Experiments in Object Detection on Thermal Imagery in Nvidia Jetson Nano

Shikhar Mishra
u6203537@anu.edu.au

Varad Sandip Kareker
u6935175@anu.edu.au

Sri Sai Ram Poosarla
u7077213@anu.edu.au

Abstract

We implemented an object detector for thermal imagery on the Nvidia Jetson Nano. Object detection models on RGB images perform well when there is good amount of light in the images but perform worse when fail when it is dark. To overcome this as well as other isses, we trained an object detection model on thermal images which is applicable for more scenarios. Our goal in this project was to obtain a real time performing object detection model on thermal images running on an edge device like the Nvidia Jetson Nano. We used two variants of You-Only-Look-Once single shot detector (YOLOv3-spp and YOLOv3-tiny). After fine-training the models on thermal images and we performed detection on Jetson Nano and observed that YOLOv3-tiny model can be used for production level applications on the Jetson.

1. Introduction

Edge IOT device market has seen tremendous growth in recent years[23]. Cloud computing and the internet of things (IoT) have elevated the importance of edge devices, resulting in the need for more intelligent, fast computing and advanced services at the network edge[23]. Nvidia is one of the larger vendors which its various ubiquitious platforms for deep learning and AI, such as its line of GPUs, libraries like CUDA[15] and the Jetson series[19] of GPU enabled embedded computers.

Thermal imaging allows to capture an estimate of the scene in absence of visible light. An object detector that can work on thermal images is useful for various applications such as military, night driving, search and rescue, industrial temperature monitoring etc and even for extraterrestrial applications. However the lack of the colour and texture information, white-black/hot-cold polarity changes and halos that appear around very hot or cold objects make thermal imagery challenging to work with.

This paper focusses on experimenting with single-shot object detectors for processing thermal images and video streams on the Nvidia Jetson Nano. We trained the tiny variant of Yolo-v3[20] for detecting objects in thermal images, using the FLIR Thermal dataset. We achieve production use of upto 10 frames per second on the Jetson Nano.

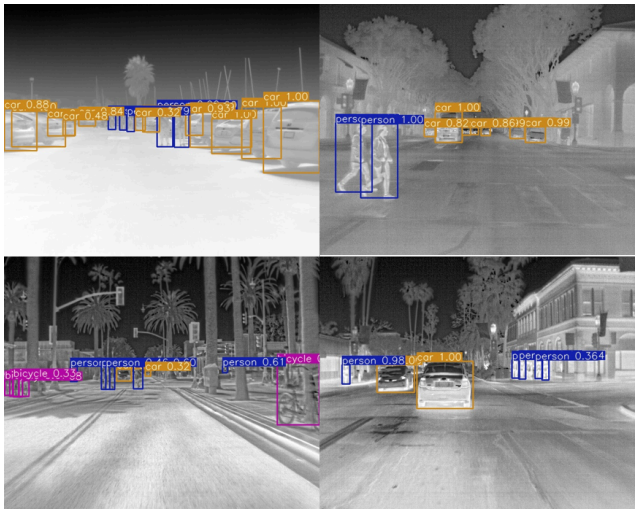

Figure 1. Example of results obtained from YOLOv3-spp trained for object detection on thermal images running on the Jetson Nano

2. Literature Review

2.1. Thermal Imagery

Infrared thermography (IRT) or thermal imaging is a technique of capturing radiations of long-infrared range (roughly 9,00014,000 nanometers or 914 μm) and producing images of that radiation, called thermograms. The amount of radiation emitted by an object is proportional to the temperature of that object, therefore thermographs allows one to see variations in temperature of that object. Thermal imaging captures an estimate of a scene in absence of visible light, making it useful for night or low light applications. Capturing thermographs requires a specialised thermal sensing camera such as the Lepton series of thermal cameras from FLIR company[7].

Thermal images can be expressed using different palette structures, to name a few, iron palette, black/white/gray palette or rainbow palette etc. Some of the thermal cameras can simultaneously capture overlapping normal RGB

and thermal streams[1].

