In-Vehicle Occupancy Detection with Convolutional Networks on Thermal Images

Farzan Erlik Nowruzi, Wassim A. El Ahmar, Robert Laganiere
University of Ottawa
{fnowr010, welahmar}@uottawa.ca, laganier@eecs.uottawa.ca

Amir H. Ghods
SMATS Traffic Solutions
amir@smats.ca

Abstract

Counting people is a growing field of interest for researchers in recent years. In-vehicle passenger counting is an interesting problem in this domain that has several applications including High Occupancy Vehicle (HOV) lanes. In this paper, present a new in-vehicle thermal image dataset. We propose a tiny convolutional model to count on-board passengers and compare it to well known methods. We show that our model surpasses state-of-the-art methods in classification and has comparable performance in detection. Moreover, our model outperforms the state-of-the-art architectures in terms of speed, making it suitable for deployment on embedded platforms. We present the results of multiple deep learning models and thoroughly analyze them.

1. Introduction

High Occupancy Vehicle (HOV) traffic lanes are reserved for vehicles with a certain number of passengers (passenger number varies between states and countries). HOV lanes are used to encourage carpooling and the use of public transportation. Current methods used to enforce HOV lanes include police officers being physically present and visually monitoring the HOV lanes, and penalizing the offenders. Alternatively, some states encourage commuters to report HOV offending vehicles. It can hence be seen that counting the number of passengers in cars at all times is an important task in the process of enforcing HOV lanes.

Counting the number of humans present in a certain area has several applications, especially in urban environments. It can be used for congestion analysis at certain places (i.e. popularity of certain products in a supermarket, visitors of a sculpture in a museum, traffic through a certain entrance of a mall) [3]. Counting the number of passengers is an integral part of the enforcement of HOV lanes process. Automating the process of occupancy detection by installing road-side cameras on HOV lanes has not been very successfull. This is mainly due to significant changes in the lighting and visibility conditions. In addition, an external camera will not be able to reliably detect all passengers, especially those in the back seats. A possible solution is to use seat sensors that determine whether or not there is a passenger sitting on a seat by measuring the weight exerted on it. However, this approach is not feasible as there is no way to differentiate between an actual passenger and a heavy object placed on the seat. In addition, these sensors are usually installed for the front seats only.

Installing regular cameras inside a vehicle to detect the number of passengers at all times might be a feasible solution. However, this method raises privacy concerns as passengers will not be comfortable with cameras recording their activities at all times [16]. In addition, it is challenging for a visible camera to distinguish humans from the human-like dummies that have been often used to cheat the system. To address these problems, we propose the use of thermal sensors installed in vehicles to reliably detect the number of passengers using deep learning models without significantly compromising their privacy. Thermal data conceals most of the distinctive visual features, thereby ensuring the anonymity of passenger identities. Further, the proposed model is capable of running on edge-devices without the need to transmit any recordings to the server side. Our method could be used as part of a complete system where cars are registered in a database by their plate number. An embedded system would detect passengers using our model and update the number of passengers in real-time with the database accessible by authorities.

Deep learning models have dominated the field of computer vision and image processing since the introduction of AlexNet [12]. Deep models have been applied to many fields ever since including classification [12][28][29], recognition [19][25], detection [21][15], scene understanding [26], and geometric analysis [17]. In this work, we developed a neural network solution for passenger counting with thermal imaging, while taking into consideration its potential application on embedded and low-powered platforms. We also provide a comprehensive evaluation of most prominent deep learning methods on our dataset.

The rest of this paper is organized as follows. Section 2 provides a literature review of related work. In section 3 we give detailed information related to our dataset. In section 4 we explain details of our algorithms and deep neural networks tested. Section 5 presents our experimentation results and we conclude this paper in section 6.

2. Literature Review

[20] detects people from aerial thermal images using a particle filter based detector and traditional local features. Authors show that heat signature of human body is unique in these environments. However, problems arise when the heat signature is close to the background temperature.

[18] provides a comprehensive study on pedestrian detection using local features and conventional machine learning methods. They found that thermal imaging is a reliable source of information, and it provides similar performance in comparison to visual images. Both of these methods are relying on the traditional features and learning methods that are widely outperformed by novel deep learning models.

