REAL-TIME HUMAN DETECTION ANALYTICS IN CONSTRAINED IMAGE INPUTS

Heshan Fernando

(188026x)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

January 2022

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of my knowledge and beliefs, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Moreover, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic, or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Signature	Date
Mr. Heshan Fernando	……..…	…….

The above candidate has carried out research for the Master's dissertation under our supervision.

Name	Signature	Date
Prof. Indika Perera	……..…	…….
Dr. Chathura De Silva	……..…	…….

ABSTRACT

Real-time video surveillance is a growing trend today. Our surrounding is being monitored daily by an increasing number of surveillance camera systems. Analyzing human movement can be used for the wellbeing of humans. There are a set of analytical tools and algorithms which can be used to detect, track, and analyze humans in images. Human movement analytics has various subdomains including human detection, human recognition, human tracking, human localization, human reidentification, human behavior analysis, and abnormal activity detection. Human detection is the most crucial step among them, and which helps to derive other sub domains.

Human detection analytics in constrained lighting conditions would be a challenging task to apply due to the low contrast of the image context. Currently available systems focused on the daytime. The background light is an essential factor in the camera images, which rigorously affects the quality of the image. We can identify considerable differences if we compare two images at the rich light condition and constrained light condition. Fewer features of the objects can be extracted in constrained light conditions than rich light conditions. Illumination of the background context is an important factor if we focus on such applications. Currently, most researchers have used human detection analytics in visible light. RGB image shows a clear view when there is sufficient light existing, and it is highly sensitive to visible light conditions compared to infrared. In this research, we considered infrared images as constrained image inputs.

Our proposed methodology contains a novel human detection approach based on machine learning and a motion dynamic model. Here we have addressed the problem using a combination of Deep Convolutional Neural Networks (DCNN) for human detection and Kernelized Correlation Filters (KCF) for human tracking. MobileNet pre-trained model is used for frame-wise human detection as the first step. Then the KCF object tracking algorithm is used to increase the human detection accuracy while tracking the human in the context. Furthermore, we applied some preprocessing techniques to reduce the noise effects. Currently, the progress made by this research-based project is sufficient to initiate the development of a complete human detection analysis solution based on live CCTV camera footage. This solution provides the core functionality of human detection analytics and it can be easily adapted to different domain solutions such as customer behavior analytics in a supermarket or worker movement analytics in an industrial premise.

Keywords: Human Detection, Human Tracking, Deep Neural Networks, MobileNet, Kernelized Correlation Filters, Infrared, Realtime Video Feed, Histogram Equalization

ACKNOWLEDGMENT

The research project “Real-time Human Detection Analytics in Constrained Image Inputs” has been a novel experiment done as the Master of Science (Research) project of Eng. Heshan Fernando from the Department of Computer Science and Engineering. First and foremost, my sincere gratitude goes to the Department of Computer Science Engineering for facilitating this opportunity through the curriculum for the MSc. Engineering degree program. Special gratitude to the University of Moratuwa, Sri Lanka for funding this research project under SRC long-term grant SRC/LT/2016/11.

My sincere gratitude is extended to Prof. Indika Perera and Dr. Chathura de Silva, Senior Lecturers, Department of Computer Science and Engineering for supervising the project. The progress of the project made would not have been a possibility unless for their intense supervision and support. The valuable feedback provided to us by Dr. Charith Chitraranjan and Dr. Anjula de Silva during the initial project proposal presentation and progress evaluations were of utmost usefulness for the development of project outcomes. My sincere gratitude is extended to them.

The opportunity provided to us by Dr. Chandana Gamage and Dr. Sulochana Sooriyaarachchi to present the research prototype at Techno 2019 Exhibition Sri Lanka gave us valuable feedback to further polish our research. My sincere gratitude is extended to them. My special thanks go to various support given by all the staff members including academic and non-academic of the University of Moratuwa. My gratitude is also extended to students of the Department of Computer Science and Engineering who took part in the IRANALYITICA dataset and the numerous supports provided to me through the research period.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
DCNN	Deep Convolution Neural Network
IP	Internet Protocol
KCF	Kernelized Correlation Filters
CCD	Charge Coupled Device
HOG	Histograms of Oriented Gradients
SVM	Support Vector Machine
FLIR	Forward Looking Infrared
YOLO	You Only Look Once
ABMS	Adaptive Boolean Map based Saliency
BMS	Boolean Map based Saliency
SCA	Stel Component Analysis
CWBTFs	Cumulative Weighted Brightness Transfer Functions
IR	Infrared
SAD	Sum of Absolute Differences
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Feature
LBP	Local Binary Pattern
RoI	Region of Interest
MIL	Multiple Instance Learning
IoU	Intersection over Union
FLIR	Forward Looking Infrared

1 INTRODUCTION

This thesis “Real-time Human Detection Analytics in Constrained Image Inputs” is a research-based project conducted for the human analytics of computer vision systems. Human movement analytics has a wide range of subdomains and applications including human detection, human recognition, human tracking, human localization, human re-identification, human behavior analysis, abnormal activity detection, etc. There are lots of applications of human movement analytics like sports analytics, traffic analytics, business intelligence, surveillance systems, health analysis, etc. Human detection is a popular subdomain that identifies the whole body, upper body, lower body, and body parts objects of the human from the image. There are so many applications such as pedestrian detection, surveillance, human behavior detection, fraud detection, etc. Human tracking is in line with human detection and which helps human analytics nowadays. Currently, people are using visible light camera systems and RGB images for human movement analytics. In our research, we considered human detection and proposed a novel methodology based on both machine learning and motion tracking method.

With CCTV cameras being deployed almost everywhere, surveillance based on computer vision has become a crucial application nowadays. Most of the surveillance systems are working on visible light imaging, but performance-based on visible light imaging is limited due to the variation in light intensity during the daytime. The background light is essential in the camera images, which rigorously affects the quality and information content of the image. If we compare the two images at rich light and constrained light conditions, there is a considerable difference. Images captured under constrained lighting conditions, usually expose fewer features compared to those captured under rich light conditions. Therefore, illumination of the background context is a crucial factor if we focus on such applications. The matter of concern lies in the need for processing images in low light, such as in the need for nighttime surveillance.

The cameras are taking RGB (Red, Green, Blue) images in the daytime, which will not provide clear images in the nighttime. Background light condition depends on various environmental factors and it varies with time. Under low light conditions, most

cameras respond poorly to color-contrast resulting in a narrow histogram. Due to the low contrast in the image context, it is more challenging when we are working with constrained light conditions.

In our research, we propose a novel human detection system based on infrared images. First, we applied pre-processing techniques on raw images to minimize the noise effects. Human detection will be applied as the second step. Our novel model consists of machine learning and object tracking methods for better human detection in images. Taking the camera feeds, processing them in real-time, and analyzing them would not be easy when considering limited computational power. For this purpose, a methodology is proposed hereby to analyze human detection in real-time using a fixed camera installed in the monitored environment. The monitored environment shall be an open system thus allowing humans to walk in and out of the environment.

1.1 Research Question

The lighting condition is a very important factor in the image quality and it severely affects the accuracy and recall of human detection. RGB images are highly sensitive to lighting conditions. Due to this, our research considered infrared images that are less sensitive against illumination changes of background light [37]. Since the currently commercially available surveillance cameras have both RGB and infrared image feeds, we would be able to integrate them with the proposed methodology.

1.2 Problem Statement

Developing a real-time image processing and machine learning technique for human detection with high accuracy in constrained image inputs.

1.3 Background and Motivation

Human detection analysis has diversified applications based on sports analytics, business intelligence, surveillance, health analytics, traffic engineering, etc. Most human analytics researches are based on visible light conditions. Current research has low precision and low recall in constrained lighting conditions. In the daytime, they considered RGB camera images for the human detection analysis which fails in the nighttime. There is a great opportunity to work on human detection analysis in

constrained light conditions to introduce robust solutions while enhancing precision and recall.

1.4 Research Objectives

1.4.1 Implement a robust human detection analytic system in both daytime and nighttime

Currently, most of the research has been focused on daytime human detection analytics. They relied on visible light and which varied with time. Due to the lack of visible light in the nighttime, they did not address the real problem robustly. This research is focusing on a robust solution that can be applied both daytime and nighttime.

1.4.2 Develop computer vision techniques to enhance current human detection analytic methods

The effective usage of computer vision techniques is essential since this research considers an image processing-based solution. Here we considered suitable preprocessing methods to enhance human detection. As a result, we enhanced the performance of the total solution.

1.4.3 Implement machine learning approaches to enhance human detection analytics

The research problem can be addressed using both machine learning and mathematical models related to image processing. Here we focused on machine learning-based solutions while taking advantage of image processing methods. Machine learning techniques are more robust to noises than image processing methods.

1.5 Project Deliverables and Outcomes

1.5.1 Novel real-time human detection model

- Effective preprocessing on the raw images to compensate for noise effects.
    - Addressed currently available related works and chose the most suitable image processing techniques to remove the noise effects.
- Deep convolutional neural network-based human detection.

    - Proposed enhanced MobileNet [1] DCNN using domain-specific dataset.
- Adaptive human tracking model to decrease miss-detections of the DCNN model.
- Real-time working prototype with a concurrent program.

1.5.2 IRANALYTICA Infrared Dataset

In this research, we collected our dataset which contained 28,453 images. Images were captured using a 2 Megapixel IP (Internet Protocol) camera at 30 frames per second rate. 30 persons (19 males and 11 females) have contributed to this dataset. All the images were captured at the former software architecture lab, Department of Computer Science and Engineering, University of Moratuwa.

1.5.3 Research paper "Real-time Human Detection and Tracking in Infrared Video Feed" [2]

Fernando, H., Perera, I., & de Silva, C. (2019, July). Real-time human detection and tracking in the infrared video feed. In 2019 Moratuwa Engineering Research Conference (MERCon), 2019, pp. 111-116.
Paper published link: https://ieeexplore.ieee.org/document/8818862

2 LITERATURE REVIEW

Analyzing human detection using computer vision has become a very attractive and effective topic in business intelligence applications. By using analyzed data, relevant authorities will be able to make decisions to develop businesses. Almost every business activity involving humans, i.e. employees or customers, is now linked with a feedback system that would lead to business decision-making through analytics. The technology for generating analytics on the behavior of virtual visitors (web visitors, email readers) is well matured today. Such technology is capable of monitoring user behavior in fine acuity including minor activities of users such as screen scrolling, click events, and viewing articles. However, the extent of maturity on the analytics of physical visitors is very less. In most cases, the analytics are generated from recorded business activities undertaken by visitors such as purchases or payments.

2.1 Existing Solutions

The proposed system will be divided into human detection analytics in both daytime and nighttime. Currently, commercially available high-end cameras may provide this feature as an inbuilt feature. However, in our solution, we can convert any economical surveillance camera into a high-end product. We can categorize cameras into 3 types based on the captured image. Such as RGB camera, Infrared camera, and Kinect camera. In RGB cameras it will provide 3 channel images that were taken with color features in the context. It might provide unclear images in low-light conditions. In infrared images, they considered infrared light to illuminate the context and capture the released infrared waves relative to environment temperature. Infrared images are less sensitive than RGB images against illumination variations. The Kinect camera provides depth images that have 3-dimensional features of the context. Here are the example images of RGB, infrared, and depth.

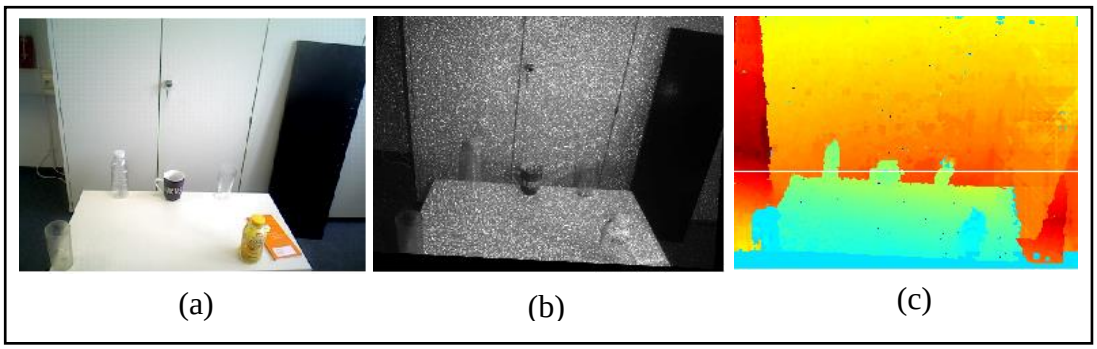


Figure 1: Types of images (a) RGB Image (b) Infrared Image (c) Depth Image
Source: http://www.cs.cornell.edu/~hema/rgbd-workshop-2014/papers/Alhwarin.pdf

2.2 Related Works

This research involves knowledge from literature related to “preprocessing”, “human detection”, “human recognition”, “human localization”, “human tracking”, “human re-identification”, and “human pose estimation”. The human movement analysis process is very challenging due to different factors. It depends on the features of humans like shape, color, pose, clothes, and hair. In addition to that, the type of the camera, the angle of the camera, distance to the human, background context also affects the final result. When identifying related work on each section, we could find a considerable number of researches already carried out on human detection and human tracking. However, the number of researches on constraint light conditions was fairly less compared to rich light conditions.

2.2.1 Preprocessing

Preprocessing is the backbone of computer vision. It targets to reduce the noise effects and complexity of raw images and smoothen the main computer vision processes. Applying noise filters, intensity thresholding, histogram equalization, histogram normalization, cropping, resizing, gray scaling, and morphological operations are the famous techniques that have been used in previous research. The research done by Junfeng Ge et al. [3] have used a Gaussian filter to reduce the noise effect in CCD and vibrations of the camera. They have applied a 5 x 5 median filter twice to decrease the variation of the pedestrian area and preserve the structural properties in the image. They proposed a hysteresis thresholding method that helped to segment the pedestrians from the background using high and low threshold levels.

Furthermore, they have used morphological operations to eliminate the regions which are smaller than the mask and candidate filtering method based on the height, width, aspect ratio, and bottom position of the region. These preprocessing steps caused more reliable and accurate detection in their proposed methodology. Another interesting research done by I. Riaz et al. [4] have used their dataset containing nighttime video scenarios with infrared images. They applied cropping, flipping, and scaling operations for the preparation of the dataset.

2.2.2 Human detection

Human detection approaches can be listed as single image processing and image sequence processing. Most of them have used low-level features like shapelet features, local binary patterns, Haar-like features, HOG features, etc. There are different kinds of applications done using human detection algorithms. For example, the automotive industry, video games, science simulations, sports activities, biometrical identification systems, and collision detection systems in intelligent cars. Furthermore, it provides us with useful information, such as human position identification, gesture recognition, motion direction, and 3D modeling of the human body. Human detection would be a challenging research topic due to the different appearances and various poses of humans. It would be more complex when the background is cluttered. There is a need for robust feature extraction for better human detection. Here we have listed some of the approaches found in the literature.