The FLIR Lepton 3.5[7] which we wanted to use was not available to be shipped in a practical timeframe, and was an expensive piece of equipment. Hence we chose to use the next best solution that is directly using well-labelled thermal images and videos captured on the camera. FLIR company provides these in the form of FLIR Thermal Imaging Dataset[5], which uses a black-white-gray palette and it can be used for object/person detection.

Figure 2. Thermal cameras in action. The FLIR Thermal dataset that we use has a black/white/gray palette

2.2. Object Detection

Object detection has been a subject of interest in the field of computer vision and robotics. It involves object localization and classification of detected objects, and is a well-established field of Computer Vision, of which person/facial detection is one important subset. There are many methods for object detection. Before the advent of deep learning based object detectors, there were hand crafted feature detectors such as Haar cascades by Viola and Jones[27] or gradient based methods like Histogram of Oriented Gradients (HoG[6]).

Deep learning based object detectors can be roughly categorised into two classes, viz. multi-stage or single shot detectors. Both classes have their advantages and disadvantages. For example, multi-stage detectors usually employ some kind of region proposal, the quintessential example being R-CNN [8]. These type of detectors while being more accurate than single shot, are processing heavy because region proposal happens over many stages and require lots of memory, hence are not able to used on edge devices with low power constraints very effectively.

Single shot detectors on the other hand are much faster because they can detect multiple objects in one pass. Released in April 2020, YOLOv4 [4] has become the state-of-the-art algorithm for single-shot object detection. It can be implemented in constrained edge devices for real-time performance in its many variants, such as YOLO-tiny. We utlised the previous version YOLOv3[20] as that has had over three years of support and readily available backbone weights for custom training. There exist other fast object detectors such as SSD [17].

2.3. Jetson Nano

Nvidia Jetson Nano is a low-power portable computer for edge deep learning and computer vision applications developed by Nvidia. It has an embedded Tegra processor with ARM architecture for central processing. With a low power mode of 5W, the Jetson series is designed to be run on batteries as well as mains. It has 4GB of shared memory, so it can run small to mid-tier neural networks and is ideal for DIY deployment purposes due to its compact form factor, as well as low price of USD$99 (there is another model with 2GB memory, but we did not use that version).

Due to the COVID-19 pandemic, many innovative applications have been proposed using the Jetson including using it as a pandemic drone to monitor the health of people [16] and as a sneeze detection or combining it with UVC light to sanitise exposed surfaces [25] using deep learning assisted thermal cameras. Thermal imaging applications include surveillance and in pandemics [26].

This study runs all of the models on Jetson Nano for live and in situ applications.

2.4. Relevant Research

A method for detecting pedestrians in thermal images was proposed back in 2010[13]. Thermal images are useful under different weather and lighting conditions, however, the lack of the texture and colour information makes it harder to generate significant features and descriptors for human and object detection[28]. Previously, a Shape Context Descriptor (SCD)[2] was used with the Adaboost[22] cascade classifier framework and multi-layer boosting algorithm. Their method of rectangle feature cascading on the filtered images allowed them to sum up pixel intensities of the images in a rectangular box. However, the high computation of SC descriptor hinders this algorithm to work for real-time applications [28].

More recently, Gong et al.[9] proposed a real-time method to detect electrical equipment in thermal images. They used a regression-based detection framework using YOLO to predict the various characteristics of the equipment, such as its class type, orientation angle and coordinate location. Further they also integrate the orientation consistency prior to the model, by which it achieved 93.7 percent mean average precision running at 20 fps [9].

Warm camera pictures captured using a drone have been used to distinguish and track objects in the ocean[16]. This item recognition calculation utilised depends on static chan-

nel boundaries and thresholds. Their classifier depended on the object area, the average object temperature, and its general shape. However, there are various situations where this order would be testing, e.g., when movement obscure is available or when the object is moving over a picture with differing sensor force, which can be brought about by a lopsided scene brilliance or sensor commotion. Along these lines, a profound learning calculation could be a more successful instrument for the item characterization, since it can deal with minor departure from the pictures influenced by natural changes, however long these impacts are broadly present in the dataset. Enhanced Object Detection[24] uses analytical fusion to combine the long wave infrared and colour cameras and Gaussian Mixture Model and the non-parametric kernel-based density estimator in the model.

