IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 3, 2020, accepted March 12, 2020, date of publication March 23, 2020, date of current version April 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982539

Pedestrian Detection in Severe Weather Conditions

P. TUMAS[1], (Member, IEEE), A. NOWOSIELSKI[2], AND A. SERACKIS[1], (Senior Member, IEEE)
[1]Department of Electronic Systems, Vilnius Gediminas Technical University, LT-10223 Vilnius, Lithuania
[2]Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, 70-310 Szczecin, Poland
Corresponding author: P. Tumas (paulius.tumas@vgtu.lt)

ABSTRACT Pedestrian detection has never been an easy task for computer vision and the automotive industry. Systems like the advanced driver-assistance system (ADAS) highly rely on far-infrared (FIR) data captured to detect pedestrians at nighttime. The recent development of deep learning-based detectors has proven the excellent results of pedestrian detection in perfect weather conditions. However, it is still unknown what the performance in adverse weather conditions is. In this paper, we introduce a 16-bit thermal data dataset called ZUT (Zachodniopomorski Uniwersytet Technologiczny) as having the widest variety of fine-grained annotated images captured in the four biggest European Union countries captured during severe weather conditions. We also provide a synchronized Controller Area Network (CAN bus) data, including driving speed, brake pedal status, and outside temperature for future ADAS system development. Furthermore, we have tested and provided 16-bit depth modifications for the YOLOv3 deep neural network (DNN) based detector, reaching a mean Average Precision (mAP) up to 89.1%. The ZUT dataset is published and publicly available at IEEE Dataport and Github.

INDEX TERMS FIR pedestrian detection, 16bit, Yolo, bad weather, ADAS.

I. INTRODUCTION

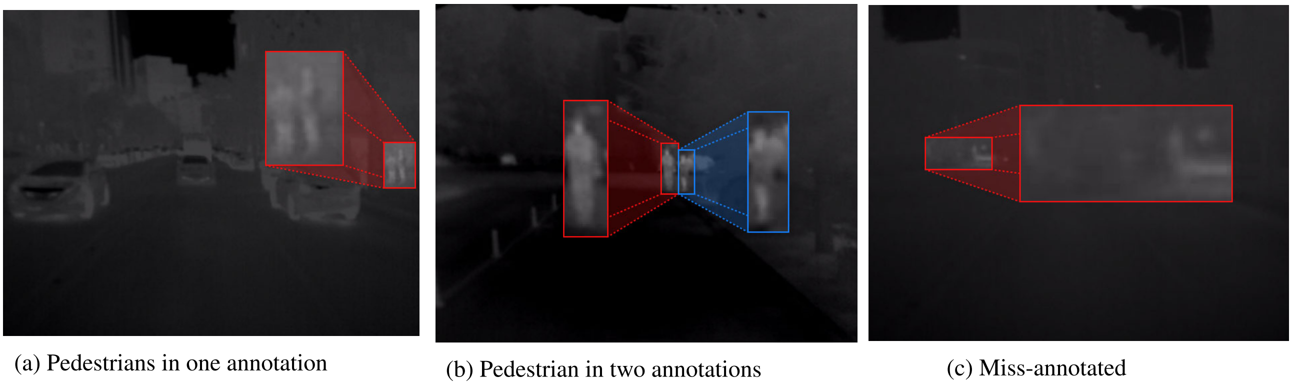
The World Health Organization (WHO) each year announces the statistics of people injured in traffic accidents. In 2018, annual road traffic deaths reached 1.35 million [3], where half the traffic accidents belong to the category of road users, cyclists, and pedestrians. Even though the European Union has the safest roads in the world, there are more than 25 000 [4] people who lose their lives every year, and many more are seriously injured. One of the causes of traffic accidents is the bad weather condition. Rain, fog, snow, and wind are factors affecting visibility and act via a psycho-physiological function of the driver [5], [6], increasing the traffic accident rate by up to 13% [7], [8]. The leading countries in traffic safety are the United Kingdom, Denmark, and Ireland. On the other hand, the highest fatality rates were in Romania, Bulgaria, Latvia, and Croatia. To prevent accidents, the EU introduces new safety measures in cars, lorries, and buses for advanced driver assistance systems (ADAS). The new systems support a new feature like intelligent speed assistance, advanced emergency braking and lane-keeping systems, frontal protection systems, driver drowsiness, attention monitoring, and event (accident) data recorder. Buses and lorries will be capable of detecting vulnerable road users.

Pedestrian detection has never been an easy task for automotive applications. Systems highly rely on data captured from various sensors like radars, visual/thermal spectrum cameras, and vehicle speed sensors to prevent collision with a pedestrian. In many cases, sensors are connected to transmit data through a Controller Area Network (CAN bus), and then individual modules receive the message to react to the data. For example, the ABS module collects data from each car wheel speed sensor via CAN bus and adjusts brake pad pressure on not slipping/slipping wheels accordingly.

Pedestrian detection systems usually use visual and thermal camera data. The visible spectrum cameras are mainly used during daylight hours, providing detailed features of pedestrians, clothing colors, and patterns. At night or when other conditions decrease visibility, manufacturers like Toyota try to enhance visibility by emitting near-infrared light through headlight projectors, and then a camera captures that reflected radiation. However, visibility distance is only up to 250 meters [9].

Another approach to increase visibility during low-light hours is to use thermal cameras. In a comparative study [10] on human performance versus a thermal imaging–based

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.


(a) Pedestrians in one annotation

(b) Pedestrian in two annotations

(c) Miss-annotated

FIGURE 1. **Examples of KAIST dataset.**

automatic system in severe lighting conditions, the second turned out to be better at detecting pedestrians. In many cases, participants of conducted experiments did not notice pedestrians at all, in contrast to a computer system that analyzed the thermal data.

Automakers like Audi, BMW, and Daimler offer Autoliv designed FLIR Pathfinder nighttime driving assistance. Such a system is based on a far-infrared (FIR) spectrum FLIR camera having resolution 324 × 256 with a refresh rate of 30Hz [11]. However, not many details are available except on publication [12] in which it mentioned that the detector was based on a Cascade classifier, and the dataset was collected driving eight years, four seasons in various locations and about one million miles were driven. The official user manual [11] is also limited by details of the system. We tried to find accuracy measures, but the only found is a single statement: "Depending on conditions and ambient temperatures, the detection algorithms may work poorly or not at all during the daytime."

According to research [13], [14], the accuracy of the detector may vary on the variety of dataset samples, detector input/type used, and implementation details. Currently, the trend is to use DNN, which requires thousands of various pose featured images. There are many datasets available for pedestrian detection in the visual spectrum. One of the most popular among the DNN training source is PASCAL VOC 2012 [15] dataset. The research showed that Fast R-CNN [16] can reach up to 72.0% mAP, YOLO [14] 63.5% mAP, SSD512 [17] 88.5% mAP in person detection. KITTI [18] dataset is also frequently used to test accuracy and detectors like improved YOLOv3 [19] can reach up to 82.95% mAP and have improved SSD [20] 68.1% mAP, RCNN [21] averaging 52.17% mAP.

