Honour School of Engineering Science
University of Oxford
3rd Year Project

Imaging System Design for UAVs

- Human Casualty Detection in High Altitudes -

Student: Jin Yeob Chung
Supervisor: Dr. Jonathan Gammell

Contents

Please note that this essay *Imaging System Design for UAVs* (30 pages) was a part of a wider group project *System Design for UAVs in Mountain Search and Rescue* (150 pages). For the complete report please contact me at jinyup100@gmail.com.

1 Imaging and Detection System

1.1 Introduction

In this chapter, the SAR mission in a high altitude environment is viewed from the perspectives of imaging and detection system. Given the range of data that includes the images obtained from the cameras and the GPS positions and velocities of each drone, the imaging and detection system elicits useful information. Processed information serve as a powerful guide to address problems associated with the SAR mission, in particular, those related to scanning of search coverage area for potential human casualties and detection of those casualties.

1.2 Design Objectives

The main purpose is to design a robust imaging and detection system that guarantees a certain level of performance within a specific set of requirements. The idea is to design an imaging and detecting system that meets the **Environmental Specification** and the **User Requirements**, and operates compatibly with the **Hardware**, **Control**, and **Communications** systems.

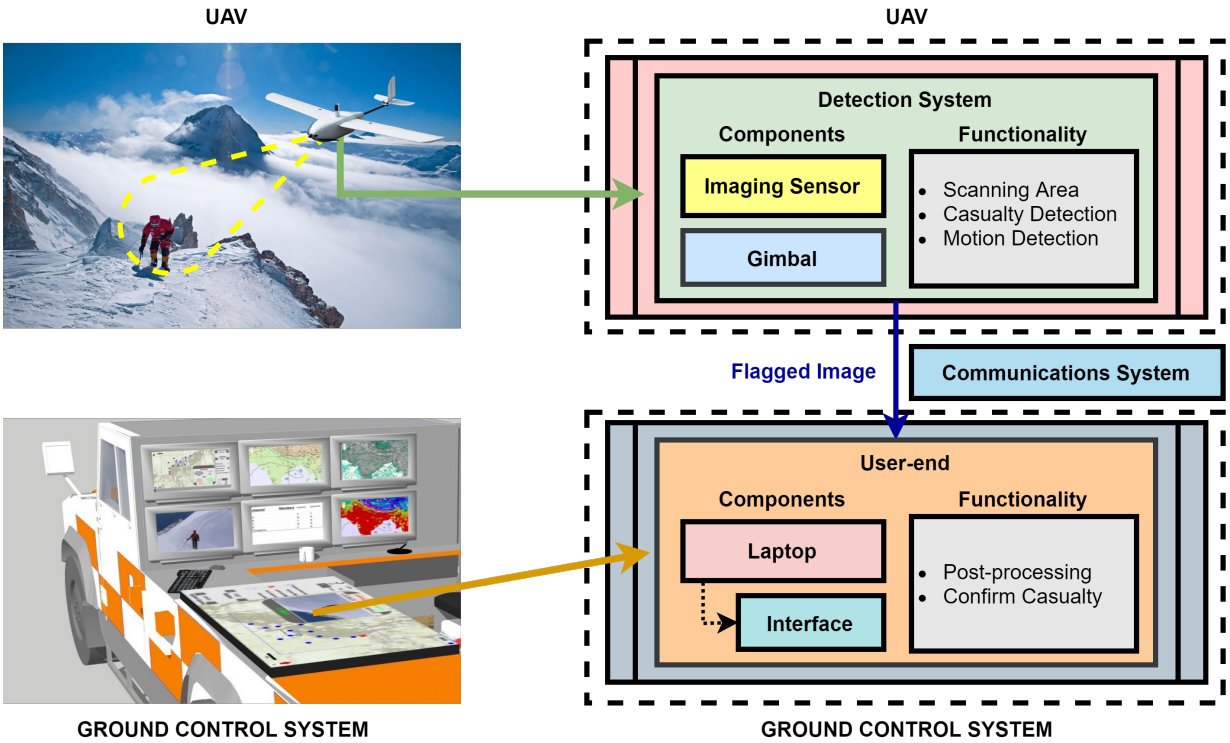

Figure 1: System Architecture

At in a higher level design, the interactions between the detection system and other systems are considered. For the hardware, the consideration of the payload means there are limits to the mass of

the detection system and the power required to run the system. For the purpose of communicating information, an image flagged by the detection system is required to be sent from the UAV to the *User Interface* at the *GCS*. This is to ensure that the *user* makes the safety-critical decision on the presence of human casualty. Taking the *user requirements* into account, we look for ways to make the process of making the decision easier and more convenient for the *user*. It is the decision made by the *user* that allows the control system to initiate the confirmed casualty protocol. The system architecture from the perspectives of the imaging and detection system is shown above in Figure 1.

At a lower level, the aim is to design a robust imaging and detection with the following capabilities:

1. A system with sufficient scanning area
2. A system capable of detecting a potential human casualty
3. A system capable of detecting any motion

A system with sufficient scanning area is desired in order to carry out the search mission within the specific search coverage area noted in the Environmental Specification Section. Different types of imaging sensors are evaluated in Section 1.3.1 to select an appropriate model, taking into consideration the objectives of both the search mission and the detection task. Human casualty detection is the single most important functionality of the imaging and detection model, which is the primary objective of any SAR mission.

The process of designing the human casualty detection system in Section 1.3 is divided into 3 parts. The first step encompasses pre-processing the sequence of image frames, the quality of which can be significantly compromised by severe weather conditions. The second step explores different feature extraction methods to elicit important features from the given image frames. The final step is to analyse the accuracy of various image classification, localisation and segmentation models in successfully flagging the presence of human casualty based on certain evaluation metrics.

Motion detection is a prominent capability that not only aids the detection of potential casualties, but also allows the *user* to determine the degree of injury of those casualties. State-of-the-art motion detection algorithms that take into account the movement of the imaging sensor are analysed in Section 1.4.

By considering both the higher level and the lower level design, a combined detection model is proposed in Section 1.5 as an effective way to improve the precision of the overall system as a whole, but also to successfully perform both casualty detection and motion detection.

Before the design and implementation of the imaging and detection system, it is necessary to formulate the general requirements for the overall system, in order to inform the design decisions specific to the imaging and detection system.

1.2.1 Requirements Formulation

The general requirements relevant to the imaging and detection system are derived from the environmental specification, the user requirements, and the other chapters of the report.

Flight speed of 22ms^{-1} has been set by the hardware of the system in the Hardware Section. The value will be used in Section 1.5 to determine the time scale that a target object stays within the field of view (FOV) of the imaging sensor.

The User Requirements Section specifies that there are limits to both the time that the operator can spend in processing information and the amount of information that the operator can process at any one time. In the worst case scenario, flagged images are transmitted from each of the 8 UAVs (see the Control Section) to the user-end interface. The requirement for the maximum number of flagged images transmitted from a single UAV has been set to 4 images per minute to allow sufficient time for the operator to verify each image. The precision of the finalised imaging and detection system in Section 1.5 has to be high enough such that the requirement is met.

In addition to the general requirements relevant to the imaging and detection system, specifications for the camera and performance specifications for the detection system are determined.

1.2.2 Camera Specification

1.2.2.1 Flight Height and Pixel-wise Size of Target Object

Flight height of 80m above ground level has been set not only to avoid potential obstacles such as trees, but also to ensure that a relatively close distance to objects on the ground is maintained. Indeed, the size of a target object in pixels relative to the size of image frame can significantly affect the precision of the detection system. In the context of our design, P. Dollar et al. [1] suggests that human casualties of size smaller than 80 pixels can lead to a drop in precision by more than 80%. Pixel-wise size of a target object has therefore been set to 120 pixels to minimise potential detection failures. The regulations regarding UAVs outlined by Civil Aviation Authority of Nepal, which restrict any UAV flight height to below 120m, have also been considered [2]. Since the same restriction on flight height is used in Europe [3], our system is operable outside the base environmental model.

1.2.2.2 Focal Length, Sensor Size and Image Resolution

The focal length of a lens is the distance between the camera lens and the sensor when the target area is in focus. Depending on the size of the focal length, camera lenses are categorised into *super telephoto*, *telephoto*, *normal*, *wide angle*, and *extreme wide angle*. For the task of scanning

the coverage area, wider angle of view and higher depth of field are preferred over magnification. Extreme wide angle of view is unsuitable, however, as it would lead to decreased pixel-wise size of a target object. Therefore, *wide angle* camera lens with the focal length in the range of 24mm and 36mm have been set as a design requirement.

The size of the sensor within a camera is an important factor as it determines how much light is used to generate each image frame. To be more specific, an imaging sensor array consists of millions of individual light-sensitive spots called photosites. Each photosite senses a small amount of light coming through the photographic lens and records the corresponding data. It is in this respect that large sensors effectively increase the image resolution of the cameras by holding a greater number of photosites. Relatedly, sensor size of 2/3" inches and image resolution of 5 megapixels (MPs) have been decided as minimum specifications for the camera.