[5] uses infrared thermal images to classify objects. They use a random forest with a depth of 2 and branching factor of 2. At each node, a Convolutional Neural Network (CNN) is trained to classify between four classes. Classification accuracy is increased by fusing these methods. However, this comes at the cost of having multiple CNN models and increasing the computational complexity of the model.

[11] proposed a multi-spectral region proposal network based on Faster R-CNN [23] by fusing infrared and visual images. They show that adding BDT Classifier [32] improves the region proposal network performance even further. In contrast, we rely on training the model solely on thermal images to respect the privacy of the passengers.

[24] learns to synthesize an image into visual domain from the thermal images. It learns the mapping through a generative adversarial network [4] that includes local fiducial regions to provide more discriminative features in reconstructed images. Once the image is reconstructed, it is matched against a dataset of known visual images for recognition.

[6] proposes using models that are previously trained on visual images and adapt them to the infrared domain. To achieve this, models are fine-tuned using infrared images. It is shown that simple preprocessing infrared images boosts the overall performance significantly. Under this framework, inversion provides largest benefit among different preprocessing methods. Similarly, we have noticed the value of simple preprocessing techniques but with a different goal. Instead of trying to fit thermal data to exhibit a similar behavior as visual images, we extract a mask based on average human body heat signature and use it to guide the training of our model.

[13] uses deep convolutinoal models to detect roughly estimated regions to count individual objects in the image. Their method relies on three major stages. Feature extraction is performed using ResNet [7]. Neighborhood detection done by utilizing specialized loss function to encourage single blob detections. And finally, a line splits and watershed splitting methods are used to divide large blobs. In essence, this method is very relevant to our detection approach. In our approach, we use gaussian like density blobs that cover the bounding box region. Later, the generated blobs are post processed with a simple technique to generate bounding boxes. We are not using a specific loss for split learning. However, the gaussian blobs provide the required basics internally to encourage easily separable density map generation. Further, unlike [13] our goal is to detect bounding boxes. And finally, our system works on thermal images rather than camera images.

We summarize the main contributions of this paper as follows:

- We introduce a new, comprehensive, and annotated thermal image dataset of in-car environments.
- We design a custom convolutional neural model that is able to surpass the classification performance of larger models, while keeping the model small enough to be suitable for embedded applications. We adapt our model to perform each of classification and detection tasks.
- We retrain well known object detection methods on this dataset and provide a comprehensive evaluation on an embedded platform.

3. Dataset

Our dataset was captured with a FLIR One Pro thermal imaging camera [1]. The camera can detect temperatures between $-20°C$ and $400°C$ with an accuracy of $\pm3°C$. The raw recording is in $16-bit$ integer format. Dividing the raw thermal reading by 100 results in the *Kelvin* scale equivalent. The thermal image resolution is 640×480. Several locations for positioning the camera inside a vehicle were tested. The ideal location is found to be under the rear-view mirror (figure 1), so that it completely covers the area inside the vehicle with minimal obstruction especially to passengers in the backseat. An Android mobile application was developed using the FLIR One SDK to capture raw thermal data.

In total 1284 images were captured with number of passengers varying between 0 and 5. Different vehicle types (SUV, Sedan, Hatchback) and passenger seating positions were adopted to provide a comprehensive dataset.

[1] www.flir.com

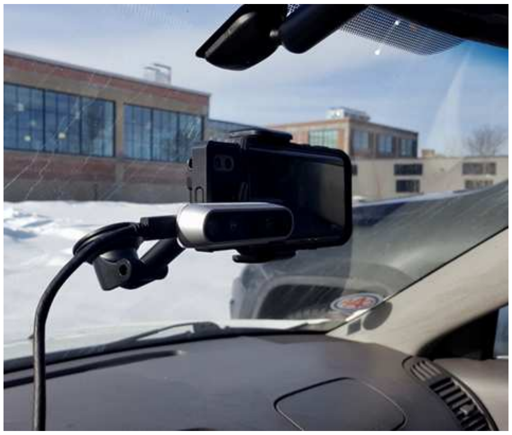

Figure 1. Thermal camera and its location for data capture.

#Passenger	0	1	2	3	4	5
#Images	65	314	402	276	253	3

Table 1. Data distribution. Number of images per passenger count is shown in this table.

