ANOMALY DETECTION ON THE EDGE USING SMART CAMERAS UNDER LOW-LIGHT CONDITIONS

Yaser Abu Awwad
Cardiff University
Cardiff, UK
abuawwadyw@cardiff.ac.uk

Omer Rana
Cardiff University
Cardiff, UK
ranaof@cardiff.ac.uk

Charith Perera
Cardiff University
Cardiff, UK
pererac@cardiff.ac.uk

ABSTRACT

The number of cameras utilized in smart city domains is increasingly prominent and notable for monitoring outdoor urban and rural areas such as farms and forests to deter thefts of farming machinery and livestock, as well as monitoring workers to guarantee their safety. However, anomaly detection tasks become much more challenging in environments with low-light conditions. Consequently, achieving efficient outcomes in recognizing surrounding behaviours and events becomes difficult. Therefore, this research has developed a technique to enhance images captured in poor visibility. This enhancement aims to boost object detection accuracy and mitigate false positive detections. The proposed technique consists of several stages. In the first stage, features are extracted from input images. Subsequently, a classifier assigns a unique label to indicate the optimum model among multi-enhancement networks. In addition, it can distinguish scenes captured with sufficient light from low-light ones. Finally, a detection algorithm is applied to identify objects. Each task was implemented on a separate IoT-edge device, improving detection performance on the ExDark database and nearly one-second response time across all stages. The code to implement our proposed design is available here.

***K*eywords** Anomaly Detection, Low-Light Image Enhancement, IoT-Edge Devices, Object Detection.

1 Introduction

Deep Learning (DL) has been involved as a key point in many applications to replace human efforts in video surveillance systems with the aid of Computer Vision (CV) domains, which is one of the dominion fields to extract local and global features through processing digital images and live-streaming videos [1]. Traditionally, authorities relied on implementing Closed-Circuit Television (CCTV) to monitor human-objects behaviours in private and public environments. However, implementing these systems is high-priced due to the installation procedure, the number of cameras needed depending on a particular scenario, and the need for cloud infrastructure to record and store captured data for analysis and decision-making. Many sectors are inclined to replace traditional video surveillance systems as well as non-vision sensors with off-the-shelf cameras (also known as vision sensors) to build intelligent surveillance systems for security and other purposes.

Camera sensors are becoming increasingly accessible due to their affordability—low cost, offer high resolution and consume less power. Moreover, they can detect, track, and identify object behaviours and send alerts in an automated manner without human interaction. However, one of the ongoing and disputable challenges studied in Video Surveillance System (VSS) domains in the last decades regards the location of processing and analyzing the obtained

data. Indeed, streaming or transmitting acquired data to the cloud showed a heavy workload on the communication network in 2017, which accounts for 74% of the total network [2] [3]. Many prior studies relied on processing the data on Cloud, Edge or both [4]. However, strategies that rely on computing services provided by cloud-based, whether on their own or in conjunction with edge-based implementations, suffer from delay-sensitivity, bandwidth limitations, privacy and storage, which have crucial effects on increasing the computational cost of offloading data and affording hardware infrastructure, on processing and maintain high-volume recorded videos and still images [5].

Object detection is an essential task within the field of computer vision. It has been applied in numerous real-world situations along with video surveillance systems, including but not limited to autonomous driving [6] and road detection [7]. The resultant output identifies distinct instances (e.g. person, car, table, etc.) for detection and tracking objectives. However, detecting objects can be complex when operating in degraded conditions such as inadequate night, dawn and dusk illumination. Further, additional conditions might contaminate captured images in low-light settings with noise (e.g. salt-and-pepper), weather conditions (e.g. rainy, foggy, etc.) and blur. Indeed, such conditions can negatively impact object detection performance, which can become problematic for applications involving anomaly detection.

Consequently, it is crucial to implement certain methodologies that possess the capability to enhance image quality and facilitate feature recovery (e.g. texture, edges, etc.). Also, it is especially essential for high-level tasks (e.g. object detection, segmentation, classification, etc.) where improved performance and high true positives are desired.