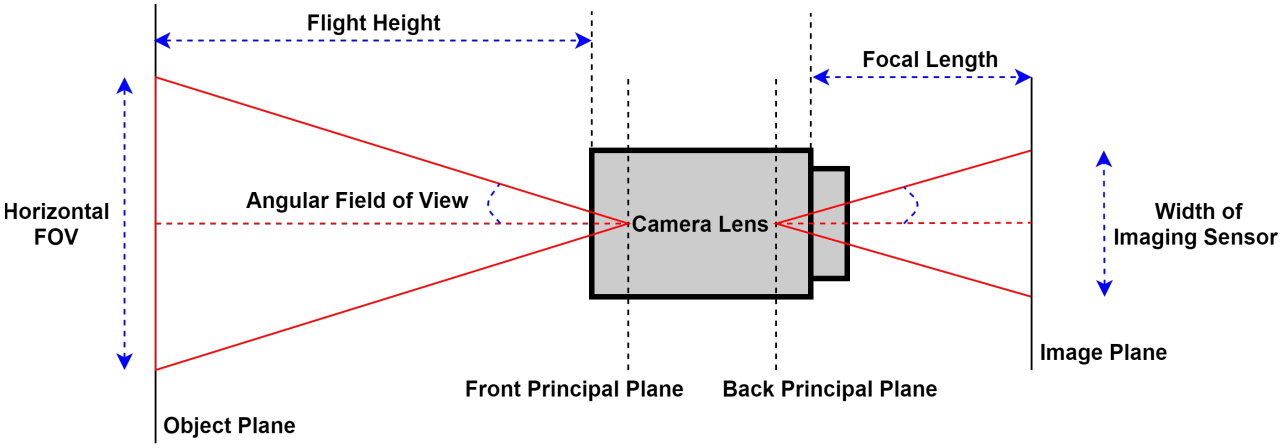

Figure 2: Relationship between FOV, Sensor Size, Focal Length, and Flight Height

Referring to Figure 2, the angular field of view (AFOV) can be used to derive the relationship between flight height and horizontal FOV, and focal length and width of sensor shown in Equation 1:

$$AFOV = 2 \times \arctan\left(\frac{Width\ of Sensor}{2 \times Focal\ Length}\right) = 2 \times \arctan\left(\frac{Horizontal\ FOV}{2 \times Flight\ Height}\right) \quad (1)$$

1.2.2.3 Horizontal Field of View and Target Area Width

Determined specifications for the size of a sensor and the range of focal length have been used to calculate appropriate range of AFOV. Combined with the value of flight height, the corresponding range of horizontal FOV has been calculated using Equation 1. That *Horizontal FOV* plays an important role in the search mission is also considered, as maximising the value would reduce search time. Based on these factors, the exact value of *target area width* is determined in Section 1.3.1.2 using the full specification of the camera.

1.2.2.4 Exposure Time

The exposure time of a camera is the length of time that the sensor is exposed to light from the surrounding environment. For the purpose of the imaging and detection system where a camera is attached to the UAV, it is important to have the highest *shutter speed*. A high specification is necessary in order to avoid potential motion blur due to the velocity of the UAV and as well as the turbulence resulting from changes in air pressure. Exposure time of less than 4000^{th} of a second has been set as a required specification. At the maintained flight speed, Fulton suggests the required specification would lead to a motion blur of less than 0.5 pixels [4].

1.2.2.5 Frame Rate

Frame rate refers to the number of image frames per second of a video that a camera captures . Ben Software suggests 6 FPS as the minimum requirement to be able to apply motion detection algorithms on the sequence of image frames [5]. A more rigorous requirment has been set for frame rate considering that a conventional real-time video has frame rate of 24 FPS [6]. The highest possible exceeding the real-time video frame rate is desired in order to sample as many images as possible, given the severe weather conditions of the base environmental model.

1.2.3 Evaluation Metrics for Detection System

In addition to camera specification, performance specification of the detection model is established by considering the general requirements in Section 1.2.1 as well as the mission statement.

		Predicted Labels		
		Positive (Human)	**Negative (No Human)**	
Actual Labels	**Positive (Human)**	True Positive (TP)	False Negative (FN)	**Recall** $\frac{TP}{TP+FN}$
	Negative (No Human)	False Positive (FP)	True Negative (TN)	**Specificity** $\frac{TN}{TN+FP}$
		Precision $\frac{TP}{TP+FP}$	**Negative Prediction** $\frac{TN}{TN+FN}$	**Accuracy** $\frac{TP+TN}{TP+TN+FP+FN}$

Figure 3: Metrics

Referring to Figure 3, it is of the highest priority to have high *recall* for the detection system. This is because we do not want a case where it fails to recognise a human casualty when there is indeed a casualty. Such case would defeat the whole purpose of having the detection system and thus we have set a high performance specification of 82% for *recall*. The value is also set recognising the performance of the state-of-the-art large FOV classifier that has recorded 82.3% recall [7].

Indeed, a higher number of *false positives* relative to *false negatives* is expected in a mountainous environment due to the presence of rocks, trees, and animals that can lead to potential misclassification. A high false positive rate can significantly compromise the overall system as it is not possible to downlink all images that are deemed to contain a casualty to the user at all times. There is also a limit on the number of images that the operator at GCS can process, which has been emphasized in Section 1.2.1. However, given a sequence of image frames within a short time interval, it is probable that the consecutive image frames are highly correlated and that the pixels contributing to the decision of the detection system are repeated. That is, the detection model that has been tested on an image dataset can significantly improve its *precision* on a video dataset by recognising such correlation and repetition. The improvement can certainly be achieved by placing a threshold on the number of flags before flagging an image as deeming to contain a human casualty. Since the threshold can lower the false positive ratio of the overall detection system, a relatively lower standard of 78% is set for *precision*. An apparent trade-off between *recall* and *precision* has also been considered. As with any other detection model, a degree of *accuracy* also has to be guaranteed. Accuracy of 80% has been set by taking the average of *precision* and *recall*.

Having established the evaluation metrics for the performance, it is equally important to set specification for computation time and memory usage required to perform detection. It is necessary for the computation time taken to classify one image frame to be less than the frame rate specified in Section 1.2.2.5 to prevent any delays and image frames being stacked in the memory. In other words, by specifying computation time to be less than the frame rate, the detection task can be performed in real time. Total memory usage during the computation should be within the standards of commercially available on-board CPUs such as Intel's GA-IMB models [8]. With reference to the HCI section, the laptop at the user-end provides more computational power and allows the detection system to be pre-trained at GCS. It is such pre-trained detection system that can be used for the SAR mission.

Various implementations of the detection model are tested and validated against image dataset based on the metrics summarised in Table 1. Having established the camera requirements and the performance metrics, it is now possible to design and implement the casualty detection system.

Requirement	Measure	Description
Accuracy	$> 80\%$	Overall accuracy of the system
Recall	$> 82\%$	Higher specification for the number of false negatives
Precision	$> 78\%$	Lower specification for the number of false positives
Computation Time	< 0.03 s	Time required to process one image frame
Memory Usage	< 5GB	Memory usage required to process one image frame

Table 1: Performance Specification for Detection System

1.3 Human Casualty Detection

The process of designing a human casualty detection system is divided into 3 main steps. Firstly, different types of imaging sensor such as standard RGB cameras, infra-red thermal cameras, and lidars are considered. Having decided on the type of imaging sensor, multi-criteria analysis is used to select the sensor best suited for the mission. Secondly, pre-processing methods are analysed in order to remove any noise added to the image frames due to adverse weather conditions. Thirdly, various implementations of classifier are explored with consideration of feature extraction methods for the models requiring distinct set of features from an image. Different stages of design implementation are shown in Figure 4.

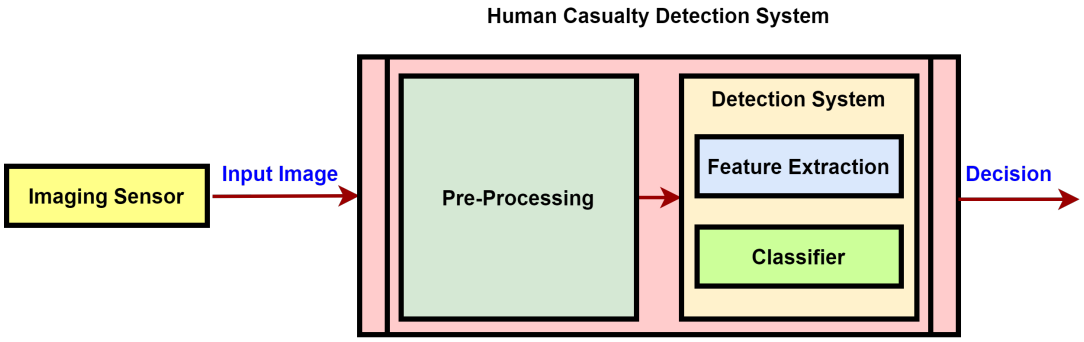

Figure 4: Human Casualty Detection System

1.3.1 Choice of Imaging Sensor

1.3.1.1 Types of Imaging Sensor

Thermal Infra-Red Camera

A thermal infra-red (IR) camera is an imaging system that captures infra-red radiation, which has a wavelength in the range of $1\mu m$ and $14\mu m$ It not only measures a temperature distribution across an image frame, but also provides an accurate temperature of an object. As suggested by Setjo et al., [9], temperature variation and particular temperature of a human casualty can be used to perform accurate casualty detection. The viability of thermal imaging in adverse weather is emphasized in [10] by demonstrating a precise classification in rainfall conditions.

Application of thermal IR camera in a low temperature environment, however, is limited. The performance of any thermal imaging system is significantly compromised by low emissivity, which is the ratio between the infra-red radiation of a target object and that of a perfect radiator. Considering the fact that casualties would be wearing thick layers of clothing to prevent a drop in body temperature, it is highly unlikely that the temperature of a human casualty would be captured by a thermal camera. It is also improbable that radiation emitted from small gaps in clothing would provide sufficient features of a human casualty for robust detection.

Lidar

A Light Detection and Ranging (Lidar) is a type of remote sensor that collects information by measuring the amount of energy that is reflected from the ground. It uses illumination in the form of a pulsed laser and records the reflected light to measure range of distances. Combined with the position and orientation of the UAV, the range of distances can be used to generate a group of elevations called a point cloud. The construction of point cloud is useful not only in generating a highly accurate map of the surroundings of a UAV, but also in detecting a human casualty [11].

The performance of lidar, on the other hand, is vulnerable to severe weather conditions. That is, the configuration of point cloud is difficult in rainfall conditions due to a distortion of data forming the point cloud. Rain droplets and snow partially reflect the emitted pulsed laser back to lidar and this causes added noise known as echo and partial gaps to the data [12]. Kutila et al. [13] further emphasizes that adverse weather conditions can result in up to 50% reduction in target detection performance.