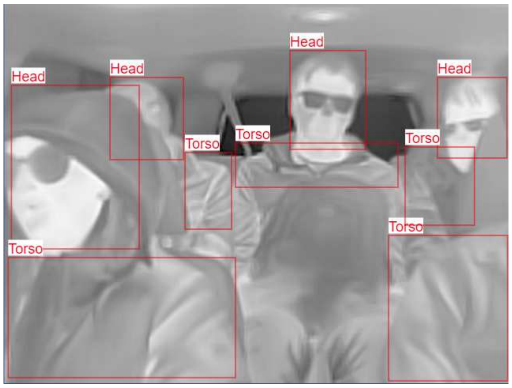

Figure 2. A sample of an annotated image.

Apart from changing cars and relocating passengers, air-conditioning in vehicles is used to introduce further environmental variations. Heating or cooling the vehicle interior causes a drastic shift in the distribution of the thermal values. The distribution of image classes is shown in Table 1.

Thermal images are manually labeled for visible portions of passenger head and torso. The torso region is usually hidden behind the front seats. These annotations are stored in a *JSON* file that is provided with the dataset. Figure 2 shows a sample annotated image from our dataset.

The clothing of the passengers is a major source of variation in thermal images. They cover the body heat, and sometimes could reflect higher intensities depending on their material. The head is mostly visible and is not covered; this provides a better opportunity for detecting passengers. Therefore, the labels for the head are used in our experiments. To count passengers positioned in the vehicle, we simply count the number of detected heads.

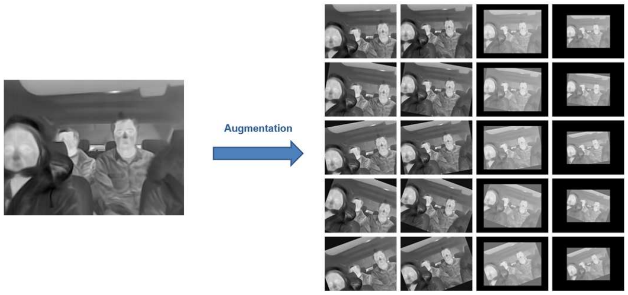

Figure 3. Original image from dataset and its augmentations.

The dataset and model definitions and source code are available to download through the following link: http://www.site.uottawa.ca/research/viva/projects/thermal-passenger-detection/.

4. Proposed Method

In this section, the data augmentation strategy and description of tested neural models are provided. We break down the proposed method in two settings and explain the parameters for each.

4.1. Augmentation and Preprocessing

Deep learning methods are extremely data hungry. In order to satisfy this requirement, we augment the dataset by applying a combination of rotation and scaling transformations on the raw data. Images that are scaled down in augmentation are padded with 0 valued pixels in order to preserve the image dimensions for training. We follow the same padding procedure used in the Tensorflow Object Detection API [9]. Sample results of this procedure is shown in figure 3.

Scale. Each thermal image is rescaled with ratios of $[0.8, 1.0, 1.2, 1.4]$. These ratios are chosen to achieve invariance against variations that might rise from changing the size of the vehicle.

Rotation. Passengers in a car tilt or rotate their heads. In order to achieve invariance against this, all thermal images are transformed with rotation angles of $[-20, -10, 0, 10, 20]$.

In total, this process generates 20 augmented samples from each thermal image. This way, the final data set size is increased from 1284 to 25680 images which was found to be sufficient for training and testing neural models.

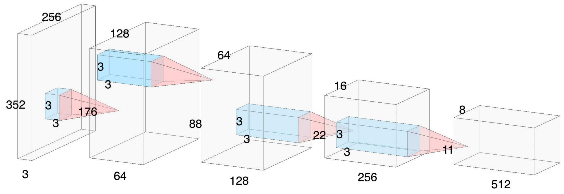

Figure 4. Proposed core model used as the backbone in all the learning tasks.

4.2. Core Model

We briefly introduce the neural networks and corresponding parameters that have been used for benchmarking on this dataset. Before processing images with these networks each image is resized to a smaller scale and ratios to match dimensionality of $90k$ pixels per image.

The core of the proposed method is a four layerd convolutional model that is inspired by [12]. All the core convolutions are performed with 3×3 kernels and strides of 2, except the third layer that has stride of 4, which drastically reduces the size of the feature map. This quick reduction in feature map size reduces the computational complexity of the model. First convolution consists of 64 kernels, and the number of kernels is doubled after every convolution. We propose two tasks on top of our core model. One is classification (*C4S-C*), and the other is multi-task learning for classification and bounding box detection with (*C4S-CD*). The core model is shown in Figure 4.

