Appendix. These locations were chosen as they represent streets where implementing the automated crosswalk safety system would likely be beneficial. Each experiment consists of a one-hour time interval, or 3600 predictions. For each location, the program records the total number of pedestrians and cyclists that are passing by and waiting to cross the street in real-time. Based on the recording of the program beeps, it can be determined whether the pedestrian prediction model correctly identified the person’s street crossing (or not) behavior. These metrics are also broken down separately for pedestrians and cyclists. Finally, the experiment identifies the number of absence triggers that occur during the one-hour span, where a pedestrian was predicted when the model should identify a street.

**Results**

In this study, there are two different types of accuracies that are analyzed: accuracy $a_p$ when a pedestrian/cyclist is present and accuracy $a_a$ when a pedestrian/cyclist is absent. When measuring $a_p$, the accuracy must take into account the true positive of triggering the signal when a pedestrian crosses the street, the false positive of triggering the signal when a pedestrian passes by, the true negative of not triggering the signal when a pedestrian passes by, and the false negative of not triggering the signal when the pedestrian crosses the street. The accuracy is calculated as:

$$a_p = \frac{TP + TN}{TP + FP + TN + FN}$$

Although the accuracy $a_p$ fluctuated based on the location and number of people at the location, the average value of $a_p$ across all locations was 0.8496. The average accuracy of predicting pedestrians (or cyclists) without intentions to cross the road (TN) was 0.7943 whereas the average accuracy of predicting pedestrians (or cyclists) with intentions to cross the road (TP) was 0.8845 (Table A.4, online Appendix). These results suggest that if a pedestrian or cyclist were to try to cross a street similar to the ones in the study, there will be a 0.8845 chance the program will recognize and trigger the pedestrian signal before they leave, helping an average of 81.67 pedestrians per hour in this study. If a pedestrian or cyclist were to pass by the camera without intentions to cross the street, there is a 0.7943 chance the program ignores them and a 0.2057 chance the program falsely activates the crosswalk, occurring an average of 12 times per hour in the experiment. Figure 3 shows more detailed statistics regarding the prediction accuracies.

The accuracy $a_a$ only takes into account whether the crosswalk was triggered or not when there were no pedestrians. $a_a$ can be calculated as:

$$a_a = \frac{T}{T + F}$$

The average value of $a_a$ for all locations was 0.99963 (Table A.5, online Appendix). This indicates there is a 0.00037 chance of an absence trigger per prediction. Since the model currently makes predictions with a 1-second delay for a total of 86,400 predictions per day, roughly 32 of these predictions are expected to be absence triggers, resulting in the pedestrian signal being activated when no pedestrians are nearby.

As seen in Figure 3, the summer school yielded the highest overall accuracy of $a_p = 0.9035$. The shopping mall had the second highest overall accuracy of $a_p = 0.8472$. The park had the lowest of the three accuracies, with an overall accuracy of $a_p = 0.7982$. Furthermore, the accuracy trends for the different prediction types were generally the same across the three locations, where cyclists not crossing the
The results above indicate that the proposed CNN-based system has the ability to detect pedestrian and cyclist intentions at various real-world locations. The system has an average accuracy rate of $a_p = 0.8496$ for when a pedestrian or cyclist was present and an average accuracy rate of $a_d = 0.99963$ for when pedestrians or cyclists were absent. Compared to the reference model by Tomé et al. (2016), the proposed model yields an $a_p$ increase of 0.0486, potentially translating to hundreds more true positives each day. This improvement can be attributed to a more advanced 22-layer architecture, custom data, and a thresholding system better suited for waiting pedestrians and cyclists. Unlike similar pedestrian signal models like the ones by Ng and Kwok (2020), this study offers additional insight into cyclist accuracy and provides multiple real-world accuracy measurements rather than simulations, demonstrating its versatility and capabilities if implemented.

Based on the data from Figure 3, we conclude the method of transportation influences the accuracy of predictions. Across all three locations, the cyclists were predicted more accurately than pedestrians by up to 0.1875 when both did not have intentions to cross the street, whereas pedestrians were predicted more accurately than cyclists by up to 0.1161 when both had intentions to cross the street. The contrast in accuracy is likely due to differences between biking and walking, such as speed, that work better with various thresholds but worse in others. Cyclists are less likely to be detected for multiple images since they are faster, which increases the accuracy when they are passing by but decreases the accuracy when they cross the street. The threshold selection was optimized to produce the highest overall accuracy, distinct thresholds might be optimal for cyclists and pedestrians.

Another result from this experiment is how individual pedestrian behavior influences the accuracy of predictions. In the shopping mall location, the program accurately predicted pedestrians passing the crosswalk with a 0.7975 accuracy, which is much higher than the 0.7059 accuracy from the park crosswalk. For pedestrians crossing the street, however, the shopping mall had a lower accuracy of $0.8984$ compared to the park accuracy of 0.9143. The summer school’s accuracy remained slightly higher than that of the shopping mall and park. After reviewing the footage from each location, we believe this difference in accuracy can be attributed to the volume and behavior of people at each location. For instance, pedestrians taking a stroll around the park usually walked slower than the shopper and students, which decreases the park’s accuracy when pedestrians pass by the camera slowly but increases the accuracy when they cross the road slowly. Other details unique to each location include the summer school students being more cautious when crossing the road and the shopping mall pedestrians having to stand in front of the camera longer before crossing the street to wait for the pedestrian signal. These differences present distinct challenges that affect the accuracy of each location, and the results could likely be improved significantly if the model was optimized to account for behavior in each location.

