Vehicle-Integrated Anti-Abandonment System (VIAAS)

A neural network-based software for in-vehicle detection of unattended pets and children

Giorgio Di Loro
University of Modena and Reggio Emilia
Modena, Italy
278606@studenti.unimore.it

***Abstract*—This paper describes the implementation of the software for a Vehicle-Integrated Anti-Abandonment System (VIAAS) designed to prevent deaths caused by forgetfulness. This idea is motivated by the fact that many children, as well as pets, die every year because of vehicular heatstroke: vehicles require safety systems that allow to detect dangerous scenarios and avoid these preventable deaths. VIAAS operations rely on neural networks for object detection and classification, face detection, age estimation; the system is designed in order to be potentially deployed in commercial cars as it is lightweight and only requires a camera to collect the pictures of the interior of the vehicle. For instructional purposes I decided to analyze different architectures that perform the operations required by VIAAS. The paper has the following structure: an introduction where I discuss the reasons and ideas behind the project; a summary of related works and the general framework of VIAAS-like systems; a description of the approach used for developing and implementing the system; the experimental results obtained using my VIAAS software; a bibliography with all the references used for developing the project.**

***Keywords*—Convolutional Neural Networks, automotive, detection and classification, artificial intelligence, safety**

I. INTRODUCTION

On average, 38 children die from heatstroke inside hot vehicles each year. During the period from 1998 to 2019 that's a total of 849 Pediatric Vehicular Heatstroke (PVH) deaths that have been documented in the United States only [1]. The research documented in [1] and [2] shows that deaths caused by forgetting children inside cars peak at ~200 cases per month during summer and that the average number of deaths per year is slowly increasing year by year. These casualties belong mainly to the age range between 0 and 6 years old, but some victims can also be found between older individuals, the oldest being 16 years old. Vehicular heatstroke is also the cause of deaths for many household pets, in particular dogs, that are locked inside their owner's car out of forgetfulness [3]. The relevance of this matter stands in the fact that these deaths are totally preventable, but they happen nonetheless because of the ways in which our brain works [4]. Psychological research [5] shows that, when under stress, our brain can forget even the most important things; moreover, it states that forgetting a child inside one's car is something that could happen to everyone, especially considering the stressful times we are currently facing. This is the reason why the "Hot Cars Act of 2019" [6] of the U.S. congress states that "all new passenger motor vehicles…should be equipped with an alert system to detect the presence of an occupant (e.g., a child or domestic animal) in a rear designated seating position after the vehicle engine is turned off.".

VIAAS is designed with the specific purpose of showing a prototype example for a system that provides the functionalities required in order to avoid further unacceptable deaths, while exploring the possible architectures and software features that allow to achieve this goal.

II. RELATED WORKS

A. *Object detection and classification*

VIAAS relies heavily on object detection and classification since it must be able to detect, correctly recognize and locate both persons and pets inside an image. These tasks have been historically addressed by countless researchers and neural network architectures, respectively. The paper [7] depicts a rather interesting and comprehensive timeline of convolutional neural networks that work on images: AlexNet [8], GoogLeNet [9], VGG [10] and ResNet [11] are just some of the most relevant examples. For what concerns object detection and classification, in particular, we can identify two main categories: region proposal-based networks (such as R-CNN [12], Fast R-CNN [13] and Faster R-CNN [14] for example) and regression/classification-based networks (such as AttentionNet [15], G-CNN [16] and the YOLO [17] family). The first class of networks is based on a two-step process that resembles the attentional mechanism of the human brain, which gives a first scan to the whole scenario and then focuses on regions of interest; the latter is based on global regression/classification, mapping straightly from image pixels to bounding box coordinates and class probabilities. The part of VIAAS dedicated to pet detection implements an architecture belonging to the regression/classification class, the YOLOv3 architecture [18].

B. *Face detection*

Face detection is an important task of the system for two main reasons: first of all, to be able to identify the position of the face inside the image, then to have exact information on how to crop out the face from the image, which must be fed to the part of the system that detects the age from a face. Face detection is once again one of the most studied and researched tasks that neural networks (and convolutional neural networks specifically) can tackle: some examples in literature are modifications based on R-CNN or Faster R-CNN architectures [19][20], DenseBox [21], multi-task convolutional neural networks (MTCNN) and others. The network for face detection implemented in VIAAS is a MTCNN inspired by [22].