To train DNN on thermal imagery data, it is only possible to find just ten datasets like: CVC-09 [22], CVC-14 [23], FLIR-ADAS [24], KAIST [25], KMU [26], LSIFIR [27], OTCBVS [28], RISWIR [29], Terravic Motion IR [30] and one of the recent SCUT [31]. SCUT, unlike others, contains images captured from driving a car in areas like downtown, suburbs, campuses, and expressway roads. It is also the biggest dataset in terms of a number of frames and annotations (containing 211k frames and 477k annotations). The resolution is another important aspect. The SCUT dataset is captured with 384 × 288 sensors and interpolated to 720 × 576. Finally, the strictly predefined labeling protocol was followed by six classes (walk person, squat person, ride person, people, person, and combined annotation person/people).

KAIST is the second biggest available dataset containing multi-spectral images captured in visible and thermal domains, having 95k frames of resolution 640 × 480 taken from on a vehicle-mounted camera. Recordings were taken in multiple areas like campuses, cities, and outskirts. All image domains were manually annotated with three classes (person, people, and cyclist) for a total of 103,128 annotations and 1,182 unique pedestrians.

Despite the existing dataset, each of them has some limitations. First of all, none of the datasets contain information about exact weather conditions while recording. For example, when it rains, the water and dirt coats cameras lenses, which causes an effect of a blurry image without precise contours and lowered intensity of pedestrian. SCUT authors mentioned that they were recording during December in Guangzhou, China, where the average rainfall is only 32 mm [32]. Secondly, there is no tracking of the outside the temperature, which affects the image details. For example, during the cold winter days, the pedestrians look very bright, however during the hot summer nights, the pedestrians are blended with the background. Finally, it is crucial to keep annotation quality by complying with the same annotation protocol. We have checked more than 50k annotations of the KAIST dataset and found that in some cases a group of people is marked as one annotation 1(a), sometimes as separate 1(b) or some annotations looked misslabeled 1(c) having no context.

The main contributions of this paper can be summarized as follows: 1) Introduction of a new benchmark database which outperforms the existing ones in several key issues (much greater data diversity, the inclusion of car CAN data, information of weather conditions and temperature, extended annotation classes, 16 bit depth images). 2) Introduction of a normalization procedure for 16 bits data, taking into account the temperature of the environment. 3) Evaluation of two

TABLE 1. Comparison of data diversity on KAIST, SCUT and ZUT datasets.

	KAIST	SCUT	ZUT
frames	95k	216k	110k
classes	4	6	9
annotations	103k	448k	122k
road scene	3	4	10
pedestrian distance	2.4 m - 61 m	4.6 m - 132 m	10 m - 100 m
data depth	8bit	8bit	16bit
temperature	not measured	not measured	-0.5 to 12 °C
maximum speed	not measured	80 km/h	180 km/h

leading deep learning network architectures on the proposed database together with checking the impact of the normalization procedure.

The rest of the article is organized as follows: we provide an overview of the dataset, the methodology of how the dataset was collected, the description of annotations available, details of modifications done to Darknet DNN implementation to support 16bit depth images, YOLOv3 and Tiny YOLOv3 (TINYv3) configuration changes, and the results of dataset evaluation. The paper ends with final conclusions.

II. MATERIALS AND METHODS

Since none of the datasets reported in scientific literature contain a sufficient variety of samples to create a detector capable of detecting pedestrians in bad weather conditions, we have decided to collect a dataset called ZUT-FIR-ADAS [1], [2] or ZUT in this paper. We used FLIR SC320 thermal camera with a spatial resolution of 320 × 240, capturing 16bit frames at 30fps. In addition to thermal data, we have synchronized and extracted a Skoda Fabia MK2 Green Line 1.4 TDI CAN bus data. The CAN data includes car speed, the brake status (brake released, foot on the brake, brake press), which was captured from the ABS module, and the outside temperature from the instruments cluster.

For the recording location, we selected four European Union countries: Denmark, Germany, Poland, and Lithuania starting in the middle of autumn and finishing in the middle of winter. Law limitations were based on the selection criteria and General Data Protection Regulation (GDPR) rules, typical weather conditions based on season, car accident statistics, and traffic infrastructure. For example, we wanted to record in Austria. However, it is entirely illegal to use a camera there, and it is possible to face fines of 26,000 Euros [33], [34]. Fortunately, Denmark, Germany, Poland, and Lithuania are camera friendly. Only in Germany is it required to mask car number plates, and people face for data publication. Fortunately, these restrictions do not apply to thermovision spectrum imagery.

Denmark was selected because it has up to 19 days of rainfall in November, is the top country in traffic safety records, and has traffic infrastructure designed for cyclists. Germany is also one of the top countries in traffic safety, but it is rich in traffic infrastructure/regulations and has unlimited speeds on the Autobahn. The remaining countries, Poland and Lithuania, are rich by nature, have forest-surrounded roads with a high probability of animals' being present, are surrounded by small villages across the main highways, and are traffic heavy on transitive routes. The dataset contains frames driven through various weather conditions like fair weather, cloudy with a chance of rain, mild rain, heavy rain, and fog. The dataset includes ten road scenes: city center, old town, roundabouts, tunnels, city outskirts, one-way roads, two-way roads, highways, Autobahns. The ZUT also includes images driving with speeds up to 180km/h, driving through capital cities (Berlin, Copenhagen, Warsaw, and Vilnius) and during morning and evening rush hours. The temperature is ranging from −0.5 to 12 degrees Celsius. The detailed comparison is visible in table 1.

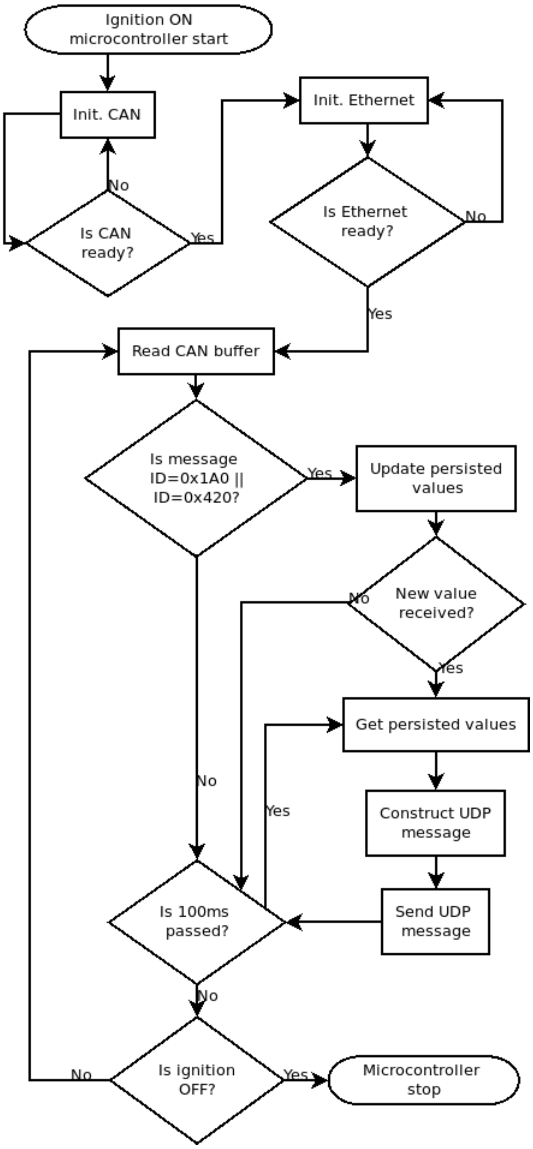


FIGURE 2. The flowchart of CAN bus capturing program.