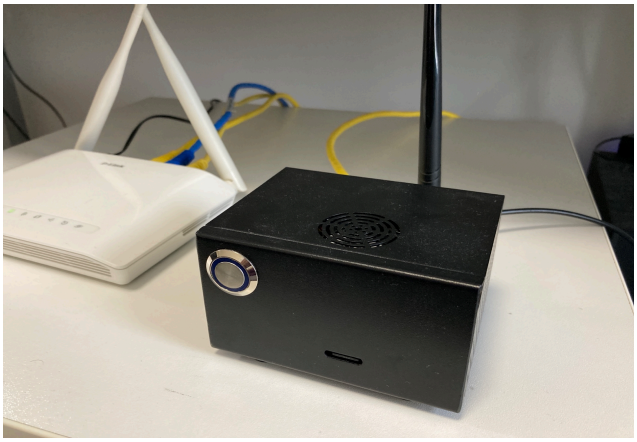

Figure 3. Jetson Nano

2.5. Measuring Accuracy: Precision Recall and IoU

Three of the most widely used metrics for measuring accuracy of an object detector are Precision, recall, and Intersection Over Union (or IoU). Precision measures how much of the detected bounding box contains the actual object, whereas Recall measures how much of the object is within the detected bounding box. These two must be used in conjunction with each other because it is possible to have high precision and low recall, and vice versa. Hence there exists the IoU metric which is a combination of both precision and recall, and is high when both the values are high. We are using Generalised IoU during our training of the yolov3-tiny model, which is just a function of IoU. For further details please refer to the GIoU paper [21].

3. Methodology and Our Approach

Initially, our plan was to create a Jetson based thermometer using a thermal camera for real-time inputs and processing. Such a 'thermometer' could then be used for low cost, contactless handheld fever detection 'gun' for screening COVID-19 patients. Gradually we realised the infeasibility of this goal, and changed our approach accordingly.

3.1. FLIR Thermal Dataset

FLIR Thermal Dataset provides 14K thermal and raw RGB images with ground truth labeling[5]. The dataset is designed for training algorithms for thermal sensors in driverless cars, and has a large number of people and cars in it. Some images with ground truth annotation can be seen in figure 2. These images are captured by a Lepton 3.5 Micro Thermal Camera Module, which has a dynamic range of -10 to 140 degree Celsius and spectral range of 8 to 14 micrometers. The images used in the experiment are of size 512x640 pixels.

Figure 4. Sample images from the FLIR Thermal Imaging Dataset in the black/white/gray palette

3.2. Jetson Challenges

While there were many challenges during the project, such as debugging various python and CUDA errors, most of the challenges arose due to using the Jetson platform, which sounds great in theory but is hard to work with in practical usage. While Nvidia has been developing this platform for the previous 4-5 years, there are still early adoption issues, such as lack of documentation and functionalities. Secondly, the Jetson is unstable out of the box due to the shared memory management between the CPU and Tegra GPU. It utilises a unique form of compressed RAM called ZRAM[14], and in short, it needs several configuration changes for stability (fig 5). Even then we had random crashes several times a week, which set us back on experimentation. Finally, the last hurdle was the ARM architecture of the Jetson. ARM is a type of RISC architecture commonly found in all sorts of mobile and IoT devices, however it is has insufficient support for desktop level productivity required for most deep learning tasks.

jetsonConfigChanges.txt

```
#!/bin/bash

#1st arg - file location where you swap file will be.
#i.e put it on the external drive.

#usage instructions
if [ $# -lt 4 ]
then
        echo "Usage: $0 fully-qualified-swapfile"
        exit
fi

#sudo prompt.
if [ $EUID != 0 ]; then
     sudo "$0" "$@"
     exit $?
fi

#stopping this service to fix the zram compression issue
systemctl disable nvzramconfig.service

#creating a larger swapfile on the external drive connected to the jetson.
dd if=/dev/zero of=$1 bs=1024 count=16777216
chmod 600 $1
mkswap $1
swapon $1
echo $1 swap swap defaults 0 0>> /etc/fstab
```

Figure 5. Configuration changes made to fix zram compression issue on Jetson Nano

3.3. Our models - yolov3spp and yolov3tiny

We chose to use the YOLOv3-SPP (Spatial Pyramid Pooling[10]) and YOLOv3-tiny[18] 'flavours' of YOLOv3 for running object detection models on the Jetson. For the SPP variant we directly used pretrained weights available for thermal images[11]. Further, we trained YOLOv3-tiny on the FLIR Thermal Dataset using the MLCV server, starting with the default weights for tiny on normal RGB images[12].