2 Motivation

This research is based on the 5G Wales Unlocked project funded by the Government of Welsh Department for Digital, Culture, Media & Sport (DCMS). The project utilized 5G technology to enhance different aspects of rural and semi-rural regions in Wales. Different scenarios incorporated the integration of multiple cameras and sensors to collect data and improve understanding of the surrounding environment. The video surveillance system implemented consists of multiple cameras (Meraki MV72X) installed in four distinct locations to monitor and ensure the safety of lone workers at a farm in Monmouthshire, in the northern region of Wales. The primary goal is to identify unknown people and vehicles to avert the theft of livestock and machinery and monitor the farmers' activities to ensure their well-being and security. Raglan Castle in Monmouthshire, to identify instances of vandalism and prevent children from scaling walls and detect individuals trespassing in prohibited zones. The primary objective of these use cases was to detect anomalous events in either behaviour or appearance, thereby allowing for prompt human intervention and decision-making processes to ensure the safety and security of the premises. Moreover, the study focused on analyzing customer behaviour within transport services by tracking the number of individuals boarding and alighting and assessing the availability and occupancy of bus seats. Additionally, in Blaenau Gwent, a region located in South-East Wales, the same camera system implemented in the farm and castle scenarios was utilized to monitor parking lots, to detect available and occupied spaces, and to individuals at bus-stop.

The cameras employed in all the use cases possess and provide different functionalities and capabilities, which are listed as follows: (1) A tiny machine-learning algorithm detects only people and vehicle classes along with the (a) timestamp, (b) object Id, (c) bounding box, (d) class confidence (%) (e) class name. (2) Audio level (dB), (3) Lux value, (4) A REST API to request screenshots and publish the meta-data with an (5) MQTT Broker.

The data generated by the cameras (referred to as *"pre-identify single"*) was considered and utilised to portray any possible anomalies within the scene. When the edge camera detected such an event, a screenshot was obtained via the API and transmitted to a cloud computing environment. The cloud-based employed various state-of-the-art object detection algorithms to scrutinise and identify instances within images. Afterwards, results are shared with relevant authorities for decision-making, sending alerts and visualisations.

As was stated before, image quality plays a crucial role in producing reliable inputs for object detection tasks, leading to superior outcomes. The quality of the data source is impacted by several factors, including the communication channel, which results in the loss of some image features during data transmission, thereby degrading the image quality. However, the most significant hurdles during the project assessment pertained to the external low-light and weather conditions encountered, especially at night, see Figure 1. In most cases, the lux value approached zero, producing noisy images that were challenging to interpret, especially when detecting anomalies. The subsequent section presents a comprehensive overview of diverse methodologies associated with anomaly detection. These studies primarily concentrate on tasks like detection and tracking, utilizing distinct location-based computing paradigms such as edge, edge-fog, and edge-cloud computing. Furthermore, they share a mutual objective of identifying instances involving pedestrians and vehicles in prohibited zones, disregarding safety protocols, and violating regulations, among other

(a) High-Light, *"Farm"*.

(b) Rainy Weather, *"Car Parking"*.

(c) Low-Light and Noise, *"Farm"*.

(d) Low-Light and Noise, *"Castle"*.

Figure 1: The 5G Wales Unlocked project scenarios with various natural conditions through several representative instances. (a) vehicle in the farm scenario with bright lighting from a sun source partially covering it. (b) depicts the car park scenario during rainy weather, producing finely-detailed imagery. (c)(d) images captured in the farm and castle scenarios exhibit poor illumination and significant noise.

factors. However, it is worth noting that the training and evaluation datasets used in these studies were typically composed of high-quality images or videos captured in ideal conditions with ample lighting, which may lead to developing approaches that perform poorly in scenarios where the data is contaminated with noise, such as low-light conditions, also known as poor illumination.