Our model is designed with consideration of low computational requirements. This resulted in a small model that is not only capable of fast inference, but also has a very short training session compared to other models. Each training session for our model takes on average around 15 minutes. This provides us with the opportunity to perform a comprehensive space search for the hyper-parameters. In order to limit the space search complexity and model size, we bound the number of layers between 2 to 6, and the strides are set to 1, 2, or 4.

4.3. Classification Task

We redefine the counting problem as a classification problem and use the number of passengers as the class labels for each image. An input image of size 352×256 is passed to our proposed four layered convolutional nework to extract the feature maps. At layer five, instead of flattening the 22×16 feature map and using a fully connected layer, we flatten the maps using a single branch factorized convolution [30] with one 1×1, two 3×3, and two consecutive groups of 1×3 row-wise and 3×1 column-wise convolutions. This further improves the performance compared to using a fully connected layer and reduces the model size by 20%. First convolution at this layer employs a 1×1 kernel with a stride of 1 followed by two 3×3 convolutions with a stride of 2. Finally, row and column-wise convolutions are applied. Drop-out [27] with a keep rate of 0.5 is applied on extracted feature maps from the core convolutions prior to passing them to the factorized convolution layer. The flattened image is passed to fully connected layers of size 256 and output layer of 6 probabilities representing the class labels. After calculating the features at each layer, Rectified Linear Units (ReLU) and batch normalization [10] are used. We have evaluated various loss functions including softmax cross-entropy, online hard example mining (OHEM) [31] with softmax cross-entropy, and focal loss functions [14]. We define the loss function by OHEM loss as it provides a few percentage points better classification accuracy than the regular softmax cross-entropy function. The concept of OHEM relies on taking the top k softmax cross-entropy loss from a batch of images. In other words, the hardest samples to classify are used when calculating the total loss for backpropagation. In our implementation, we set k to be half of the batch size. Momentum optimizer with the initial learning rate of 0.01, momentum value of 0.9 and decay factor of 0.1 after every 5000 steps is utilized as the optimizer for this model.

4.4. Multi-Task Learning

To perform bounding box detection, we use a similar method to [13]. Instead of using single pixel label, we use a gaussian masking on coarser sized image to perform bounding box detection. The output of the core convolutional layers at one side are fed into a classification model similar to *C4S-C*, and on the other side is passed to deconvolutional layers. We call this model *C4S-CD*. The deconvolutions upsample the image by strides of 2. The first one has a feature map of size 512 and the last one produces the predicted heatmap. To calculate the loss, we resize the input image to the shape of 44×32 and create a mask that represents the head of each detected passenger using an intensity gaussian distribution with σ set at one fifth of the ground truth bounding box size. $L2$ function is used to calculate the loss for the predicted and target heatmaps.

Generating bounding boxes from the heatmap requires further attention. Close objects have tendency to be merged with each other. Using gaussian masking instead of the binary masking in the learning process has addressed this problem to some extent. The boundaries of the heads in output heatmap contain smaller values compared to the center. We employ a simple thresholding mechanism to further split the connected regions of multiple targets. The thresholding value is set to 40 in our tests to create the binary masks. Once the regions are split, we use blob (binary large object) detection from the OpenCV library [2] to detect the connected regions in the binary heat map. Since our network also outputs the number of passengers n in the vehi-

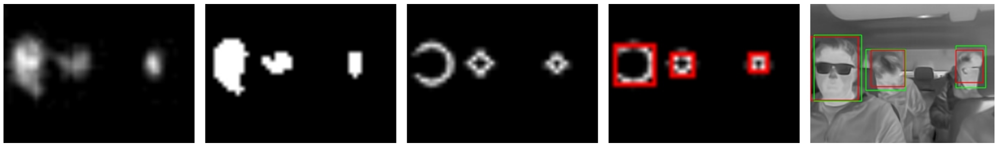

Figure 5. Stages of splitting the heatmap to connected regions and applying blob detection to get the enclosing bounding boxes.

cle, we sort the detected blobs by size and take the n biggest blobs as the predicted head locations. This way we eliminate noisy masked clouds from the data. Once we generate a list of accepted blobs, enclosing bounding boxes are generated for them. The boxes are then rescaled to the original image representing the locations of the detected heads. Figure 5 shows the various steps of generating bounding boxes from the output heat maps. Since our passenger counts are output by the classification branch of *C4S-CD*, we only report the precision/recall value for the heatmap based detections.