RGB Camera

A standard RGB camera is equipped with a complementary metal oxide semiconductor (CMOS) sensor through which visible red, green, and blue (RGB) light are combined to produce an image frame. A key advantage of a RGB camera is the provision of images with high resolution. Whereas even the best thermal cameras provided by FLIR Systems have a resolution of 0.3 megapixels [14], a conventional high definition RGB camera has a resolution greater than 8 megapixels. In addition to high resolution, it is expected that a visible camera would provide sufficient information to recognise a human casualty. The notion is based on the fact that the figure and clothing of a casualty would be well differentiated from the environment. It is in this light that numerous research have been contributed to pedestrian detection based on images processed by a visible camera [15].

Research efforts on different types of imaging sensors show that either RGB cameras or its combination with thermal IR cameras are best suited to the purpose of the mission. While a standard RGB camera would provide high resolution images, a thermal IR camera would supply an alternative set of images in adverse environmental conditions. In practice, many of the commercial UAVs manufactured by FLIR Systems or DJI accomodate either a high definition RGB camera alone or a camera capable of supplying both visible and thermal images, however, at the cost of lower resolution. Considering the low temperature of the base environment which significantly affects the performance of thermal IR cameras, a high performing RGB camera has been selected over a camera capable of both thermal and visible light imaging. In order to account for adverse weather conditions, pre-processing methods specific to RGB images are explored in Section 1.3.3.

1.3.1.2 Multi-Criteria Analysis

Following the decision on the type of imaging sensor to be used, multi-criteria analysis (MCA) is used to select the commercially available RGB camera model best suited for detection. The decision factors outlined in Section 1.2.1 are assigned higher values of weight in the analysis as they represent the necessary requirements. Additional factors such as *mass*, *operating temperature* and *power* are used for the evaluation. *Mass* and *power* are important factors for the total payload of the UAV underscored in the Hardware Section and *operating temperature* is a prominent consideration given the harsh environmental specification of Annapurna I. However, these additional factors are assigned relatively lower values of weight because the total payloads for mass and power have been met in the Hardware Section with sufficient power provided by the battery and the presence of heating mechanism. The multi-criteria analysis on potential imaging sensors is shown by Table 2.

Sensor	Exposure	Sensor Size	Resolution	Frame Rate	Mass	Power	Temperature	Score
GoPro Hero8	1/125 s	1/2.3”	8.3 MPs	24 FPS	0.13 kg	10W	$0-50°C$	64
L1D-20C	1/8000 s	1”	8.3 MPs	30 FPS	0.15kg	10W	$-10-40°C$	78
LM11059	1/8000 s	1”	0.3 MPs	5 FPS	0.80 kg	7.2W	$0-50°C$	40
Nex-7	1/4000 s	1”	1.5 MPs	10 FPS	0.30 kg	7.2W	$0-40°C$	45
FLIR Oryx	1/4000 s	1/1.8”	8.8 MPs	60 FPS	0.50 kg	12.3W	$0-70°C$	82
Zenmuse X7	1/8000 s	Super 35	8.8 MPs	30 FPS	0.18 kg	10W	$0-70°C$	**86**
Function	-5log(E)	Sensor Size	2 Res	1/2 FR	M	P	—min(T)—	

Table 2: Multi-Criteria Analysis

The multi-criteria analysis shows that *Zenmuse X7* best matches our desired specification and as such it has been chosen as the imaging sensor for the system. The full specification of *Zenmuse X7* in the 3rd column of Table 3 satisfies the requirements for the sensor derived in Section 1.2.1 summarised in the 2nd column. Moreover, the target area width and height of 60.8m and 45.6m are calculated each respectively using the finalised value of flight height, focal length and sensor size. The design is to attach the camera to the UAV using the *X7* gimbal, which also allows the imaging sensor to pan from -300° to +300° and roll from -20° to +20°. Such capabilities of the gimbal are desired to account for the changes in flight altitude due to pathing.

Requirement	Measure	Actual	Description
Flight Height	$= 80$m	80m	Maintained flight height above ground
Target Size	> 120 pixels	153 pixels	Pixel-wise target object size at ground
Focal Length	24mm - 36mm	35mm	Distance between the lens and imaging sensor
Sensor Size	$> 2/3” \equiv 8.8$mm	23.5mm	Width of imaging sensor
Image Resolution	> 5 MPs	8.8 MPs	Number of pixels within an image frame
Horizontal FOV	19.5 - 29.3 °	28.7	Horizontal FOV of the Camera
Exposure time	$< 1/4000$ s	1/8000s	The length of time sensor is exposed to light
Frame Rate	> 24 FPS	30 FPS	The rate at which images appear on display

Table 3: Requirements and Actual Specification for the Imaging Sensor

1.3.2 Dataset

It is necessary to create an image dataset in order to train, test and validate various implementations of the detection model. There are publicly available benchmark datasets for pedestrian detection, such as the *Caltech Pedestrian Dataset*, and *Penn-Fudan Database*. However, given that these datasets consist of images in an urban environment, it is uncertain as to how well the detection models trained over these datasets would perform over images of a mountainous environment.

Taking this into consideration, a new image dataset has been created using the application programming interface (API) provided by *Google* and *Bing*. Based on search queries such as *Annapurna* and *Annapurna Climber*, 750 images containing a human in a high altitude environment, and 750 images purely containing the mountainous environment have been collected. The dataset is then split into training, testing, and validating dataset according to the ratio 8:1:1.

There are various problems associated with the collected images. The APIs do not always provide images relevant to the search queries. The images relevant to the search queries vary in size and resolution and the distance between the camera and the casualty varies according to images. In order to address these issues, images have been deleted and cropped where necessary. All image frames were resized to 1000 by 1000 to not only ensure a degree of consistency but also to reduce redundancy.

In addition to a new image dataset, several UAV footages of a mountainous environment have been collected from *Youtube* using relevant search queries. Video datasets have been made because the ultimate aim is to create a detection model that performs robustly over streams of images taken by the imaging sensor of the UAV. Details of image and video datasets are shown in Table 4.

Image Dataset				
Name	Number of Images	Training	Testing	Validating
Image Dataset 1	1500	1100	200	200
Video Dataset				
Name	Number of Images	Length	Frame Rate	Labels
Video Dataset 1	6000	6 mins 40 s	24	Combined
Video Dataset 2	3000	3 mins 18 s	24	Human Only
Video Dataset 3	3000	3 mins 43 s	24	No Human

Table 4: Dataset

Referring to Table 4, video datasets 2 and 3, each respectively containing human only and the environment only, have been collected as an alternative way to measure the false negative rate and the false positive rate and thereby evaluate the performance of the detection model. The performance of the finalised detection model has been tested against video dataset 1, which consists of image frames containing all labels.

1.3.3 Pre-Processing Techniques

1.3.3.1 Median Filter

A median filter is a type of non-linear filter that is used to reduce random noise, especially when the noise probability amplitude density has large tails [16]. The process is achieved by sliding a pre-determined size of kernel across the image and taking the median of all the pixel values at the location of the centre of the kernel. As an edge is crossed by the kernel, the sharp contrast on each side of the border leads to sharp contrast between the edge and the neighbouring pixels. While median filters are effective in maintaining important features of an image, they are also computationally efficient means to remove any long-tailed noise in an image such as snow.

1.3.3.2 Adaptive Median Filter

Research on adaptive median filter considers various aspects of median filter that can be improved. The centre weighted median (CWM) filter is a weighted median filter assigning more weight only to the central value of each kernel [17]. The ranked-order based adaptive median filter (RAMF) is based on a test for the presence of noise in the central pixel itself [18]. The process is then followed by the test for the presence of any residual noise in the output of a median filter to ensure a relatively wider distribution of noise are removed. The impulse size based adaptive median filter (SAMF) is a median filter that varies the size of the kernel by recognising the size of the impulse noise. The performance of RAMF has been verified on sample test images in the 3rd column of Figure 5.

1.3.3.3 Multi-guided Filter

Guided filter is a type of filter that performs edge-preserving image enhancement of an input image using the content of a second image called a guidance image. Recognising the difference between background edges and rain or snow streaks in terms of brightness and bluriness, Zheng et al. [19] decomposes an input image into low frequency and high frequency part each respectively in terms of pixel intensity. By using the decomposed high frequency and low frequency parts of an image as guided filters, impulse noise due to snow and rain streaks are removed. The method can then be combined with image enhancement techniques to give a final output in Figure 5.

1.3.3.4 Evaluation

Figure 5 shows the applications of median filter, adaptive median filter and multi-guided filter on test images. Results show that RAMF outperforms the remaining filters. The final decision on the choice of pre-processing method for the overall imaging and detection system is discussed in Section 1.5.