To fill the gap of prior studies, we develop an image enhancement technique that can be employed in low-light conditions for object detection tasks on various edge devices (referred to as *"Nodes"*). As proposed in [8], a lightweight detector and enhancer were implemented to distinguish between captured blurred and clear images. Its objective was reinstating the sharp details if the detector considers the captured images blurred. In addition, the whole system was explicitly designed to adhere to the edge computing requirements. Similarly, our proposed technique is designed to select the most effective enhancement method among multiple low-light enhancement networks by considering the characteristics and features of the input images. In summary, the main contributions of this research are as follows:

- Our proposal entails a lightweight dynamic classifier that can determine the optimal enhancement technique from a range of techniques based on the input's features and illumination level. Since low-light environments may accumulate additional noise and enhancement techniques, performance and capabilities vary depending on many factors (e.g. training data, model architecture, etc.). Thus, the classifier determines the most effective approach after the feature extraction and selection phases instead of forwarding inputs to static or specific techniques based on assumptions.

- Evaluate the efficacy of image enhancement techniques in object detection tasks involving the identification of instances such as people, cars, buses, motorcycles, etc. Specifically, consider the most effective pre-processing stage for the proposed design by analyzing the performance of various image enhancement techniques.

- To demonstrate the feasibility of our proposed system, we created a proof-of-concept by implementing it on a range of resource-constrained devices (edge-only), including the Raspberry Pi and Jetson Nano Developer Kit. In addition, we measured edge devices' processing speed, detection accuracy and other relevant computational resource metrics.

3 Related Work

Anomaly detection in digital data refers to behaviours that deviate from typical ones in space, time, or both [9]. In smart city applications, anomaly detection is crucial for real-time through single or multiple cameras for inspecting vast quantities of recorded videos and still images. Researchers have exploited cloud-based services to handle computationally intensive tasks like action recognition, tracking, etc, to identify anomalies. However, a few studies recently investigated detecting anomalies using the edge computing paradigms [9].

In [10], the authors proposed real-time video streaming for human detection and tracking by extracting low-level features using edge and fog computing for performing high-level human and action recognition tasks. Moreover, a Histogram of Orientation (HOG) [11] with a Support Vector Machine (SVM) classifier [12] was used to extract object representations and classify them as human and non-human on a Raspberry Pi 3. Finally, features extracted are passed on to a Kernelized Correlation Filter (KCF) [13] tracker, placed at the fog stratum (Laptop) to estimate pedestrians' future positions and construct their trajectories.

In [14] proposed a hybrid edge-cloud computing approach that utilizes a lightweight deep learning model at the edge. To reduce the complexity of the model on the NVIDIA Jetson TX, the Depth-Wise Separable Convolution technique, as introduced by [15], was employed on the convolutional neural networks (CNNs) component of both Tiny-Yolo and MobileNetV2-SSD. Meanwhile, a centralized cloud-based system with a graphics processing unit was used to host a larger model, YoloV3, which was utilized for the detection phase.

In [16], a real-time system for detecting facial emotions was devised using a PYNQ-Z1 board. Initially, a Harr-Cascade algorithm and Binary Neural Network (BNN) were employed to extract facial features from human images and construct a feature map. These features were then utilized for training a face emotions classifier through a BNN to detect anomalies in public transportation, including shared cabs and taxis, to ensure passenger safety. As a result, anger, disgust, fear, and sadness were categorized as abnormal, while happiness and surprise were classified as normal. On the other hand, [17] demonstrated a system known as iSENSE, an intelligent surveillance system deployed at the edge. It was accomplished by developing a Lightweight Convolution Neural Network (L-CNN) based on the Depth-Wise Separable Convolution technique, as described by the work of [15], to extract human features. In addition, an SSD-head was incorporated to produce bonding boxes, probabilities, and classes. The proposed L-CNN was evaluated in conjunction with three tracking algorithms: Karman Filter (KF) [18], Kernelized Correlation Filter (KCF) [13], and Background Subtraction [19]. The system's overall tracking performance was tested on various Single-Board Computers (SBC), including the Raspberry Pi 3 and Tinker board.