C. *Age estimation*

For age estimation, lots of examples can be found in literature as well: these models were developed thanks to the huge amount of labeled data collected in databases such as the FaceTracer Database [23], which separates ages in 5 classes (baby, child, youth, middle aged, senior), or the Adience Dataset [24], which separates ages in 8 age ranges (0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60-). Example networks that perform age estimation are described in [25] (two-level system that classifies samples into overlapping age groups and then estimates the apparent age with local regressors, whose outputs are then fused for the final estimate) and [26] (a two-level CNN architecture including feature extraction and classification itself trained and tuned on two different datasets). The neural network implemented in VIAAS is inspired by [27] and consists of a simple CNN that will be described later.

D. *Anti-Abandonment systems*

VIAAS is, to my knowledge, a one-of-a-kind system. I have not been able to find similar projects or even designs (even if my research were not extensive and maybe someone could be currently working on similar implementations), whereas the child detection task is already tackled by laws [28] or systems like [29], which consist in third-party car seats for kids that can alert the driver or in sensors and actuators embedded in the car's seats themselves to perform the same function. With the increasing pervasiveness of electronics inside the car and the implementation of new driver/passenger monitoring systems, VIAAS could be a valid alternative to seat-mounted sensor detection which would not require additional hardware.

III. DESCRIPTION OF THE APPROACH

The pipeline of the system is shown in Fig.1 below. The general architecture consists of 3 macro layers: the first level neural network layer, where two separate NNs perform feature extraction from the input image (the first one is dedicated to detecting the presence of any dog or cat inside the vehicle, whereas the second one performs face detection and localization); the second level NN layer, where age is estimated for the input faces detected in the first layer; the output layer, where information extracted at the previous steps is processed in order to issue an alert if pets or kids have been detected. The following paragraphs describe the different building blocks of the architecture in detail. Before feeding the image to the first level networks, the software uses the functions of the OpenCV and Pillow modules to open the image that has to be analyzed.

A. *Pet detection*

The neural network used for pet detection (object detection and classification) is a regression/classification-based network whose architecture is inspired to the YOLOv3 model [18]. The choice of a regression/classification-based network was motivated by different reasons: it is a lightweight network that is able to infer predictions in a small amount of time; it is pretty accurate on various image datasets, while also being able to detect a lot of different classes of objects; it is suitable for the instructional purposes of the project, since we didn't investigate this particular architecture during our course and it was an interesting occasion to study some alternative implementation. The network is composed by 53 total layers, depicted in Fig.2.

	Type	Filters	Size
	Convolutional	32	3 × 3
	Convolutional	64	3 × 3 / 2
1×	Convolutional	32	1 × 1
	Convolutional	64	3 × 3
	Residual		
	Convolutional	128	3 × 3 / 2
2×	Convolutional	64	1 × 1
	Convolutional	128	3 × 3
	Residual		
	Convolutional	256	3 × 3 / 2
8×	Convolutional	128	1 × 1
	Convolutional	256	3 × 3
	Residual		
	Convolutional	512	3 × 3 / 2
8×	Convolutional	256	1 × 1
	Convolutional	512	3 × 3
	Residual		
	Convolutional	1024	3 × 3 / 2
4×	Convolutional	512	1 × 1
	Convolutional	1024	3 × 3
	Residual		
	Avgpool		Global
	Connected		1000
	Softmax		

Fig. 2. Architecture of the pet detector NN (YOLOv3 based)

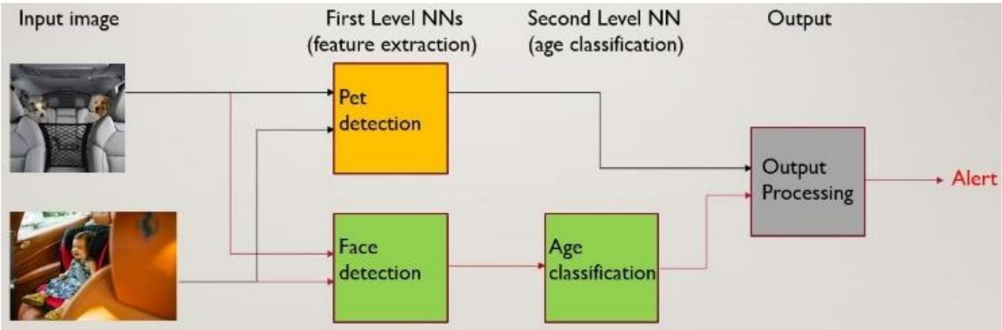

Fig. 1. High-level architecture of the VIAAS project

It is important to notice that similar, but newer and more refined architectures, are already available (YOLOv4 [30]) or under development (YOLOv5) and could be used to further improve VIAAS performance, even though the YOLOv3-based architecture allows to perform the required operations nonetheless, as we will see in the experimental results section. For this stage, a pretrained model has been used for detecting just cats and dogs (but the number of classes can be broadened by passing a parameter to the network class) since the purpose of the VIAAS is constrained to detecting pets and children.