The edge-oriented histogram is commonly used in previous related works. Dalal and Triggs [5] introduced the Histograms of Oriented Gradients (HOG) which showed greater performance than other edge detection methods. HOG uses the details of the intensity of the gradient for the featured nine directions between 0 to 180 degrees. It will analyze the deviation of the pixels in a certain cell in each gradient direction. They proposed the HOG features with the SVM classifier on their dataset named INRIA RGB image dataset. The HOG approach is most suitable for detecting frontal views of persons at eye height. HOG detector invariant to photometric transformations. However, the HOG detector is not very suitable for detecting side views or inclined views of people. For instance, HOG has a high recall, but the low precision of this output is a matter of concern. Therefore, there should be extra filtering

or feature extracting techniques used to enhance the performance of the HOG. Furthermore, it would be hard to determine a suitable threshold parameter to achieve good precision and recall. Another drawback of HOG is that its results have no consistent gap between detector bounds and actual person boundary. It is not viable to estimate human body coordinates based on detector boundaries. However, the HOG approach is sufficiently fast for real-time solutions. Suard et al. [6] have used the HOG descriptor with Support Vector Machine (SVM) and proposed a method for pedestrian detection in infrared images. They focused on the nighttime driver assistance system. Their solution has achieved a 90% detection rate for pedestrians for their dataset. Another research done by Budzan et al. [7] introduced modified HOG features with low-level preprocessing on infrared images in their research. Their proposing solution contained a combination of pixel-gradient and body parts processing. There are three major steps such as head modeling, human modeling, and classification. They were able to reduce the execution time and false detections of the HOG descriptor, which obtained over 95% precision. Gajjar et al. [8] have mentioned a novel method for human detection in thermal images based on HOG features and SVM. They used HOG features and k-mean clustering for human tracking. Using the k - means algorithm, they obtained movement patterns of the humans in the frame and predicted human tracking. Person re-identification is another research in human movement analytics applications nowadays. Person detection and tracking will help for better person re-identification. Based on the positively detected windows, the path followed by a person in the image has been determined. Their proposed methodology achieved 83.11% of precision and 41.27% of recall on the OSU color and thermal pedestrian dataset hosted by the Ohio State University. Khandhediya et al. [9] have done another interesting research using the HOG descriptor and adaptive background subtraction for the dynamic camera. They focused on FLIR cameras. As the principle involves sensing based on thermal radiation in the near IR region, it is possible to detect humans from an image captured using a FLIR camera even in low light conditions. As proposed in their technique, a fused image was created using the raw image and the background-subtracted image. The performance of HOG on the fused image improves significantly compared to the original image with HOG features. The proposed method is applicable when there is motion existing in the camera rather than traditional background

subtraction. Adaptive background subtraction is calibrated according to the motion effect and the pedestrians are detected using their movement. Their proposed solution has achieved 76.09% of human detection precision for moving camera frames.

Viola-Jones object detection framework [10] is a generic approach for object detection, which can also be used for detecting body parts of humans. It is widely used for face detection as it provides a fast, reliable detector for frontal views of human faces. This accuracy is due to the sharp features in the human face such as nose, eyes, mouth, etc. Considerably accurate upper body detection models are also developed using this approach. However, the detector fails in handling multiple angles of persons. Furthermore, existing haar-classifiers are not suited for surveillance camera-based human detections due to the angle of the camera. Additionally, similar to the HOG detector, the Viola-Jones object detector is also subjected to variable gaps between detector boundaries and object boundaries. Dai et al. [11] and Zhu et al. [12] proposed an interesting human detection approach using an ensemble detector by replacing the Haar features using the HOG features. Due to this novel approach, they were able to keep the discriminative power of HOG features [5] as well as the detection speed advantage of the Viola-Jones object detection framework [10]. There are two types of ensemble detectors as substructure detector and detector ensemble. The substructure detector consists of a set of part detectors and constraints where each part detector focuses on specific part classification. The substructure detector can be taken as positive when all its part-detectors are positive and the constraints are satisfied. While detector ensemble is composed of a set of substructure-detectors. If at least one substructure is detected positively, it gives a positive detection. Otherwise, the detector ensemble takes it as a negative response. The detector ensemble has shown better performance against the occlusion, illumination, and rotation of the objects than the substructure detector.

The CENTRIST feature (Census Transform Histogram) was proposed by Riaz et al. [4] for human detection. It is based on histogram orientation and they used SVM as the classier and tested on infrared images. CENTRIST performed better than HOG in human detection while decreasing the training and testing time effectively. They have extracted HOG and CENTRIST features and used them to train two separate

linear SVMs for the evaluation of the proposed method. Based on that, a comparative analysis of both methodologies is carried out to evaluate them. Single frame classification-based pedestrian detection in the driving assistance system was proposed by Shashua et al. [13] and it performed well. There are 9 overlapping sub-regions generated with their positions. Each sub-region has a local image descriptor that will compute the local shifts of the image structure. It depends on the change of the pose and limbs of pedestrians. As the final step, they have combined the calculated discriminative values per sub-region using Adaboost. Instead of complex human detection approaches, an intensity thresholding-based method has been proposed by Ge et al. [3]. They used the hysteresis threshold for the segmentation and applied it to human detection. Furthermore, they have tried the Median filter and Gaussian filter for noise filtering purposes in the preprocessing step. The proposed method depends on the intensity threshold and they are subjective along with the background. It resulted in higher false positives with complex backgrounds.

Convolution Neural Network (CNN) is the most sophisticated way to solve object detection, natural language processing, and pattern recognition problems nowadays. It is a multilayer feed-forward neural network and contains neurons, learnable weights, and biases. The neurons receive several inputs taken from the vector and respond to an output vector by going through weights and activation functions. There are lots of applications that have been done using CNN and they show impressive performance especially in object detection. Here are some of the CNN models used in the object detection domain.

2.2.2.1 AlexNet

AlexNet CNN was designed by Alex Krizhevsky and team [28] in 2012 and was able to win the ImageNet LSVRC (Large Scale Visual Recognition Challenge). There are 8 deep layers including 5 convolution layers and 3 fully connected layers. It has more than 650,000 neurons and 60 million parameters. AlexNet was used for image classification of 1000 different categories including object categories (pencil, keyboard, mouse, etc) and animals (cats, dogs, etc). There is a ReLU activation function after every convolutional and fully connected layer. It took six days to train ALexNet on two GTX 580 GPUs using 1.2 million images.

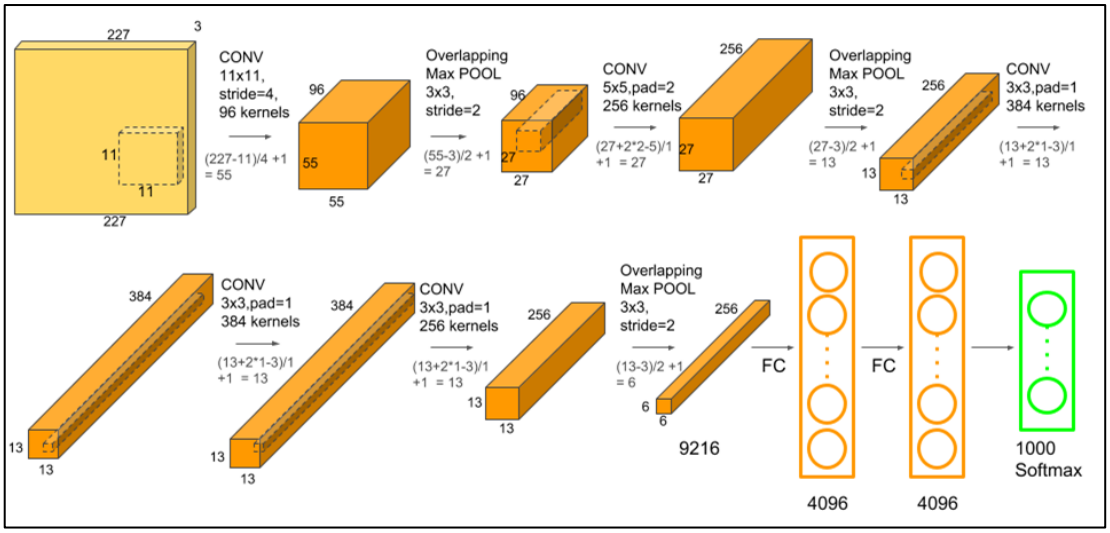


Figure 14: Neural network architecture of AlexNet

Source: https://neurohive.io/en/popular-networks/alexnet-imagenet-classification-with-deep-convolutional-neural-networks/

2.2.2.2 GoogLeNet

GoogleNet [29] was the winner of ImageNet LSVRC 2014. It was introduced by a team of Google based on LetNet CNN and there was a novel element called inception module. The inception layer consists of 3 layers (1×1, 3×3, and 5×5 convolutional layers). The proposed architecture has 22 deep layers and achieved 6.67% of the top-5 error rate.

2.2.2.3 VGGNet

VGGNet CNN has become the runner-up in ImageNet LSVRC 2014. It was proposed by Karen Simonyan and Andrew Zisserman [30] in their research paper. Their main contribution to the community was the evaluation of network architectures compared to the depth with small(3x3) convolution filters. They have considered 11,13,16 and 19 layer network architectures. Their best performance came with a 16-layers network which has a total of 138 million parameters.

2.2.2.4 ResNet

ResNet (Residual Neural Network) is the winner of ImageNet LSVRC 2015 proposed by Kaiming He and the team [31]. They tested the proposed architecture with

152 layers on the ImageNet dataset which is 8x deeper and less complex than the popular VGGNet CNN. ResNet achieved a 3.57% top-5 error rate on ImageNet.

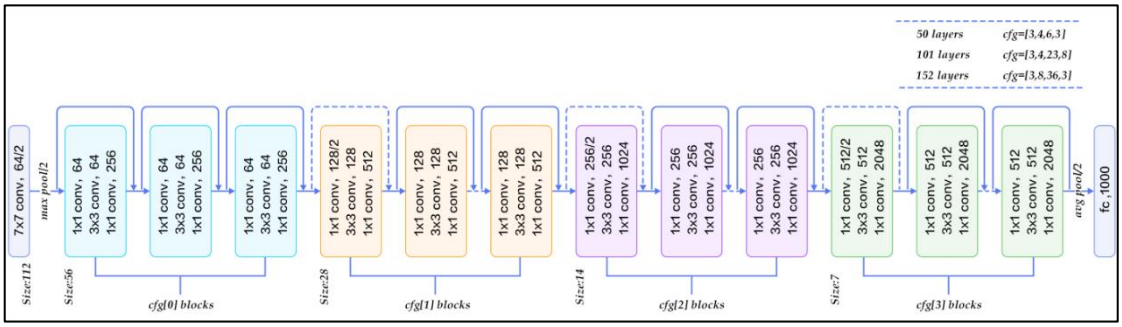


Figure 15: Neural network architecture of ResNet

Source: https://www.researchgate.net/publication/320568840_KinNet_Fine-to-Coarse_ Deep_Metric_Learning_ for_ Kinship_Verification /figures?lo=1

2.2.2.5 YOLO

YOLO (You Only Look Once) [32] has become a very famous CNN nowadays that is capable of applying real-time object detection applications. The CNN identifies the RoIs and their class probabilities from the given image in one operation instead of looking at the complete image frame. Since the whole detection pipeline is a single network, the complete process helps to optimize directly on detection performance. YOLO is faster than most of the existing object detection methods and it showed 45 frames per second real-time processing time on a Titan X GPU without batch processing. YOLO might fail to detect small objects (like a flock of birds) within the image due to the spatial constraints of the algorithm.

2.2.2.6 R-CNN

R-CNN (Region-based Convolutional Neural Network) is proposed by Ross Gishick and the team [33]. They have used a selective search algorithm to segment 2000 bounding regions from the given image and applied CNN on top of each of these regions. CNN works as a feature extractor and provides a feature vector. Then SVM was applied to all the feature vectors to predict the object using their confidence score. Since RCNN architecture considers a large number of region proposals (2000) and all of them are applied through the CNN model, it would be timely complexed, and it took

higher training time and testing time. Therefore, RCNN is not applicable for real-time applications.

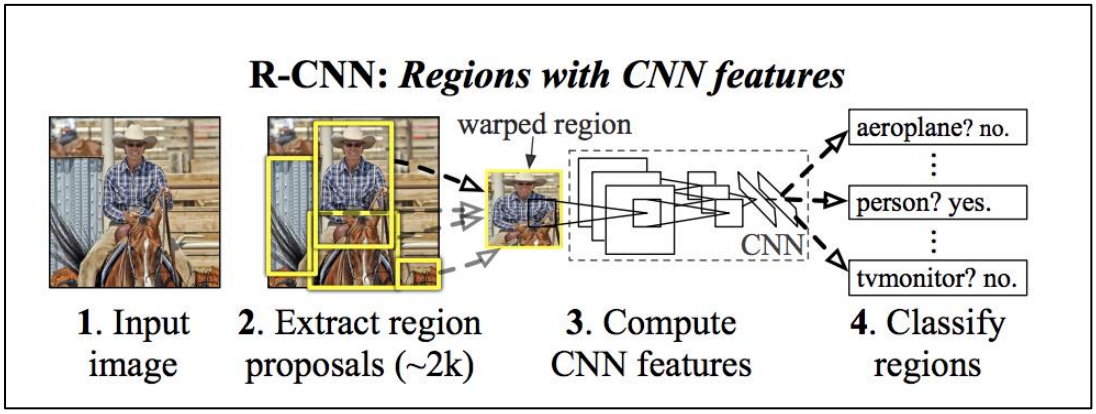


Figure 16: The architecture of RCNN

Source: https://www.kdnuggets.com/2016/09/9-key-deep-learning-papers-explained.html/3

2.2.2.7 Fast R-CNN

Ross Gishick and the team [34] proposed an enhanced R-CNN network called Fast R-CNN to overcome the drawbacks of R-CNN. It is a faster object detection algorithm than the R-CNN. The input image will be fed into CNN and there will be generated a convolutional feature map. This convolutional feature map can be used to identify region proposals. Thereafter, the RoI pooling layer was applied to reshape the

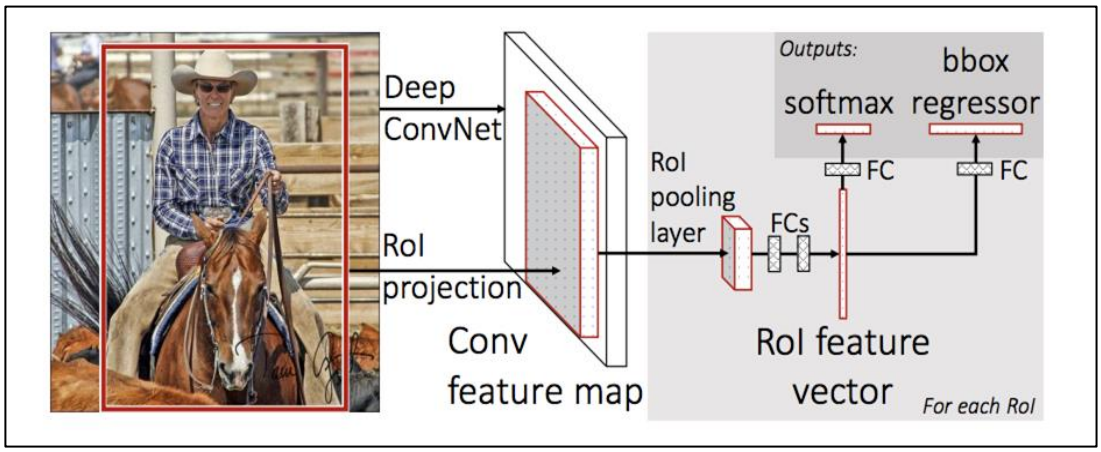


Figure 17: The architecture of F-RCNN

Source: https://www.kdnuggets.com/2016/09/9-key-deep-learning-papers-explained.ht ml/3

region proposals into a fixed size. A fully connected layer has been applied to those fixed-size region proposals and the SoftMax layer has been used to predict the class of the region. Since Fast R-CNN doesn't use 2000 region proposals, it reduced the complexity of selective search and enhanced the time of training and testing over R-CNN.

2.2.2.8 Faster R-CNN

The Faster R-CNN is another enhanced version of R-CNN models. It was proposed by Shaoqing Ren et al. [35] in 2015 which is the same as Fast R-CNN. But it replaced the selective search using RPN (Region Proposal Network). RPN is also another convolution neural network that is capable of identifying region proposals more effectively than selective search. As a result, it solved the bottleneck of having higher time consumption for the selection of region boxes and achieved real-time applicable solutions. Thereafter RoI pooling layer was applied to reshape the region proposal. Finally, it classifies the object within the bounding boxes.