Furthermore, [20] introduced a real-time method for performing video analytics on an NVIDIA Jeston TX2, an edge device. This approach is augmented with a graphics processing unit (GPU) to facilitate the detection of individuals in restricted areas through the utilization of YOLOv5s [21], which generates a range of object outputs such as bounding boxes, class names and probabilities. The DeepSORT tracker [22] is then applied to enable the tracking of detected objects and assign a unique identification number to each object.

The above studies identified anomalies based on cloud-based or partially (e.g. edge-cloud or edge-fog) computing frameworks. The detection process was also executed under favourable conditions of good visibility and ample illumination (such as daytime, sunny weather, etc.). Thus, this work conducts comprehensive operations, encompassing classification, enhancement, and detection, on several *"Nodes"* with limited resources, utilizing only the edge computing paradigm. Furthermore, this study aimed to enhance the detection phase in situations with inadequate illumination through multi-network enhancement techniques by examining input features.

4 Materials and Methods

This section describes our suggested technique for enhancing images under low-light conditions for object detection. It should be noted that the primary objective of employing these methods is to accurately identify objects in poorly lit environments rather than comparing methods for image quality improvement. Consequently, the emphasis will be

placed on the findings produced after the detection stage. To do so, the state-of-the-art object detection algorithms employed are evaluated and considered through a wide range of metrics.

The order of the following sections is determined based on the testing and evaluation phases rather than the input's journey from source to destination.

4.1 Overall System

In this subsection, we offer a description of the end-to-end proposed design, composed of three compulsory tasks; classification, enhancement and detection, as depicted in Figure 2, where steps are described as follows:

1. In the event of a probable anomaly (referred to as pre-identify signal), the Meraki Camera commences the process by obtaining a screenshot and transmitting it to the neighbouring node *"Client (1)"* for further analysis.

2. The *"Client (1)"* holds a lightweight dynamic classifier in which features are extracted and labels are assigned to denote the suitable enhancement technique based on the input's features.

3. A queue is designed for holding requested images and releasing them one at a time immediately after processing and completing the first image.

4. If images have sufficient light, E_0 is assigned, indicating that no enhancement is necessary, and they are passed immediately to the detection phase on the same node. If not, enhancement happens via E_1 or E_2 on separate nodes.

5. The *"RUAS"* and *"Zero-DCE++"* methods are used to enhance low-light images delivered to succeeding nodes with the labels E_1 and E_2, respectively and followed by the detection phase. The selected enhancement techniques are based on critical factors outlined in the subsequent section.

4.2 Low-Light Image Enhancement

The techniques introduced in this work vary in type of learning-based, architecture, data training, and framework. Table 1 illustrates the most recent open-sourced methodologies with public implementation accessible that have been explored between 2017 and 2021 [24]. Notably, the prevalent techniques depend on supervised learning, which involves feature mapping and metrics evaluation using paired images of degraded and ground truth. However, as acquiring paired data of identical scenes is still a formidable task, some researchers have resorted to creating synthetic images that emulate those taken in low-light environments, which might be utilized to formulate new techniques that address the obstacles presented by low illumination levels. Nonetheless, approaches based on synthetic data exhibit poor performance in real-world situations, leading to an increase in false positives during detecting anomalies (e.g. people, vehicles, etc.), particularly at night. Consequently, researchers are exploring a new direction for addressing the low-light issue that does not rely on prior knowledge derived from reference images with normal illumination, such as zero-reference and unsupervised learning.

In the study by [25], a sub-networks enhancement *MBLLEN* was suggested by extracting feature representation from low-light and enhanced images for subsequent feature fusion. Conversely, [26] proposed an unsupervised learning model that employed generative adversarial learning (GAN) techniques. The *UNet* [27] architecture has been chosen as the *generator* part of the network. In general, *generator* and *discriminator* in GAN networks fight each other until the *discriminator* gives up on recognizing images created by the *generator* as fool images, producing a realistic output similar to the original ones. In contrast to most known learning approaches, [28] proposed a model without using paired images based on curve methods to create a lightweight network under the name *Zero-DCE++*.