All YOLOv3 model variants utilise residual skip connections and upsampling, however their main feature is the detection at 3 different scales, due to which it can detect smaller objects quite well, which is good for us as would like to detect background objects as well. The unique feature of YOLOv3-SPP model are the SPP blocks which extract more relevant features from their inputs, which helps in increasing the class and category-specific information. For further details about the model architectures, please refer to their original papers[20, 18, 10].

4. Results and Discussion

We selected 141 continuous frames of images from FLIR thermal data set for running the model detection on Jetson. We tested using different input image sizes (416, 320 and 256 pixels width). The same set of tests were also run on desktop level GPU (NVIDI RTX 2080 Ti, 11 GB) for comparison of the performance. We summarise the total number of detections made by the Jetson for the most important object classes (Person, Car, Bicycle) (see fig 10). As there were problems in implementing an absolute ground truth comparison, we compare all the results relative to the 'largest' model run on the server (YOLOv3-spp-416).

4.1. YOLOv3-spp

As the SPP is largest and most accurate model when compared to tiny, it detected more number of objects in the images. We can see from (fig 10), as the image size is increased the model is able to detect more number of objects from the images. This result can also be visualised in (fig 6).

Figure 6. Output for different image sizes for YOLOv3-spp

4.2. False Positive/Negative Discussion

The image on the left is the output for input image width of 416 pixels and the one in right is for 256 pixels. The left image has higher number of detections, as we can see people are detected in this where on the right the people are not detected at all. We understand that comparison using total number of detections is not ideal, as we cannot account for any false positives or negatives. It is likely that the yolov3-spp has a higher false positive rate than tiny because it is detecting more object in the frame, however, we are unable to visually compare all the annotated results. The tiny model is observed to have more missed detections, hence has a higher false negative rate than spp. This may or may not be important to the real-world application of these models. For example, in some situations, it may be imperative to detect all the true occurrences, however in others it may not be necessary.

4.3. Speed: Frames Per Second (fps)

SPP gave FPS values of 3.2, 2.4 and 1.7 respectively for the image sizes as seen in (fig 10). We can say that the FPS is inversely proportional to the input image size, which is a descent performance given jetson has a small GPU. The FPS obtained when it was run on desktop level GPU are 65.9, 64.4 and 57.6. A graph for comparison is shown in fig 7.

4.4. YOLOv3-tiny

The YOLOv3-tiny is a smaller model as compared to YOLOv3-spp. The tiny architecture is designed such that it can be run in real-time on small scale GPU's such as Jetson Nano. We trained the model for 80 epochs using the Adam optimiser on the FLIR thermal image data set. As the training happens the GIoU, Objectness and Classification scores have dropped which indicates that the training is happening smoothly (fig 8).

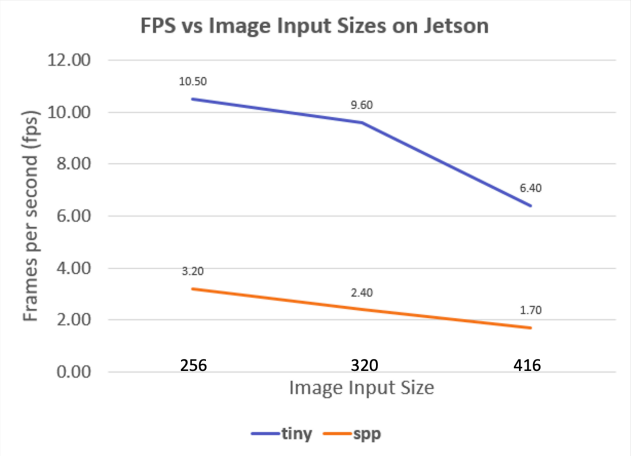

Figure 7. FPS Graph

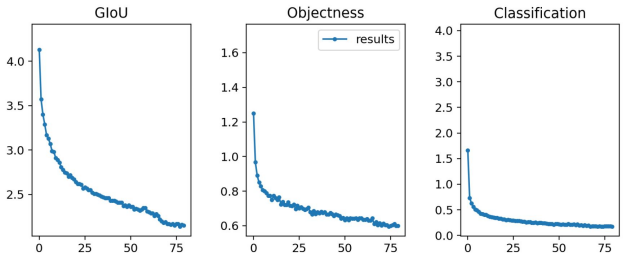

Figure 8. YOLOv3-tiny training for 80 epochs on the GPU server.