A. CAN BUS DATA CAPTURE

To collect car data, we used an Atmel SAM3X8E 32bit ARM microprocessor having a clock speed of 84MHz. To interface CAN bus, we used MCP2515 CAN Controller with SPI

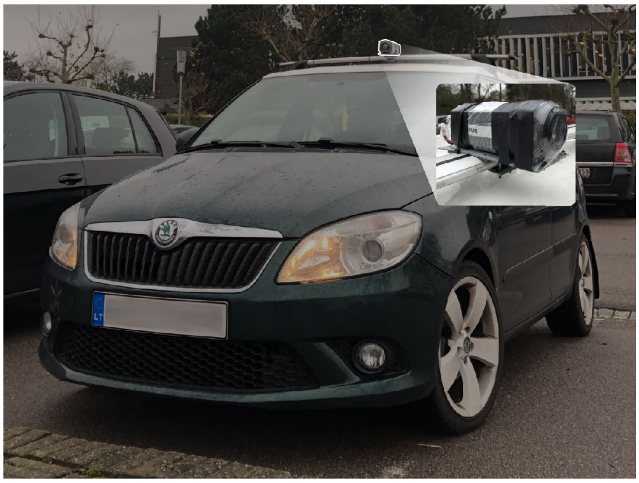


FIGURE 3. **Wrapped camera and placement.**

(a) With protection (b) Without protection

FIGURE 4. **A comparison of camera view.**

interface connected directly to instrument cluster CAN bus wiring. We have established a gigabit network in the car and broadcasted UDP packets of CAN data through ENC28J60 Ethernet SPI interface. We have chosen to use UDP packets to minimize packet size and gain speed, but the drawback is that some of the packets might be lost. Lastly, we designed the application (2), which throttles the frequency of broadcast ten times per second. Besides, there is an additional logic implementation that keeps track of data captured by persisting it in memory. For example, when the brake pedal is pressed, the message throttle mechanism is overridden, and the message is sent right away. This strategy was chosen to minimize network overflow and overhead in thread synchronization.

B. DATA RECORDING

Each recording session starts with the camera protection preparation procedure. To protect the camera and its lens against dust, rain, and dirt, every session, we have been wrapping the camera (3) with a very thin plastic film used in the food industry.

Experimentally, we have found that very thin plastic film allows thermal energy to pass through it with a minor distortion (4). The center of the roof was chosen for the location of the mounting point of the camera, in order to minimize the dirt coming from cars in front and to allow equal left/right side visibility. Additionally, we re-calibrated the camera for a better view before each recording.

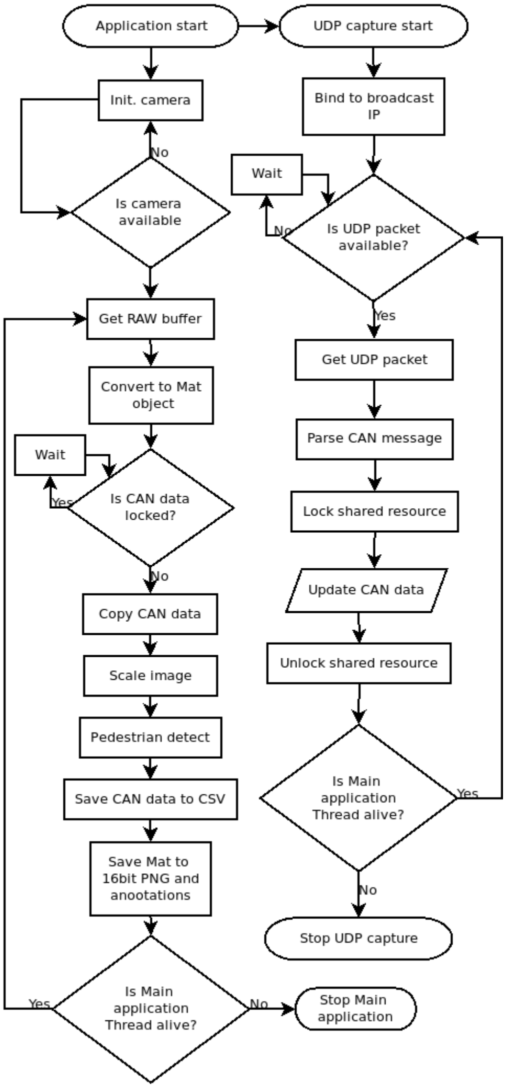


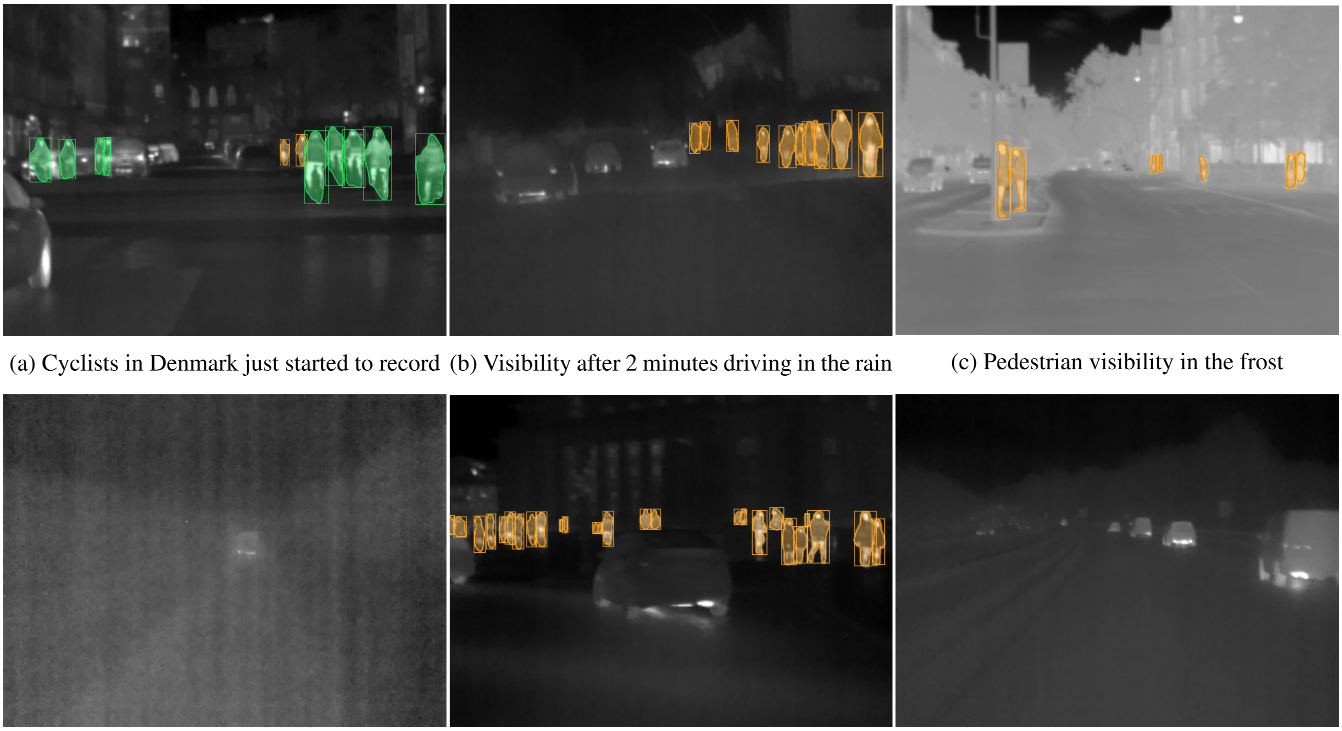
FIGURE 5. **The flowchart of data recording application.**