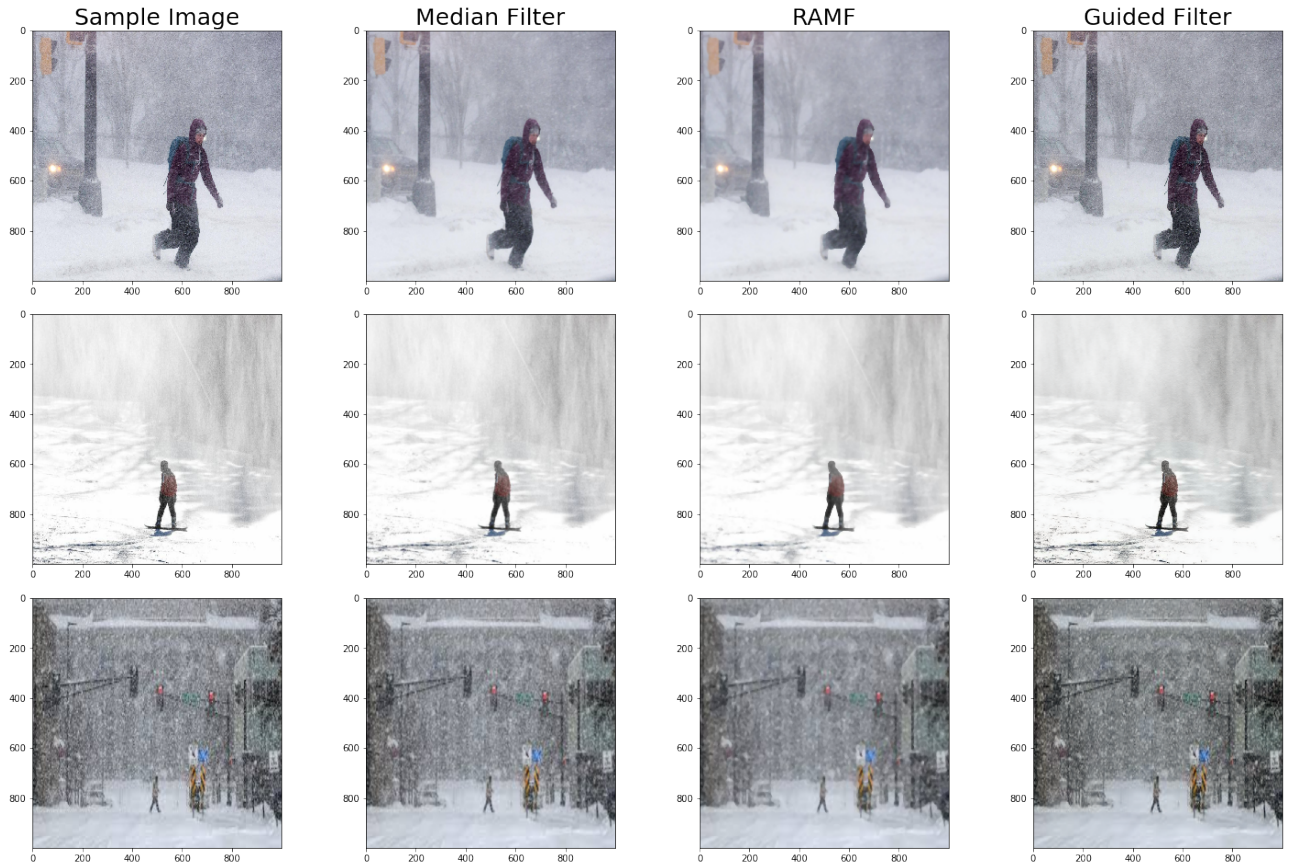

Figure 5: Performance of Pre-processing Methods on Sample Images

1.3.4 Types of Casualty Detection Models

Applications of computer vision in the domain of images include classification, localisation, and segmentation. Image classification is a relatively simple task of classifying whether or not a given image frame contains a human casualty. Image localisation is a task of classifying and identifying the region where a human casualty is deemed to exist. Localisation is done by means of creating a bounding box around such region. Image segmentation is a relatively meticulous task at a microscopic level that goes beyond classification and localisation to identify pixels that belong to that of a casualty.

In this chapter, various human casualty classifiers are implemented and tested. Based on the performance of the image classification model over the validating dataset, the best performing model is selected. Then, we use the selected model to perform image localisation by generating a bounding box in the region that is considered to contain a casualty. The model is tested against the video datasets to calculate the false positive and the false negative rates.

The implementation of an image segmentation model is a difficult task as we are constrained by a number of limitations. The dataset created in Section 1.3.2 is not labelled on a pixel level and the training of the detection model at a pixel level requires a high performing GPU. Taking the limitations into account, publicly available pre-trained models have been used to carry out segmentation.

1.3.5 Feature Extraction

An image frame contains repeating and redundant information. For instance, to classify whether there exists a casualty within a given image frame, we do not need to process every pixel to make the decision. As such, feature extraction methods can be used to elicit regions of interest (ROI) from each image for those classification models requiring such extraction methods. In this section, features of colour and histogram of oriented-gradients (HOG) are extracted from each image frame.

Colour features are extracted based on the notion that the clothing of a casualty would stand out in the base environment model where the vast majority of the background consists of black and white, due the great majority of the environment being composed of snow and rock. The histogram of oriented-gradients (HOG) is a feature descriptor that is commonly used for the purpose of object detection. With reference to [20], the feature descriptor performs efficiently in eliciting human features from an image frame and as shown in [21], it is commonly used for pedestrian detection.

1.3.5.1 Colour Extraction

All the available colour spaces including *RGB*, *HSV*, *LUV*, *HLS*, *YUV* and *YCrCB* have been tested to find the optimal space for feature extraction. For each colour space, features of an image frame are extracted by using a colour-based histogram. The bins have been set to range from 0 to 256 to reflect the scope of channel magnitude of an individual pixel and the number of bins to equal 64 for each of the 3 channels. Histogram for the extracted colour features are shown by Figure 7.

Figure 6: Image

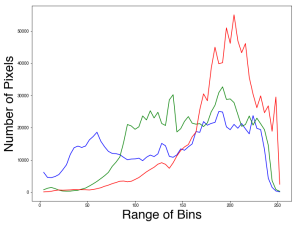

Figure 7: Histogram

Figure 8: Mask

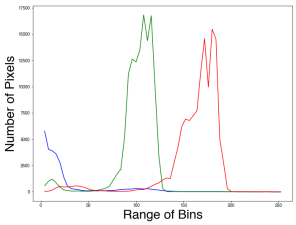

Figure 9: Histogram

As an attempt to extract prominent colour features of the pixels, an image mask is created by setting the threshold of the sum of the pixel magnitude of the three channels in the interval of 10 and 30 and by iterating dilation and erosion to minimise the unneeded areas. The process had been carried out based on the idea that the extraction of colour features from the segmented images would lead to enhanced performance from the classification models. However, few preliminary tests have shown that classifiers result in higher level of performance when trained with colour features from the whole spectrum. Having said that, the method involving an image mask is further researched in the implementation of R-CNN.

1.3.5.2 Histogram of Oriented Gradients (HOG) Extraction

HOG is a feature descriptor that focuses on the structure of an image frame and the shapes within a frame. That is, HOG determines whether or not a pixel is an edge, and further improves upon edge detection algorithms such as Sobel and Canny edge detecting algorithms by extracting gradients and orientations. For a designated colour channel, the gradient of each pixel is calculated by calculating the differences across the neighbouring pixels in the x-direction and y-direction, denoted by G_x and G_y each respectively. Then, the gradient magnitude $|\mathbf{G}|$ and orientation ϕ are calculated by:

$$|\mathbf{G}| = \sqrt{G_x^2 + G_y^2} \tag{2}$$

$$\phi = \arctan \frac{G_y}{G_x} \tag{3}$$

The next step is to elicit gradient magnitude and orientation from local regions of an image frame. This is done by dividing the image frame into cells. Over all the individual pixels in each spatial region defined by a cell, a local 1-D histogram of gradient magnitude and orientation is calculated. Each histogram assigns the weighted value of the orientations into bins, where the gradient magnitude is used as weights instead of frequency. Although the HOG features are obtained from the cells of the image, the gradients of the image obtained in such way are sensitive to illumination. Such variation in illumination can be reduced by normalising the gradients with respect to a larger spatial region defined by a block. The extracted HOG features are shown by Figure 11 and the strongest 50 features are shown by Figure 12.

Figure 10: Sample Image

Figure 11: HOG

Figure 12: Extraction

1.3.6 Classification Models

1.3.6.1 k-Nearest Neighbour (k-NN) Classification Model

A k-NN algorithm is a supervised learning algorithm that stores all the features of the training image dataset and the corresponding labels indicating the presence of humans in order to classify the testing image. That is, the k-NN classifier compares the features of the testing image to the features of each and every image stored in the training dataset. In the case where $k = 1$, the output is simply the label of the training image containing features that are most similar to those of the testing image.

The measure of proximity of one set of features to another is evaluated using vector p-norm. Given the two images with corresponding feature vectors F_1 and F_2, the L1 and L2 norm are shown by Equations 4 and 5 each respectively.

$$d_1(F_1, F_2) = \sum_{i=1}^{M} |F_1^i - F_2^i| \quad (4) \qquad d_2(F_1, F_2) = \sqrt{\sum_{i=1}^{M} |F_1^i - F_2^i|^2} \quad (5)$$

In a more general case where $k > 1$, the output label is computed by taking the vote from each of the k near neighbours, using the corresponding values obtained from Equations 3 or 4 as the relative weight of each neighbour. Whilst the k-NN classifier requires 0.00527 seconds and 1332 Mbyte to classify one testing image, the computational efficiency is heavily compromised by its performance - giving 59.5% accuracy, 98% precision, and 55.4% recall.

1.3.6.2 Logistic Regression Model

As noted by Bishop in [22], ways to address a general classification problem can be broken down into two steps. An *inference* step is one in which training data is used to learn a model for $p(\mathcal{C}_k|\mathbf{x})$ and a *decision* step is one in which we use the obtained posterior probabilities to optimally assign classes. Bishop describes 3 distinct approaches to solving the problem. A *generative* model solves the inference problem by determining the class conditional densities $p(\mathbf{x}|\mathcal{C}_k)$. The model then uses Bayes' theorem to find the posterior class probabilities $p(\mathbf{x}|\mathcal{C}_k)$. A *discriminative* model tackles the inference problem by directly determining the posterior class probabilities $p(\mathcal{C}_k|\mathbf{x})$ and thereby directly assigning class probability to $\mathbf{x}$. The final way is to directly model $y(\mathbf{x})$ called a discriminant function, which takes an input vector $\mathbf{x}$ and assigns it to one of K classes as shown by Equation 6.

$$y_k(\mathbf{x}) = f(\mathbf{w}_k^T\mathbf{x} + w_{k0}) \quad (6) \qquad \text{where } \mathbf{w}_k \text{ is a weight vector, } w_{k0} \text{ is a bias vector}$$

An input vector $\mathbf{x}$ is assigned to class $\mathcal{C}_i$ if $y_i(\mathbf{x}) > y_j(\mathbf{x})$ for all $i \neq j$, with the decision boundaries between each class formed by $y_i(\mathbf{x}) = y_j(\mathbf{x})$. In the case of the human casualty classification, an input vector $\mathbf{x}$ is a combination of HOG and colour features and the problem reduces to a binary classification where we only need to classify whether or not a given image frame contains a human casualty. For such case where $K = 2$, the posterior probability for class $\mathcal{C}_1$ can be written in the form of Equation 6 using Bayes' theorem.