We next used the weights of the custom trained YOLOv3-tiny model. We checked the detections done by this model for different input image sizes. The number of detection by the tiny are quite less when compared to YOLOv3-spp as shown in (fig 10). But the FPS we obtain with yolov3-tiny is close to real-time which is 10.5, 9.6 and 6.4 respectively to the input image size.

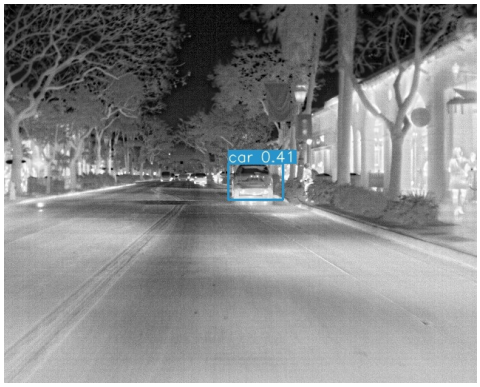

Figure 9. YOLOv3-tiny output for input image size 256.

If we compare the number of objects detected for 256 and 416 image size there is very less count difference for cars and bicycles, so if our goal is to find large objects in the image the tiny is giving near close to real time performance. We can also see for comparison in (fig 9) and (fig 6) yolov3-tiny does a decent job detecting large objects. So, if detection of the larger/closer to the camera objects is the main use-case, then using yolov3-tiny is appropriate.

	Model					
	YOLOv3-tiny (custom train)			YOLOv3-spp		
	Image Size			Image Size		
	256	320	416	256	320	416
No. of persons	12	13	23	360	473	671
No. of cars	33	49	41	542	565	588
No. of Bicycles	14	24	21	139	139	146
FSP recorded on Jetson	10.5	9.6	6.4	3.2	2.4	1.7
FPS recorded on Desktop level GPU	170.6	145	126.9	65.9	64.4	57.6

Figure 10. We observed during testing that the image input size is the most important factor that determines speed as well as accuracy of the model. Results obtained are shown. Note the general tradeoff between size of model and speed.

4.5. Demo

The results we have got on running for the 141 images on the Jetson nano are converted into video format. These videos can be seen in `http://shorturl.at/cszL6`. The FPS of all these videos is same for demonstrating the variation in detection of objects by both the models for different image sizes. Note however that in practice, the results for the larger spp models were processed much more slowly than the smaller v3-tiny models.

5. Conclusion and Future Work

The aim of our project was to get practical insight into python and Pytorch for deep learning, and well as learning about the Nvidia Jetson platform which has the potential to be widespread in the IoT-driven future ”smart city”. We were able to leverage our knowledge of object detection for application in a thermal imaging dataset. We ran a pretrained as well as our custom trained model on the Jetson to achieve upto 10 fps, which is really good given the constraints of the Jetson.

No brilliant theory can match an ingenious engineering application. Application of work in this area has a multitude of implications for fields ranging from autonomous vehicles to medical screening. While we had many ideas for the direction of the project, we weren’t able to take all of them. For example, we had thought about comparing the ground truth labels with the detected bounding box results to obtain precise IoU, precision and recall accuracy estimates. However, that task slowly became very data science intensive which was not our main injection. We would have also wanted to test the models on the full FLIR dataset, but the we found that as we increased the input image size,

the Nano overheated and randomly crashed in between processing, so we used only a subset that guaranteed us results. Finally, if we actually had one thermal camera we could have explored its practical usage.