We have created a second application (5) on the car installed computer to record data, based on an I7 eighth generation processor with an NVIDIA RTX2070 graphics processor. The application has two threads, the first for image capturing/recording and the second for reading UDP packets with CAN data from the local gigabit network.

The image capturing thread reads data from the camera using the GenICam protocol. Since this protocol is generic, there are many SDK providers for camera manipulation. In our case, we used the Baumer GAPI SDK [35] because it allowed us to get raw data, and is compatible with OpenCV. After successfully grabbing the frame, we convert it to a single-channel 16bit Mat object scaled to 640 × 480 resolution. Then, CAN data from UDP capture is copied by a thread locking semaphore. To minimize annotation work, we used the pre-annotation approach [36] based on the TINYv3 [37] neural network. The pre-annotation detector was trained by 50,000 of images scaled down to 640 × 480 resolution taken from SCUT dataset with "People-?" class excluded,

TABLE 2. Annotation count per specific class.

	Pedestrian	Occluded	Cyclist	Motorcyclist	Scooterist	Body-parts	Unknowns?	Baby carriage	Pets and Animals
Training	59649	4008	7908	173	94	16611	14	27	140
Benchamark	21083	1112	2355	49	0	9091	1	10	107
Total	**80732**	5120	**10263**	**222**	**94**	25702	15	37	**247**



(a) Cyclists in Denmark just started to record (b) Visibility after 2 minutes driving in the rain (c) Pedestrian visibility in the frost

(d) Visibility during heavy rain (e) Visibility during mild rain (f) Driving in Autobahn during the fog

FIGURE 6. Dataset examples during various weather conditions.

having 67% mAP precision. Finally, we save corresponding frame ID with CAN data to comma-separated CSV file and the captured frame with YOLO type annotation.

The CAN thread binds to the network interface and starts reading UDP packets. Then the shared resource across the recording thread is locked, and the resource is updated with received data. Finally, the resource is unlocked, and the process continues to repeat until the main application is stopped.

In this way, we have recorded more than 500GB of data; however, to minimize the dataset and have a wider variety of samples, we have taken every tenth frame for the annotation process.

C. THE ANNOTATIONS

The annotation started on pre-labeled data. We have found that generally pedestrians were pre-annotated well, but the main work was to divide objects into different classes. For this process, we used Ybat: YOLO BBox Annotation Tool [38]. The ZUT dataset contains two sets of annotations. The first set is made of annotations used for the training. The benchmarking set was used to measure the accuracy of the detector in places where it was not trained.

Each set of annotations includes fine-grained labels divided into nine classes: Pedestrian, Occluded, Body-parts, Cyclist, Motorcyclist, Scooterist, Unknowns?, Baby carriage, Pets, and Animals. The cardinality of each class or available annotations are presented in the table (2). An individual person is labeled as a pedestrian when it is walking, running, standing, or when at least 60% of the individual person's body is visible. Occluded class is used when it is impossible to distinguish an individual from a group of people. The Body-parts class is mainly used when less than 40% of the individual person's body is visible. For example, individual person is behind a car, and his/her head is visible, then the individual person is considered part of the Body-parts class. The same strategy is used with legs and hands. Cyclists, Motorcyclists, and Scooterist are labeled separately because there are a lot of scooterists and motorcyclists in Germany, but cyclists are dominant in Denmark. The Unknowns? class is mainly used for objects similar to pedestrians, like a tilted tree or a hot traffic sign or traffic light. The Baby Carriage class doesn't contain many annotations, but we saw that there is some visibility of children in it. The last class Pets and Animals, includes domestic cats and dogs

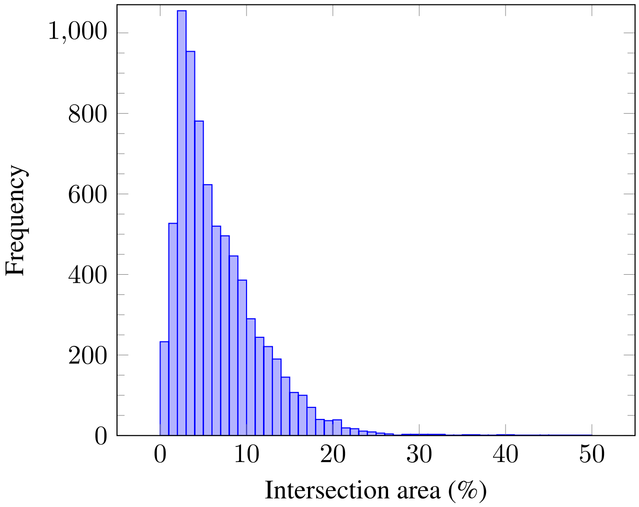


FIGURE 7. **Pedestrian annotation intersection histogram.**

mainly, but there are several foxes and rabbits annotated as well.

The database was collected during hours of driving in different real-life scenarios without any artificial arrangements. For this reason, the database reflects real situations encountered on the road, and the number of object instances is fewer in some classes than in others. For example, compared to pedestrians pets and animals are extremely rare in the city environment.

Figure (6) presents pedestrian visibility changes through different weather conditions. When the recording started, the camera is clear, and pedestrians are very clearly visible 6(a). However the visibility becomes indistinct after driving several minutes in the rain 6(b), where only the warmest are visible. Similarly, the view looks darker during the mild rain 6(e) and driving in the fog 6(f). During the heavy rain, there are almost no thermal objects visible 6(d). The opposite situation is visible during the frost. The background and the people are very bright, and it is hard to distinguish further than 60 meters 6(c).