B. *Face detection*

The neural network used for face detection is a multi-task convolutional neural network (MTCNN), which is one of the possible types of region proposal neural networks described before. The architecture of the network is inspired to the one described in [22] for performing face detection with bounding boxes and facial landmark detection. This VIAAS component consists of cascaded CNNs designed for real time performance and divided in 3 stages:

1. Proposal network (P-Net), a fully convolutional network used to obtain the candidate windows and their bounding box regression vectors.
2. Refine network (R-Net), a CNN that takes the candidates from P-Net, rejects a large number of false candidates and performs calibration with bounding box regression.
3. Output network (O-Net), similar to the second stage, that describes the face in more detail, detecting five facial landmarks' positions.

The CNN architectures employed are shown in Fig.3 below.

C. *Age classification*

The neural network used for age classification is a CNN composed by 6 layers: 3 convolutional layers and 3 fully connected layers. The architecture of the network is inspired to [27] and depicted in Fig. 4. The input of the network is the crop of the face obtained thanks to the bounding box coordinates obtained by the MTCNN in the previous step. Convolutional layer 1 consists of 96 filters of size 3x7x7 pixels, followed by a Rectified Linear Unit (ReLU), a max pooling layer (MP) taking the maximal value of 3x3 regions with two-pixel strides and a local response normalization layer. Convolutional layer 2 contains 256 filters of size 96x5x5 pixels, followed again by ReLU, MP and a local response normalization layer. Convolutional layer 3 applies a set of 384 filters of size 256 x 3 x 3 pixels, followed by ReLU and MP. Fully connected layer 1 receives the output of the third convolutional layer and contains 512 neurons, followed by a ReLU and a dropout layer. Fully connected layer 2 receives the 512-dimensional output and again contains 512 neurons, followed by a ReLU and a dropout layer. Fully connected layer 3 maps to the final class for age classification and its output is fed to a soft-max layer to obtain the probability values. The estimation itself is made by taking the class with the maximal probability for the given test image.

