Pedestrian Detection

Aljoša Koren, Žan Gostič

November 15, 2022

Abstract

abstract.

1 Introduction

In this article we will present our work in making an application for the Mobile Sensing course addressing pedestrian detection with the phone's camera placed on the windshield of the vehicle. We made a driving assistant application, which alerts the driver when it detects pedestrians on the road with a beeping sound. The phone needs to be placed on the windshield and the main camera should be focused on the road. When the app detects pedestrians crossing the road, it alerts the driver by producing a distinct sound alert, which produces a response of the driver to pay more attention on the road.

More than 7000 pedestrians are killed yearly, about one every 75 minutes. [CDC] With our application we are planning on reducing this enormous number of casualties. As many car owners cannot afford newer cars with additional expensive equipment, we are planing that with our application anyone with a smartphone could download our application and make roads safer.

2 Related work

3 Methods

4 Evaluation

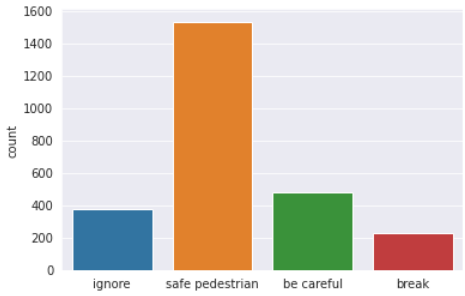
4.1 Data

Our main data is the CityPersons dataset [ZBS17] which is a new set of annotations targeting pedestrian detection. It is based on the Cityscapes dataset [COR+15, COR+16] which contains stereo images recorded in 27 European cities. Cityscapes dataset contains 5000 images with fine pixel-level annotations and 20000 images with coarse semantic labels. CityPersons contains only the fine pixel-level annotated images with 5 different classes: ignore regions, pedestrians, riders, sitting persons, other persons with unusual postures and a group of people. Ignore regions represent hard negatives or areas that can be easily misinterpreted as humans for example posters with people on it, reflections etc. The dataset contains 35000 persons (20000 unique) and 13000 ignore labels. Images contain an average of 7 persons. 83% of the annotated persons are pedestrians, riders represent 10% of the labels and sitting persons 5%. Images have a height of 1024 and width of 2048 pixels.

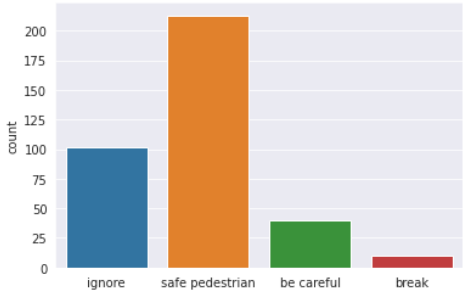
Additionally, we created a script that allowed us to add annotations to each marked pedestrian. We decided on 4 labels. First label annotates the ignore regions. Second is for pedestrians that are safely on the sidewalk or very far away. The third label marks pedestrians that look like they plan on crossing the road, the cyclist driving near and pedestrians crossing the road down the road. The last label is for the pedestrians that are crossing the road in front of the vehicle and the car would hit them if it did not break immediately. We can than classify the images in one of four classes by giving them the class of the most dangerous individual on the image.

First we split the training set into train set and the validation set. We decided to use images from two towns as the validation set and other 16 as the train set. In the test set there were images from 3

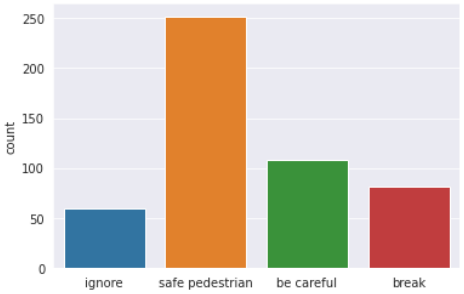
never seen before cities. Train set contained 2610, validation set 365 and test set 500 images. In Figure 1 we can see that the train, test and validation dataset have similar distributions, as they all contain majority of images containing pedestrians that are safely on the sidewalks. However, validation dataset contains larger percentage of images where there are no pedestrians in sight and test set contains a larger percentage of images where pedestrians are in danger if the car doesn't break. Images were manually annotated so there might be mistakes and the subjective bias integrated in the classes. In Figure 2 we can see four examples of newly labeled data, where pedestrians are labeled with four different classes.



(a) Train dataset distribution.



(b) Validation dataset distribution.



(c) Test dataset distribution.

Figure 1: Classes distributions

Figure 2: Examples from the dataset.