2.2.2.9 SSD

W. Liu et al. [36] proposed a paper on SSD (Single Shot Detector) in 2016. It achieved great speed gains over Faster R-CNN with real-time speed. The specialty of SSD is that it does the generation of the region of interest using region proposal network and region classifications in a single shot. A convolution network applies to the image and calculates the feature map. Then the 3x3 convolutional kernel is used to predict the bounding rectangles on the feature map and classify the objects with probability. SSD predicts the bounding boxes and classifies them using multiple

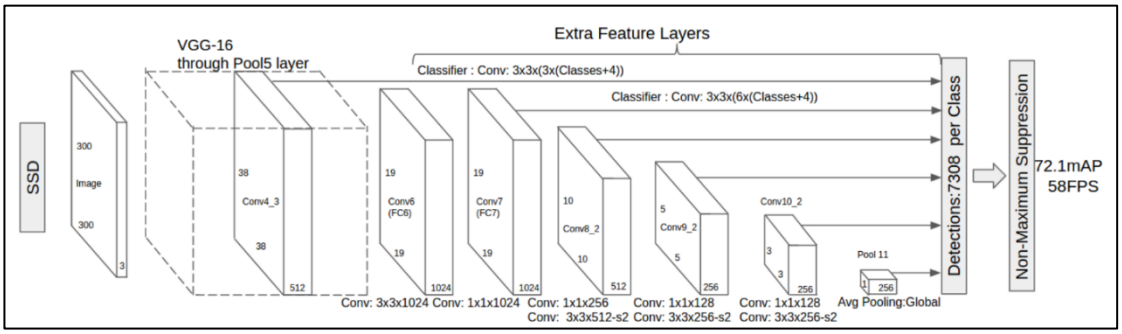


Figure 18: The network architecture of SSD

Source : https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox- detector-for-real-time-processing-9bd8deac0e06

convolution layers. It is capable of detecting multiple objects since those multiple convolution layers can operate at different scales. SSD achieved 72.1% mAP at 58 FPS for 300x300 input and 75.1% mAP for 500x500 input on VOC test on Nvidia Titan X GPU processor.

2.2.2.10 MobileNet

MobileNet was proposed by A. Howard et al. [1] from Google. It is a lightweight pre-trained CNN and is mostly suitable for mobile devices, embedded systems, and computers without GPU. It is applicable for computer vision applications including image classification and object detection. The specialty of the MobileNet is that it uses both regular convolution and depthwise separable convolution. The depthwise separable has two stages as depthwise convolution and pointwise convolution. The depthwise separable has two stages as depthwise convolution and pointwise convolution. First, the depthwise convolution operation will process each channel and create an output vector that has the same number of channels as input. The pointwise convolution will be applied using a 1x1 kernel which is the same as regular convolution. As a result of depthwise separable convolution, there is a significant reduction in the number of parameters and it reduces the total number of multiplication operations with less computational power. It is about 8 to 9 times less computation than standard convolution and faster than traditional CNN. But there is a small reduction in accuracy and latency.

Object detection problems can be extended to human detection. Heo et al. [14] proposed a CNN-based pedestrian detection algorithm on infrared images by considering the season. They used CNN called You Only Look Once (YOLOv2) on top of Adaptive Boolean Map-based Saliency (ABMS) for human detection in their methodology. Due to the season, the emitting energy percentage of pedestrians is varying. The proposed solution contained two Boolean Map-based Saliency (BMS) models according to the season and they achieved 87.12% of precision and 62% of recall. Even though CNN-based object detection methods provide a better performance, computational expensiveness would be a major disadvantage.

2.2.3 Human localization

Human localization is a widespread application in computer vision. It is originated from human detection. As we mentioned before, human localization is a sub-domain of object localization and it will define the person's body relative to image context. The region of interest of the human will be identified in human localization. This would help to get a better idea of a person's location and can be extended to human tracking for better human movement analytics applications.

2.2.4 Human recognition

Human recognition would be a very challenging topic in computer vision. It depends on the testing data and there should be high-resolution images to recognize the person correctly. If we take thousands of people, there might be differences among them. However, there are considerable similarities among them rather than differences. In regard to CCTV camera images, they have relatively low quality and the human body represents a small part of the total context. And, according to the camera position and field of view, the human face will not contain considerable features to detect people uniquely. Due to these factors, it would be a very challenging task to recognize humans using CCTV camera images.

2.2.5 Human tracking

Human tracking would be an important step in human movement analysis which will identify changes in the human body/body parts with time. In most cases, tracking applies to a sequence of image frames like a video stream. There should be a strong algorithm to track the human body/body parts in consecutive frames. The problem of nighttime object detection and tracking using video surveillance data was addressed by Nazib et al. [15] in their research. They applied additional statistical information to the robust contrast model to detect objects. Thereafter, an object tracking model was applied based on the Kalman filter. The correlation factor between the present and previous frames will be used to estimate the motion state or parameters of the object movement. Another research done by Xu et al. [16] have used the SVM classifier, Kalman filter, and mean-shift tracker for their solution. They used a single night-vision video camera installed on the vehicle and focused on pedestrian detection

and tracking. The hotspot occurred due to humans identified and classified as RoI of humans. There were two types of SVM classifiers used in the proposed solution, a single classifier capable of detecting all types of pedestrians and multiple classifiers for each type of pedestrian detection. In the evaluation, they observed that the single classifier showed better performance compared to multiple classifiers.

Kernelized Correlation Filter (KCF) [17] tracker is a decent tracker for object tracking when the target shows no occlusions or orientation changes. KCF tracker is capable of accurately determining when the tracker is lost, eliminating the possibility of false tracking. However, the tracking tends to assume it is lost at violent motions of the target even in cases where Multiple Instance Learning (MIL) [18] and BOOSTING trackers do proper tracking. Open TLD (Tracking, Learning, and Detection) [19] is another popular algorithm for tracking objects. Compared to other tracking approaches, TLD is comparatively better in handling occlusions. It is also able to detect when the tracker is lost. Another special feature of the TLD tracker is that it resizes the tracker bounds as the size of the subject changes. However, in comparison to other trackers such as KCF, TLD is slower and less stable.

2.2.6 Human Re-identification

Person Re-identification is the process where a previous visitor is re-identified as the same person using video feeds from the same camera or a different camera. The re-identification problem can be classified into 3 subsections based on the time (short-term[48][50] and long re-identification[49][50]), background (contextual[50] and non-contextual re-identification[50]), and the dataset (open set [50] and close-set[50] re-identification). The emergence of video-based re-identification which was intended for tracking in videos saw foreground information and color information being used for re-identification. Lately, the advancements in deep learning related to image classification have affected person re-identification mostly.

2.2.6.1 Short-term and Long-term Re-identification

Once the same person is identified on the same day or within a short time interval it is known as short period re-identification [48][50]. There is a key assumption that the targeted person will wear the same clothes within the

reidentification time interval. A more unique or permanent identity can perform long period re-identification where a human can be identified within different days with different clothing (with changed appearance). Most of the state of art research tackles the short period re-identification problem. Long-term re-identification [49][50] requires more expertise and processing and it is still largely an unresolved area.

2.2.6.2 Contextual and Non-contextual Re-identification

Contextual re-identification [50] makes use of contextual information such as other camera positions and landmarks on the fields of view of cameras to identify people moving between (transitions between) views of multiple cameras. Conversely, non-contextual re-identification [50] doesn't use contextual information. It will stick to non-contextual information such as facial features and body features. Since we have to support tracking as well, we may need to select a contextual re-identification method. A best matching non-contextual state-of-the-art system can be implemented if we are to support it as an independent re-identification system.

2.2.6.3 Open set Re-identification and Closed set Re-identification

Person re-identification has two types called "Open Set Person Re-identification [50]" and "Closed Set Person Re-Identification [50]" based on the previous knowledge related to the people in the sample dataset. This prior knowledge may include information such as trained models on images of persons, face recognition models, measurements of considered persons' body structures or gait details of the persons, etc. If we consider a group of people with prior knowledge for the re-identification, we can define it as "Closed Set Person Re-Identification [50]". The most matching person from the given closed set will be identified as the detected person in this approach. If there is a person who is a stranger to the probed dataset, the system would return the most closed match from the dataset which would be a false match. It is unable to indicate that person as a stranger. Closed set person re-identification is a largely researched topic in the computer vision community. However, it can be applied only for specific scenarios where all the probable visitors are previously known. Thus, it does not apply to public spaces. "Open Set Person Re-identification [50]" would be a challenging topic where it does not restrict to a particular dataset. It will detect and

gradually learn the new persons and try to figure out an unidentified person who is previously known.

In the RGB – RGB re-identification problems, we consider the identical features of the person in the images including colors. The trained dataset and tested dataset contained RGB images. Bhuiyan et al. [20] have worked on RGB-RGB re-identification in their research. The appearance of the person varies across the camera due to background illuminations and viewpoint changes. They have addressed these concerns by introducing Stel Component Analysis (SCA) and Cumulative Weighted Brightness Transfer Functions (CWBTFs) for the RGB-RGB person re-identification. RGB-RGB Re-identification is very challenging in surveillance due to variations in lighting conditions. RGB images seem unclear and less informative at low light conditions compared to Infrared (IR) images. Therefore, infrared cameras are more effective than RGB cameras in surveillance applications nowadays. Most of the commercially available surveillance cameras can move from RGB to IR mode automatically in the dark. It would be worth studying RGB-IR cross-modal Re-ID when considering 24-hour surveillance applications. Wu et al. [21] have proposed a solution for RGB-IR cross-modality re-identification in their research. They have considered the existing cross-domain neural network models such as one-stream, two-stream, and asymmetric FC layers. Then they evaluated the concerns among them and proposed zero-padding to train the one-stream network by considering domain-specific nodes in the network for RGB-IR cross-modality matching.

Scale Invariant Feature Transform (SIFT) [22] and Speeded Up Robust Features (SURF) [23] are very popular approaches to solve object detection and recognition in computer vision applications. They were gradient-based features as HOG. Both SIFT and SURF have high discriminative power to segment the objects from context and they are robust against rotation and scale variations of objects. They were barely applied in real-time applications due to higher computational expensiveness than other feature descriptors. Jungling et al. [24] proposed a 3d model for the person re-identification using maximum occurrences in SIFT features [22] extracted from an image sequence. Hamdoun et al. [25] proposed a solution for human re-identification in multi-camera systems using SURF features [23]. They measure the

similarity of interest points in several images using the Sum of Absolute Differences (SAD) during short video sequences. Finally, they achieved 82% precision and 78% recall for their dataset.

2.3 Summary of Literature Reviews

In the context of human detection, preprocessing is a crucial step that can be applied to remove noises and keep image processing and machine learning approaches smoothly. There are some basic operations like resizing, grayscaling, cropping, and noise removal filters. Noise removal filters have been used to eliminate the noise. Gaussian filter, median filter, and adaptive filter are some of them. Furthermore, we found some morphological operations like eroding and dilating to enhance the object regions. Histogram equalization and histogram normalization are the better way to enhance the image quality while preserving edges and boundaries.

Several techniques have been used to detect humans based on computer vision and machine learning. Intensity thresholding is a basic operation that can be used to segment objects from the background. Hysteresis thresholding is an interesting way addressed in literature [12]. But the thresholding will be background subjective and will not be good for complex backgrounds. HOG [4] is a robust edge detection algorithm that performs well against illumination changes but fails with scale and objects variations. HOG is a feature and it is used along with a descriptor like SVM in some research [5][6][7]. CENTRIST features [10] showed better performance with human detection experiments HOG while reducing the training and the testing time significantly. Haar-features classifier proposed by Viola et al. [13] is another interesting way to address object detection. It is commonly used in face detection but can be extended into human full-body detections. But currently, pre-trained classifiers are not suitable with surveillance camera feed due to the angle. There was a novel approach based on haar classifier called ensemble detector [14][15]. They substituted the Haar features with the HOG features and keep the speed advantage of Viola's object detection framework [13] as well as the discriminative power of HOG features [4].

Background subtraction [8] is another way to detect moving objects. It can be applied to human detection when the camera is static or dynamic. But it would be complex when shadows are existing with objects. Pedestrian detection in the driving assistance system is one of the applications in human detection. Shashua et al.[3] the paper suggested a novel way to detect a person by diving into candidate regions and calculating local features. Finally, features among sub-regions and across sub-regions were combined and classified using Adaboost. Nazib et al. [9] have considered object detection and tracking in infrared images. After person detection, they applied a tracking model based on the Kalman filter. Furthermore, Xu et al. [11] have presented pedestrian detection and tracking using a single night-vision video camera installed on the vehicle. They have used an SVM classifier for the people detection and Kalman filter and mean shift tracker for the tracking.

CNN is the most sophisticated way to apply object detection in computer vision. It shows great performance against noisy and complex backgrounds. But training and testing a CNN would be a timely and computationally complex process. Furthermore, we need a large number of data if we create a new CNN. AlexNet [38], GoogleNet[29], VGGNet[30], ResNet[31], YOLO[32], R-CNN[33], Fast R-CNN[34], Faster R-CNN[35], SSD[36] and MobileNet[1] are the some of available CNN model which we found in literature. They have been used for various research problems including object detection and classification. Heo et al.'s [16] paper addressed pedestrian detection in Infrared images based on the season. They used Adaptive Boolean-Map-based Saliency to map the background based on the particular season. They used YOLOv2, which differed from conventional classifier-based methods.

3 DATASET

3.1 Existing Datasets

The image data can be listed in a single dimension or multi-dimensions. They might be found as still images or a sequence of images. Furthermore, they might be taken from a single camera at one background or different backgrounds in the same angle or different angles, view from multiple cameras at different angles, or multi-dimensional data from a scanner. Many publicly available image datasets have been used for computer vision researches. Those images might contain humans, animals, and objects. We can differentiate image datasets into 3 categories based on channels of image data.

3.1.1 RGB image datasets

RGB images have red, green, and blue 3 channels which make more informative data by colors. Most of the available image datasets are RGB image datasets. ImageNet is the famous RGB image dataset that has played an important role in the deep learning revolution. ImageNet is widely used in object recognition research problems. ImageNet has more than 14 million images based on 22 thousand categories which are publicly available for research purposes. In regard to human images, there are considerable datasets that contain a large number of human images.

3.1.1.1 COCO dataset

COCO dataset [38] is a very popular dataset containing 330,000 images including 80 object categories and 91 item types. There are more than 250,000 people with key points that were very useful in human detection and human recognition applications. It is a good object detection dataset due to its characteristic of super-pixel segmentation that has been used for object recognition, segmentation, and captioning.

3.1.1.2 VIPeR dataset

VIPeR dataset [39] has been used in many human detections, human recognition, and human re-identification researches which has 632 images from 2 outdoor scenes. Those images contain humans with full-body visibility. They were affected by considerable resolution changes due to different camera views.

3.1.1.3 i-LIDS-static dataset

i-LIDS-static [40] was mostly used in human re-identification research. It has 479 image pairs taken from four cameras at London Gatwick airport. There are pedestrians with luggage with full-body visibility.

3.1.1.4 PRID2011 dataset

PRID2011 dataset [41] was commonly used for multiple detections for person re-identification problems. Some images contained pedestrians in street views in two different views. They have captured 856 persons from one view and 475 from the other view, 245 persons are appearing in both views.

3.1.1.5 INRIA person dataset

The INRIA dataset [42] is a very useful source in the computer vision community which is commonly used for pedestrian detection researches. The research done by Dalal et al. [5] proposed the HOG feature detector and contributed the INRIA dataset for the community. All the images were taken around the Ohio State University campus. They have not labeled all the images, skipped to label the ambiguous cases. Therefore, information about some persons is missing.

3.1.2 Infrared image datasets

3.1.2.1 ETHZ thermal infrared dataset

The ETHZ dataset [43] was contributed by the Swiss Federal Institute of Technology Zurich. There are 4381 thermal images containing humans, horses, and cats. Furthermore, there are 2418 background images for reference. All the images were taken from a thermal IR camera with a resolution of 324 x 256 pixels. Both 8-bit and 16-bit images can be found in the dataset. There are sets of image sequences that are very useful in human tracking applications.

3.1.2.2 KMU-PD dataset

Keimyung University Pedestrian Detection (KMU-PD) dataset [44] containing moving pedestrian images. It includes partial or full body pedestrians at nighttime, pedestrians with different speeds, and activities. The images were taken from a 30 Hz thermal camera set up on the vehicle which was moving from 20 km/h to 30 km/h

when the outdoor temperature is about 30°C. The authors are proposing 13 video sequences taken from different conditions based on pedestrian movement and vehicle moving speed.

