Face detection in thermal images with YOLOv3*

Gustavo Silva[1], Rui Monteiro[1], André Ferreira[2], Pedro Carvalho[3], and Luís Corte-Real[1,3]

[1] Faculty of Engineering, University of Porto, Portugal
{silva95gustavo,ruipauloaraujomonteiro}@gmail.com, lreal@fe.up.pt
[2] Bosch Car Multimedia Portugal, S.A., Braga, Portugal
andre.ferreira2@pt.bosch.com
[3] INESC TEC, Porto, Portugal
pedro.carvalho@inesctec.pt, lreal@fe.up.pt

Abstract. The automotive industry is currently focusing on automation in their vehicles, and perceiving the surroundings of an automobile requires the ability to detect and identify objects, events and persons, not only from the outside of the vehicle but also from the inside of the cabin. This constitutes relevant information for defining intelligent responses to events happening on both environments. This work presents a new method for in-vehicle monitoring of passengers, specifically the task of real-time face detection in thermal images, by applying transfer learning with YOLOv3. Using this kind of imagery for this purpose brings some advantages, such as the possibility of detecting faces during the day and in the dark without being affected by illumination conditions, and also because it's a completely passive sensing solution. Due to the lack of suitable datasets for this type of application, a database of in-vehicle images was created, containing images from 38 subjects performing different head poses and at varying ambient temperatures. The tests in our database show an AP_{50} of 99.7% and an AP of 78.5%.

Keywords: thermal imaging · face detection · computer vision · deep learning · YOLOv3 · transfer learning

1 Introduction

In an autonomous driving environment, solutions for interior vehicle monitoring become a necessity, namely to monitor occupants and car interior, tasks that are mainly associated to the driver in modern transportation systems.

A possible approach for in-vehicle interior sensing considers the visible domain, namely the use of RGB cameras. These images greatly depend on external conditions, namely light. This introduces a considerable limitation if we aim to monitor the vehicle during the 24 hours of a day. With this in mind, other modalities are being explored, that can be used independently or in conjunction. An example is near-infrared (NIR), robust to lack of lighting but requires

* Supported by Bosch Car Multimedia Portugal, S.A. and INESC TEC Porto, Portugal

a dedicated source of IR light and filters and is also sensible to different lens exposures [6]. The modality we explore in this document, thermal images, is a passive solution robust to any external light conditions.

Face detection is an object detection task with the specific goal of detecting faces in images. It is the first and one essential step for other more complex tasks, such as face verification and person identification, and so it can be used in many areas such as bio-metrics, security and entertainment.

In the visible domain, traditional approaches include the Viola-Jones framework [10], developed in 2004 and capable of performing in real-time. In the thermal domain, researchers were able to enhance and apply the same method to thermal images [1]. They employed and tested different types of features and concluded that the performance of the system was better using LBP features [7]. Recently, deep convolutional neural networks (CNNs) are being used to improve those results. In [4], transfer learning using a pre-trained Inception model [9] on visible images was successfully applied to thermal face detection, achieving a Positive Predictive Value (PPV) of 99.5%.

2 Dataset

In order to have training and test data for the algorithms described in this document, and due to the lack of labeled and suitable datasets for this application in the thermal domain, a database was created by capturing images inside a vehicle. This dataset is not limited to data important for the purposes of this paper, but also includes subjects performing activities that are relevant for the development of other monitoring algorithms.

The setup consisted of a camera operating both in the infrared thermal and in the visible light spectrum. For this purpose, the FLIR ONE Pro camera was chosen, mainly due to the fact that it combines both modalities in a small device and ensures calibration between frames. This camera has a thermal resolution of 160x120 and an RGB resolution of 1440x1080, capturing frames at a rate of 8.7 per second with a FOV of 55º ×43º ±1. The thermal sensor operates in the 8 – 14μm waveband, measuring temperatures between -20ºC and 400ºC with a thermal sensitivity of 0.15ºC. The camera was placed in front of the passenger using a folding arm, connected via USB to a Linux-running machine (NVIDIA Jetson TX2). Since the camera was designed to be controlled with a smartphone, a driver had to be developed to connect it to an embedded device, and also to extract raw temperature values provided by the capturing device with the highest thermal resolution (14bit). The described setup can be seen in Fig. 1.

The participants were asked to perform some specific actions and activities, namely head movements in multiple axis, simulating facial expressions, simulating fatigue, wearing glasses, smoking, entering the vehicle and leaving the vehicle. Additionally, in the middle of the session, the air conditioning system of the vehicle was adjusted to change the cabin temperature.