Some pedestrian detection applications, like detecting [39], [40] pedestrians in the crowd is need to have an indicator of pedestrian annotation intersection. For this reason, we have provided a histogram (7) which shows intersected area distribution through the dataset.

D. TRAINING

For the training, we decided to use two versions of YOLO DNN - YOLOv3 and TINYv3, since both can be used for real-time performance and are regarded as state-of-the-art detection/recognition approaches. To increase the accuracy of YOLO it was decided to train DNN by using 16bit depth images data, which provide up to 256 times more information than a regular 8bit images. To support this feature, the latest Darknet implementation was taken (maintained by AlexeyAB [41]) and the following changes were applied:

1) The image loading function

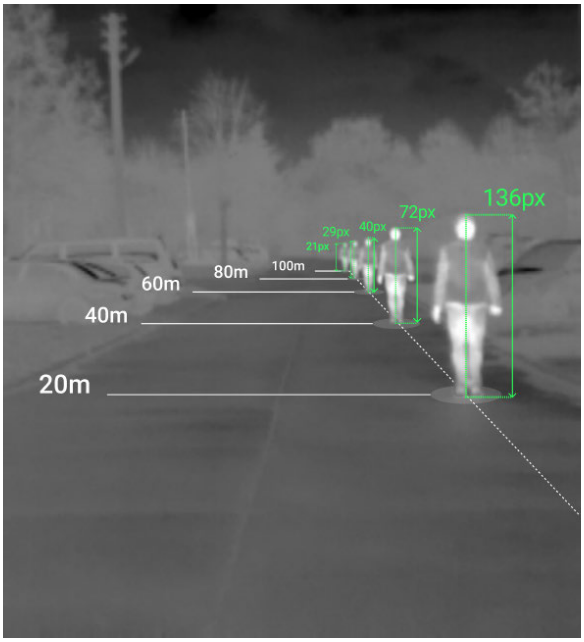


FIGURE 8. **Pedestrian height distribution per distance.**

TABLE 3. **Annotation distribution according to weather conditions.**

Country	Dataset	Drizzle	Frost	Rain	Cloudy	Fog	Clear sky
Denmark	Training	**20886**	0	**37064**	3051	0	0
Denmark	Benchmark	**16291**	0	0	0	0	0
Germany	Training	0	10206	13	0	1535	0
Poland	Training	212	0	0	15657	0	0
Poland	Benchmark	0	0	1153	6441	0	752
Lithuania	Benchmark	3687	0	25	5459	0	0

2) The normalization function
3) The data augmentation function

The image loading function was changed in training and testing phases to load 16bit images. The normalization function was changed by dividing the intensity value of over 65535 instead of 255. Finally, the augmentation function was changed by adding a low pass filter, which filters intensity by the temperature in which the image was captured. To design a low pass filter, all dataset annotations were used. The corresponding temperature averaged the maximum and minimum pixel intensity, and the quadratic function was fitted upon the lowest maximum intensity points. In case the maximum intensity was below the function, the value is kept without applying the filter. This filter resulted in hot objects like tires, disk brakes, exhaust pipes, windows, and chimneys to be less bright and provide more contour and pattern information. This also enhanced the visibility of ‘‘hot‘‘ pedestrians who, for example, had driven the long distances in warm cars.

The remaining YOLO configuration was left unchanged except the input resolution enhanced to 640 × 480, recalculated the anchors by k-means algorithm, and changed the input channel number to 1. The ZUT training set was divided by 80% of images used for the training and 20% for the testing. Additionally, we excluded classes like Occluded,

TABLE 4. ZUT training results.

Resolution	Version	Depth	Loss	mAP(0.5)	mAP(0.25)	Iteration
416×416	YOLOv3	16bit	0.2448	80.5	91.5	251000
416×416	TINYv3	16bit	0.2954	66.3	86.0	243000
640×480	YOLOv3	16bit	0.2514	85.4	92.5	220000
640×480	TINYv3	16bit	0.2681	79.1	92.3	250000
640×480	YOLOv3+Low pass	16bit	**0.1514**	**89.1**	**95.4**	383000
640×480	TINYv3+low pass	16bit	**0.1681**	**82.3**	**94.2**	420000
640×480	YOLOv3+Low pass	8bit	0.1914	**79.6**	92.3	123000
640×480	TINYv3+low pass	8bit	0.2414	**71.1**	89.1	78000

TABLE 5. Annotation distribution per pedestrian distance per best detector mAP.

Distance interval	Traininig (t)	Benchmark (b)	YOLOv3 (t)	TINYv3 (t)	YOLOv3 (b)	TINYv3 (b)
From 81m to infinity	7050	6725	**38.7**	**38.6**	1.8	0.6
From 61m and 80m	**39339**	**15790**	**82.9**	**71.5**	52.4	22.8
From 41m and 60m	**24689**	**7104**	**92.9**	**75.6**	76.5	59.4
From 21m and 40m	8212	2085	91.4	70.9	79.5	51.7
From 0m and 20m	9334	2104	91.9	68.9	69.7	23.2

TABLE 6. Detection evaluation on "Body-parts", "Unknowns?" and "Baby carriage" classes.

Version	Set	mAP	IoU	TP	FP	FN	Average IoU	Precision	Recall	F1-score
YOLOv3	Traininig	**13.2**	50%	743	1021	3657	27.64	0.42	0.17	0.24
TINYv3	Traininig	**3.3**	50%	368	1600	4032	11.37	0.19	0.08	0.12
YOLOv3	Traininig	21.3	25%	186	166	694	32.73	0.53	0.21	0.30
TINYv3	Traininig	21.6	25%	940	824	3460	32.32	0.53	0.21	0.30
YOLOv3	Benchmark	**3.1**	50%	118	394	2109	14.81	0.23	0.05	0.09
TINYv3	Benchmark	**0.78**	50%	83	665	2144	6.80	0.11	0.04	0.06
YOLOv3	Benchmark	5.8	25%	159	353	2068	18.2	0.31	0.07	0.12
TINYv3	Benchmark	3.83	25%	178	570	2049	11.64	0.24	0.08	0.12

TABLE 7. Modified SCUT training results.

Resolution	Version	Depth	Loss	mAP(0.5)	mAP(0.25)	Iteration
640×480	YOLOv3	8bit	0.1456	**86.4**	89.3	269000
640×480	TINYv3	8bit	0.1786	**79.3**	83.3	168000

Unknowns?, Baby carriage, Pets and Animals from the dataset and merged the remaining classes into one category.