3.1.2.3 BU-TIV dataset

BU-TIV thermal infrared dataset [45] consists of 63,782 frames, recording thousands of objects including pedestrians, cars, bicycles, flying animals, etc. The images were taken using FLIR SC8000 cameras in 1024 x 1024 resolution. The purpose of the benchmark for the various computer vision-based researches includes single object tracking, multi-object tracking in different views, analysis of motion patterns, and censusing wild animals.

3.1.3 RGB-Infrared image dataset

3.1.3.1 KAIST dataset

Korea Advanced Institute of Science and Technology (KAIST) [46] multispectral pedestrian dataset contains 95000 color-thermal pairs taken from a moving vehicle carrying a color camera and a thermal camera. They covered diversified perspectives of the world captured in the day and night time including sunrise, morning, afternoon, sunset, night, and dawn. All images were manually annotated and there are 1,182 unique pedestrians.

3.1.3.2 OTCBVS benchmark dataset

OTCBVS [47] is a publicly available dataset for testing and evaluating computer vision-based algorithms. There are 13 datasets based on different types of domains and taken from different cameras and different locations. Several researchers and students have used them as a benchmark for their research. OTCBVS dataset contains images and video data which spread within and beyond the visible spectrum. There are visible, NIR (Near Infrared), FIR (Far Infrared), and depth images. All of them are available for free to all researchers in the computer vision community.

3.1.3.3 SYSU-MM01 dataset

SYSU-MM01 dataset [21] contains both RGB and IR images which were taken from 6 different cameras (2 IR cameras and 4 RGB cameras). There are 491 unique

persons from 15,792 infrared and 287,628 RGB images. This benchmark was published and used for the RGB-infrared cross-modality human re-identification.

TABLE I. COMPARISON OF CURRENTLY AVAILABLE IMAGE DATASETS

Dataset	Images	RGB	IR
COCO	330,000	yes	no
VIPER	1,264	yes	no
iLIDS	476	yes	no
CAVIAR	610	yes	no
INRIA	5213	yes	no
ETHZ	4,381	no	yes
KMU-PD	9,412	no	yes
BU-TIV	63,782	no	yes
KAIST	95,000	yes	yes
SYSU-MM01	287,628/15,792	yes	yes

3.2 IRANALYTICA infrared image dataset

For our research, we are proposing an infrared dataset called IRANALYTICA [2] taken from 30 unique personals including both genders. There are 28,453 infrared images which contain 19 males and 11 females. We captured the data from the 2 Megapixel IP camera at 30 frames per second. First, we captured the video feed and converted them into thousands of images. There were some distorted, noisy images in the collected dataset. We had to clean them up by the manual removing process. Later, we annotated all images with their ground truth. The dataset has been divided into a training sample with 22763 images and a testing sample with 5690 images. Here are some of the images of the IRANALYTICA dataset.

(a)

(b)

(c)

Figure 2: Images of IRANALYTICA Dataset

3.3 Experimental limitations in current datasets and IRANALYTICA

The datasets we found in the literature have some limitations. Most of the datasets covered object classification problems. They had multiple classes and multiple same object occurrences. Those data were not suited for our research since we focused on single human detection. In our research, we wanted to address a robust solution for Infrared images. Therefore, we had to neglect famous RGB image datasets like COCO, VIPeR, and INRIA. We considered Infrared only or RGB-Infrared Image datasets. As we noticed, the currently available datasets were not suited for our research problem since we focused on CCTV camera images. Furthermore, those images were taken in different fields of view and the human body was too small compared to the overall image context. Due to that, we missed some important human features. Moreover, the annotation of the image data was not matched with our requirement. By considering these concerns, we got feedback to create a domain-specific dataset that covered our dataset requirement to experiment with the proposed methodology. If we elaborate on our dataset IRANALYTICA, images were taken from a CCTV camera in the same field of view. We captured Infrared images in different lighting conditions while capturing the human body as much as clear. We were able to capture a single person in the images since we considered single-person detection. Moreover, we could capture both genders from different angles when they were moving.

4 METHODOLOGY

This section explains the proposed methodology with the modules integrated with the final solution. The overall solution has been defined by addressing a set of subproblems in the human movement analysis domain. The latter section elaborates those sub-problems, identifies alternative solutions, and states the selected approach.

4.1 Overall Solution Breakdown

Here we are elaborating on the main approaches we have done in our research on human analytics in constrained light conditions in figure 3. The main functionalities have been assigned to 4 main units as follows,

1. The camera feed agent
2. The image processing agent
3. The human detection agent
4. The human tracking agent

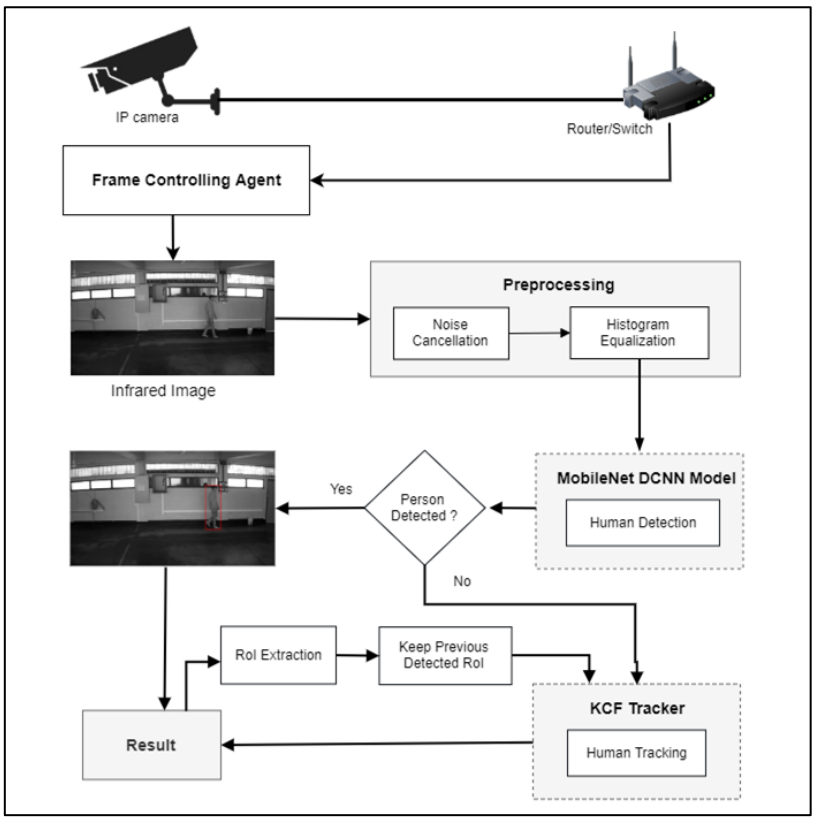


Figure 3: The overall architecture of the proposed system.

The camera feed agent will capture the background context from a 2MP IP camera and transmit the camera feed into the image processing agent. We had to do some preprocessing steps on the raw image frame before applying human detection. Therefore, we applied noise filters and histogram equalization to enhance image quality. After the preprocessing stage, human detection will be applied using a deep convolution neural network in the human detection agent. In this stage, the DCNN identifies humans from the input image and predicts the region of interest of the detected person. The video input is a sequence of image frames. Since our research focused on infrared surveillance videos, we considered human detection and human tracking by taking consecutive image frames from the video inputs. The KCF tracker-based adaptive motion model has been proposed here to improve the human detection accuracy of the overall system by reducing the missed detections of DCNN. Our proposed solution has been performed well with testing samples.

4.2 Subproblems and Evaluated Alternative Solutions

As we explained in the overall solution breakdown, the solution we have developed is a multi-staged flow. Each stage of the solution involves a set of subproblems of the overall problem. Thus, the overall solution combines solutions to a set of sub-problems. Alternative prototypes were developed to evaluate each subproblem and the most suitable alternative was selected. Here are the identified sub-problems,

1 Obtaining camera inputs.
2 Pre-processing of camera inputs.
3 Human detection.
4 Feature extraction
5 People tracking.
6 People re-identification.
7 Deploy and set up the solution in a real-time environment.

4.2.1 Obtaining camera inputs

Obtaining inputs from the camera was the first sub-problem we identified in our research. Capturing image feed and transmission through the network is a very

crucial part of the overall solution. Therefore, all inputs should be captured through the camera and processed thereafter. In the testing environment, the inputs may be fetched from stored media. In the actual production environment, the system should be able to obtain inputs from USB camera drivers or IP camera networks. We focused on real-time video stream processing at this step. Therefore, our expected output was a stream of image frames. Furthermore, we targeted infrared images in this stage of the project. Therefore, the output would be a **n x m** array of infrared images.

4.2.1.1 Native OpenCV implementation available for C++

OpenCV can be identified as a famous standard library written in C++ and used for image processing purposes nowadays. This is an open-source library and it is available for different operating systems such as Windows, Linux, Mac OS, iOS, and Android across different platforms such as x86, x64, and ARM. OpenCV supports video streaming via the VideoCapture class which provides a uniform interface for streaming video content from disk, external peripherals, or sockets.

4.2.1.2 Native Python wrapper for OpenCV

OpenCV has a native Python wrapper developed alongside the C++ library by the original authors. This wrapper is integrated with the NumPy library, which is another famous standard library used in Python for numerical processing. Additionally, the features of Python such as tuples and inner functions make it a preferred programming language for rapid prototyping. However, for developing compute-intensive end products, Python may not be a good option since it is an interpreted language. OpenCV Python wrapper supports video streaming via the VideoCapture class which provides a uniform interface for streaming video content from disk, external peripheral, or sockets.

4.2.1.3 Java wrapper for OpenCV - JavaCV API

OpenCV has a native Java wrapper developed alongside the C++ library by a third-party developer. This library is available under Apache license 2 and GNU general public license 2. JavaCV wrapper supports video streaming via the VideoCapture class which provides a uniform interface for streaming video content from disk, external peripheral, or sockets.

4.2.1.4 Third-party .net wrapper on OpenCV - EmguCV API

EmguCV is a .net wrapper for the image processing library OpenCV. It is available for Microsoft Windows via .Net Framework and for Linux and Mac OS via Mono. This library is developed by a third-party developer EmguCV and is available under a dual license agreement. For non-commercial purposes, the library can be used for free. However, for commercial purposes, a separate license is needed to be obtained. EmguCV wrapper supports video streaming via the VideoCapture class which provides a uniform interface for streaming video content from disk, external peripheral, or sockets.

4.2.2 Pre-processing on camera input

Raw images captured through the camera, which have noise due to various factors such as network delays, hardware issues, background illuminations, etc. Due to this, the preprocessing of raw images is essential before we apply human detection. This preprocessing is undertaken to reduce false positives and false negatives. Additionally, pre-processing can enhance the performance of the human detection process. The input was a **n x m** vector image before the preprocessing stage and the expected output would be a denoised image. Here the preprocessing techniques are undertaken.

4.2.2.1 Manual removing distorted images

Due to network delays and issues with the streaming of the system, we obtained some distorted images. Figure 4 shows some of the distorted images we found.

Figure 4: Distorted images of the collected dataset

These types of images negatively will influence the overall performance of the system in the training stage. We had to follow the manual removal process while annotating the images of the dataset. Since our dataset has more than 30,000 images, this manual process became so challenging.

4.2.2.2 Resizing of the input frame

We have used the 2-megapixel IP camera for data collection. We reduced the resolution to 1280 x 720 by considering the performance of the system and the required frame rate. This approach significantly enhanced the processing time of the camera feed while it slightly affected the detection of faraway humans due to lack of resolution. Therefore, there was an insignificant impact on the precision of the mapped location of the person while the processing speed became high.

4.2.2.3 Grayscale conversion

The captured infrared images have 3 channels that consume more memory bandwidth and processing workload than a grayscale image with only 1 component. In many detection applications for 3 channel images, it is required to calculate the pixel intensity values, which is an extra compute step compared to grayscale images. Therefore, it is computationally heavier to process raw infrared images over grayscale images. As a result, a grayscale copy of each frame is obtained for detection purposes.

4.2.2.4 Denoising using noise filters

Noise is a widespread factor when the camera does not receive sufficient light in images. This noise is observed as a snowy texture of dots on video frames. By reducing the noise effects on the input image frames, we could reduce the false positive and false negative detections. The usage of the denoising algorithms has a performance impact which reduces the frame rate to about 2 - 3 frames per second. However, the performance impact of denoising does not have an impact on the results severely. Different types of noise filters can be applied in our use case. The Mean Filter, Median filter, Min/Max filter, Adaptive filter are some of them. We have tried some of the noise filters for the raw images.

4.2.2.4.1 Mean filter

Mean filter is the simplest noise filtering method in computer vision applications that can smooth the image while reducing the intensity variation among adjacent pixels. For a given vector, each pixel will be replaced by its mean(average) value. The pixels which are not correlated with their surroundings will be eliminated as a result of the mean filter.

The mean filer is another simple noise filtering method used in signal processing applications. It is a convolution filter and there should be a kernel to convolute the image. The kernel reflects the shape and size of the neighborhood of the image and a 3×3 square kernel is the most common one. However, based on the level filtering we can use larger ones like a 5×5 square kernel for severe smoothing. Here is an example of mean filtering. Here we will consider the 6x6 vector and 3x3 kernel. We will extend the border values by outside boundary values. Then apply 3x3 convolution and assign the mean value for the particular pixel. Figure 5 shows an example of a mean filter.

The mean filter will create a certain amount of blurring effect on the image. Due to this, there will be lost important features of the original image. However, the mean filter can be used to reduce different types of noises like Gaussian, uniform, or Erlang.

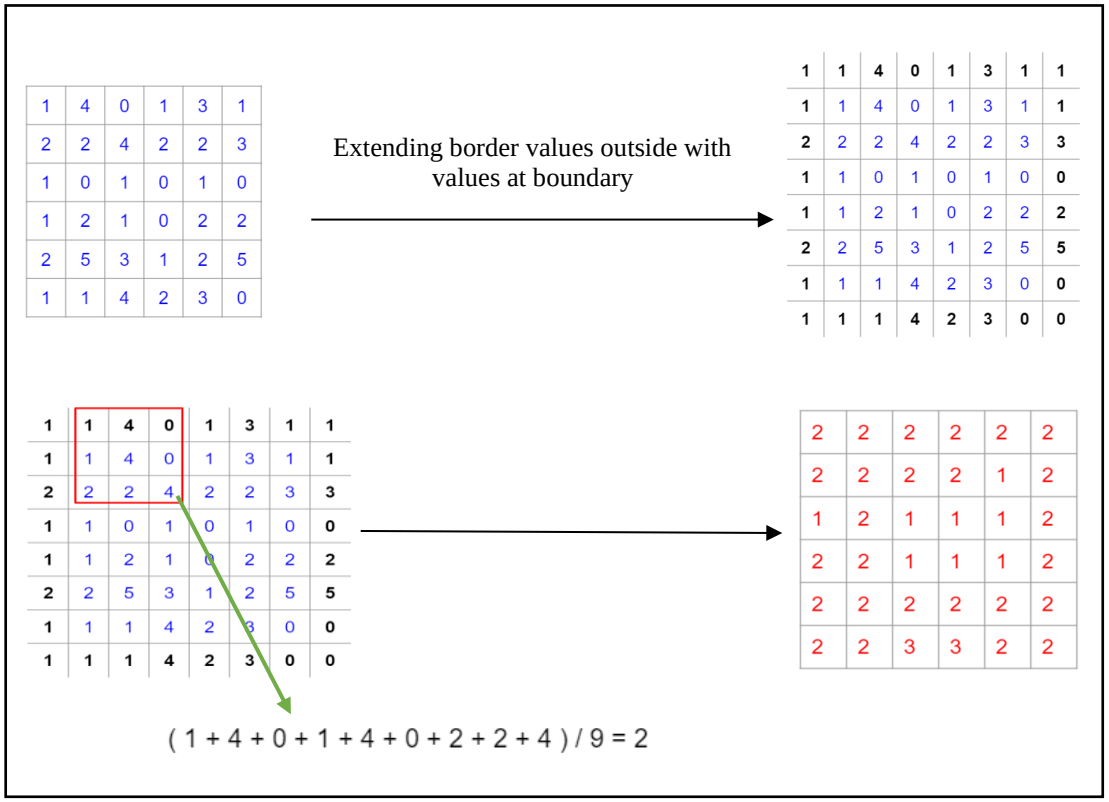


Figure 5: Mean filter example

4.2.2.4.2 Median filter

The median filter is also a very famous approach that can be used to reduce the outlier noises in images. It moves through the pixel by pixel of the given vector and replaces each pixel with the median value of its neighboring pixels. When we compare with the mean filter, the median filter will remove noise components while preserving important features of the original image. It effectively removes outlier noises like “salt” and “pepper” type noises.