Fig. 1. The recording setup.

In total, the database contains recordings of 38 subjects. In terms of gender distribution, data was captured from 33 males and 5 female subjects of white ethnicity. The average age of the recorded population is 28.8 with standard deviation of 10.1 years, and the average height is 1.76m with a standard deviation of 0.10m. Regarding hair size, there were 3 bald subjects, 27 with short hair, 3 with medium hair and 5 with long hair. Furthermore, 63% of the subjects had a beard and 66% had a mustache.

In total, the database contains 87286 frames, where 5361 images were auto-labeled with facial bounding boxes using an RGB face detetor [11] and manually filtered to remove incorrect labels generated by the automatic labeling process. Example images can be seen in Fig. 2.

3 Implementation

Our face detection algorithm is based on the YOLOv3 [8] real-time general object detector. The feature extractor of YOLOv3 is pre-trained on a large amount of visible images from ImageNet [2] and the full object detection framework is then trained on COCO [5] database. In order to perform face detection in thermal images, we take advantage of those pre-trained weights and adapt them to our scenario where the input is a single-channel temperature matrix and the output is the bounding boxes of all faces. We studied and compared different ways of adapting the network to our input, which are further discussed in this article. In all our experiments, the pre-trained were loaded and all the layers were trained.

In order to retrain the network, not only facial images are required, but also negatives. In the context of thermal imaging, these usually are high-temperature

Fig. 2. Example images taken (a) in a "cold" environment and (b) in a "hot" environment. Additionally, ground truth labels of the dataset are shown (facial landmarks, facial expression and glasses usage).

objects that could be confused with our target class. Most of the databases of infrared images have a clean background (cold) and are not suitable for good learning of negatives. Therefore, we collected and hand-labeled a total of 2075 additional images (some including faces and others not) in multiple scenarios, but ensuring that no appearing subject is included in the data reserved for testing. These images, together with the images from the database described in section 2, were used as training data in our experiments. The validation of all models reported in this section was performed using the Monte Carlo method for subject-wise cross-validation (leave-group-out validation, with a group size of 8) and only the best model was chosen to be tested on the test set.

For the pre-trained output to match the objective of face detection, the network should be adjusted so that each bounding box only predicts one class. Additionally, since facial bounding boxes have a certain aspect ratio, we run

the K-means algorithm to cluster the sizes of all faces in the dataset and generate anchor boxes that are better tuned for the use case, unlike the pre-trained version of YOLOv3 which is prepared to receive multiple classes of objects of varying size.

3.1 Model selection

Since the input of YOLOv3 is the three RGB channels of visible images, our single-channel thermal input needs to be adapted. We have experimented and compared different ways of performing this. One possible method is to apply a color palette to the input, so that the number of channels matches the input of the network. We chose for this purpose a palette where the hottest pixels are orange, yellow or white, similar to the facial skin color. The expectation is that it better mimics the colors of the pre-trained version of YOLO in RGB images. This model achieved an AP_{50} of 99.64% and an AP of 78.41% in the test set.

Our next experiment was to feed the network directly with the thermal image, without preprocessing it with a color palette. A disadvantage of the previous model is that color mapping the temperatures does not introduce any new information when compared to an input composed solely of the pixel temperatures. Therefore, we experimented tripling the single-channel input in order to match the number of input channels of the pre-trained network, without applying any palette or making any other change to it. We take advantage of the fact that all the images in our dataset contain temperature information, and we do not perform any kind of equalization to avoid losing that important data, considering that the facial temperatures have a limited expected range [3]. This experiment improved the accuracy of the model when compared to the initial attempt with YOLOv3 using a color palette, resulting in an AP_{50} of 99.89% and an AP of 79.21%.

To avoid the need of tripling the input, we experimented passing only one channel and making the necessary adjustments to the network. This channel corresponds to the temperature matrix captured by the thermal camera. In order to prepare the network for the new input, it is important to understand how the first convolutional layer of YOLOv3 works and how it can be adapted to accept the new input. The output of a convolution layer is visualized in Fig. 3. In YOLOv3, the first layer contains 32 filters, also known as kernels, of size 3x3, which means that each value in the output convolved feature is a linear combination of the pixel values in a 3x3 square around it. A kernel is therefore defined by 9 trainable weights and, considering 3 color channels, there are $32*3 = 96$ kernels.

Since the input shape suffered a reduction in the number of channels from 3 to 1, we discarded the weights corresponding to the kernels of the first convolutional layer and initialized them with random values. Comparing the results of this model to the triple thermal input version, we noticed a decrease in AP_{50} to 99.59%, and in AP to 76.28%, but also a decrease of 6ms in inference time, due to the smaller number of input channels.