Time permitting, we could have trained other variants of single shot detectors, such as YOLOv3-tiny-prn[3]. Further, we mostly experimented with changing the image input sizes, but more experimentation is warranted in the other hyperparameters of the network. Finally,

Yet another idea for future work in this area would be investigate simulation of RTSP/IP camera streams, and statistically quantify the number and size of streams that the Jetson can handle. We can also evaluate on image/video characteristics, such as noise or contrast. Further, deep-learning models are usually optimised for the target hardware they are deployed on. For the Jetson lineup, this involves the TensorRT and tkDNN packages, as well as low level programming experience, which we were not able to do.

It is also worth mentioning that there are more powerful versions of the Jetson, such the Jetson AGX Xavier, which has 32 GB of memory and an Nvidia Volta class GPU. It would be interesting to observe how its performance compares to that of its little brother.

References

[1] Thermal cameras explained, howpublished= https://www.grainger.com/know-how/equipment-information/kh-thermal-imaging-applications-uses-features-345-qt.

[2] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. *Advances in neural information processing systems*, 13:831–837, 2000.

[3] Alexey Bochkovskiy. Yolov4 by alexeyab `https://github.com/AlexeyAB/darknet#pre-trained-models`.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[5] FLIR Corporation. Flir thermal dataset`https://www.flir.com/oem/adas/adas-dataset-form/`.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[7] FLIR. Flir lepton thermal camera series `https://lepton.flir.com/`. *Lepton Thermal Cameras*.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014.

[9] X. Gong, Q. Yao, M. Wang, and Y. Lin. A deep learning approach for oriented electrical equipment detection in thermal images. *IEEE Access*, 6:41590–41597, 2018.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[11] Joe Hoeller. joehoeller/object-detection-on-thermal-images `https://github.com/joehoeller/Object-Detection-on-Thermal-Images`. *Joe Hoeller GitHub*.

[12] Glenn Jocher. Yolov3 model weights `https://drive.google.com/drive/folders/1LezFG5g3BCW6iYaV89B2i64cqEUZD7e0`.

[13] Vijay John, Seiichi Mita, Zheng Liu, and Bin Qi. Pedestrian detection in thermal images using adaptive fuzzy c-means clustering and convolutional neural networks. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 246–249. IEEE, 2015.

[14] Kagaalow. Jetson hacks add swap `https://www.jetsonhacks.com/2019/11/28/jetson-nano-even-more-swap/`.

[15] David Kirk et al. Nvidia cuda software and gpu parallel computing architecture. In *ISMM*, volume 7, pages 103–104, 2007.

[16] F. S. Leira, T. A. Johansen, and T. I. Fossen. Automatic detection, classification and tracking of objects in the ocean surface from uavs using a thermal camera. In *2015 IEEE Aerospace Conference*, pages 1–10, 2015.

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[18] Qi-Chao Mao, Hong-Mei Sun, Yan-Bo Liu, and Rui-Sheng Jia. Mini-yolov3: real-time object detector for embedded applications. *IEEE Access*, 7:133529–133538, 2019.

[19] NVIDIA. Autonomous machines `https://developer.nvidia.com/embedded-computing`. *NVIDIA Developer*, Sep 2020.

[20] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[21] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.

[22] Margaret Rouse. Adaboost, Oct 2020.

[23] Margaret Rouse. What is an edge device? `https://searchnetworking.techtarget.com/definition/edge-device`. *SearchNetworking*, Oct 2020.

[24] Louis St-Laurent, Xavier Maldague, and Donald Prévost. Combination of colour and thermal sensors for enhanced object detection. In *2007 10th International Conference on Information Fusion*, pages 1–8. IEEE, 2007.

[25] Viral Thakar, Himani Saini, Walid Ahmed, Mohammad M Soltani, Ahmed Aly, and Jia Yuan Yu. Efficient single-shot multibox detector for construction site monitoring. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–6. IEEE, 2018.

[26] Rita Tse, Tianchen Wang, Marcus Im, and Giovanni Pau. Privacy aware crowd-counting using thermal cameras. In

Twelfth International Conference on Digital Image Processing (ICDIP 2020), volume 11519, page 1151916. International Society for Optics and Photonics, 2020.

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[28] Weihong Wang, Jian Zhang, and Chunhua Shen. Improved human detection and classification in thermal images. In *2010 IEEE International Conference on Image Processing*, pages 2313–2316. IEEE, 2010.