The window is the pattern of neighbors, which is used to slide each pixel over the whole image. The pixel values of the given window will be sorted in ascending order and replaced by the target pixel by the middle value of the window. Here is an example of a median filter. From the pixel values of the window, “0” & “15” intensity values are abnormal than others. Seems they are not relevant to the context. They are impulse/outlier noises. Figure 6 shows an example of the median filter.

The low pass filters are less effective compared to the median filter in removing impulse noises since spatial smoothing may cause image-independent noisy components with the original signal. On the other hand, the median filter will remove the signal independent outliers while keeping edges and important features of the image.

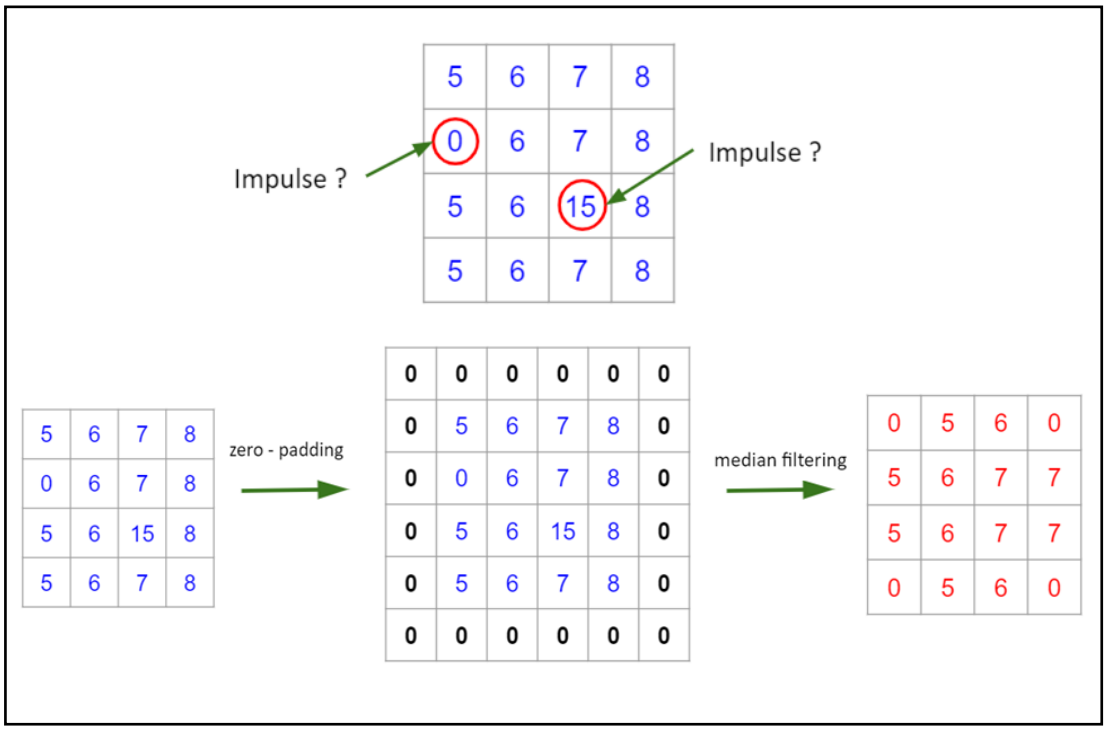


Figure 6: Median filter example

4.2.2.4.3 Adaptive filter

The adaptive filter can be called a self-adjusting filter. Because it can adapt itself to the local features and structure of the given image. It is not another filter as the kernel-based filter. Adaptive filters show better performance in the process of removing impulse noises than median filters. The filtering operation is not purely uniform and depends on the local characteristics of the image. Adaptive filters are effectively applicable when we know the nature and characteristics of the signal being processed.

Adaptive filters can be used against "speckle" noise, which suffers from coherent imaging systems like ultrasound. There are different ways of statistical models of speckle noises and adaptive filters can be applied accordingly.

$$g(x,y) = f(x,y) + f(x,y)n(x,y) \tag{1}$$

Let take ***f*** as the original image content and ***g*** as the corrupted image. If we represent the noise component, it would be another function of ***f*** and it would be like ***f(x,y).n(x,y)*** where ***n(x,y)*** would be the zero-mean Gaussian distribution. Speckle noises depend on the magnitude of the image component ***f***. They negatively affect both spatial and contrast resolution.

4.2.2.4.4 Histogram equalization

Histogram equalization is a popular method to process signals. It is also used in image processing applications to modify the contrast of the image by altering the intensity levels. It is a monotonic and non-linear function that creates a uniform distribution of intensity levels in the output image by re-assigning the intensity values of pixels in the input image.

Let ***f*** as the intensity function of the original image and ***P*** as the normalized histogram function of the input image. Here we can define the equation of P_n as follows in (2),

$$P_n = \frac{number\ of\ pixels\ with\ intensity\ n}{total\ number\ of\ pixels} \tag{2}$$

This ***f*** function is ranging from ***0*** to ***L – 1*** where ***L*** is the number of possible intensity values. For an 8-bit grayscale image, ***L*** would be 256. The histogram equalized image(***T)*** can be defined by (3),

$$T(k) = floor\ (\ (L-1)\sum_{n=0}^{k} P_n) \tag{3}$$

Let take the intensity functions of ***f*** and ***T*** as continuous and random variables that are bounded with [***0, L − 1***]. By assuming ***T*** is differentiable and invertible, we can define the ***T(A)*** function as follows by (4) where ***P*** is the probability density function of ***f***. ***T*** is the cumulative distributive function of ***P*** multiplied by ***(L − 1)***.

(4)

$$T(A) = (L-1)\int_0^A P_A(x)dx$$

The preprocessing stage generated clear views of the images with finer edges than raw images. It helped with the features extraction step and enhanced the detection rate by 5.1%. Here we are listing the raw image in figure 7(a) and the resulting image after histogram equalization in figure 7(b).

(b)

Figure 7: The preprocess using histogram equalization (a) raw image (b) enhanced image after histogram equalization.

4.2.2.4.5 Histogram normalization

Histogram normalization is a linear operation and is commonly used in image processing applications to improve the contrast levels of the image. It causes stretching in the distribution of intensities into a discrete distribution of probabilities. Due to this, it is called "contrast stretching" or "histogram stretching". If we take the input image pixel intensity as ***I(x,y)*** and the resulting pixel after applying histogram normalization as ***R(x,y)***. The normalized pixel can be defined by the following equation (5),

$$R(x,y) = \frac{(I(x,y) - I_{min}) \times (L_{max} - L_{min})}{(I_{max} - I_{min})} + L_{min} \qquad (5)$$

Where I_{max} - Maximum intensity value of the image

I_{min} - Minimum intensity value of the image

L_{max} - Maximum targeted image intensity (Typically L_{max} = 255)

L_{min} - Minimum targeted image intensity (Typically L_{min} = 0)

Although the technique is quite useful and simple to implement, Histogram normalization is at risk of outlier noises. Suppose the image has all the pixels in the range [200 - 255] except one pixel which has a value of 0. Then the Histogram normalization will not work at all.

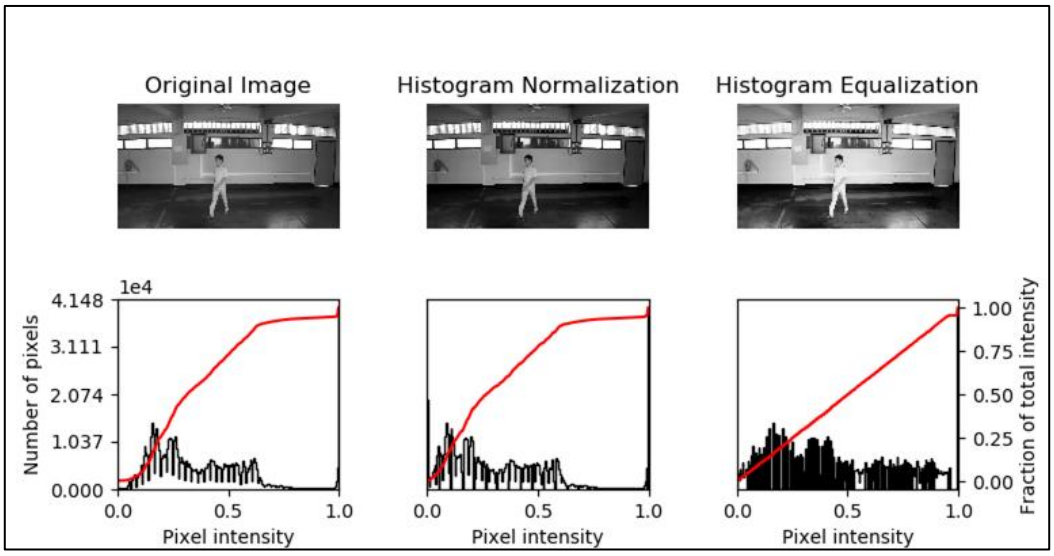


Figure 8: Comparison of the histogram equalization and histogram normalization

4.2.2.4.6 Morphological transformation operations

Morphological operations are commonly used in computer vision which processes images based on shapes. These operations are only relying on the relative order of the pixels rather than their intensity values. They are suitable for binary image processing. However, they can be used for grayscale images too. Morphological operations probe the input image with a small component called a structuring element. The structuring element will be compared with the corresponding neighborhood of pixels. There are two types of morphological operations called dilation and erosion.

Dilation introduces another layer of pixels to the inner and outer boundaries of the regions while erosion removes pixels from the context. As a result of dilation,

objects have become more visible and small holes in objects have been filled. The erosion removes islands and small objects so that only substantive objects will be retained. The shape and the size of the structuring element severely affect the number of pixels added or removed from the object boundaries. Figure 9 elaborates on the steps of morphological operations applied in the segmentation process of the sheep on a gray image.

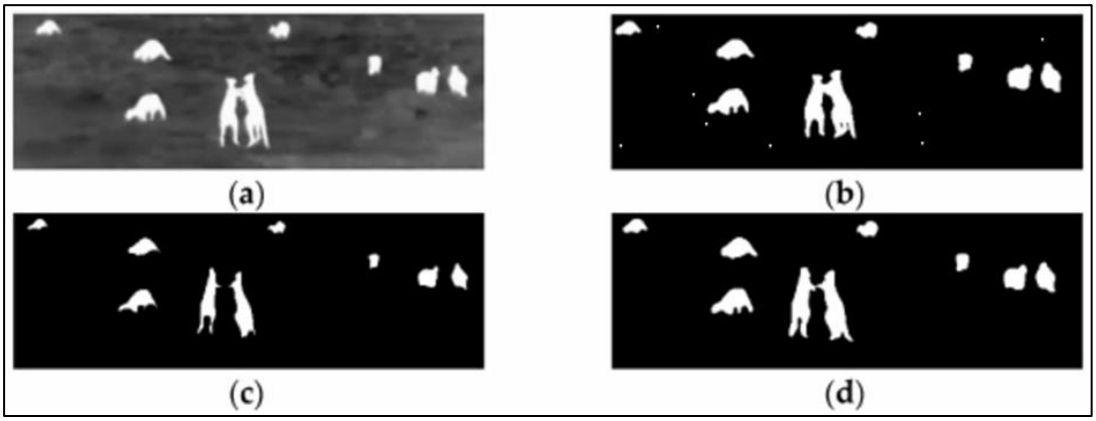


Figure 9: Morphological operation steps (a) original image, (b) after global threshold, (c) after applying erode, (d) after applying dilate

4.2.3 Human detection

It is necessary to identify a special feature point of the person that can be used to pinpoint his/her location. An intermediate step of determining this feature point could be identifying a tight boundary box around the body of each person in the frame. Afterward, by analyzing inside this boundary, one can identify the location of a special feature point of a person which can be used to pinpoint the location of the person. In this stage, we will take the preprocessed image as the input and we will get the detected persons with their bounding rectangles. Listed below are the alternative solutions considered for this problem.

4.2.3.1 Histogram of Oriented Gradients for human detection

Histogram of Oriented Gradients (HOG) utilizes a classifier trained on gradients observed on the human body to detect humans. The algorithm utilizes a moving window alongside the classifier to detect the boundaries of humans. For our experiments, we used the trained HOG detector available in OpenCV to detect persons.

This detector demonstrates decent precision and recall in detecting humans. However, it often fails when body parts are occluded. Another drawback of HOG Detector is that existing trained detectors are not suitable for CCTV Camera angles. A major drawback of the HOG detector is that it does not provide a tight boundary box to the detected person. Sometimes, it provides a boundary enclosing the actual tight boundary with a large gap. On some occasions, it provides a boundary not fully covering the people. This inconsistency means that it is not possible to directly estimate the position of a special feature point on a detected person by using ratios from the boundary box.

A major advantage of HOG Detector is that it is very fast and therefore real-time compatible. A non-GPU accelerated version of the algorithm can easily exhibit 20 - 30 FPS in a modern computer. Additionally, this algorithm is relatively lightweight compared to some other techniques we considered. These advantages are of significant importance since our research is on real-time human detection analytics. Figure 10 elaborates the results we obtained when trying the HOG detector on RGB images and infrared images.

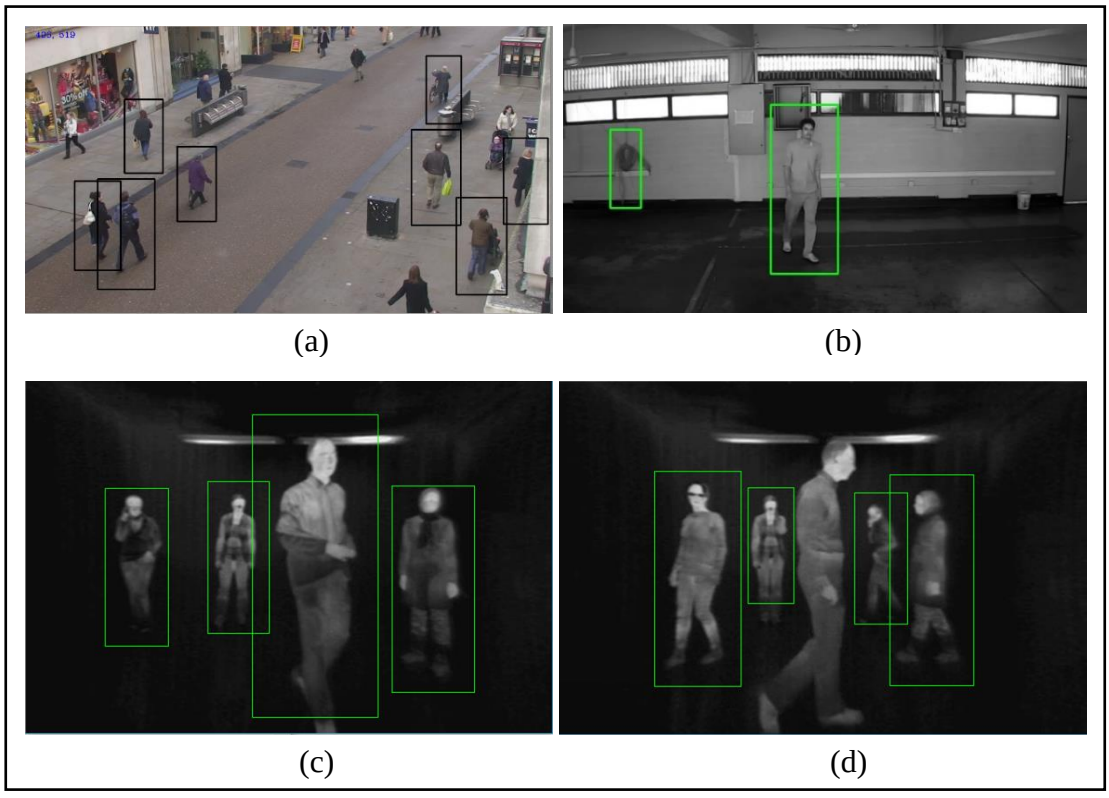


Figure 10: Human detection using HOG detector in (a) RGB image (b) Infrared image (c) Infrared image (d) Infrared image

4.2.3.2 Haar cascade detector

Haar cascade classifier is a famous method in the computer vision community. It is based on Haar-like features and the Adaboost learning algorithm. Paul Vola and Michael Jones proposed the Haar-like features by improving the idea of Haar wavelets. They contain a sequence of rescaled square-shaped features. It considers adjacent rectangular regions at a specific location and calculates the difference between the sum of pixel intensities in each region. This difference can be used to segment the subsections of the image. Haar classifiers can solve pattern recognition problems in computer vision like object recognition. OpenCV provides pre-trained sets of haar classifiers based on the Viola-Jones Object Detection Framework. They are available for real-time object recognition applications including human detection, face detection, etc.