III. RESULTS

The training results are presented in table (4). We provide information on experiment configuration and obtained results. Two variants of mAP (for different IoU: 0.5 and 0.25), loss and number of iteration are referenced. Initially, the best performance was registered for unmodified YOLOv3 DNN, reaching the accuracy of 80.5 mAP. The TINYv3 acheived only 66.3 mAP after 243k iterations, which indicates that the network cannot extract more features from the dataset. In this case, we have increased input resolution to 640 × 480, and YOLOv3 improved to 85.4 mAP, which outperforms the initial YOLOv3 by 6%. However, the TINYv3 reached accuracy almost the same as YOLOv3 with an input resolution of 416 × 416. Furthermore, the low pass filter additionally increased accuracy by 4.3% for YOLOv3 and 4.1% for a TINYv3. Finally, we converted the training set to 8bit images with low pass filter to compare the 16bit images versus 8 bit images. The 8 bit TINYv3 reached 71.1 mAP and we stopped the training at 78k iterations because the loss stopped to decreasing. The YOLOv3 reached 79.6 mAP at 123k iterations and the loss stopped decreasing as well. To sum up we increased accuracy by 10.67% with 16bit images and low pass filter.

The visibility distance is another crucial aspect for evaluation of detection accuracy. Figure (8) shows that pedestrians whose height is 1.88 meters in the 100-meter distance would be equal to 21 pixels. Pedestrians farther than 100 meters are very poorly visible. However, Table (5) shows that the training set contains about 39k of annotations at a distance between 61 and 80 meters where the YOLOv3 16bit + low pass version reached 82.9 mAP on the training set and 52.4 mAP on the benchmark. The TINYv3 16bit + low pass version performed worse and entered 71.5 mAP for the training set and 51.7 mAP for the benchmark set. The second biggest interval is from 41 to 61 meters. This interval was the best for both detectors gaining 92.9mAP for YOLOv3 16bit + low pass, 75.6 mAP for TINYv3 16bit + low pass for the training set. The benchmark set had similar results where YOLOv3 16bit + low pass got 76.5 mAP and TINYv3 16bit + low pass 59.4 mAP. The worst results were obviously from 81m going to infinity, where the object got very small.

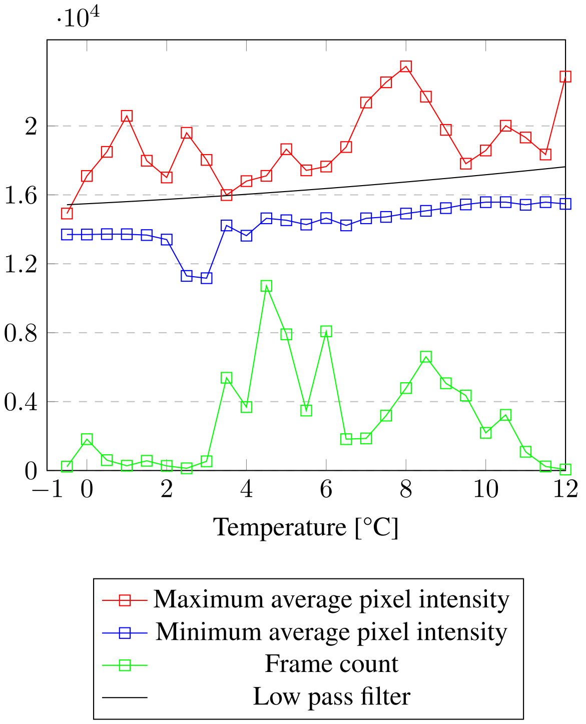


FIGURE 9. Maximum, minimum intensity and temperature distribution per dataset.

We also wanted to measure the detection precision on classes excluded from the training. Table (6) shows that the YOLOv3 and TINYv3 are making very small mAP detection on Body parts, Un-knowns? and Baby carriage classes.

IV. VALIDATION

In validating our training results, we faced two challenges. The first one is that there is no 16bit thermal dataset used for pedestrian detection application, primarily used in competitions like VOT Challenge [42]. The second issue is annotation methodology, since pedestrians can be annotated in many ways and poses, the dataset used for direct comparison should be annotated in the same way to have the most accurate results. For those reasons, we decided to use our dataset

TABLE 8. ZUT and SCUT dataset comparison.

Detector	Training source	Validation set	Samples	mAP	IoU	TP	FP	FN	Average IoU	Recall	Precision	F1-score
YOLOv3	SCUT	ZUT training 8bit + low pass	12008	**37.7**	50%	3647	2534	5874	39.27	0.38	0.59	0.46
TINYv3	SCUT	ZUT training 8bit + low pass	12008	**32.4**	50%	3022	2265	6500	39.12	0.32	0.57	0.41
YOLOv3	SCUT	ZUT training 8bit + low pass	12008	55.0	25%	4503	1678	5019	45.02	0.57	0.73	0.57
TINYv3	SCUT	ZUT training 8bit + low pass	12008	45.8	25%	3633	1654	5889	43.87	0.38	0.69	0.49
YOLOv3	SCUT	ZUT benchmark 8bit + low pass	36128	**27.7**	50%	3796	3543	9280	34.39	0.29	0.52	0.37
TINYv3	SCUT	ZUT benchmark 8bit + low pass	36128	**25.4**	50%	3483	3994	9593	32.01	0.27	0.47	0.34
YOLOv3	SCUT	ZUT benchmark 8bit + low pass	36128	37.8	25%	4495	2844	8581	38.31	0.34	0.61	0.44
TINYv3	SCUT	ZUT benchmark 8bit + low pass	36128	33.9	25%	4092	3385	8984	35.38	0.31	0.55	0.40
YOLOv3	ZUT 8bit + low pass	SCUT Validation	76384	**39.1**	50%	38059	7008	84478	56.08	0.31	0.84	0.45
TINYv3	ZUT 8bit + low pass	SCUT Validation	76384	**32.6**	50%	29597	10514	92940	46.96	0.24	0.74	0.36
YOLOv3	ZUT 8bit + low pass	SCUT Validation	76384	55.7	25%	43996	1071	78541	61.86	0.36	0.98	0.52
TINYv3	ZUT 8bit + low pass	SCUT Validation	76384	57.0	25%	38011	2100	84526	55.97	0.31	0.95	0.47

TABLE 9. ZUT combined with SCUT dataset comparison.

Detector	Training source	Validation set	Samples	mAP	IoU	TP	FP	FN	Average IoU	Recall	Precision	F1-score
YOLOv3	SCUT + ZUT 8bit + low pass	ZUT training 8bit + low pass	12008	**82.7**	50%	6530	1009	2992	62.29	0.69	0.87	0.77
TINYv3	SCUT + ZUT 8bit + low pass	ZUT training 8bit + low pass	12008	**73.9**	50%	6882	1806	2642	56.67	0.72	0.79	0.76
YOLOv3	SCUT + ZUT 8bit + low pass	ZUT training 8bit + low pass	12008	92.1	25%	6967	572	2555	64.70	0.73	0.92	0.82
TINYv3	SCUT + ZUT 8bit + low pass	ZUT training 8bit + low pass	12008	89.8	25%	7715	971	1807	60.57	0.81	0.89	0.85
YOLOv3	SCUT + ZUT 8bit + low pass	ZUT benchmark 8bit + low pass	36128	**69.2**	50%	6705	1059	6371	60.24	0.51	0.86	0.64
TINYv3	SCUT + ZUT 8bit + low pass	ZUT benchmark 8bit + low pass	36128	**58.6**	50%	6880	2008	6196	54.02	0.53	0.77	0.63
YOLOv3	SCUT + ZUT 8bit + low pass	ZUT benchmark 8bit + low pass	36128	79.7	25%	7193	571	5883	62.92	0.55	0.93	0.69
TINYv3	SCUT + ZUT 8bit + low pass	ZUT benchmark 8bit + low pass	36128	72.5	25%	7692	1196	5384	57.86	0.59	0.87	0.70
YOLOv3	SCUT + ZUT 8bit + low pass	SCUT Validation	76384	**80.8**	50%	86512	7156	36025	71.19	0.71	0.92	0.80
TINYv3	SCUT + ZUT 8bit + low pass	SCUT Validation	76384	**78.1**	50%	85661	12426	36876	66.04	0.70	0.87	0.78
YOLOv3	SCUT + ZUT 8bit + low pass	SCUT Validation	76384	83.9	25%	87991	5677	34546	71.87	0.72	0.94	0.81
TINYv3	SCUT + ZUT 8bit + low pass	SCUT Validation	76384	83.7	25%	88500	9587	34037	67.22	0.72	0.90	0.80