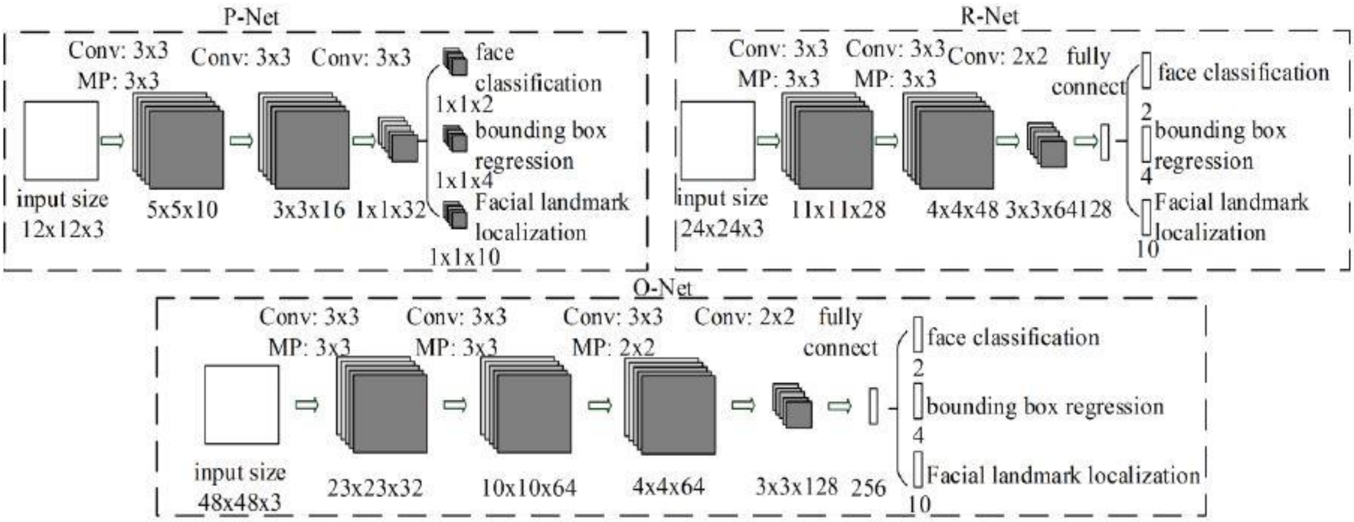

Fig. 3. Architecture of the cascaded CNNs for the face detector

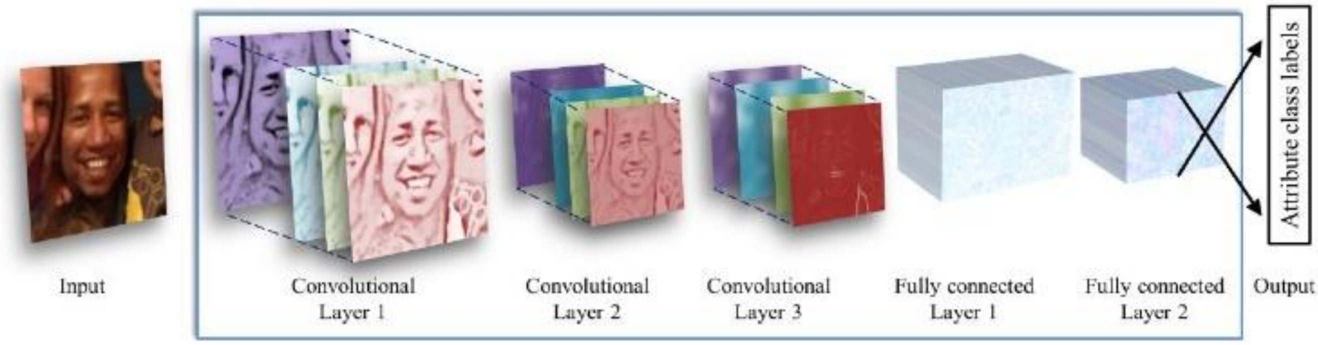

Fig. 4. Architecture of the CNN for age classification

D. *Output processing*

The output layer of VIAAS consist of the software required to take action and eventually alert the driver once that the detection and classification process is over. The pet detection network returns a Boolean flag (stored in the "pet_detected" variable) which is set to True if a dog or a cat is detected. The age estimation network returns a Boolean flag (stored in the "kid_detected" variable) which is set to True if the age range estimated is between the (0-2) and (15-20) ranges (according to the data on the subjects affected by vehicular heatstroke showed in [1] and referenced in the introduction of this paper). The software does then a check on the pet_detected flag: if it is True the system alerts the driver by displaying a specific message. The same is done for the kid_detected flag separately. If they are both False nothing needs to be signaled.

IV. Experimental Results

A. *Pet detection*

The pet detector is trained using full size images and with techniques such as data augmentation and batch normalization and its performances and accuracy versus inference time can be seen in [18]. It is able to detect quite confidently the presence of an animal inside the car, as in the example of Fig. 5 where the VIAAS shows the output displayed in Fig. 6 (probability of the detection being a cat plus its location).

Fig. 5. Example image of a pet (cat) inside the car

```
cat  :  99.33752417564392  :  [94, 49, 176, 149]
--------------------------------
ALERT: unattended pet left in the car
```

Fig. 6. Output of the VIAAS for input as in Fig. 5

B. *Face detection and age classification*

The face detector is trained on labeled images from WIDER FACE and CelebA. The performance of the face detection algorithm is then evaluated by comparing it to the state-of-the-art methods in FDDB and the state-of-the-art methods in WIDER FACE (data might be outdated, check [22] for details). The age classification layer is trained with dropouts [27] (i.e., randomly setting the output value of network neurons to zero) on the Adience dataset. This part of the system is the most critical one, since, as I will show in the next section, age estimation is the most fallible and depends on different properties of the input face. The output of VIAAS for the input face shown in Fig.7 is displayed in Fig.8.