Furthermore, the asterisk (*) located next to certain model names signifies a *"Lightweight"* architecture, which might fit constrained devices and faster inference. Since previous studies have yet to examine recently developed models, such as the CSDNet family and RED-RT, thus, these new methods are considered and joined the evaluation with the object detection task.

Indeed, during the testing phase, a few models were ignored and discarded, such as Retinex, ExCNet & RRDNet, since these techniques perform training during inference to find the optimal value of the illumination map. Further, methods such as UTVNet & DRBN accept only paired images, which are more suitable for custom data purposes. In addition, Chen et al. & REDIIRT only accept images captured in RAW format when most modern cameras produce

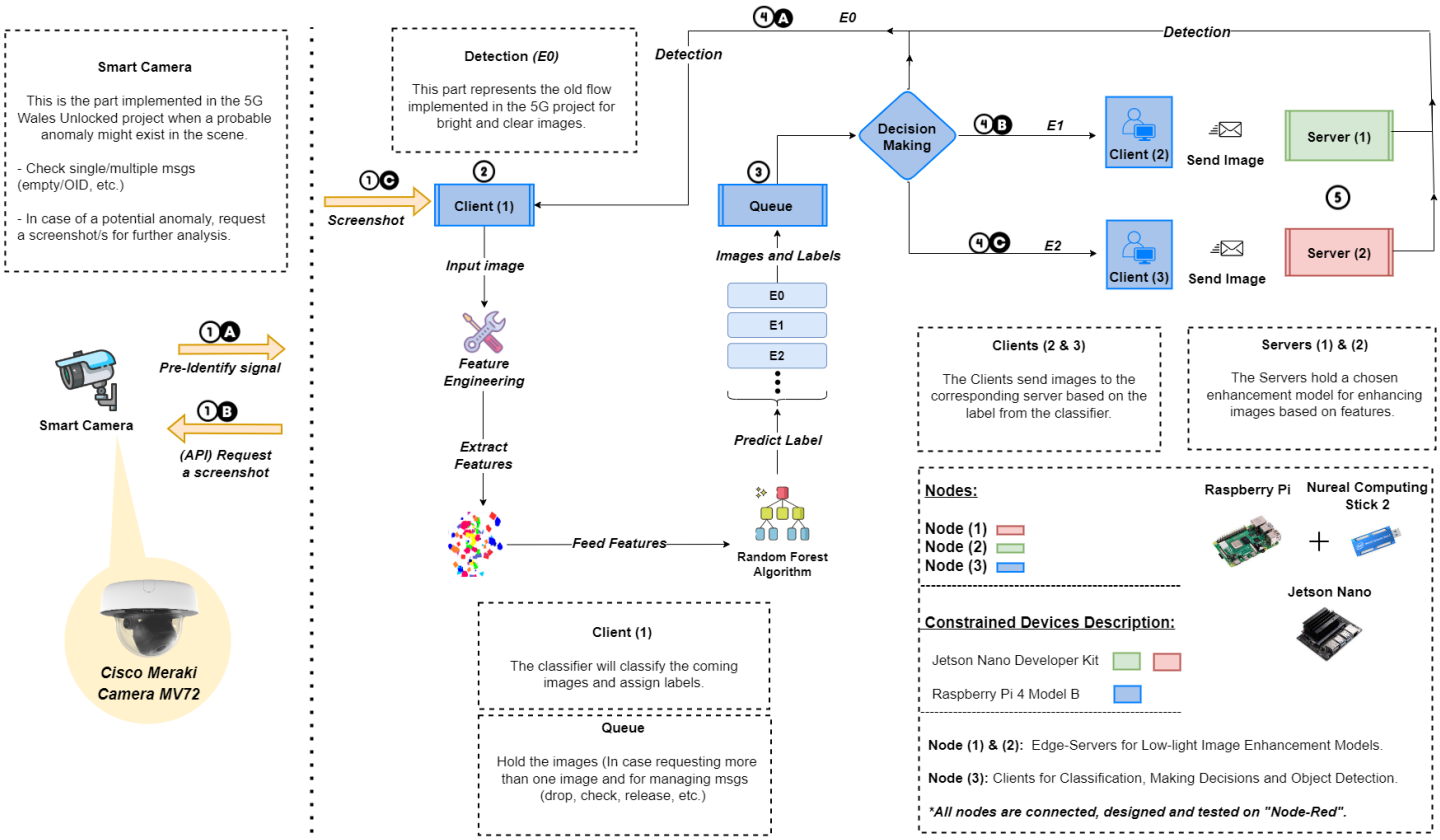


Figure 2: Our proposed end-to-end system design. We illustrate the proposed design from source to destination for detecting anomalies on multiple constrained devices when images captured by the edge camera are enhanced through multiple low-light image techniques, donated by $\{E_1, E_2, ..., E_N\}$ based on inputs local and global features. All nodes were designed, connected and tested using the *"Node-Red"* platform [23].

RBG as well as public dataset format. On the other hand, samples should be smaller for faster processing when using KinD, KinD++ & EnlightenGAN. Indeed after inspecting each model structure, training data, framework, etc., non of the models utilized the ExDark dataset (Section 4.5) for training and model creation; only the EnlightenGAN, Retinex-DIP and CSDNet employed the dataset for testing, which encourages evaluating these methods on unseen data for the detection assessment.

Ultimately, given that the primary goal is to boost object detection accuracy, any enhancement model that outperforms the existing ones may be substituted. In other words, a demonstration that these models might be suitable and effective as a pre-processing step for improving subsequent tasks (e.g. classification, detection, segmentation, etc.) while preserving output, quality and latency at the edge.

4.3 Object Detection