8bit + low pass version and compare detectors YOLOv3 and TINYv3 against the SCUT test dataset. Besides, we had deeply analyzed the SCUT dataset annotation methodology and found that there are many cases when two pedestrians touched with the hand is marked as a single annotation. Also, when there is a case of group people (two pedestrians visible, others not), it was marked as a single group of people annotation, including partially visible pedestrians. This part was taken with the care in the ZUT dataset. We marked clearly visible pedestrians as pedestrians, and only the occluded and partially visible were marked as an occluded annotation. To solve this incompatibility in annotation methodology, we have iterated through all SCUT dataset and excluded frames containing a group of people annotations and people annotations similar to the square shape. Also, to make a competition fair, we reused our YOLO 8bit configuration and trained it on the modified SCUT dataset. Important to mention, we have also scaled down all the images to 640×480 resolution, since the original source resolution was lower and merged with other classes to people class. The dataset shrunk to 78,942 frames (118,377 annotations) for the training and 76,381 frames (122,537 annotations) for the testing.

In the table (7) we have presented the training results of YOLOv3 and TINYv3 detectors of the modified SCUT dataset. We have used two thresholds for IoU, which were set to 50% and 25%. The YOLOv3 version reached 86.4 mAP, which was very close to our 16bit version and outperformed our 8bit version. The TINYv3 reached up to 79.3 mAP, better than the 8bit version but still not enough to compete with 16bit modification.

In the table (8) we have provided mAP, Average IoU, Recall, Precision, F1-score as well as True Positive (TP), False Positive (FP) and False Negative (FN) measures having two thresholds (25% & 50%) of IoU. The strategy of comparison was to take SCUT and compare it against ZUT training. Then benchmark sets and after that do oppositely for ZUT. Such comparison revealed that both sets are not performing verywell against each other since both of them were collected in different weather conditions and location and surroundings do not have much of a familiar context. The best results for SCUT were on the ZUT training set, having 37.7 mAP of 50% IoU by YOLOv3. The ZUT performed similarly on the SCUT dataset, reaching 39.1 mAP with YOLOv3, at the most.

Since both sets performed similarly, we additionally decided to join them into one set, retrain YOLOv3 & TINYv3 and repeat the same validation once again. In the table (9) we found that precision improved a lot. YOLOv3 on the ZUT training set reached 82.7 mAP, and on the ZUT benchmark 69.2 mAP and 80.8 mAP. TINYv3 also improved and achieved 73.9 mAP on the ZUT training set, 58.6 mAP on the benchmark set, and 78.1 mAP on SCUT.

V. CONCLUSION

To sum up, in this paper we have provided a ZUT dataset that contains 122k annotations and more than 79k (3) collected during the drizzle or the rain. The remaining annotations were collected during frosty and cloudy conditions. Only 752 annotations were observed when the sky was clear. In addition to this, the dataset includes car CAN data, which can be used for creating ADAS systems for thermal image based detectors.

The CAN data can contribute to better system performance. Although we have investigated in the paper, only one of its components - the temperature of the environment - there are many more potential applications. We proposed the normalization procedure, which utilizes the temperature information, and we registered a few percent improvements. Besides this, the speed of the car can be used for a great advantage to modify tracker parameters. According to the speed or driving pattern, the threshold of detection probability can be adjusted. Driving pattern alone can be determined on frequencies of using break and acceleration pedals (for a city, there are many brakes and accelerations, Authobans characterize with long brakes and accelerations).

Furthermore, the proposed modifications show that using 16bit images instead of 8bit improves detection accuracy by 10.67%. The low pass filter also gives improvements in increasing accuracy by four percent, however the more complex filter could improve the accuracy further because the current proposition is based on average results, and in this dataset, we had not enough samples in the temperature range between −1.5 to 4 °C. Also, the onboard precipitation sensor would help in adjusting the intensity of the image, because currently we cannot apply any further real-time enhancements in regards to the rain or fog.

Finally, the comparison of SCUT and ZUT databases showed that a wider variety of annotations made a much stronger detector, which is capable of work in severe and good weather conditions. This also concludes that another dataset should be collected during the spring and summer seasons to add more samples for better detection accuracy.

ACKNOWLEDGMENT

The authors would like to thank Sandra Tumiene for her contribution to the annotating process and given quality.

REFERENCES

[1] P. Tumas, A. Nowosielski, and A. Serackis, "ZUT-FIR-ADAS," *IEEE Dataport*, 2020. Accessed: Apr. 3, 2020, doi: 10.21227/7f37-hx89.

[2] *Zut-Fir-Adas*. Accessed: Jan. 19, 2020. [Online]. Available: https://github.com/pauliustumas/ZUT-FIR-ADAS

[3] *Global Status Report on Road Safety 2018*. Accessed: Jan. 3, 2020. [Online]. Available: https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/

[4] *2018 Road Safety Statistics: What is Behind the Figures?* Accessed: Jan. 3, 2020. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/MEMO_19_1990

[5] A. K. Jägerbrand and J. Sjöbergh, "Effects of weather conditions, light conditions, and road lighting on vehicle speed," *SpringerPlus*, vol. 5, no. 1, p. 505, Dec. 2016, doi: 10.1186/s40064-016-2124-6.

[6] G. Zhang, K. K. W. Yau, X. Zhang, and Y. Li, "Traffic accidents involving fatigue driving and their extent of casualties," *Accident Anal. Prevention*, vol. 87, pp. 34–42, Feb. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001457515301159

[7] C.-F. Lio, H.-H. Cheong, C.-H. Un, I.-L. Lo, and S.-Y. Tsai, "The association between meteorological variables and road traffic injuries: A study from macao," *PeerJ*, vol. 7, p. e6438, Feb. 2019.