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \qquad \text{where } e^{-a} = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$
$$= \frac{e^a}{e^a + 1} = \frac{1}{1 + e^{-a}} = \sigma(a) \quad (7)$$

Therefore, the posterior probability of class $\mathcal{C}_1$ can be written as a logistic sigmoid function σ, acting on a linear function of the feature vector $\mathbf{x}$. Such model is known as a logistic classification model. Then, the maximum likelihood estimators are used to determine the parameters of the model. Equation 8 shows the derivative of the logistic sigmoid function, from which we derive the log-likelihood function. For an image dataset consisting of N images each with the corresponding features labels t_n, the likelihood function is shown by Equation 8.

$$\frac{d\sigma}{da} = \frac{d}{da}\left(\frac{1}{1+e^{-a}}\right) = \sigma(1-\sigma) \tag{8}$$

$$p(\mathbf{t}|\mathbf{w}) = \prod_{i=1}^{N} y_i^{t_i}(1-y_i^{t_i}) \tag{9}$$

Using the property of a logarithmic function that it is a strictly increasing function, the cross entropy loss function is obtained by taking the negative of the logarithm of Equation 9.

$$e(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{i=1}^{N} t_i \ln y_i (1 - t_i \ln y_i) \tag{10}$$

The aim is to maximize the posterior probability and hence minimize the loss function. This is essentially an optimisation problem with various possible approaches. Referring to [23], a stochastic average gradient (SAG) method can be used to optimize the sum of smooth convex functions in Equation 10. [24] proposes yet another incremental gradient (SAGA) method by taking a more generalised variance reduction approach. As demonstrated in [25], limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method is an optimisation method that solves an unconstrained optimisation problem by applying quasi-Newton algorithm. The iterative algorithm shown in [26] makes full use of Newton-conjugate gradient procedures to solve the problem.

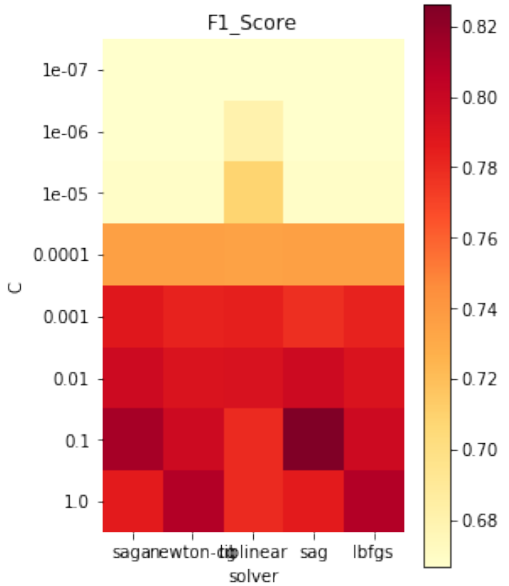

Figure 13: Optimisation Method

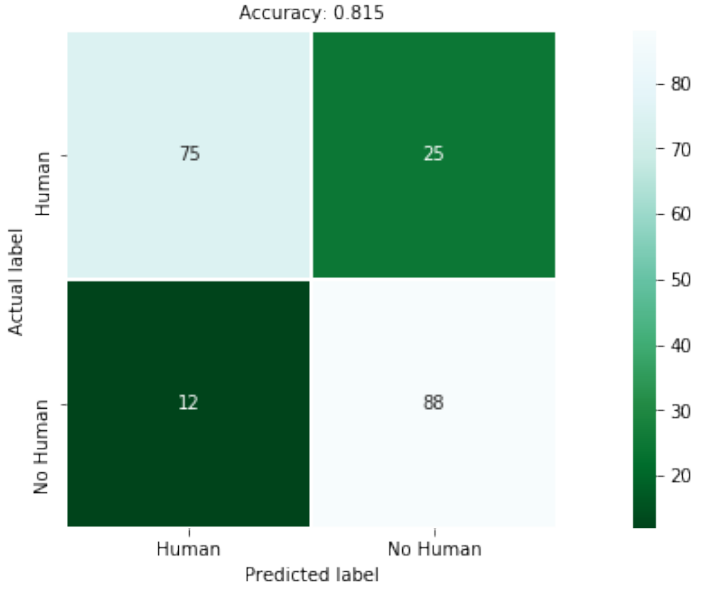

Figure 14: Confusion Matrix

The `scikit-learn` library has been used to implement the logistic classifier and the `parfit` library has been used to find the highest performing optimization method that gives the highest F1- score. Figure 13 shows the efforts that have been made to find the logistic classifier with the appropriate hyperparameters and the most efficient algorithm to solve Equation 9. The SAG method has proved to the most accurate, with the corresponding logistic classifier resulting in 81.5% accuracy, 88% precision, and 77.9% recall as shown by Figure 14. The logistic classifier requires 0.00005 seconds and 4813 Mbytes of memory to carry out the task.

1.3.6.3 Support Vector Machine (SVM)

Having seen that the posterior probability of class $\mathcal{C}_1$ can be written as a logistic sigmoid function, a hinge loss function and a logarithmic loss function can be defined as shown in Equations 11 and 12.

$$L_2(p(\mathcal{C}_1|\mathbf{x})) = \max(0, 1-a) \qquad (11) \qquad L_1(p(\mathcal{C}_1|\mathbf{x})) = -\ln\left(\frac{1}{1+e^{-a}}\right) = \ln(1+e^{-a}) \qquad (12)$$

Using the hinge loss function introduced in Equation 11 and knowing that a represents a scoring function that measures the log ratio of the probabilities for each class, the loss function that we want to minimise becomes:

$$L = \sum_{i=1}^{N} \max(0, 1 - t_i(w_i^T x_i)) \qquad (13) \qquad L = \sum_{i=1}^{N} \max(0, 1 - t_i(w_i^T x_i)) + \lambda \sum_{j=1}^{N}\sum_{k=1}^{N} |w_{j,k}|^2 \qquad (14)$$

The problem with the loss function presented in Equation 13 is that although a certain set of parameters $\mathbf{W}$ may correctly classify every example such that $L = 0$ and hence all the margins are minimised, this particular set of parameters $\mathbf{W}$ may not be necessarily unique. For instance, assuming that a certain set of parameters $\mathbf{W}_c$ correctly classify all the training dataset, all the multiples of the set $\lambda\mathbf{W}_c$ would also correctly classify the training dataset. This is because the linear transformation λ uniformly stretches all the parameters, also giving $L = 0$.

Therefore, there is a need to make a decision on the preference of certain set of weights over other sets. This can be achieved by introducing a regularisation loss function to the original loss function as shown in Equation 14. The regularisation loss function in the form of vector-2 norm essentially shows preference for lighter weights over larger weights. The hyperparameter λ essentially determines the degree of margin between the closest set of data to the decision boundary.

Equation 14 is a primal form of the constrained optimisation problem from which the Lagrangian dual form can be derived. Referring to [27], the minimisation problem in Equation ?? can be transformed into a maximisation problem using a duality principle. The dual form of the optimisation prob-

lem is one that can be solved using both linear and quadratic programming [28]. The implemented SVM has resulted in 78.0% accuracy, 78.0% precision, and 78.0% recall, requiring 0.0001 seconds and 3632 MBytes of memory.

1.3.6.4 Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a type of neural network that consists of neurons, the corresponding weights and biases of which can be trained. It is composed of an input layer, an output layer and multiple hidden layers, different combinations of which have led to various research findings. Whilst InceptionNet proposed in [29] consists of 22 layer deep CNN with over 60 million parameters, VGGNet proposed in [30] consists of 16 convolutional layers requiring computation of 138 million parameters.

As such various implementations of CNN architecture are possible. The problem with the complex architectures proposed above is that they require high computational power to an extent where over four Nvidia GPUs are having to be used over three weeks. It is in this light that we have decided to design an architecture of CNN that can be trained on the laptop at the user-end. In this way, CNN can be continually trained with images directly obtained from the environment. The images used for calculation have been resized to 150 x 150 to account for the pixel-wise target object size in Section 1.2.2.1.

For the design, *tensorboard* has been used to explore the performances of different CNN architecture with varying depth of layers. That is, the performances of different architectures with a maximum of 7 convolutional layers have been recorded and tested. Figure 15 shows the architecture of the top performing CNN based on validation accuracy.

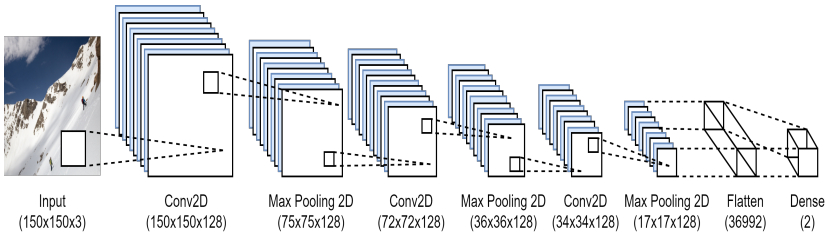

Figure 15: Architecture

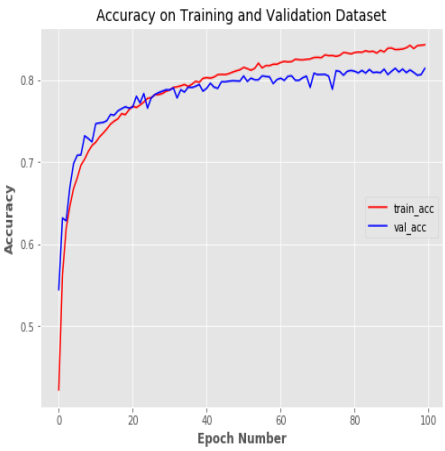

Figure 16: Accuracy

Figure 16 shows the performance of the optimal architecture of CNN that has given the highest validation accuracy of 0.9149. Then, the performance of the trained CNN has been evaluated against the testing dataset. The final performance of the trained CNN has given 83.2% accuracy, 88.3% precision, and 80.2% recall, requiring 0.002 seconds and 1940 MBytes of memory.