5 Results

We tested the classification on pretrained MobileNetV2 [SHZ+18] on the ImageNet dataset [DDS+09]. The first model $PD_{v1}^{MobileNetV2}$ has frozen pretrained weights on the base MobileNet network. On top we added additional global average pooling layer and 3 dense layers with 64, 32 and 4 neurons. Between the dense layers we added dropout layers with 20% dropout rate for regularization. Two layers have ReLU activation function while the last has the softmax activation function. As per learning rate we tried the 0.00001, with Adam optimizer and sparse categorical cross-entropy. We decided to use the batch size of 8. Firstly, we reached the problem of the unbalanced dataset. The model reached the accuracy of 50% which was close to the majority classifier. We continued by making the class weights of the minority classes bigger. By doing this we got a better accuracy by 1% and also the classifier produced more variety of inputs instead of just the majority class. We also introduced data augmentation to increase the dataset without the need of classifying new images. We added horizontal flips, random rotation up to 10 degrees, randomly zooming in image up to 20% and shifting vertically and horizontally the images to up to 10% of the image. However, this augmentations did not drastically increase the overall accuracy. As we were resource constrained we decided to train on images with height and width of 224 pixels. Images were therefore 4.5 and 9.1 times smaller in height and width respectively. This reduced the amount of memory and training time significantly.

$PD_{v2}^{MobileNetV2}$ had the same parameters as the $PD_{v1}^{MobileNetV2}$, however, we decided to train it for more epochs (40). This managed to increase accuracy from the 51% to 57% which is 7% better then the majority classifier (Table 1). We still had problems with the over-weighted class as the ignore images were always classified as the majority class and the be careful images were also classified as the majority class in 96% of cases (Table 2). The majority class (safe pedestrian) had a great recall of 99% and precision 55% (F1-score 71%). Break class had a precision of 88% and recall 43% (F1-score 58%). In Figure 3 we can see that the accuracy on the validation set did not increase largely after a few beginning epochs. The loss still looks like it is decreasing meaning probabilities for classes are getting better but the most probable class is not accurate.

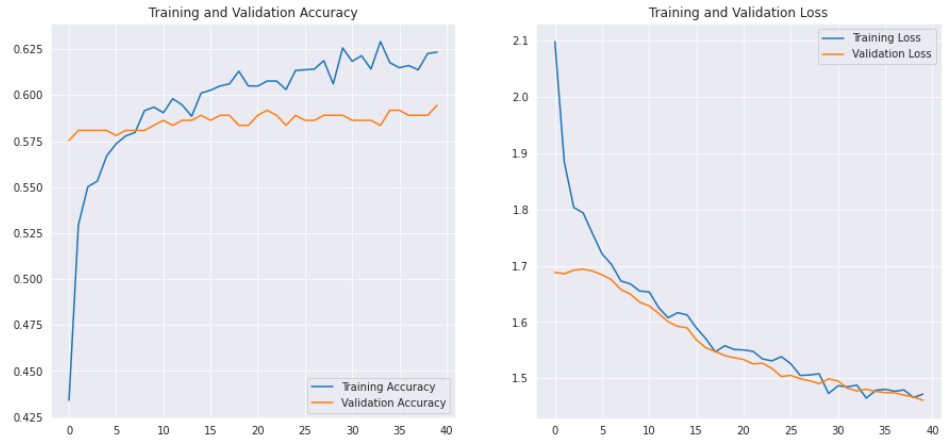


Figure 3: Accuracy and loss on train and validation set during training.

	precision	recall	f1-score	support
Ignore (Class 0)	0.00	0.00	0.00	59
safe pedestrian (Class 1)	0.55	0.99	0.71	252
Be careful (Class 2)	0.00	0.00	0.00	108
Break (Class 3)	0.88	0.43	0.58	81
accuracy			0.57	500
macro avg	0.36	0.36	0.32	500
weighted avg	0.42	0.57	0.45	500

Table 1: Model scores.

	Ignore (Class 0)	safe pedestrian (Class 1)	Be careful (Class 2)	Break (Class 3)
Ignore (Class 0)	0	**59**	0	0
safe pedestrian (Class 1)	0	**250**	1	1
Be careful (Class 2)	0	**104**	0	4
Break (Class 3)	0	**43**	3	35

Table 2: Confusion matrix.

$PD_{v3}^{MobileNetV2}$ had a trainable base model. The training time increased approximately by 4 times comparing to the models with frozen weights of the MobileNetV2 [SHZ+18] base model. Despite the increased time, we observed the accuracy increased by 2% (59%). In Figure 4 we can see that the model was only trained on 10 epochs because of the resource and time constraints. We can see that the validation accuracy would probably increase a bit further if we trained for a bit more epochs. In Table 3 we can see the increase in F1-scores for all classes. The confusion matrix in Table 4 confirms that $PD_{v3}^{MobileNetV2}$ is a better model as the break class gets identified in the majority of cases correctly and other classes that were always miss-classified get a few correct classifications.