The 2D object detection algorithm is an emerging field using deep learning neural networks. The generated output is a bounding box wrapped around the detected object with numerical coordinates, confidence (%) and class name. The specifics of this output are determined by the dataset used to train these detectors and identify desired objects. For example, most of the techniques developed in this area have been trained on image datasets such as ImageNet [48] and MS COCO [49], which include thousands of object classes. Fine-tuning and transfer learning are popular methods to re-train the algorithm on a custom dataset or only to detect specific class name(s) [50]. The one-stage and two-stage detectors are the two primary categories that make up the many types of detectors. One-stage procedures are much quicker in the training and inference stages. In contrast, two-stage procedures require more time and effort to train but provide accurate results. Thus, one common trade-off that focuses on speed against accuracy may be made when

Table 1: Learning types for low-light image enhancement techniques, **SL:** Supervised Learning, **USL:** Un-Supervised Learning, **SSL:** Semi-Supervised Learning, **ZSL:** Zero-Shot Learning.

Method / Learning	SL	USL	SSL	ZSL
LLNet [29]	✓			
LightenNet [30]	✓			
RetinexNet [31]	✓			
MBLLEN[25]	✓			
Chen et al [32]	✓			
DeepUPE [33]	✓			
KinD [34]	✓			
KinD++ [35]	✓			
EnlightenGAN* [26]		✓		
ExCNet [36]				✓
Zero-DCE [37]				✓
DRBN [38]			✓	
Xu et al. [39]	✓			
TBEFN [40]	✓			
RRDNet [41]				✓
DSLR [42]	✓			
Zero-DCE++* [28]				✓
RUAS* [43]	✓			
Retinex-DIP [44]				✓
UTVNet [45]	✓			
CSDNet [46]	✓			
CSDGAN [46]		✓		
LiteCSDNet-LOL* [46]	✓			
LiteCSDNet-UPE* [46]	✓			
SLiteCSDNet-LOL* [46]	✓			
SLiteCSDNet-UPE* [46]	✓			
RED-RT* [47]		✓		

deciding between the two kinds based on the application requirements. One of the prevalent models for single-stages is known as YOLO, which stands for "You Only Look Once" and its several modifications (from Yolo to Yolov8) until this paper was written. At the same time, the Region-Based Convolution Neural Networks (RCNNs) are gaining interest for two-stage models, which produced various versions including RCNN, Fast-RCNN, and Faster-RCNN [51].