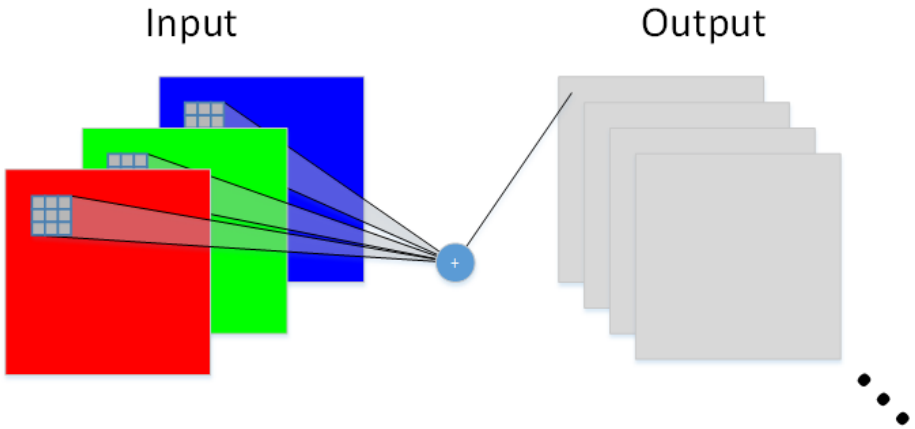

Fig. 3. Convolutional layer connected to RGB input.

To improve those values, we tried to take advantage of the old weights of the convolutional kernels of the first layer. Since there are 3 kernels per filter (one for each color channel), it is necessary to properly combine the weights of those kernels into one. Convolutional layers calculate the output values of each filter according to the formula

$$h_j^n = max(0, \sum_{k=1}^{K} h_k^{n-1} * w_{kj}^n), \tag{1}$$

where h is a feature map, n is the index of the convolutional layer in the model, j is the index of a filter, K is the total size of the kernel and w is a weight matrix. Note that the output of a convolutional layer for a multi-channel input is related to the sum of the convolution operation on each channel, and not to its mean. For this reason, we sum the weights of each kernel for each filter to adapt from a multi-channel to a single-channel input, in an attempt to feed similar data to the rest of the network, and therefore taking as much advantage as possible from the previously learned weights. This resulted in an AP_{50} of 99.75%, and AP of 78.26%.

3.2 Optimizing for speed

In the context of this paper, we are not very interested in detecting small objects, as we assume a minimum size of the faces of the vehicle occupants and distance to the camera (<60cm). Therefore, it is possible that parts of the network are not contributing to the overall accuracy, because we do not have small objects in our dataset. To test this hypothesis, we grabbed the weights of our single-channel predictor with adapter weights and pruned part of the network. The high-level architecture of YOLOv3 is represented in Fig. 4.

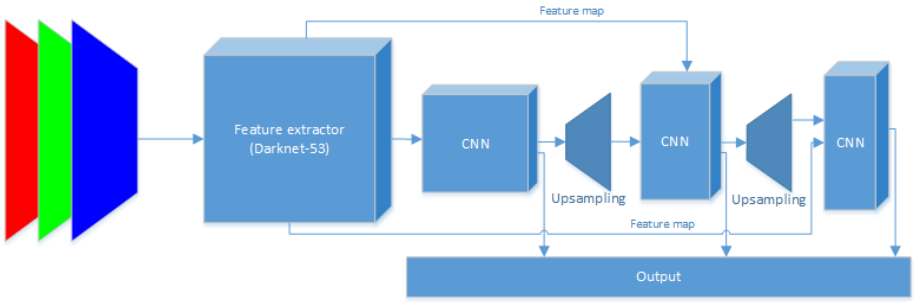

Fig. 4. High-level architecture of YOLOv3. Note that its output is generated from 3 different parts of the network.

The first output of YOLOv3 is given by the 82nd layer and the second output (medium-sized objects) is given by the 94th layer, so the rest of the network can be eliminated for our purposes. Inferences with this pruned model result in a similar accuracy, scoring minus 0.03% in AP_{50} and minus 0.02% in AP. Indeed, the last few layers of the network are not a high contribution to the prediction accuracy. The big advantage of making this conclusion is that we can predict in the pruned model, which means a considerable improvement in terms of speed with a low sacrifice in accuracy. In our implementation, the full model takes 35ms to predict one frame of resolution 416x416, while the pruned one only takes 25ms, which means we get a reduction of 29% in inference time. We also experimented ignoring the output of the second output layer, but the results deteriorated.