5. Experiments

We employ n-fold cross-validation to remove any bias towards dataset distribution from experiments. To perform these experiments, four folds have been employed. In each fold, the dataset is divided into two mutualy exclusive sets of train and test. For training, we have used 90% of the raw data and then augmented them with the methods proposed in section 4.1.

The remaining 10% of images are used as test images. We do not augment the test dataset. There are two reasons behind this decision. The first is that testing augmentations of the same image do not contribute much in reflecting the accuracy of detection models. An image and its augmentations most likely have the same detection results. Augmentation enriches the train set and makes it more robust, but it does not add variety to the testing set. The second reason is that we aim to get results on images that would reflect real situations.

In classification, we compare our model to Mobilenet. In the detection task, we use Single Shot Multi-box Detector (SSD) [15] once on top of Mobilenet and once with Inception V2. We use the tensorflow object detection api [9] to compare our model performance with pre-existing methods in object detection using SSD. [9] shows that Inception V2 [30] achieves the best performance in terms of mean average precision (mAP) compared to other deep CNN methods such as [7] when used as a backbone for SSD [15]. They also show that Mobilenet [8] does not lag too much behind Inception V2 in terms of mAP while having a lower processing time. This is also confirmed in our tests. We have opted to test with the SSD [15] framework for detection as it is a fast detection model. SSD does the classification and localization in one pass, making it an efficient solution.

SSD-Inception. Single Shot Multi-box Detector (SSD) proposed by Liu *et al.* [15] is another network that uses a single convolutional neural network for object detection. It is composed of the VGG classification network truncated before any classification layers and replaced by 6 convolutional layers. The 6 final layers and respective anchor box scales gradually decrease in size, allowing for the detection of objects of different scales. Szegedy *et al.* [29] introduced the inception model for classification. Instead of using one kernel of fixed size for a convolution, the inception model applies 3 filters of sizes 1×1, 3×3, and 5×5, in addition to a max-pooling operation. Later, this has been updated by factorization to provide faster computation [30]. Results of these operations are concatenated to form the final output of an inception layer. Since the inception model learns more parameters, we have decided to test its performance as the base layer for SSD.

SSD-MobileNet. Howard *et al.* [8] introduced Mobilenets, that are ideal for mobile and embedded systems applications. Instead of standard convolutions, Mobilenets use a combination of depthwise convolutions followed by 1×1 pointwise convolutions for optimization. Depthwise convolutions apply a single filter to each input channel. Pointwise convolutions are then used to combine the outputs of the depthwise convolution. This architecture separates the filtering and combining operations and drastically reduces the model size and computational requirements, while not significantly compromising the model accuracy. Mobilenet is composed of 19 layers (depthwise then pointwise convolutions) and a fully connected layer fed into a softmax layer for classification. While inception networks as base for SSD generate more robust feature maps, this comes at a heavy size and computational cost. Since our goal is to have a model that is capable of running on limited power and memory devices, a small model size and computational efficiency are vital. We tested the use of Mobilenet truncated before the classification layers as a base model for SSD.

To compare these models against each other, accuracy, speed, and precision-recall measures are used.

5.1. Counting Accuracy

In the task of counting the number of passengers in a car, we define accuracy as the number of images for which the head count is correctly computed given the test dataset. At this point, the measure is agnostic of the passenger locations, and only the final count is valuable. Table 5.1 shows the best accuracies.

C4S-C provides significantly better classification based counting accuracy than Mobilenet in both cases. OHEM loss introduces more weight on hard examples hence could generalize better on the test set. For HOV lanes, it is of utmost importance to detect if there are two or more passengers in the vehicle. To address this, we introduce binary accuracy. Figure 6 presents the class confusion matrices for various approaches. Through confusion matrix analysis, it is concluded that the majority of misclassifications occure between adjacent numbered classes. Therefore the binary accuracy for all the models is usually larger than their exact counting accuracy.