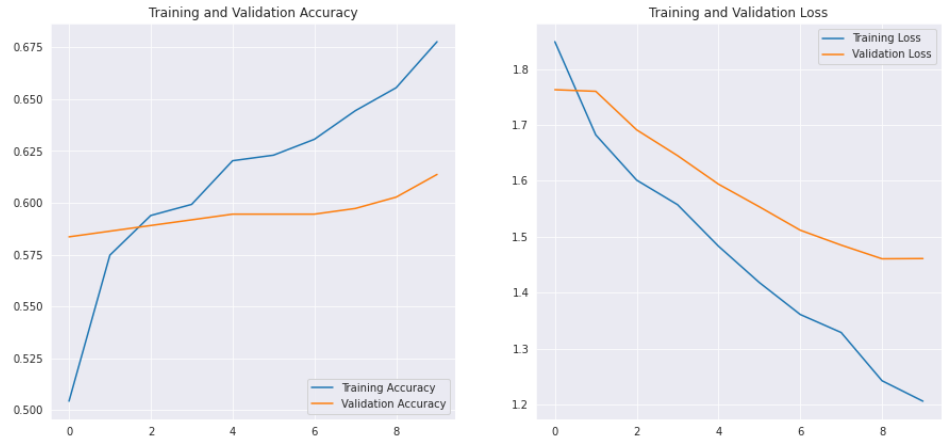


Figure 4: Accuracy and loss on train and validation set during training.

	precision	recall	f1-score	support
Ignore (Class 0)	0.47	0.15	0.23	59
safe pedestrian (Class 1)	0.57	0.94	0.71	252
Be careful (Class 2)	0.50	0.04	0.07	108
Break (Class 3)	0.81	0.54	0.65	81
accuracy			0.59	500
macro avg	0.59	0.42	0.42	500
weighted avg	0.58	0.59	0.51	500

Table 3: Model scores.

	Ignore (Class 0)	safe pedestrian (Class 1)	Be careful (Class 2)	Break (Class 3)
Ignore (Class 0)	9	**50**	0	0
safe pedestrian (Class 1)	7	**238**	3	4
Be careful (Class 2)	3	**95**	4	6
Break (Class 3)	0	36	1	**44**

Table 4: Confusion matrix.

For the $PD_{v4}^{MobileNetV2}$ we decided to change the input size of the images from the 224x224 to 512x512. We also decreased the batch size from 8 to 4. Because of the increased input the training

of 1 epoch took around 7 times more time than for $PD_{v3}^{MobileNetV2}$ and around 28 times more than $PD_{v2}^{MobileNetV2}$. This is why we were forced to decrease the training to only 5 epochs. However, even with such short training we were able to increase the accuracy by another 2% to 61% (Table 5). In Table 6 we can see that the model increased the number of correctly classified be careful and break classes whuch are the most important for our application as they are the ones where we have to alert the user.

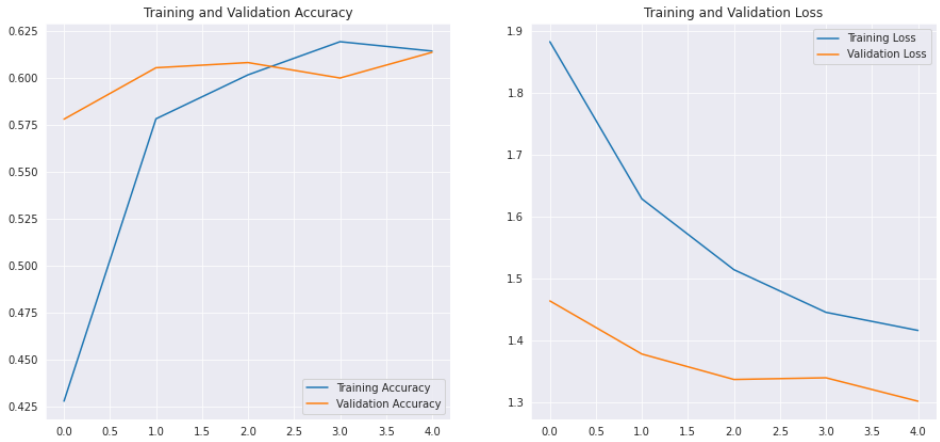


Figure 5: Accuracy and loss on train and validation set during training.

	precision	recall	f1-score	support
Ignore (Class 0)	0.00	0.00	0.00	59
safe pedestrian (Class 1)	0.60	0.94	0.73	252
Be careful (Class 2)	0.50	0.23	0.32	108
Break (Class 3)	0.85	0.54	0.66	81
accuracy			0.61	500
macro avg	0.49	0.43	0.43	500
weighted avg	0.55	0.61	0.54	500

Table 5: Model scores.

	Ignore (Class 0)	safe pedestrian (Class 1)	Be careful (Class 2)	Break (Class 3)
Ignore (Class 0)	0	**59**	0	0
safe pedestrian (Class 1)	0	**238**	11	3
Be careful (Class 2)	0	**78**	25	5
Break (Class 3)	0	23	14	**44**

Table 6: Confusion matrix.

6 Discussion

In the future we are planning on making a much better classifier by testing different architectures and deep learning hyper-parameters.

7 Acknowledgments

References

[CDC] Transportation Safety: Pedestrian safety. https://www.cdc.gov/

transportationsafety/pedestrian_safety/index.html. Accessed: 2022-10-14.

[COR+15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on The Future of Datasets in Vision*, 2015.

[COR+16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[SHZ+18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018.

[ZBS17] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. *CoRR*, abs/1702.05693, 2017.