1.3.6.5 Classification Evaluation on Image Dataset

Table 5 summarizes the performances of all the classification models that have been explored thus far based on accuracy, precision, recall, memory usage and computation time. From the table, it can be seen that the performance of CNN outperforms that of any other classifier within a reasonable level of memory and time required for computation. As a result, CNN has been selected as the model most fitting for the task of human casualty classification.

Model	Accuracy	Precision	Recall	Memory	Time
k-NN	59.5%	98.0%	55.4%	1332 MBytes	0.00527 s
Logistic Regression	81.5%	88.0%	77.9%	4813 MBytes	0.00005 s
SVM	78.0%	78.0%	78.0%	3632 MBytes	0.0001 s
CNN	83.2%	88.3%	80.2%	1940 MBytes	0.0001 s

Table 5: Classification Evaluation

An aspect of CNN that differs from other classification models is that CNN does not require manual feature extraction, such as those based on colour and HOG. That is, the convolutional layer of the CNN extracts the feature map from the pixels of the original image, which are eventually used by the fully connected layer to carry out the classification task. Relatedly, there has been a growing interest in the interpretability of CNNs. It is indeed reasonable to be able to explain the decisions that have been made by the classification model and critical to know that the model is performing efficiently and accurately in such safety-critical applications such as the SAR. This is considering perspectives of both the user-end as well as the 3rd party willing to use our product.

To recognise the importance of the explainability of the system, we have decided to take into account the explainability of the classification model. The LIME technique introduced in [31] uses local exploration of neighbouring pixels to find an interpre representation understandable to humans. Such interpretable representation is a binary vector indicating the presence or absence of contiguous patch of similar pixels called *super-pixels*. It is via the sampling of these *super-pixels* that the author identifies the regions of an image contributing towards the decision of the classifier. Figure 17 shows all the images that have been accurately labelled by the CNN, and the first and the last images in Figure 18 show that the pixels belonging to those of human casualty have contributed to making a correct decision.

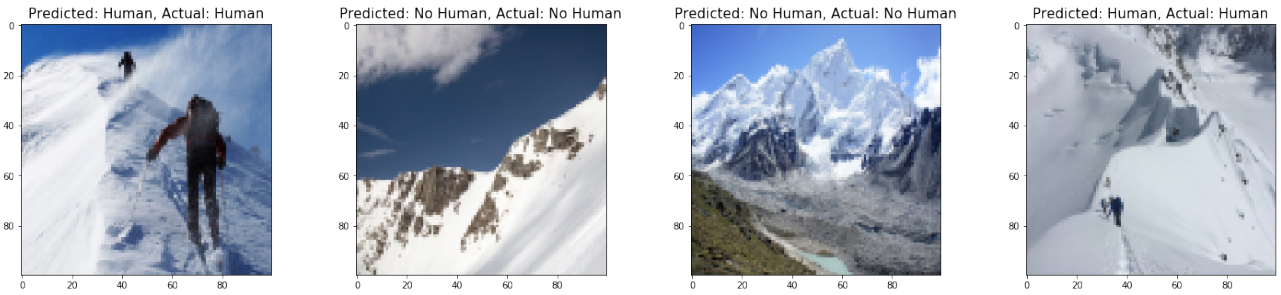

Figure 17: Correctly Labelled Images

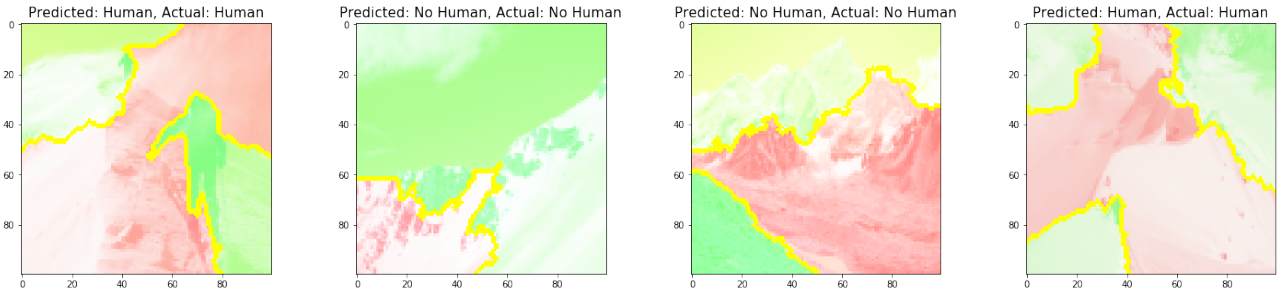

Figure 18: Explainability using LIME

1.3.7 Localisation and Segmentation Models

1.3.7.1 Localisation based on implemented Convolutional Neural Network (CNN)

One way to implement a localisation model is by using the trained CNN and taking a sliding window approach across the given image frame. An image frame can be divided into smaller image frames, using pre-determined window sizes. In this particular task, all the image frames obtained from the video dataset have been resized from 4096x2140 to 1000x1000 in order to reduce the required computation time. The window sizes of [140, 150, 160] have been used based on the fact that the CNN was trained with image size of 150x150 recognising the target object size is 153 pixels as defined in Section 1.3.6.4. Each window-sized segments of a 1000x1000 image frame are then used as inputs to the CNN for classification. For the segments that are deemed to contain a human casualty, the corresponding window size and the position within an image frame are stored. In the case where multiple windows are stored, a threshold is set in a form of heatmap to elicit the region that is most likely to contain a human casualty. The steps taken are summarised in Figure 19 and the application on the sample testing images in video dataset 1 are shown in Figure 20.

There is, however, a problem with the model. The problem relates to the pre-determined window sizes, the optimal values of which can vary according to the environment. That the window sizes are pre-determined also means that the detection model can be sensitive to the distance between the UAV and the human casualty. The performance would therefore depend on the ability of the UAV to maintain cruise height.

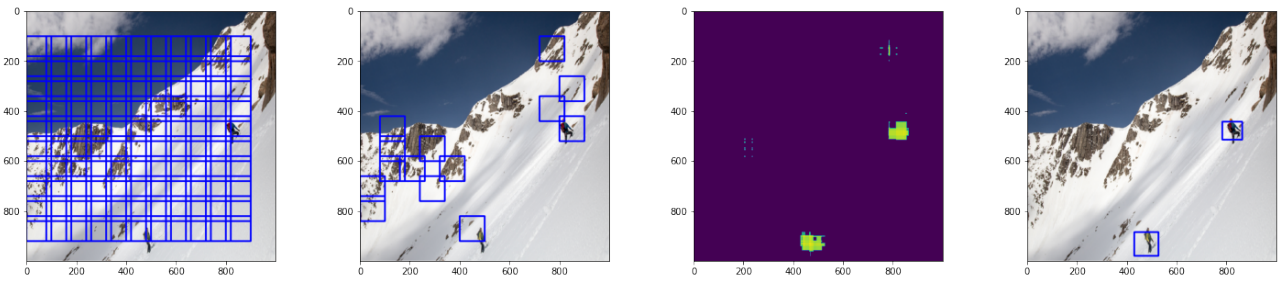

Figure 19: From Classification to Localisation

1.3.7.2 Region-based Convolutional Neural Networks (R-CNNs)

Instead of manually taking sliding window approach, **RCNN** proposes the application of selective search for generating class-independent region proposals. A Faster R-CNN proposed in **FRCNN** makes use of CNN to extract features from the proposed regions and perform object detection. The task is therefore essentially divided into a combination of a region proposal task and a classification task. The regional proposal task reduces the dimensions of the original image frame into computationally scalable regions, and the classification task evaluates the state of the object by creating a bounding box.

Relatedly, the proposed architecture of the model involves the separate usage of the two neural networks that form the basis for region proposal and classification tasks. For the region proposal task, a ResNet-18 model that is trained on ImageNet is required. As noted in [32], a residual network is essentially a deep convolutional neural network with 18 layers with residual mapping performed by shortcut connections and element-wise addition. The notion is based on the fact that it is computationally efficient and accurate to optimize the residual mapping function relative to the original, unreferenced mapping function. Pre-trained ResNet-18 from OpenCV's Deep Neural Network module has been used to implement a Faster R-CNN.

In addition to ResNet-18, an IoU-net, proposed in [33], is designed for an estimator prediction as an alternative to conventional bounding box regression techniques. An IoU-net consists of Feature Pyramid Network (FPN), which is a model developed for building high-level semantic feature maps proposed by Lin et al., and Region Proposal Network (RPN), which outputs a set of rectangular region proposals based on the feature maps provided by the FPN. With reference to [34], the FPN for RPN is realized by a 3 x 3 convolutional layer followed by two sibling 1 x 1 convolutions for classification and regression. Then, the object and non-object criterion and bounding box regression target are defined with respect to a set of reference boxes called anchors and according with the paper, a total of 15 anchors are used over the pyramid. Figure 20 shows the result of Faster R-CNN over sample test images.

Mask R-CNN is another variant of R-CNN that is suggested in [35] as an image segmentation model. In addition to the existing sequence of layers for bounding box recognition, Mask R-CNN adds another sequence of layers for prediction of object mask that works compatibly. While the combination of localisation and segmentation lead to an accurate detection of human casualty, the model is significantly compromised by computation and is only capable of running at 5 FPS.

1.3.7.3 Evaluation of Localisation and Segmentation Models

Figure 20 shows the applications of localisation and segmentation models on samples test images on video dataset 1. In terms of *recall* the performances of CNN-based localisation model and Mask RCNN outperformed that of Faster RCNN.