They worked fine with normal camera images, not with surveillance camera images due to some factors like the number of objects, camera to object distance, the angle between the camera and object, and the environmental conditions during image acquisition. The camera-object distance is a very important factor, and it is not proportional to the precision and recall of human detection. Furthermore, it is difficult to detect humans when they are not facing the front view since the classifiers were trained using front view images. If there are any other objects similar to the human body shape, there can be false detections. The infrared shadows formed from the reflection on glass or painted walls are some examples that make false detections. We

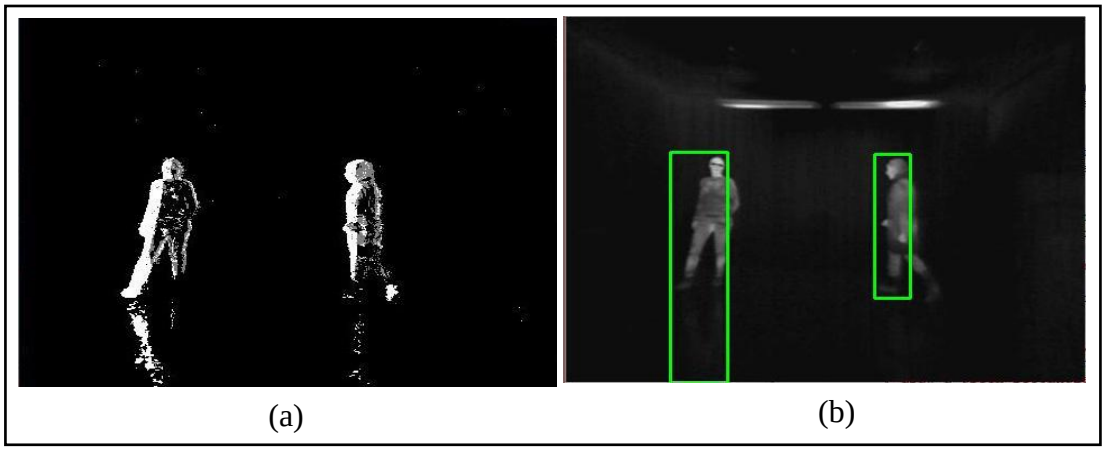
(a) (b)

Figure 11: Human detection using (a) haarcascade_mcs_upperbody.xml (b) haar cascade_fullbody.xml classifiers

have applied the pre-trained classifiers provided by OpenCV (haarcascade_upperbody.xml haarcascade_fullbody.xml, etc.) to detect our collected infrared images. They were not able to detect humans successfully. Figure 11 shows the results we obtained.

4.2.3.3 Background subtraction

Background subtraction is a popular way to detect changes in image sequences in computer vision applications. The background subtraction can be applied when there is a motion of the object or the camera. That can be used for human detection and it is simple compared to currently available methods. The difference between the current frame and the background image will be taken to identify the moving object from the image context. Background subtraction is mostly appropriate for object detection in a sequence of images. It can be extended to different applications in computer vision like human tracking, human behavior detection, and pose estimation. In most of the research, the background subtraction algorithm was used when the camera was fixed. However, Khandhediya et al. [8] tried something different than traditional background subtraction. They proposed the solution by introducing adaptive background subtraction for moving cameras. They focused on the motion of the camera and applied adaptive background subtraction to detect pedestrians from the moving camera feed. The major disadvantage of background subtraction is that the detection process would be more complicated when shadows are existing with the object. Because when the object moves in, the object's shadow also moves relatively.



(a) (b)

Figure 12: Background subtraction (a) frame difference image (b) detected RoIs of humans

4.2.3.4 OpenPose detector

OpenPose is a real-time human keypoint detection library written by a group of researchers from Carnegie Mellon University, Pittsburgh, USA. It is an open-source project which was written in C++. OpenPose is capable of detecting multiple persons alongside 18 key feature points on them by analyzing video frames. This algorithm uses trained models on Caffee with CUDA GPU acceleration to provide real-time performance. This library is also highly accurate when compared to other alternatives. It initially identifies feature points independently and then uses a greedy algorithm to map the points to obtain a person. Therefore, it can detect persons without the full body is visible. Additionally, the algorithm does not slow down based on the number of people in the image frame, making it very suitable for public spaces with large crowds. Another advantage of using the OpenPose library is that it provides human feature points. This feature eliminates the separate special feature point estimation logic if the required feature point is visible. Additionally, the availability of a set of feature points on the human body makes it possible to do pose estimation based on the relative position and visibility of the feature points. Additionally, by using the provided feature points, an accurate tight bound on the screen space occupied by a detected person can be determined. OpenPose consumes more computing power and memory than other alternatives. Therefore, when compared to other alternatives there is the necessity of GPU acceleration.

(a) (b)

Figure 13: Examples of human detection using OpenPose

Source: https://www.youtube.com/watch?v=pW6nZXeWlGM

4.2.3.5 Convolution neural network

Convolutional Neural Network (CNN) is another type of deep neural network, commonly used to recognize patterns with fewer processing steps. They are very famous in the computer vision and signal processing domain. CNN is very close to regular Neural Networks and it contains a large number of neurons with learnable weights and biases. Each of them gets inputs and performs the convolution operation. Unlike regular Neural Networks, there are 3 dimensions such as height, width, and depth. There are some distinct types of layers called Convolution, Fully Connected, ReLU, Pooling layers, etc. Each layer is capable of transforming an input 3D volume into another output 3D volume. As we identified in the literature, there are a bunch of CNN models like AlexNet, GoogleNet, VGGNet, ResNet, YOLO, R-CNN, Fast R-CNN, Faster R-CNN, SSD, and MobileNet. Most of them were applied to object detection and object classification problems in the previous research. But some research [14][51] used CNN to address the human detection problem.

4.2.4 Feature extraction

Feature extraction is a very important step in machine learning applications. For feature extraction, there are several extraction methods called descriptors. A feature descriptor is another algorithm that extracts useful information from the input image and encodes them into a series of numbers called feature vectors.

4.2.4.1 HOG feature

HOG [5] is a famous feature in the computer vision community, used for object detection applications. It can be generalized for pattern recognition problems too. In the implementation, it decomposes an image into small squared cells, then calculates HOG features. Finally, it normalizes the output using a block-wise pattern. Due to the distribution of the local intensity gradients or edge directions, the local appearance and the shape of the object can be varied. HOG is commonly used in object detection use cases like human detection and vehicle detection. The gamma and color functions of the input image should be normalized before applying the object searching to increase the efficiency. The sliding window will go through the region by region of the image frame and search objects in the small areas. HOG is invariant to photometric

transformations. That would be a plus point when considering illumination changes of background. However, it would not be good when rotation and scale variations exist in the object. The shape of the human body is varying with time. And there are scale variations & rotations when moving. Due to this, HOG is not a good option when we are considering a robust solution for human detection.

4.2.4.2 SIFT feature

D. Lowe et al. [22] proposed SIFT (Scale Invariant Feature Transform) features to represent local features of the image. It is commonly used for image matching and object recognition applications in the computer vision domain. It consists of a method to detect points of interest of gray-level images in their original definitions. In the implementation of the SIFT descriptor, it is calculated from the intensity of the image around interesting locations in the image domain that can be called points of interest, or as key points. There are 4 main steps in the SIFT computation. The first one is the calculation of the scale-space extrema using the difference of Gaussian. As the second step, the key point candidate localization will be done by skipping the low contrast points. The key point orientation assignment will be considered as the next step. Finally, the local image descriptor will be calculated for each key point based on image gradient magnitude and orientation. The SIFT algorithm can be applied for multi-scale images. SIFT features are invariant against image scaling, transition, and rotation. Furthermore, they are robust against illumination changes, occlusion, and clutter. SIFT is computationally more expensive than HOG and hardly used in real-time applications.

4.2.4.3 SURF feature

Speeded Up Robust Feature (SURF) is a local appearance feature descriptor, commonly used for object matching and object recognition problems as SIFT. H. Bay et al. in 2006 [23] proposed the SURF and it approximates Laplacian of Gaussian with a box filter. The convolution with box filters is faster than the average image when the integral image is used. It is more effective when there are different scales. The value of the determinant Hessian blob detector and Haar wavelet response around the pixel is applied in the implementation of the SURF. The neighborhood around the given key

point is selected and segmented into subregions. The wavelet responses of each sub-region are used in the calculation of the descriptor. SURF performs well against rotation, occlusion, and blurring, but not against illumination variations and viewpoint changes. Moreover, it is 3 times faster than SIFT while performing competitive results against SIFT.

4.2.4.4 Haar features

Paul Viola and Michael Jones proposed Haar-like features in 2001 [10] based on Haar wavelets and cascade function. Haar classifiers can be used for different kinds of object detection problems. They were commonly used for face detection and human detection. The sum of the pixel intensities in adjacent rectangular regions at a specific point was considered and calculated the difference among them. That difference is the key measure when the segmentation is applied. When considering real-world facts, the regions of the eyes are darker than the region of cheeks in human faces. This feature can be represented using two adjacent rectangles in the eye region in the face. We need to train our classifier using a specific dataset according to the targeted detection. If we create a classifier to detect edges and lines, it will be capable of identifying clear edges and lines in the given image. Not for other features. If there is any manipulation of the face such as covering up the eyes with sunglasses, the face detector may fail to detect human faces. The accuracy of the detection of objects is not related to the number used of Haar features. It depends on the training. The accuracy and error margins can be updated in the training stage. Using a simple classifier, those measures can be fine-tuned. The training inputs should define the ground truth of the bounding rectangle of the target object with their positive and negative samples. In the training stage, unique features related to the object are identified and build the model. The OpenCV provides a pre-trained set of classifiers for the detection of a face, eyes, human body, etc. But they showed poor performance with the infrared images. Those pre-trained haar classifiers did not work with current CCTV camera footages due to the angle and field of view of the image frames.

4.2.4.5 Color-based features

Color-based features in images are very common and simple. Considered RGB (Red, Green, Blue), YCbCr (Luminance, Blue difference chroma, and Red difference chroma components), and HSV (Hue, Saturation, Value) color spaces when they were implemented. In most of the research, they have considered color-based features for feature extraction which can be implemented easily. And it is less computationally expensive and very sensitive to noises. The quality of the feature depends on the lighting condition of the external environment.

4.2.4.6 Local Binary Patterns

Local Binary Pattern (LBP) features represent the textual characteristics of the object surfaces. Taking the binary image by applying a threshold to the neighborhood of each pixel would be a very effective way to address the local features. The texture pattern probability can be summarized into a histogram and LBP values will be generated for all the pixels. By considering the distribution of the LBP histogram, texture regularity can be determined. LBP features are robust against the monotonic gray-scale changes like illumination variations. Furthermore, they are less computationally complex than SIFT, SURF and it is possible to analyze images and detect objects in real-time conditions.

4.2.5 People tracking

In human tracking, we tried to solve the misdetections in consecutive image frames. For the tracking, here are the currently available tracking algorithms we found and tested.

4.2.5.1 TLD tracker

TLD [19] has 3 main steps as tracking, learning, and detection. The tracker can split the long-term tracking tasks into short-term tacking problems along with learning and detection. It detects and follows the object in each frame and localizes the object's appearance. The learning step follows the estimation of detection errors and error reduction for better performance. TLD tracker has a strong recovery mechanism that can work under occlusion over a sequence of frames. Furthermore, it works fine against scale variations.

4.2.5.2 Kalman filter

The Kalman filtering algorithm [15] uses the prior motion information of the moving object to predict the location of the object in the next frame. It is a popular algorithm to use on the prediction correction model in linear and time-variant or time-invariant systems. There we should have to define the dynamic model of the targeted object. The constant velocity (CV) model is the commonly used model in most applications which assumes that the velocity of the targeted object is constant d the time interval. Most tracking systems used process noise (zero-mean white noise in the dynamic model) which was conducted empirically.

4.2.5.3 MIL tracker

The MIL [18] is using a small neighborhood around the current location to detect several potential positives rather than the single point of the object. It uses the appearance model including image patches and they can identify the object of interest precisely. As a result, it achieved robust tracking results with fewer parameters. MIL tracker performs poorly against full occlusions.

4.2.5.4 KCF tracker

KCF [17] stands for Kernelized Correlation Filters and it is another type of correlation filter. The correlation between the two samples will be high when they are matched in the correlation filters. The correlation between translated versions of the patch containing the target and the patch at the same location in the next frame will be considered when applying this concept with tracking problems. KCF tracker used HOG features and ridge regression in the implementation and it shows good performance in tacking with faster results. But it shows poor performance against the scale variations of the objects.

TABLE II. COMPARISON OF CURRENTLY AVAILABLE TRACKING MODELS

Tracker	**Advantages**	**Disadvantages**	**Comment**
Tracking, Learning, and Detection (TLD) Tracker [19]	Support for occlusion. Automatically scales tracker bound when object resizes.	Occasional mistracking	Most suitable for long-duration tracking due to the resizing of the tracker.
Multiple Instance Learning (MIL) Tracker [18]	Relatively fast and accurate.	No tracking failure notification. No recovery from full occlusion. No resizing of tracker bounds.	Suitable for short-duration tracking. (e.g. between frames until detector response)
Kalman Filter [15]	Simple and versatile	No recovery from full occlusion. Noise sensitivity is higher than other trackers.	Suitable for short duration tracking
Kernelized Correlation Filters (KCF) Tracker [17]	Fastest of the list and highest accuracy. Good Tracking failure notification.	No recovery from full occlusion. It does not automatically scale the tracker bound when the object resizes.	The tracker does not resize. Therefore, it is not suitable for long-range tracking. Tracker occasionally interprets properly tracked situations as failures. Therefore, it is not suitable for short-term tracking.

In addition to the mentioned cons, it must be noted that none of the above trackers scale well when many tracking labels are present. The system significantly slows down if too many trackers are present in a frame.

4.3 Proposed Solution

For the evaluation of the proposed methodology, we developed a prototype as a minimal viable product. We have set up the system with 4 subunits as a camera feed agent, image processing agent, human detection agent, and human tracking agent. Each step connected with the next unit and helped to increase the accuracy of the final result.