The difference in location also resulted in different absence trigger results. For instance, the park location resulted in 3 absence triggers, two more than the shopping mall and three more than the summer school. Although these differences seem minor, they can amount to a difference of dozens of absence triggers per day. This difference is potentially due to an inadequate amount of data that features a park as its background. For the CNN model to produce less absence triggers for different locations, it is therefore important to have adequate training data with a similar background to the deployed location.

The dataset used to train the CNN model could be improved in various other ways. The dataset is relatively small, and a larger dataset might improve the accuracy. The evidence from the park crosswalk experiments indicates that images that are specific to a particular location will likely increase the accuracy of that location. External factors, such as the weather or light levels might have an effect on accuracy. Our current model uses almost exclusively daytime images, so subsequent studies will need to be performed to analyze the effectiveness of the prediction system at night. A crosswalk safety system may be even more important for more extreme weather conditions, so it could also be useful to gather ample data to account for such situations.

The selected threshold of $(n, t) = (5, 3)$ was set to maximize the overall accuracy for cyclists and pedestrians. However, the experimental results indicate that individual accuracies may be improved if different values of $n$ are used to identify cyclists and pedestrians. As a result, it will be useful to consider a detector that combines distinct systems for different methods of transportation. Such an approach can account for minor differences, such as speed, that might be currently limiting the accuracy.

As for the setup of the prediction system, the current model is meant to assist the conventional pedestrian signal button; thus, the camera is mounted at street level. However, in certain locations with obstacles in the way, the...
same model can be adjusted to be mounted higher in the air so there is less obstruction. By changing the dataset to include data frames from a higher perspective, the model will be retrained correspondingly. The current model can be seen as a worst-case scenario, yet the accuracy of $a_p = 0.8496$ indicates that it will still accurately predict pedestrian behavior most of the time.

It may also be useful to consider other machine learning models, particularly ones designed to track objects like the YOLO algorithm (Redmon et al. 2016). The current model has some difficulty when there are consecutive pedestrians passing by the camera without using the crosswalk, since the model cannot distinguish between the individuals. Another issue is when a pedestrian stands in front of the camera for a prolonged time without crossing the street. Since detection algorithms can track the position of individuals, subsequent experiments can use the threshold system and position tracking to lower the chances of incorrect predictions. Furthermore, object detection algorithms will be able to distinguish between cyclists and pedestrians when both appear in frame, providing greater clarity in the output.

Funding

The author(s) reported there is no funding associated with the work featured in this article.

ORCID

Mark Stamp http://orcid.org/0000-0002-3803-8368

Data availability statement

The dataset used to train the models is available from the authors upon request.

References

Albawi S, Mohammed TA, Al-Zawi S. 2017. Understanding of a convolutional neural network. Paper presented at: International Conference on Engineering & Technology (ICET); August 21–24, Antalya, Turkey.

Antonio JA, Romero M. 2018. Pedestrians’ detection methods in video images: a literature review. Paper presented at: IEEE International Conference on Computational Intelligence and Computing Research (ICICIC); December 13–15, Madurai, India.

Center for Disease Control. Pedestrian Safety. 2022. [accessed 2022 June 21]. https://www.cdc.gov/transportationsafety/pedestrian_safety/index.html

Chen L, Chen C, Ewing R. 2012. The relative effectiveness of pedestrian safety countermeasures at urban intersections – Lessons from a New York City experience. Transportation Research Board. [accessed 2022 August 13]. https://nacto.org/docs/usdg/relative_effectiveness_of_pedestrian_safety_counter_measures_chen.pdf

Governor’s Highway Safety Association. 2022. Pedestrian traffic fatalities by state – 2021 preliminary data. [accessed 2022 August 12]. https://www.ghsa.org/sites/default/files/2022-05/Pedestrian%20Traffic%20Fatalities%20by%20State%20%E2%80%93%202021%20Preliminary%20Data.pdf

Hardesty L. Explained: Neural networks. MIT News Office 2017. [accessed 2022 May 30] https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

Kingma DP, Ba J. 2014. Adam: a method for stochastic optimization. arXiv.org. doi: 10.48550/arXiv.1412.6980.

Ng S, Kwok CP. 2020. An intelligent traffic light system using object detection and evolutionary algorithm for alleviating traffic congestion in Hong Kong. Int J Comput Intell Syst. 13(1):802–809.

Paszke A, Gross S, Massa F, et al. 2019. Pytorch: an imperative style, high-performance deep learning library. Paper presented at 33rd Conference and Workshop on Neural Information Processing Systems (NeurIPS); December 8–14, Vancouver, Canada.

Radhakrishnan P. 2017. What are hyperparameters? And how to tune the hyperparameters in a deep neural network? Towards Data Science. [accessed 2022 June 3]. https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a

Redmon J, Divvala S, Girshick R, Farhadi A. 2016. You only look once: Unified, real-time object detection. Paper presented at: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 27–30, Las Vegas, United States.

Sermanet P, Kavukcuoglu K, Chintala S, Lecun Y. 2013. Pedestrian detection with unsupervised multi-stage feature learning. Paper presented at: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 23–28, Portland, United States.

Stamp M. 2022. Introduction to machine learning with applications in information security. 2nd ed. Boca Raton (FL): Taylor & Francis/CRC.

Szegedy C, Liu W, Jia Y, et al. 2015. Going deeper with convolutions. Paper presented at: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 7–12, Boston, United States.

Tomè D, Monti F, Baroffio L, Bondi L, Tagliasacchi M, Tubaro S. 2016. Deep convolutional neural networks for pedestrian detection. Signal Process Image Commun. 47:482–489. 10.1016/j.image.2016.05.007

van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T, scikit-image contributors. 2014. scikit-image: Image processing in Python. PeerJ. 2:e453.