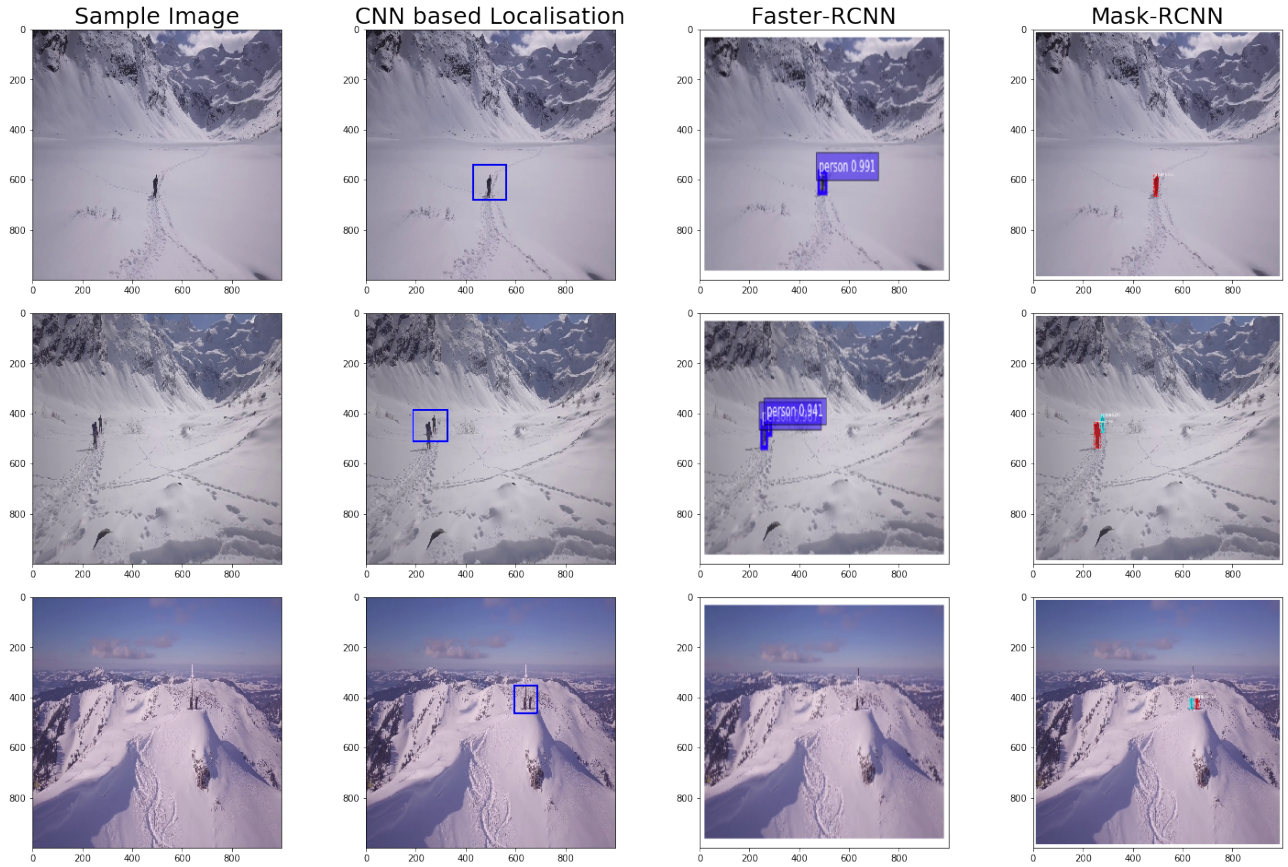

Figure 20: Performance of Localisation and Segmentation Models on Sample Images

It is based on Table 6 entailing performance specifications and requirements that the decision is made on the classifier for the overall architecture of the imaging and detection system in Section 1.5.

Model	Accuracy	Precision	Recall	Memory	Time
CNN	86.4%	87.7%	84.4%	5628 MBytes	0.0132 s
Faster R-CNN	80.3%	76.4%	77.5%	7878 MBytes	0.5856 s
Mask R-CNN	88.8%	82.3%	92.2%	12452 MBytes	1.8654 s

Table 6: Localisation Evaluation on Combined Video Datasets

1.4 Motion Detection Models

Performing motion detection from UAV is a difficult because the task is to be performed from a moving frame of reference. This means that conventional motion detection algorithms such as frame differencing methods cannot be applied, as these algorithms fail to take into consideration the movement of the UAV.

1.4.1 Velocity Transformation

One way to take into account the motion of the UAV on image frames is by using the velocity transformation matrix. Referring to Figure 21, velocities v_1, v_2, and v_3 and time t_1, t_2, and t_3 can be obtained from the control system. From the obtained values of velocities, the according transformation matrices T_1 and T_2 can be calculated since Equation 15 holds.

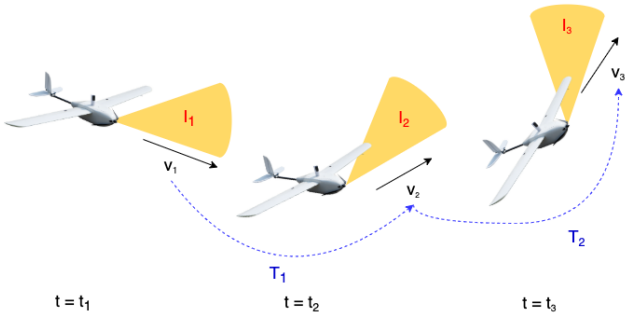

Figure 21: Velocity Transformation

$$v_3 = T_1 v_2 = T_1 T_2 v_1 \quad (15) \qquad I_3^A = T_1 I_2 \quad (16) \qquad I_3^B = T_1 T_2 I_1 \quad (17)$$

Assuming that there is a minimal variation in flight height, the 2-D velocity transformation matrices T_1 and T_2 can be applied to image frames by applying them to the pixel variations in the x-direction and those in the y-direction. The resulting images I_3^A and I_3^B are regarded as images that result purely from the motion of the UAV, and hence the average value can be subtracted from the original I_3 to find any motion that arises from a moving target. The problem with this particular method owes to the lacking information of the velocity of the UAV in the available datasets to fully validate the accuracy and feasibility of the method.

1.4.2 Optical Flow Methods

Consider a sequence of image frames $I(x, y, t)$ where (x, y) denote the location within an image frame, and t denotes time. Many optical flow methods based on calculating differentials assumes that the grey values of an object in subsequent frames do not change over time. That is, for a small time interval it is assumed that Equation 18 holds.

$$I_x u + I_y v + I_t = 0 \quad (18) \quad \text{where } u = \frac{dx}{dt}, v = \frac{dy}{dt} \text{ is the displacement field called optical flow}$$

The single equation in the form of Equation 18 is not sufficient to calculate the unique values of u and v. Lucas and Kanade made an additional assumption that the optical flow vector is constant within a certain regions of size ϵ. Using the assumption, it is possible to specifically determine constants u and v at particular location and time using a weighted least square minimisation.

$$E_{LOCAL}(u, v) = K_\rho * (I_x u + I_y v + I_t)^2 \tag{19}$$

In addition to the consideration the local movements of the pixels, Horn and Schnuck [36] suggested the application of global energy function to capture the global movements of the pixels. The minimisation of Equation 20 is the dense flow estimates that account for such global movement.

$$E_{GLOBAL}(u, v) = \int (I_x u + I_y v + I_t)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) dx\ dy \tag{20}$$

By solving the combined minimisation problems of Equations 19 and 20, the movement of the dynamic foreground can be differentiated the static background. Recent research [37] has further suggested the application of Histogram of Oriented Optical Flow (HOOF) as a way to more accurately distinguish the two, by categorising the direction and movement of each grid of pixels. Figure 22 shows the ability of HOOF method to detect motion of the moving casualty perhaps at the cost of also elciting regions of interest based on motion.

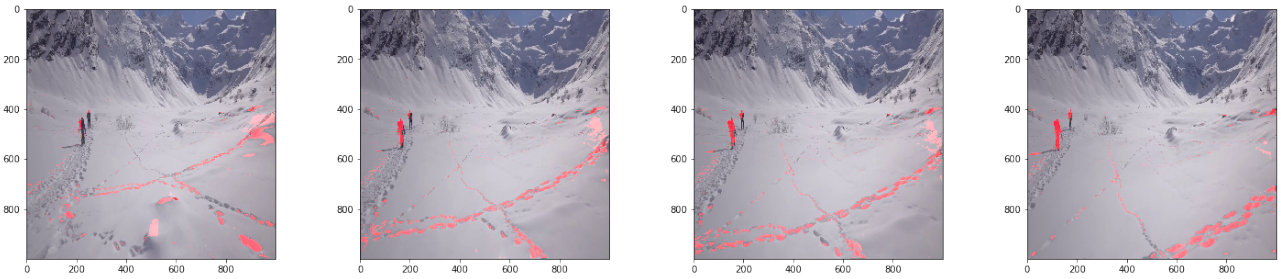

Figure 22: Histogram of Oriented Optical Flow

1.5 Overall Architecture of Imaging and Detection System

Consideration of effectively combining the casualty detection model and motion detection model, and more importantly, boosting the precision, *Mixture of Experts* technique has been used as the finalised architecture for the overall imaging and detection system. The key idea is to use image feature maps and motion feature maps to make a final decision on the presence of human casualty. At the same time the technique allows each detection model to focus on predicting the right decision for the cases where it is outperforming the other expert. It is such combination of different modes of information that allows the overall imaging and detection system to perform better than each individual expert.