4.3.1 The camera feed agent

The purpose of this unit was to capture the camera feed and transmit it for image processing. For capturing the image feed, we used a 2MP IP (Internet Protocol) camera with a 30fps frame rate. The captured image buffer was transferred using a wired local area network. We have used wifi networks initially for communication purposes. But we noticed that there were some network lags and due to that, there were some distorted images in our initial dataset. So, we have moved to wired LAN as a better communication bridge. The ability to develop rapid prototypes is a significant benefit for research-based projects. Considering the support for rapid prototyping and availability across platforms, Python Wrapper on OpenCV was chosen for obtaining video inputs. When the commercially viable end product is being developed, we are proposing the OpenCV C++ library which contains a similar layout to the Python wrapper. Hope it will perform less computational time than the Python wrapper. Controlling stream buffer while processing image feed was another fact we considered. This would be a very challenging part of the image feed agent. Capturing real-time feed and applying processing on image buffers would not work using a single thread. Initially, our program crashed due to this high computational operation. We have used a concurrent programming method to sort out this problem. Two separate threads have been used and it solved the crashing issue. One thread for the image capturing and syncing the buffer. The other thread was used to do the rest of the image processing operations. We found that it took an average time of 0.09 seconds for the image processing operations in our testing machine (Core i5 2.4 GHz, 16GB RAM, Lenovo ThinkPad laptop). Even though the IP camera captures the feed with 30fps. We had the capability of processing the feed at 6.7 fps. Due to this, our proposed MVP will not capture and process all the frames grabbed by the camera. Since consecutive image

frames did not show the considerable movement of a human, there is no need for processing every frame. We skipped some of the consecutive frames and applied human detection and tracking. Here are the high-level operations of the two threads we used in the proposed solution.

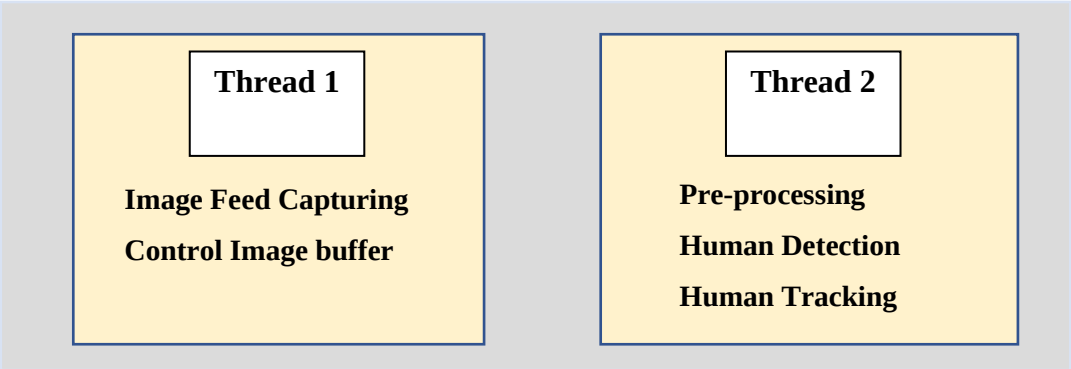


Figure 19: Concurrent video stream processing

4.3.2 The image processing agent

Removal of noise components from the raw images is a very important part of the overall process. As we mentioned previously, noise effects occurred due to various reasons like network issues, hardware faults, and background lighting conditions. This unit focused on the removal of such noses and applying the preprocessing on the raw images before applying human detection. For the removal of distorted images in the captured training dataset, we had to use the manual removal process. Since we had more than 100,000 images in the dataset, it was a very time-consuming process. We have used the 2-megapixel IP camera for the data collection. For better performance of the system, the input image should be in 1280 x 720 px resolution. So, we have resized the images. Then, we have used computer vision techniques on images to reduce noise effects. We have used median filters to reduce the effect of outlier noises and histogram equalization in our proposed methodology. Histogram equalization is a better way to improve the contrast levels of an image in computer vision. It will alter the intensity histogram of the image and generate a uniform distribution. Histogram equalization is a monotonic and nonlinear function when compared to histogram

normalization. It redefines the image with finer edges and enhances the feature extraction process.

Figure 20: The preprocessing using histogram equalization (a) raw image (b) enhanced image after histogram equalization

4.3.3 The human detection agent

The resulting image after applying the preprocessing stage was applied for human detection at this step. We have used a DCNN in this unit and tried to identify the RoI (Region of Interest) of the detected person. We used the transfer learning approach to train the DCNN model using TensorFlow [23] object detection API. The proposed solution should have to apply to the real-time application and therefore, we chose MobileNet [1] DCNN. MobileNet is a lightweight pre-trained DCNN model which has 28 convolution layers. It is a combination of standard convolution and depthwise separable convolution. We transfer learned the model using our own dataset IRANALYTICA with Adam optimizer. After applying depthwise separable convolution layers, we obtained a 7 x 7 x 1024 feature map. We have replaced the bottom layers of the DCNN using an average pooling layer and a fully connected layer with the ReLu function. Figure 21 represents the network architecture of the proposed DCNN model and Table III shows the layer structure of the modified MobileNet model. The RoI of the detected person in the given image was received as the output after the softmax layer of the DCNN.

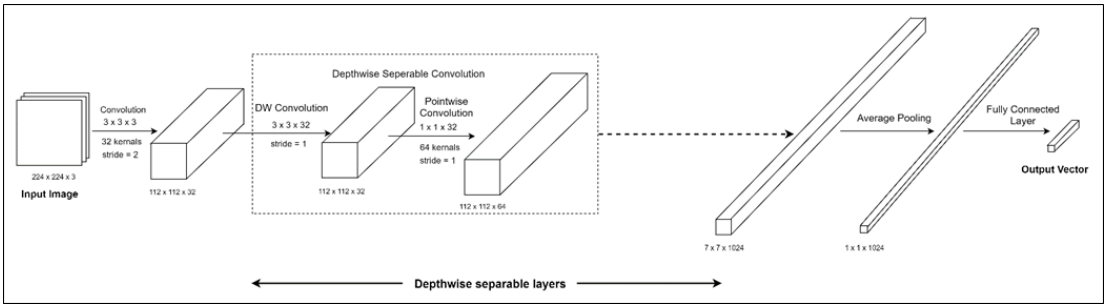


Figure 21: The neural architecture of the proposed MobileNet CNN

TABLE III. THE LAYER STRUCTURE OF THE MODIFIED MOBILENET DCNN MODEL

Layer	Stride	Filter	Input
Conv	2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw[a]	1	3 × 3 × 32 dw	112 × 112 × 32
Conv	1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw	2	3 × 3 × 64 dw	112 × 112 × 64
Conv	1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw	1	3 × 3 × 128 dw	56 × 56 × 128
Conv	1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw	2	3 × 3 × 128 dw	56 × 56 × 128
Conv	1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw	1	3 × 3 × 256 dw	28 × 28 × 256
Conv	1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw	2	3 × 3 × 256 dw	28 × 28 × 256
Conv	1	1 × 1 × 256 × 512	14 × 14 × 256
5 x Conv dw 5 x Conv	1 1	3 × 3 × 512 dw 1 × 1 × 512 × 512	14 × 14 × 512 14 × 14 × 512
Conv dw	2	3 × 3 × 512 dw	14 × 14 × 512
Conv	1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw	1	3 × 3 × 1024 dw	7 × 7 × 1024
Conv	1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Average Pool	1	Pool 7 × 7	7 × 7 × 1024
FC	1	1024 × 4	1 × 1 × 1024
Softmax	1	Classifier	1 × 1 × 4

a. dw – depthwise

4.3.4 The human tracking agent

We applied an adaptive tracking model to increase the human detection accuracy at the final stage. This dynamic model was able to detect the bounding rectangle of humans in the given frame based on the previously successfully detected RoI. In some instances, DCNN failed to detect humans even though there were them. By applying this dynamic model, we were able to reduce the miss detections of the DCNN. We have used the KCF tracker algorithm for the implementation of this dynamic model which is a very good discriminative classifier to detect and track the target from the surrounding environment. It uses HOG features [4] with a ridge regression-based algorithm. KCF tracking shows great performance against benchmarks and it is good for fast-tracking. But it does not perform well against scale variations of the targeted Objects. Figure 22 shows the flow diagram of the proposed adaptive detection model.

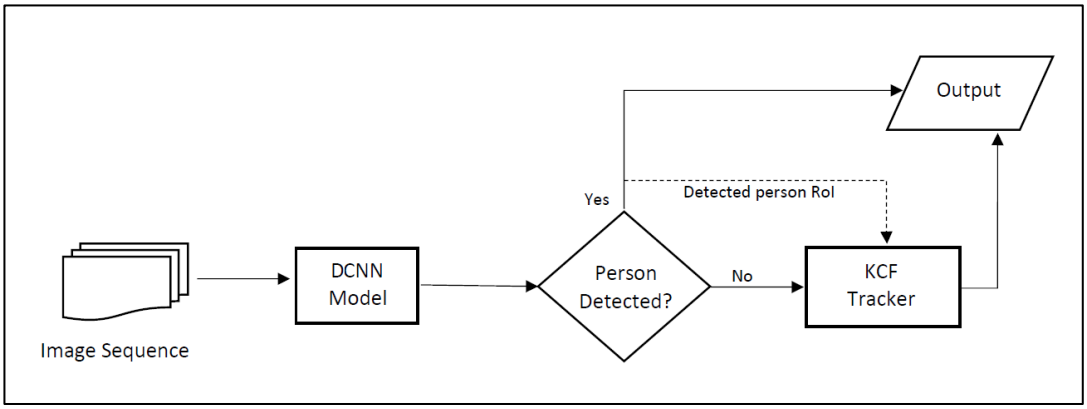


Figure 22: Motion based adaptive detection model

4.3.5 Deploy and set up the solution in the real-time environment

Figure 23: Proof of concept displayed at Techno 2019

4.3.6 Research assumptions and limitations

In our research, we considered constrained image inputs which are defined when the low light conditions. As we identified, RGB image feeds did not suit our research topic since they were highly sensitive to illumination changes. Therefore, we considered infrared images since we wanted to propose a solution for both rich and constrained lighting conditions. Since infrared images can capture the feed-in both times, we could apply this method at any daytime. Furthermore, we were limited to single human detection problems since we trained the model for a single human body. Moreover, our collected and trained dataset has some limitations like using the same background and field of view in the images. Since we used a pre-trained model using the COCO dataset, the proposed DCNN can detect humans in different backgrounds and different fields of view with moderate detection accuracy.

5 EXPERIMENTAL EVALUATION AND DISCUSSION

5.1 Dataset

We are proposing a robust way to address human detection against different lighting conditions. Therefore, we focused on infrared images that are less sensitive to the variation of lighting conditions compared to RGB images. Here we have collected and prepared our dataset named “IRANALYTICA” which contains 28,453 infrared images. To build up the DCNN model, we transfer learned the model using our prepared dataset. All the images were taken from 30 persons (19 male and 11 female candidates). A 2MP IP camera (30 FPS frame rate) is used to capture the vision feed under different lighting conditions and walking poses of the people. The feed was captured and saved in video format. Later they were converted into JPEG image format and constructed the dataset. There were some noisy, distorted images and we manually removed them from the dataset. All the remaining images were annotated with ground truth. The dataset has been divided into a training sample with 22,763 images and a testing sample with 5690 images.

Figure 24: The IP camera used for the dataset creation

5.2 Experimental Setup

For the experiment and evaluation purposes, we have used a testing sample of our collected dataset IRANALYTICA. All the images were annotated and divided images into training samples and validation samples using K-fold cross-validation.

5.2.1 Transfer learning the DCNN

The training dataset was annotated with ground truth and transfer learned using Tensorflow object detection API [25]. We used the ssd_mobilenet_v2_coco.config file which is provided by Tensorflow to transfer learn MobileNet v2 coco model. For the training process, we have used a virtual server machine (Intel Core i5 2.4 GHz CPU, 16GB RAM, Nvidia GTX 1050 GPU, Linux operating system) which was taken from Google Cloud. It took 4.5 hours to train the model.

5.2.2 Evaluation of the model

We tested the enhanced MobileNet model using the validation sample in the same virtual machine and tried to decrease the error rate by adam optimizer. Later for the demonstration purpose, we have created a prototype using a wired LAN connection with Intel Core i5 2.4 GHz CPU, 16GB RAM Lenovo ThinkPad laptop which had Windows 10 64-bit operating system.

5.3 Intersection Over Union

Intersection over Union (IoU) is a standard measure to evaluate the detected bounding rectangle of the person. IoU can be defined as the ratio between the overlap area (Intersection) and the combined area (Union) of the detected rectangle and the ground truth rectangle. Let **A** as the detected boundary rectangle and **B** as the ground truth boundary rectangle. The IoU can be defined as the following equation,

$$IoU = \frac{\text{Overlap Area of A and B}}{\text{Combined Area of A and B}}$$

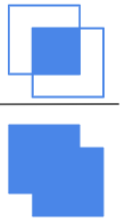
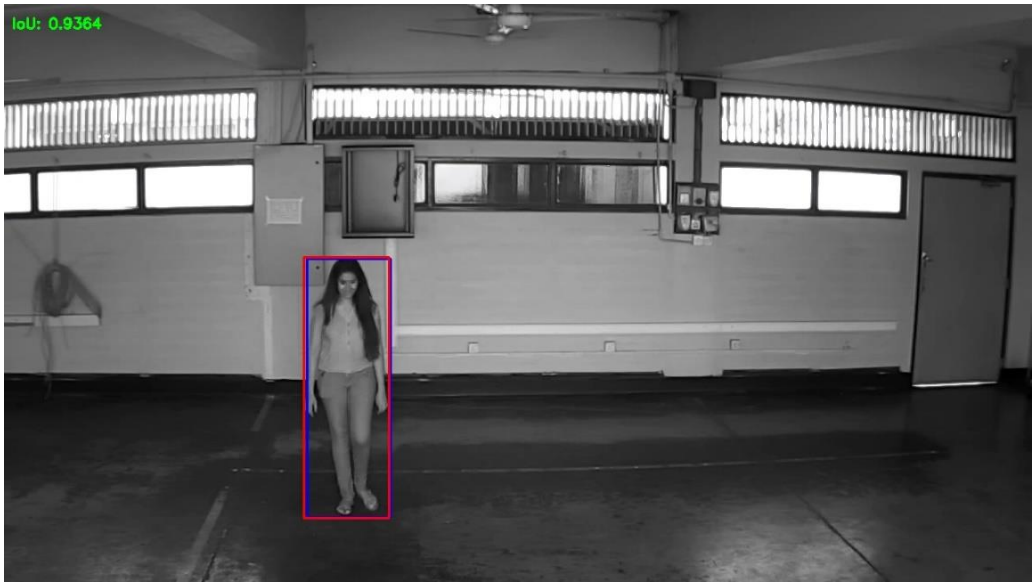


Figure 25: Intersect over Union calculation

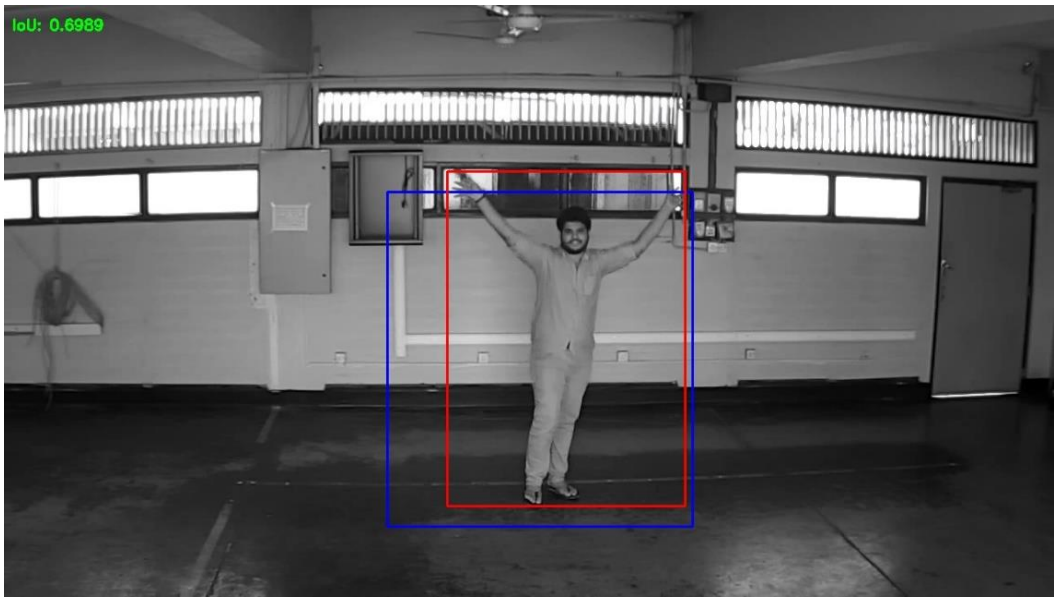
We took the valid detection if the IoU of a given image is greater than 60%. The following figures show the resulting image solution including detected RoI, ground truth RoI, and IoU value after applying our proposed. The blue color rectangle represents the detected RoI and the red color rectangle shows the ground truth RoI of the human. In the left corner of the images, we have represented the IoU value of each

result. In Fig. 25(a) and Fig. 25(b), the detected rectangle is very close to each other, therefore we obtained IoU close to 90%. But in the figure. 25(c), IoU became 69.89 % due to the unfamiliar posture of the person. But detected RoI covered the person's body within the rectangle with surplus space. Since the IoU value is still greater than 60%, we can consider that case as a valid detection.