5.2. Precision-Recall

SSD-MobileNet and SSD-Inception are designed to localize the objects in an image. We perform the object detection test in order to compare the effectiveness of the models in identifying the location of objects in the image. We can't solely rely on precision-recall or accuracy in order to identify the better performing method. We could have a method with high accuracy, but with a lower precision due to the fact that all the false positives happened in few images. Or inversely, we could have a higher precision or recall but lower accuracy, due to the spread of false positives or false negatives between multiple images.

Figure 7 shows precision vs recall curve for the methods with confidence thresholds ranging from $[0.05, 0.9]$ and intersection over union (IOU) threshold set to 0.5. Mobilenet, which performs relatively poorly in classification, performs better in object detection when used as base of SSD. There are better classification models such as InceptionResNet V2 [28] that perform poorly with SSD. However, they provide good results with other detectors such as Faster RCNN [23]. This is due to the fact that they are tuned to function effectively with their respective detectors. Similarly, Mobilenet performs slightly worse and much slower with Faster RCNN compared to SSD. Our model is suffering from this issue. Specially, the rapid reduction of the feature map dimensions results in such features that limits the capabilities of SSD.

To alleviate this problem we use the C4S-CD. The deconvolution and blob detection modules add a negligible overhead to the system. However, while it produces better results than SSD, it still lags behind other methods.

5.3. Speed.

Finally, to decide on which method is more appropriate for embedded platforms, we compare the execution performance. All of the methods are implemented in *Tensorflow* [1] and are benchmarked on a *Nvidia Jetson TX-2* platform. The Jetson is one of the fastest and most power-efficient computing device developed for AI embedded systems applications. Although it does not compare to the computing power of regular GPUs, it has a decent performance which makes it a better fit for embedded systems. Since this solution would be used in vehicles, the final goal is to embed the model on an edge device with much lower available power than Jetson. However, Jetson would provide a better perspective over the performance of each of the models.

The proposed C4S-C outperforms all the others in this case with a whopping 63 frames per second processing speed on the Jetson GPU. Tables 3 and 4 show the speed comparison of classification and detection methods respectively.

The fast training and testing speed of our proposed network allowed us to perform a comprehnsive network parameter tuning in order to choose the best fitting settings for the task in hand. This is an obvious advantage of small and fast models against large ones.

6. Conclusion

In this paper, we have introduced a new thermal image dataset for counting number of passengers in vehicles. We introduce a data augmentation model to increase the amount of data and build models that are robust against variations such as rotation and scale. We propose two models based on one core architecture for classification and detection tasks. The classification model outperforms the state of the art with a comfortable margin with almost half the computational complexity. Further, we have compared the results of various object detection models, and proposed a new method based on blob detection to detect passengers. Our proposed models are designed with the main consideration to run on a low powered edge device. This results in having a very small and fast model that is comparable in performance to the state-of-the-art. One aspect that we have not explored in this paper is techniques to prune and quantize our models. This will further optimize our model to comfortably run on a low powered device.

7. Acknowledgments

This research was conducted in collaboration with SMATS Traffic Solutions[2]. We thank our colleagues at SMATS for the insights and resources they provided for this research.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow:

[2] www.smatstraffic.com

Classification		
Method	Counting Accuracy	Binary Accuracy
Mobilenet	84.54%	96.67%
C4S-C (SMCE)	89.70%	97.24%
C4S-C (OHEM)	**91.03**%	98.0%

Detection		
Method	Counting Accuracy	Binary Accuracy
SSD-MobileNet	92.56%	98.67%
SSD-Inception	**95.04**%	99%
SSD-C4S-C	84.73%	97%

Table 2. Comparison of counting accuracy. Counting based on classification and detection results are presented at the top and bottom of this figure, respectively.

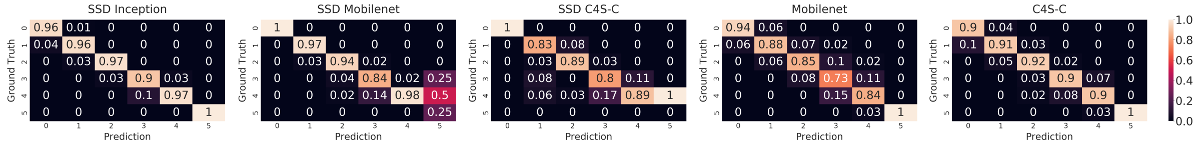

Figure 6. Confusion matrices for tested models.