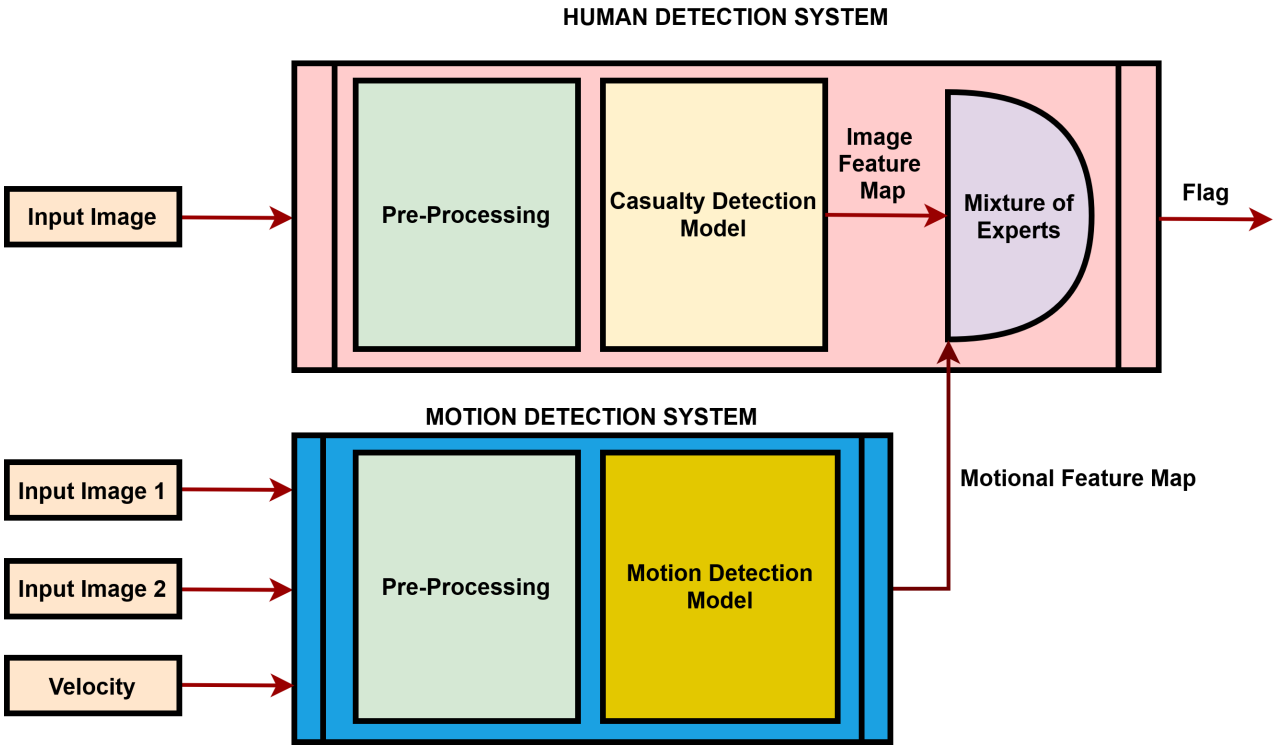

Figure 23: Architecture of Imaging and Detection System

One of the key differences between an image dataset and a video dataset is that in the case of the latter, image frames obtained across adjacent time interval have high correlation. To further reduce the false positive ratio and thereby to increase precision, a flag indicating the presence of the human casualty has been raised if and only if there exists overlapping set of windows with intersection greater than 75% across 20 image frames. 10 image frames have been determined by taking into account the cruise speed of the UAV, which is equivalent to the 1/3 of the time taken by the UAV to travel the vertical target area of the camera.

To conclude, using the *RAMF* filter as the pre-processing method in Section 1.3.3, *CNN based localisation* model in Section 1.3.7.1, and *HOOF* method in Section 1.4.2, we propose the final architecture of the overall imaging and detection system shown in Figure 23. Over video dataset 1, the finalised detection and imaging system has achieved 90.0% accuracy, 91.5% precision, and 88.6% recall, requiring 0.022 seconds and 6345 MBytes of memory per image.

1.6 Conclusion

In this chapter, the design of imaging system is considered by formulating the requirements for the overall system and by exploring different implementations of image classification and localisation models, as well as motion detection models. The localisation of the human casualty and the consideration of the degree of injury through motion detection have been of the primary achievements of the chapter with the reflections on the interactions between the system and the user.

References

[1] P. Dollar, C. Wojek, B. Schiele and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[2] (1st May 2020). "Operating uav in nepal," [Online]. Available: http://caanepal.gov.np/drones.

[3] (1st May 2020). "Civil drones (uav)," [Online]. Available: https://www.easa.europa.eu/easa-and-you/civil-drones-rpas.

[4] (1st May 2020). "Picture blur, motion and camera shake," [Online]. Available: https://www.scantips.com/lights/shake.html.

[5] (1st May 2020). "How to achieve effective motion detection," [Online]. Available: http://bensoftware.com/blog/how-to-achieve-effective-motion-detection/.

[6] (1st May 2020). "Understanding frame rate and its importance in live streaming," [Online]. Available: https://www.dreamcast.ae/blog/frame-rate-in-live-streaming/.

[7] A. Angelova, A. Krizhevsky and V. Vanhoucke, "Pedestrian detection with a large-field-of-view deep network," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 704–711.

[8] (1st May 2020). "Intel cpu onboard," [Online]. Available: https://www.gigabyte.com/Motherboard/Intel-CPU-Onboard.

[9] C. H. Setjo, B. Achmad and Faridah, "Thermal image human detection using haar-cascade classifier," in *2017 7th International Annual Engineering Seminar (InAES)*, 2017, pp. 1–6.

[10] M. Ivašić-Kos, M. Krišto and M. Pobar, "Human detection in thermal imaging using yolo," in *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*, ser. ICCTA 2019, Istanbul, Turkey: Association for Computing Machinery, 2019, 20–24, ISBN: 9781450371810. DOI: 10.1145/3323933.3324076. [Online]. Available: https://doi.org/10.1145/3323933.3324076.

[11] M. Hammer, M. Hebel and M. Arens, "Person detection and tracking with a 360° lidar system," in *Electro-Optical Remote Sensing XI*, G. Kamerman and O. Steinvall, Eds., International Society for Optics and Photonics, vol. 10434, SPIE, 2017, pp. 179 –185. DOI: 10.1117/12.2278215. [Online]. Available: https://doi.org/10.1117/12.2278215.

[12] S. Hasirlioglu, A. Kamann, I. Doric and T. Brandmeier, "Test methodology for rain influence on automotive surround sensors," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 2242–2247.

[13] M. Kutila, P. Pyykönen, H. Holzhüter, M. Colomb and P. Duthon, "Automotive lidar performance verification in fog and rain," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1695–1701.

[14] (1st May 2020). "Camera resolution and range," [Online]. Available: https://www.flir.com/discover/marine/technologies/resolution/.

[15] W. Cheng, "Pedestrian detection using an rgb-depth camera," in *2016 International Conference on Fuzzy Theory and Its Applications (iFuzzy)*, 2016, pp. 1–3.

[16] M. Ohki, M. E. Zervakis and A. N. Venetsanopoulos, "3-d digital filters," in *Multidimensional Systems: Signal Processing and Modeling Techniques*, ser. Control and Dynamic Systems, C. Leondes, Ed., vol. 69, Academic Press, 1995, pp. 49 –88. DOI: https://doi.org/10.1016/S0090-5267(05)80038-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0090526705800386.

[17] S. Khan and D.-H. Lee, "An adaptive dynamically weighted median filter for impulse noise removal," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, Dec. 2017. DOI: 10.1186/s13634-017-0502-z.

[18] H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithms and results," *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 499–502, 1995.

[19] X. Zheng, Y. Liao, W. Guo, X. Fu and X. Ding, "Single-image-based rain and snow removal using multi-guided filter," in *Neural Information Processing*, M. Lee, A. Hirose, Z.-G. Hou and R. M. Kil, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 258–265, ISBN: 978-3-642-42051-1.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467360.

[21] Y. Ma, X. Chen and G. Chen, "Pedestrian detection and tracking using hog and oriented-lbp features," in *Network and Parallel Computing*, E. Altman and W. Shi, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 176–184, ISBN: 978-3-642-24403-2.

[22] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006, ISBN: 0387310738.

[23] M. Schmidt, N. Le Roux and B. Francis, "Minimizing finite sums with the stochastic average gradient," *Mathematical Programming*, 2017.

[24] A. Defazio, F. Bach and S. Lacoste-Julien, *Saga: A fast incremental gradient method with support for non-strongly convex composite objectives*, 2014.

[25] R. Bollapragada, D. Mudigere, J. Nocedal, H.-J. M. Shi and P. T. P. Tang, "A progressive batching l-bfgs method for machine learning," *ArXiv*, vol. abs/1802.05374, 2018.

[26] C. W. Royer, M. O'Neill and S. J. Wright, "A newton-cg algorithm with complexity guarantees for smooth unconstrained optimization," *Mathematical Programming*, pp. 1–38, 2020.

[27] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition.," in *ESANN*, 1999, pp. 219–224. [Online]. Available: http://dblp.uni-trier.de/db/conf/esann/esann1999.html#WestonW99.

[28] Q. Wu and D.-X. Zhou, "Svm soft margin classifiers: Linear programming versus quadratic programming," *Neural Comput.*, vol. 17, no. 5, pp. 1160–1187, May 2005, ISSN: 0899-7667. DOI: 10.1162/0899766053491896. [Online]. Available: https://doi.org/10.1162/0899766053491896.

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

[30] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2014.

[31] M. T. Ribeiro, S. Singh and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 1135–1144.

[32] K. He, X. Zhang, S. Ren and J. Sun, *Deep residual learning for image recognition*, 2015.

[33] B. Jiang, R. Luo, J. Mao, T. Xiao and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *The European Conference on Computer Vision (ECCV)*, Sep. 2018.

[34] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, *Feature pyramid networks for object detection*, 2016.

[35] K. He, G. Gkioxari, P. Dollár and R. Girshick, *Mask r-cnn*, 2017.

[36] B. K. Horn and B. G. Schunck, "Determining optical flow," USA, Tech. Rep., 1980.

[37] R. Chaudhry, A. Ravichandran, G. Hager and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1932–1939.