The Faster-RCNN (Detectron2) base model with default configuration was used for benchmarking and evaluating the low-light enhancement on the cloud paradigm [52]. Additionally, Detectron2 was exclusively employed for all use cases in the 5G Wales Unlocked project. On the other hand, for limited-resource devices (e.g. Raspberry Pi), Yolov5-tiny with a Neural Computing Stick 2 (NCS2) was used for a faster inference [53].

4.4 Lightweight Dynamic Image Classification

4.4.1 Feature Engineering

The classification task aims to produce a label describing the content of an image that is assigned with a percentage probability. Feature engineering and data-driven methods are two categories of classification techniques. Feature engineering involves applying filters or kernels directly to the input image, resulting in a new image of the same size

denoted by $\acute{f} = I \bigodot K$, where "i" refers to the input image, "K" to the kernel, and "$\acute{f}$" to the features that define and differentiate inputs based on texture, edges, luminance, and other factors. However, selecting appropriate filters can be challenging since certain features may be difficult to identify, necessitating an extensive search of descriptors to determine the best filters and their hyper-parameters. On the other hand, modern approaches such as Convolution Neural Networks (CNNs) are part of the deep learning era by performing tasks without human intervention in selecting the appropriate kernels and their values [54]. In summary, feature engineering permits selecting of suitable filters to be applied to data for feature extraction. In contrast, contemporary Convolution Neural Networks (CNNs) achieve this task intuitively without human intervention in selecting kernels and their values. As a result, traditional methods are helpful in building models suitable for resource-constrained devices since they are lightweight and inferred quickly.

1. Original Pixels

In the context of feature extraction, the pixel values play a crucial role in bag-of-feature representation. As these values offer the original characteristics of colour intensity, including RGB or grayscale colour space, texture, brightness, and other features must be included before applying convolution kernels to ensure accurate feature extraction.

2. Gabor Filter

Gabor is a conventional tool utilized in image processing for texture analysis, edge detection, and feature extraction, specifically for image classification and segmentation purposes. It acts as a band-pass filter, enabling the transmission of specific frequencies while blocking all others, unlike high-pass and low-pass filters that solely allow the transmission of high and low frequencies, respectively. Generally, Gabor is a product of a sinusoidal signal of a certain frequency and orientation modulated by a Gaussian wave [55]. The equation representing the filter is displayed as Equation 1.

$$g(\mathbf{x}, \mathbf{y}, \sigma, \alpha, \theta, \lambda, \gamma, \phi) = exp(-\frac{\acute{x}^2+\acute{y}^2\gamma^2}{2\sigma^2}) * exp[i(2\pi\frac{\acute{x}}{\lambda} + \phi)] \tag{1}$$

Where $\acute{x}$ and $\acute{y}$ are expressed as:

$$\acute{x} = x\cos\theta + y\sin\theta \tag{2}$$

$$\acute{y} = -x\sin\theta + y\cos\theta \tag{3}$$

Gabor filter relies on several parameters, mainly focused on orientation and wavelength, to govern the kernel direction and frequency. These parameters include *(x, y)* for Kernel size, σ for Standard Deviation, θ for Kernel Angle, λ for Wavelength, γ for Aspect Ratio and ϕ for Kernel Offset from the original center *(0,0)*. The filter encompasses several parameters that produce a variety of kernels with different sizes, orientations, positions, and others, resulting in a multitude of filters with distinct values, collectively known as the Gabor filter bank. These filters can extract local or global features from images for subsequent tasks [56]. In addition, these parameters were subjected to value comparisons during the feature extraction phase to generate a filter bank. These filter banks are subsequently used to train a model for classification tasks. While having numerous features may seem advantageous, it is not always the case. More specific features with fewer options generally perform better.