Fig. 7. Example image of a kid with a plush inside the car

```
Found 1 faces
Age Range: (4, 6)
ALERT: unattended kid left in the car
```

Fig. 8. Output of the VIAAS for input as in Fig. 7

C. VIAAS

The overall system performs the operations as intended, successfully detecting pets and kids, as well as not going off for possible false positives such as dolls, in a lot of occasions. The main critical aspects, however, are those related to the age classification part of the network: variations in posture, presence of glasses and other variables can negatively influence the estimation resulting in both false positives and false negatives. An example of partially wrong output is given in Fig. 10 where the input image is Fig. 9: here the dog is correctly detected (and the alert is fired) but the age of the child is completely off (25-32 range) because of the glasses she is wearing, that make the network unable to classify her correctly. Improvements are certainly possible with different networks, more extensive training or other several variations to the project. The case of an avoided false positive is shown in Fig.11 and Fig.12

Fig. 9. Example image of a kid and a dog inside the car

```
dog  :  99.37427639961243  :  [334, 142, 475, 328]
--------------------------------
Found 1 faces
Age Range: (25, 32)
ALERT: unattended pet left in the car
```

Fig. 10. Output of the VIAAS for input as in Fig. 9

Fig. 11. Example image of a cat plush inside the car

```
Everything ok

Process finished with exit code 0
```

Fig. 12. Output of the VIAAS for input as in Fig. 11

V. Conclusions

Even if VIAAS is still fallible in some of its parts, the results achieved with this rather simple system are significant and show an interesting application of technologies that are nowadays deployable even in mobile devices. Moreover, this project allowed me to deepen the topics treated during the course and to individually research a lot of information on neural networks, their history, the ideas behind their development and the general programming framework to which they belong. As a final note, I thank the professors of the course for giving us the tools to understand these technologies and to allowing us to develop a project independently.

REFERENCES

[1] J. Null, “Trends and Patterns: Pediatric Vehicular Heatstroke Deaths 1998-2019”

[2] KidsAndCars.org, “U.S. Child Hot Car Deaths from the KidsAndCars.org National Database (1990-2019)”

[3] M.Torres, “What's Happening In Your Brain When You Forget Your Kid In The Car”

[4] PETA.org, “31 Animal Companions Endured Heat-Related Deaths in 2020… That We Know Of”

[5] D. M. Diamond, “When a child dies of heatstroke after a parent or caretaker unknowingly leaves the child in a car: How does it happen and is it a crime?”

[6] 116th U.S. Congress, “H.R.3593 - Hot Cars Act of 2019”

[7] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu and Xindong Wu “Object Detection with Deep Learning: A Review”

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions”

[10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”

[13] R. Girshick, “Fast r-cnn”

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards realtime object detection with region proposal networks”

[15] D. Yoo, S. Park, J.-Y. Lee, A. S. Paek, and I. So Kweon, “Attentionnet: Aggregating weak directions for accurate object detection”

[16] M. Najibi, M. Rastegari, and L. S. Davis, “G-cnn: an iterative grid based object detector”

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection”

[18] J. Redmon, A. Farhadi , “YOLOv3: An Incremental Improvement”

[19] H. Wang, Z. Li, X. Ji, and Y. Wang, “Face r-cnn”

[20] X. Sun, P. Wu, and S. C. Hoi, “Face detection using deep learning: An improved faster rcnn approach”

[21] L. Huang, Y. Yang, Y. Deng, and Y. Yu, “Densebox: Unifying landmark localization with end to end object detection”

[22] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks”

[23] N. Kumar, P. N. Belhumeur, and S. K. Nayar, "FaceTracer: A Search Engine for Large Collections of Images with Faces"

[24] E. Eidinger, R. Enbar, and T. Hassner, “Age and gender estimation of unfiltered faces”

[25] F. Gurpinar, H. Kaya, H. Dibeklioglu, A. Salah, “Kernel ELM and CNN Based Facial Age Estimation”

[26] O. Agbo-Ajala, S. Viriri, “Deeply Learned Classifiers for Age and Gender Predictions of Unfiltered Faces”

[27] G. Levi, T. Hassner, “Age and Gender Classification using Convolutional Neural Networks”

[28] UL.com, “Italy Becomes First EU Country to Mandate Anti-Abandonment Car Seat Technology“

[29] Corporate.leasys.com, “Children in the car: everything you need to know about anti-abandonment devices”

[30] Z. Yang and R. Nevatia, “A multi-scale cascade fully convolutional network face detector”

[31] Python.org, “Python documentation”

[32] Pytorch.org, “Pytorch documentation”

[33] Opencv.org, “OpenCV documentation”

[34] ImageAI.org, “ImageAI documentation”