3.3 Training without last output

Since we have concluded that the last layer of YOLOv3 is not helpful in our use case, we were able to increase its speed at inference time, but it is also possible to totally prune it during training so that it no longer contributes to the total loss (and decreasing training times). The loss function used for training of the model is defined in equation 2.

$$\begin{aligned}
&\sum_{l=0}^{L}(\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2 \\
&+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2] \\
&+\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}-\hat{C}_i log(p_i(C_i)) \\
&+\lambda_{noobj}\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj}-\hat{C}_i log(p_i(C_i)) \\
&+\sum_{i=0}^{S^2}\mathbb{1}_{ij}^{obj}\sum_{c\in classes}-c_i log(p_i(c)))
\end{aligned} \tag{2}$$

As in previous versions of YOLO, the loss function takes into account the correctness of the center and dimensions of the predicted bounding boxes, the confidence given to objects, the confidence when there are no objects and the correctness of classification when there is an object. The changes with version 3 of YOLO are in the way the last three components of the loss function are calculated, using logistic regression instead of the previous squared difference, and the enclosing sum ($\sum_{l=0}^{L}$) that corresponds to the output of each layer of the feature pyramid network that follows the feature extractor (Fig. 4). This sum is responsible for adding the individual losses of each output at each scale. Therefore, if we decrease the number of output layers, L, from 3 to 2, we are effectively excluding the loss of the last output in the overall loss function. Our experiments show a reduction in training times of around 26%, but a decrease in AP_{50} of 2.3% and 4.1% in AP. For this reason, we decided not to exclude the last layer from the training process.

3.4 Comparison of results

Table 2 presents a comparison of results between the different transfer learning techniques experimented, each being briefly described in Table 1. In terms of inference time, models C and D are 6ms faster than the others, due to the smaller number of input channels. In accuracy, model C has the worst performance, which leads to the conclusion that adapting the weights to different images is better than random initialization. The results we obtained during cross-validation were similar, and we decided to choose model D due to the good compromise between speed and accuracy. Then, for testing, we used the model D that performed best during cross-validation, reaching an AP_{50} of 99.71% and an AP of 78.52%. A prediction example is provided in Fig. 5.

After manual observation of the output of the face detector, it is noticeable that the score on the AP metric is limited by the fact that the labeling process

Model	Description
A	YoloV3 with palette
B	YoloV3 grayscale with 3 channels
C	YoloV3 grayscale with 1 channel and random weights
D	YoloV3 grayscale with 1 channel and reused weights

Table 1. Model description

Model	AP25	AP50	AP75	AP	Inference time
A	99.71%	99.60%	97.16%	78.38%	31ms
B	99.89%	99.78%	97.34%	79.20%	31ms
C	99.78%	99.52%	95.37%	76.25%	25ms
D	99.85%	99.75%	97.33%	78.25%	25ms
best D	99.85%	99.71%	97.14%	78.52%	25ms

Table 2. Mean of test results obtained with different transfer learning techniques, predicting without the last output layer. The last row reports the test score of the model D that performed best during cross-validation.

was automatic and the face detector used for RGB images sometimes produces bounding boxes with slightly incorrect boundaries. For this reason, although the face detector reaches 97.14% in AP_{75}, it is harder for the algorithm to exactly match the ground truth and reach such values when the IoU (Intersection over Union) threshold is higher.

4 Conclusions

Our results show that it is possible to develop an accurate face detector in thermal images using transfer learning with neural networks developed for RGB images. Furthermore, one possible reason to why the detector does not score higher in the AP metric is the fact that the ground truth information is generated automatically and the limits of the bounding boxes are not perfectly defined. Overall, we argue that our work compares ways of transferring existing algorithms from RGB to thermal and demonstrates good results in a vehicle scenario.