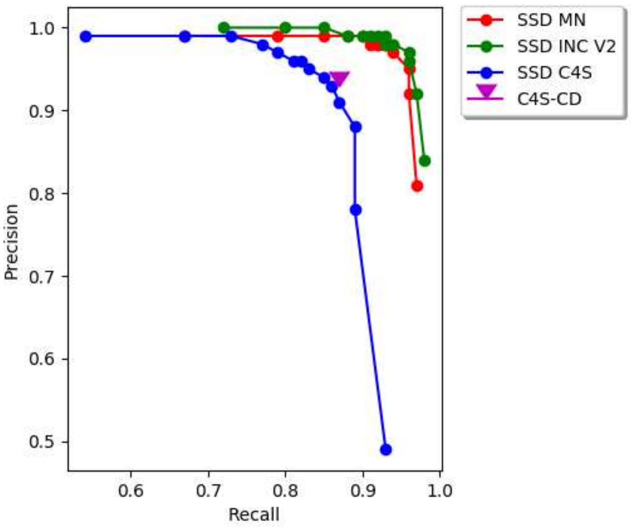

Figure 7. Precision/Recall comparison for IOU of 0.5.

A system for large-scale machine learning. In *12th Symposium on Operating Systems Design and Implementation)*, pages 265–283, 2016.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] H. Ding, J. Han, A. X Liu, W. Xi, J. Zhao, P. Yang, and Z. Jiang. Counting human objects using backscattered radio frequency signals. *IEEE Transactions on Mobile Computing*, pages 1–1, 2018.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] E. Gundogdu, A. Ko, and A. A. Alatan. Object classification in infrared images using deep representations. *IEEE International Conference on Image Processing (ICIP)*, pages 1066–1070, 2016.

[6] C. Harrmann, M. Ruf, and J. Beyerer. Cnn-based thermal infrared person detection by domain adaptation. *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything*, 2018.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[9] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[11] D. Knig, M. Adam, C. Jarvers, G. Layher, H. Neumann, and M. Teutsch. Fully convolutional region proposal networks for multispectral person detection. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 243–250, 2017.

	Mobilenet		**C4S-C**	
Jetson TX2	ms	FPS	ms	FPS
GPU	27	37	15.9	63
CPU	600	1.7	369	2.7

Table 3. Speed comparison for classification models

	SSD Mobilenet V2		**SSD Inception V2**		**C4S-CD**	
Jetson TX2	ms	FPS	ms	FPS	ms	FPS
GPU	77	13	95	10.5	19.5	51.3
CPU	500	2	1210	0.8	373	2.7

Table 4. Speed comparison for detection models

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pages 1097–1105, 2012.

[13] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 547–562, 2018.

[14] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015.

[16] D. Luo, J. Lu, and G. Guo. An indirect occupancy detection and occupant counting system using motion sensors. *WCX 17: SAE World Congress Experience*, 2017.

[17] F. E. Nowruzi, N. Japkowicz, and R. Laganiere. Homography estimation from image pairs with hierarchical convolutional networks. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 904–911. IEEE, 2017.

[18] D. Olmeda, C. Premebida, U. Nunes, J. M. Armingol, and A. Escalera. People detection and tracking from aerial thermal views. *Integrated Computer-Aided Engineering*, page 347360, 2013.

[19] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[20] J. Portmann, S. Lynen, M. Chli, and R. Siegwart. People detection and tracking from aerial thermal views. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1794–1800, 2014.

[21] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NIPS) (NIPS)*, pages 243–250, 2015.

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NIPS) (NIPS)*, pages 243–250, 2015.

[24] B. S. Riggan, N. J. Short, and S. Hu. Thermal to visible synthesis of face images using multiple regions. pages 30–38, 2018.

[25] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[26] J. Shao, K. Kang, C. Change Loy, and X. Wang. Deeply learned attributes for crowded scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4657–4666, 2015.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, pages 1929–1958, 2014.

[28] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[31] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[32] L. Zhang, L. Lin, X. Liang, and K. He. Is faster r-cnn doing well for pedestrian detection? *European Conference on Computer Vision*, pages 443–457, 2016.