[8] J. Davidović, E. Smailović, N. Marković, and B. Antić, "The influence of weather conditions on road safety," *Put i saobraćaj*, vol. 63, no. 4, pp. 13–20, May 2019. [Online]. Available: http://www.putisaobracaj.rs/index.php/PiS/article/view/88

[9] *Night View*. Accessed: Jan. 3, 2020. [Online]. Available: http://www.toyota-myanmar.com/innovation/safety-technology/safety-technology-2/safety-technology-3/radar-cruise-control-2/night-view

[10] A. Nowosielski, K. Małecki, P. Forczmański, and A. Smoliński, "Pedestrian detection in severe lighting conditions: Comparative study of human performance vs thermal-imaging-based automatic system," in *Progress in Computer Recognition Systems*, R. Burduk, M. Kurzynski, and M. Wozniak, Eds. Cham, Switzerland: Springer, 2020, pp. 174–183.

[11] *Flir Path Finder Kit*. Accessed: Jun. 11, 2019. [Online]. Available: http://www.safetyvision.com/sites/safetyvision.com/files/FLIR_PathFindIRII_User_Guide_1.pdf

[12] D. Forslund and J. Bjarkefur, "Night vision animal detection," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 737–742.

[13] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[14] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. Accessed: Jan. 5, 2020. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[16] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Apr. 2015, pp. 1440–1448.

[17] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," 2017, *arXiv:1711.06897*. [Online]. Available: http://arxiv.org/abs/1711.06897

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[19] M. Ju, H. Luo, Z. Wang, B. Hui, and Z. Chang, "The application of improved YOLO V3 in multi-scale target detection," *Appl. Sci.*, vol. 9, no. 18, p. 3775, 2019.

[20] W. Tianshu, Z. Zhijia, L. Yunpeng, P. Wenhui, and C. Hongye, "A lightweight small object detection algorithm based on improved SSD," *Infr. Laser Eng.*, vol. 47, no. 7, 2018, Art. no. 703005.

[21] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster R-CNN doing well for pedestrian detection?" in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9906. Cham, Switzerland: Springer, 2016, pp. 443–457.

[22] Y. Socarras, S. Ramos, D. Vázquez, A. López, and T. Gevers, "Adapting pedestrian detection from synthetic to far infrared images," in *Proc. ICCV*, Jan. 2013, pp. 1–3.

[23] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu, and A. López, "Pedestrian detection at Day/Night time with visible and FIR cameras: A comparison," *Sensors*, vol. 16, no. 6, p. 820, 2016.

[24] *Flir Thermal Sensing for Adas*. Accessed: Jun. 6, 2019. [Online]. Available: https://www.flir.com/oem/adas/adas-dataset-form/

[25] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "KAIST multi-spectral Day/Night data set for autonomous and assisted driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 934–948, Mar. 2018.

[26] I. Jegham and A. Ben Khalifa, "Pedestrian detection in poor weather conditions using moving camera," in *Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2017, pp. 358–362.

[27] A. Khellal, H. Ma, and Q. Fei, "Pedestrian classification and detection in far infrared images," in *Intelligent Robotics and Applications*, H. Liu, N. Kubota, X. Zhu, R. Dillmann, and D. Zhou, Eds. Cham, Switzerland: Springer, 2015, pp. 511–522.

[28] J. W. Davis and M. A. Keck, "A two-stage template approach to person detection in thermal imagery," in *Proc. 7th IEEE Workshops Appl. Comput. Vis. (WACV/MOTIO)*, vol. 1, Jan. 2005, pp. 364–369, doi: 10.1109/ACVMOT.2005.14.

[29] A. Miron, "Multi-modal, multi-domain pedestrian detection and classification: Proposals and explorations in visible over StereoVision, FIR and SWIR," Ph.D. dissertation, Babes-Bolyai Univ., Cluj-Napoca, Romania, Jul. 2014.

[30] *Terravic Motion IR Database*. Accessed: Jun. 11, 2019. [Online]. Available: http://vcipl-okstate.org/pbvs/bench/

[31] Z. Xu, J. Zhuang, Q. Liu, J. Zhou, and S. Peng, "Benchmarking a large-scale FIR dataset for on-road pedestrian detection," *Infr. Phys. Technol.*, vol. 96, pp. 199–208, Jan. 2019.

[32] *Guangzhou Weather by Months*. Accessed: Jan. 3, 2020. [Online]. Available: https://en.climate-data.org/asia/china/guangdong/guangzhou-2309/

[33] *Fine Against Individual in Austria*. Accessed: Jan. 3, 2020. [Online]. Available: https://easygdpr.eu/gdpr-incident/strafe-gegen-privatperson-wegen-dashcam/

[34] *Driver Warning—Your Dash Cam Could Land You Up to £9,000 Fine and See You Jailed Abroad*. Accessed: Jan. 3, 2020. [Online]. Available: https://www.express.co.uk/life-style/cars/998528/Dash-cam-car-Europe-fines-prison/

[35] *Baumer SDK*. Accessed: Jan. 3, 2020. [Online]. Available: https://www.baumer.com/ch/en/product-overview/industrial-cameras-image-processing/software/baumer-gapi-sdk/c/14174

[36] P. Tumas and A. Serackis, "Automated image annotation based on YOLOv3," in *Proc. IEEE 6th Workshop Adv. Inf., Electron. Electr. Eng. (AIEEE)*, Nov. 2018, pp. 1–3.

[37] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[38] *YBAT—YOLO BBox Annotation Tool*. Accessed: Jan. 3, 2020. [Online]. Available: https://github.com/drainingsun/ybat

[39] K. C. Chan, A. Ayvaci, and B. Heisele, "Partially occluded object detection by finding the visible features and parts," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2130–2134.

[40] C. Zhou and J. Yuan, "Multi-label learning of part detectors for heavily occluded pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3506–3515.

[41] *Darknet YOLO Implementation*. Accessed: Jan. 3, 2020. [Online]. Available: https://github.com/AlexeyAB/darknet/commit/dcfeea30f195e0ca1210d580cac8b91b6beaf3f7

[42] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016.

P. TUMAS (Member, IEEE) was born in 1991. He received the M.Sc. degree from Vilnius Gediminas Technical University, in 2016, where he is currently pursuing the Ph.D. degree in pedestrian detection in FIR domain. During the studies, he has published five articles which main focus is topology of pedestrian detection systems, dataset analysis, and research of deep neural networks.

A. NOWOSIELSKI was born in 1978. He received the M.Sc. and Ph.D. degrees from the Faculty of Computer Science and Information Technology, Technical University of Szczecin (now West Pomeranian University of Technology, Szczecin), in 2002 and 2006, respectively. He is an author of more than 60 scientific articles. His current research interests include computer vision, visual surveillance, digital image processing, and human–computer interaction.

A. SERACKIS (Senior Member, IEEE) was born in 1980. He received the M.Sc. and Ph.D. degrees from Vilnius Gediminas Technical University (VGTU), in 2004 and 2008, respectively. He was an Associate Professor, in 2012, and a Professor, in 2017, with VGTU. He is an author of more than 60 scientific articles, two textbooks, and a monograph. His current research interests include real-time image and signal processing, development, and application of intelligent systems.