Further work can be considered. Model A uses a color palette to map the temperatures to different colors and there is the possibility of experimenting different palettes and see how they impact the prediction accuracy. Additionally, at the moment, the face detection algorithm here presented for thermal images relies on pre-trained weights fitted to RGB images from ImageNet and adapted to work with thermal images. Instead, we could retrain the whole YOLOv3 neural network with the grayscale version of those images, so that the network is prepared from scratch to accept input in a single-channel format, therefore removing the necessity of readjusting the weights from a three-color system to single-color. Furthermore, experiments can be conducted to understand how the input resolution affects the accuracy of the predictor, and what is the expected

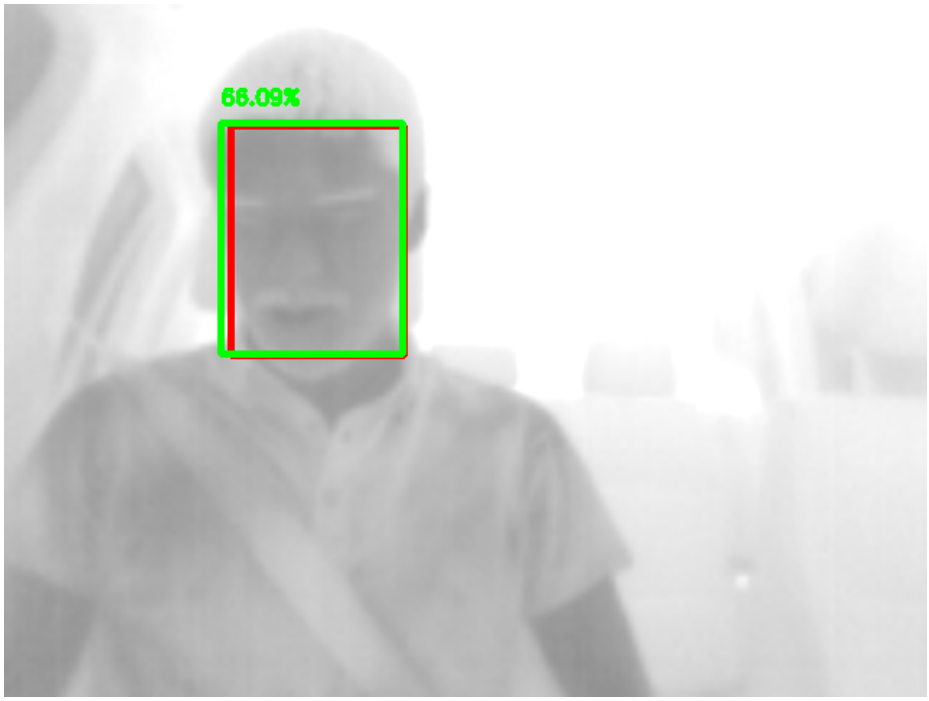

Fig. 5. Example of a successful face detection in a very hot vehicle interior. The red bounding box represents the ground truth and the green box refers to the prediction of our model, together with its confidence value.

trade-off between inference speed and quality of predictions. Moreover, in the context of this work we focused on a maximum distance of $\sim 60cm$, which means the algorithm is not prepared to handle small objects. Larger distances could be considered by using scale augmentation or adding to the database images of faces further from the camera.

References

1. Basbrain, A.M., Gan, J.Q., Clark, A.: Accuracy enhancement of the viola-jones algorithm for thermal face detection. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10363 LNAI**, 71–82 (2017). https://doi.org/10.1007/978-3-319-63315-2_7
2. Jia Deng, Wei Dong, Socher, R., Li-Jia Li, Kai Li, Li Fei-Fei: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009). https://doi.org/10.1109/CVPRW.2009.5206848
3. Korukçu, M., Kilic, M.: The usage of IR thermography for the temperature measurements inside an automobile cabin. International Communications in Heat and Mass Transfer **36**(8), 872–877 (2009). https://doi.org/10.1016/j.icheatmasstransfer.2009.04.010

4. Kwásniewska, A., Rumiński, J., Rad, P.: Deep features class activation map for thermal face detection and tracking. Proceedings - 2017 10th International Conference on Human System Interactions, HSI 2017 pp. 41–47 (2017). https://doi.org/10.1109/HSI.2017.8004993
5. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2014). https://doi.org/10.1007/978-3-319-10602-1_48
6. Nonaka, Y., Yoshida, D., Kitamura, S., Yokota, T., Hasegawa, M., Ootsu, K.: Monocular color-IR imaging system applicable for various light environments. In: 2018 IEEE International Conference on Consumer Electronics (ICCE). pp. 1–5. IEEE, Las Vegas, NV, USA (2018). https://doi.org/10.1109/ICCE.2018.8326238
7. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition **29**(1), 51–59 (1996). https://doi.org/10.1016/0031-3203(95)00067-4
8. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement (2018). https://doi.org/10.1109/CVPR.2017.690
9. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2015). https://doi.org/10.1109/CVPR.2016.308
10. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision **57**(2), 137–154 (2004). https://doi.org/10.1023/B:VISI.0000013087.49260.fb
11. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (2016). https://doi.org/10.1109/LSP.2016.2603342