(a)



(b)



(c)

Figure 26: Results obtained in proposed system

5.4 Results and Evaluation

We used precision, recall, and F1 score to evaluate the results of the proposed methodology. The results are compared with ground truth and verified by taking the IoU. If the (IoU) is greater than or equal to 60%, we considered it as a correct detection. Our proposed methodology showed the growth of the performance in each stage significantly against the infrared test dataset. We could increase the human detection accuracy while decreasing the false positives in the overall solution as an effect of the preprocessing stage and tacking-based adaptive detection model. Table II elaborates on the comparison of accuracy, precision, and F1 score in each stage of the proposed solution. We used an Intel Core i5 2.4 GHz, 16GB RAM Lenovo ThinkPad laptop along with the Windows 10 64-bit operating system for the evaluation of the proposed methodology.

5.4.1 Evaluation of camera feed methods

Initially in our dataset creating step, we obtained distorted images due to network delay. It made our dataset preparation process slower. We used a wired local area network (LAN) instead of wifi LAN. As a result, we could skip streaming issues

and be able to focus on our research main objectives. For the prototype, we have used the OpenCV Python wrapper to retrieve the image feed. The system crashed when we ran the system with a real-time feed. That was due to image buffers and the high computational complexity of human detection and other image processing operations. Since we might have to work with limited computational machines mostly and we had to move for the two-thread architecture. It succeeded and we were able to manage the system without a problem.

5.4.2 Evaluation of preprocessing methods

The purpose of the preprocessing stage was to reduce the noise effect of the raw images and prepare for human detection. There were noises in the raw images due to hardware effects and lighting conditions. We applied median filters to withdraw outlier noises, histogram equalization to enhance the contrast, resizing methods to reduce the complexity at this stage. The preprocessing methods provided clear images including finer edges for the human detection stage. Finally, we achieved 80.11% detection accuracy and 86.63% of precision before applying preprocessing on raw images. As a result of the preprocessing step on raw images, the feature extraction of the input image was improved and increased the human detection accuracy by 5.1%.

5.4.3 Evaluation of human detection methods

Human detection is the main step in our research which will predict the human body bounding rectangle from a given image. Deep convolution neural network-based human detection is the very common and best way to solve object detection in the present. Since we targeted a real-time surveillance system, the MobileNet lightweight DCNN model was transfer-learned with our infrared dataset. We achieved 85.21% detection accuracy for the DCNN model by fine-tuning it with several iterations.

5.4.4 Evaluation of human tracking methods

Since our target was to build up a solution for real-time surveillance systems, we had to consider video input processing and apply human detection to them. The video contains a sequence of consecutive image frames. As an effect of using a tracking-based detection mode, the human detection accuracy of the overall system was increased. The proposed solution could be able to achieve 89.71% human

detection accuracy against infrared images. It would be greater than the previous best detection rate of 87.12% [14] in DCNN based human detection.

TABLE IV. DETECTION ACCURACY, PRECISION, AND F1 SCORE AS RELATED TO THE EFFECT OF PREPROCESSING & MOTION MODEL

Method	**Accuracy (%)**	**Precision (%)**	**F1 Score (%)**
Without Preprocessing	80.12	87.44	88.37
With Preprocessing	85.21	90.90	91.55
With Preprocessing and Motion Model	89.71	91.99	94.22

5.4.5 Comparison of the Results with State art Methods

We have evaluated the HOG detector and YOLOv2 DCNN using our dataset. The HOG features with the SVM detector were used for human detection and we obtained 74.13% detection accuracy. Meanwhile, YOLOv2 DCNN obtained 85.80% detection accuracy against the same testing image sequence. HOG detector failed to detect the human when there was this rotation and angle view of the human while YOLO performed better detection. When we consider the HOG, YOLO, and our proposed method, real-time performance is also a very crucial factor. YOLO has a higher processing time while our proposed method took 218ms to process one frame.

TABLE V. COMPARISON OF ACCURACY, PRECISION, RECALL AND PROCESSING TIME OF HOG DETECTOR, YOLO AND OUR PROPOSED METHOD

Method	**Detection Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Processing Time (ms)**
HOG + SVM detector	74.13	85.01	83.79	232
YOLOv2	85.80	89.57	94.24	951
Proposed Method	89.71	91.99	96.56	218

6 CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Human movement analytics is a very challenging topic in the computer vision domain including image and video content management, human recognition, human behavior analytics, and driving assistance systems. In our research, we addressed human detection analytics in constrained image inputs which is the most challenging. Lightning conditions in the background are a very important factor that rigorously affects the quality of the image. When it comes to the real world, the light conditions vary with time.

Our proposed solution includes both machine learning and a tracking-based detection model for human detection and tracking in the constrained image inputs. We were able to remove the noise effect on the overall solution by applying the preprocessing step on raw images. The raw images may contain noise components due to the hardware effect and background lighting conditions. In the preprocessing stage, we applied a resizing method, noise filters, and histogram equalization. We were able to increase the human detection rate by 5.1% as an effect of preprocessing. We used a machine learning approach to human detection in the second step for the enhanced image. CNNs are the best and common way to address object detection problems nowadays. Since we targeted real-time surveillance systems, we applied the MobileNet lightweight DCNN model. By finetuning with several iterations in the training stage, we achieved 85.21% human detection accuracy for the modified DCNN model. We had to work with video inputs when considering real-world applications. Video inputs are consisting of sequences of image frames. Our proposed KCF tracking algorithm-based adaptive detection model was able to enhance the overall human detection accuracy of the system. The adaptive detection model was able to predict the bounding rectangle of the person by using predecessor frames when the DCNN fails to detect humans in the given frame. Finally, our proposed solution achieved 89.71% of human detection accuracy which is greater than the previous best detection rate of 87.12% [14] in DCNN based human detection. However, there were HOG descriptor-based related works that obtained detection accuracy of 90% in [3] and 95% [10]. Due to the different human body angles and viewpoints of the camera, they had higher false

detection rates. But we could overcome those concerns by applying the pre-processing and tracking-based detection model.

6.2 Recommendation

6.2.1 Multiple person detection and tracking

In our research, we focused on single-person detection and tracking. But when we consider the practical scenario, multiple person detection and tracking would be the next timely and important part of the research topic. So we are proposing to enhance the current methodology to detect and track multiple humans.

6.2.2 Extend the methodology with multiple camera systems

In this research, we considered the single-camera system. Even though this proposed methodology works fine with different angles and different backgrounds. In real-world applications, there are multiple cameras have been used. For a better human detection analytics system, we have to extend this methodology with multiple cameras to solve complex problems.

6.2.3 Try out different human analytics subdomains

There are different ways to address human movement analytics. Here, we focused on human detection and human tracking. But there are furthermore to extend this research in the future. As future work, we are looking forward to remaining approaches like human recognition, human behavior detection, human gait analytics, and human re-identification for the better analytics of humans.

6.2.4 Human movement analytics in RGB – Infrared cross-modality

The commercially available cameras nowadays generate both RGB and infrared feed as the output. They facilitate RGB image frames in the daytime and infrared image frames at nighttime. As future work, our team is looking to extend the research topic with the RGB-Infrared cross-modality concept by introducing correlation parameters. It would be a timely needed robust solution for real-world applications.

REFERENCES

[1] G. 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.

[2] H. Fernando, I. Perera, and C. de Silva, “Real-time human detection and tracking in the infrared video feed” in *2019 Moratuwa Engineering Research Conference (MERCon)*, 2019, pp. 111-116.

[3] J. Ge, Y. Luo, and D. Xiao, "Adaptive hysteresis thresholding based pedestrian detection in nighttime using a normal camera," in *IEEE International Conference on Vehicular Electronics and Safety*, 2005, pp. 46-51.

[4] Riaz, J. Piao, and H. Shin, “Human detection by using centrist features for thermal images,” *International Journal on Computer Science and Information Systems*, vol. 8, no. 2, pp. 1–11, 2013.

[5] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, *2005*, pp. 886-893.

[6] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian Detection using Infrared images and Histograms of Oriented Gradients," in *2006 IEEE Intelligent Vehicles Symposium*, Tokyo, 2006, pp. 206-212.

[7] S. Budzan, “Human Detection in Thermal Images Using Low-level Features,” *International Journal on Measurement Automation Monitoring*, 2015, vol. 61, no. 6, pp. 191–194.

[8] V. Gajjar, A. Gurnani, and Y. Khandhediya, “Human Detection and Tracking for Video Surveillance A Cognitive Science Approach,” in *2017 IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2805–2809.

[9] Y. Khandhediya, K. Sav, and V. Gajjar, “Human Detection for Night Surveillance using Adaptive Background Subtracted Image,” *arXiv preprint arXiv:1709.09389*, 2017, pp. 1–5.

[10] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition*, 2001, pp. 511–518.

[11] S. Dai, Y. Ming, Ying Wu, and A. Katsaggelos, “Detector Ensemble,” in *IEEE Conference on Computer Vision and Pattern Recognition,* Minneapolis, MN, 2007, pp. 1-8.

[12] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng, “Fast Human Detection Using a Cascade of Histograms of Oriented Gradients,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* New York, NY, USA, 2006, pp. 1491-1498.

[13] A. Shashua, Y. Gdalyahu, and G. Hayun, “Pedestrian detection for driving assistance systems: single-frame classification and system level performance,” in *IEEE Intelligent Vehicles Symposium,* Parma, Italy, 2004, pp. 1-6.

[14] D. Heo, E. Lee, and B. C. Ko, “Pedestrian Detection at Night Using Deep Neural Networks and Saliency Maps,” *Journal of Imaging Science and Technology*, 2017, vol. 61, no. 6, pp. 1–9.

[15] A. Nazib, C.-M. Oh, and C. W. Lee, “Object detection and tracking in night time video surveillance,” in *10th Int. Conf. Ubiquitous Robot. Ambient Intell.*, 2013, pp. 629–632.

[16] F. X. F. Xu, X. L. X. Liu, and K. Fujimura, “Pedestrian detection and tracking with night vision,” *IEEE Trans. Intell. Transp. Syst.*, 2005, vol. 6, no. 1, pp. 63–71.

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, vol. 37, no. 3, pp. 583–596.

[18] B. Babenko, M. -H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in *Conference on Computer Vision and Pattern Recognition,* Miami, FL, USA, 2009.

[19] Z. Kalal, K. Mikolajczyk, J. Matas, “Tracking-learning-detection”, in *IEEE Transactions Pattern Analysis and Machine Intelligence*, 2011, vol. 34, no. 7, pp. 1409-1422.

[20] Bhuiyan, A. Perina, and V. Murino, “Exploiting Multiple Detections for Person Re-Identification,” J. Imaging, 2018, vol. 4, no. 2, pp. 28.

[21] A. Wu, W.-S. Zheng, H.-X. Yu, S. Gong, and J. Lai, “RGB-Infrared Cross-Modality Person Re-Identification,” in *IEEE International Conference on Computer Vision*, Venice, 2017, pp. 5390-5399.

[22] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 1150-1157.

[23] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Proc. Ninth European Conference on Computer Vision*, 2006, pp. 404–417.

[24] Jungling and M. Arens, “Local Feature Based Person Reidentification in Infrared Image Sequences,” in *7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Boston, MA, 2010, pp. 448-455.

[25] “Tensorflow Object Detection API”, 2018. [Online] https://github.com/tensorflow/models/tree/master/research/object_detection. [Accessed: Nov. 12, 2018].

[26] R. Mazzon, S.F. Tahir, A. Cavallaro, "Person Re-identification Using Spatial Covariance Regions of Human Body Parts," in *7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Boston, MA, USA, 2010.

[27] B. Waber, "Bloomberg," 16 May 2013. [Online]. Available: https://www.bloomberg.com/news/articles/2013-05-16/the-next-big-thing-in-big-data-people-analytics. [Accessed: 03 June 2017].

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, 2012, pp. 1097-1105.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR*, 2015.

[31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.

[32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” in *IEEE transactions on pattern analysis and machine intelligence*, 2015, vol. 38, no. 1, pp.142-158.

[34] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.

[35] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91-99.

[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. -Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*. Springer, Cham, 2016, pp. 21–37.

[37] "VISIBLE VS. THERMAL DETECTION: ADVANTAGES AND DISADVANTAGES," 09 September. [Online]. Available: https://www.lynred.com/blog/visible-vs-thermal-detection-advantages-and-disadvantages. [Accessed: 03 November 2021].

[38] T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context”, arXiv [cs.CV]. 2015.

[39] Language and Media Processing Laboratory. Viper: The video performance evaluation resource. http://viper-toolkit.sourceforge.net, November 2011.

[40] H. O. S. D. Branch, “Imagery library for intelligent detection systems (i-lids)”, in 2006 IET conference on crime and security, 2006, pp. 445–448.

[41] M. Hirzer, C. Beleznai, P. M. Roth, en H. Bischof, “Person Re-Identification by Descriptive and Discriminative Classification”, in Proc. Scandinavian Conference on Image Analysis (SCIA), 2011.

[42] E. Maggiori, Y. Tarabalka, G. Charpiat, en P. Alliez, “Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark”, in IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017.

[43] Simon Lynen, J.P. ETHZ Thermal Infrared Dataset. 2014. Available online: http://projects.asl.ethz.ch/datasets/doku.php?id=ir:iricra2014 (accessed on 11 June 2020).

[44] M. Jeong, B. C. Ko, and J. Y. Nam, ‘‘Early detection of sudden pedestrian crossing for safe driving during summer nights,’’ IEEE Trans. Circuits. Syst. Video Technol. 27, 2017, pp. 1368–1380.

[45] Zheng Wu, Nathan Fuller, Diane Theriault, Margrit Betke, "A Thermal Infrared Video Benchmark for Visual Analysis", in Proceeding of 10th IEEE Workshop on Perception Beyond the Visible Spectrum (PBVS), Columbus, Ohio, USA, 2014.

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

[47] Bilodeau, G.-A., Torabi, A., St-Charles, P.-L., Riahi, D., Thermal-Visible Registration of Human Silhouettes: a Similarity Measure Performance Evaluation, Infrared Physics & Technology, Vol. 64, May 2014, pp. 79-86.

[48] Z. Imani, H. Soltanizadeh, en A. A. Orouji, “Short-term person re-identification using rgb, depth and skeleton information of rgb-d sensors”, Iranian Journal of Science and Technology, Transactions of Electrical Engineering, vol 44, no 2, 2020, pp. 669–681.

[49] P. Zhang, Q. Wu, J. Xu, en J. Zhang, “Long-term person re-identification using true motion from videos”, in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 494–502.

[50] A. Bedagkar-Gala en S. K. Shah, “A survey of approaches and trends in person re-identification”, Image and vision computing, vol 32, no 4, 2014, pp. 270–286.

[51] B. Yassine, G. Larbi, en L. Hicham, “Human detection in surveillance videos using MobileNet”, in 2020 2nd International Conference on Computer and Information Sciences (ICCIS), 2020, pp. 1–5.