3. Sobel Filter

The Sobel operator is a spatial domain filter that uses a convolution kernel to compute the approximate gradient of an image in the "*Gx*" and "*Gy*" directions for each pixel [57]. Unlike filters in the frequency domain, such as low-pass, high-pass, and band-pass filters (e.g., Gabor filters), which allow only specific signals to pass, the Sobel operator operates directly on the pixel values of the original image. The Sobel convolution kernel consists of two $\mathbb{R}^{3x3}$ operations, as shown below:

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \& \quad Gy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4}$$

Where magnitude and angle are calculated as:

$$G_{Mag} = \sqrt{Gx^2 + Gy^2} \quad , \quad \theta = tan^{-1}(\frac{Gy}{Gx}) \tag{5}$$

The matrices above are identical in terms of their values, but one is the other with a 90-degree rotation applied to it. The Sobel is an extension of the "*Roberts*" operator. However, the sole difference is the matrix's form, denoted by the notation $\mathbb{R}^{2x2}$. Further, various edge descriptors such as the Canny, Prewitt's, Scharr, and Laplacian operators and various additional operators have the same capability to identify edges in digital images. However, in contrast to the previous edge approaches, recent studies have shown that the Sobel operator outperforms other edge detection operators in preserving edges, reducing noise, and producing sharp edges [58].

4.4.2 Random Forest Classifier

In Supervised Learning (SL) context, Random Forest (RF) is used to predict categorical or numerical dependent variables in classification and regression problems. The RF achieves high accuracy in its predictions by constructing multiple classifiers, a collection of decision trees (Forest) picked randomly (Random), as suggested by its name. This approach addresses the limitations of the Decision Tree (DT) method, which relies on a single tree for all training data, and could lead to model over-fitting. By contrast, the RF builds several decision trees using several datasets, referred to as Bootstrapped Dataset with various *"N"* values, in order to overcome the over-fitting problem [59, 60].

Furthermore, an imbalanced dataset between bright and low-light images has led to the introduction of Upsampling [61] technique to augment the features of the low-light images, given that obtaining them is relatively effortless. Ultimately, GridSearchCV [62] was used along with the random forest for hyperparameter to find the optimal parameters and improve the model's performance.

4.5 Dataset

The Exclusively Dark dataset (known as the ExDark) comprises 7363 sample images captured with different light intensities and 13-object classes (e.g. people, cars, bus, etc.) that exhibit a range of lighting conditions from near-zero lux to partially dark [63]. These images were sourced from publicly available datasets provided by [64, 65] and obtained through digital cameras and smartphones. The dataset encompasses multiple light intensity levels, providing potential feature extraction and classification utility. The ExDark dataset was extensively utilized during the enhancement and detection stage, owing to the varied lighting conditions encountered in real-world scenarios. Further, the classifier was constructed using random samples for the ExDark as well as images captured under normal light conditions from public databases such as the Berkeley [66], Stanford [67], and MS COCO [49]. Additionally, images from the 5G Wales Unlocked project scenarios to join the training and testing phases.

5 Experiment

In this section, we initially assess the object detection process before and after implementing enhancements while determining essential metrics such as inference speed and detection accuracy. Subsequently, we demonstrate effective techniques that yield a significant number of true positive outputs, significantly contributing to the classifier's development. Moreover, we provide metrics about edge devices when evaluating the proposed design in edge scenarios. Lastly, supplementary findings are presented in the appendix section.

5.1 Enhancement & Detection

5.1.1 Data Preparation

The ExDark includes the class name and box coordinates [l, t, w, h] generated using the Matlab toolbox [68]. Since most modern object detection algorithms generate coordinates in popular formats (e.g. COCO, Yolo, etc.), all ground truth formats were converted to Yolo-normalized ones to facilitate the evaluation stage. Therefore, only the classes, people (with 609 images) and cars (with 638 images) were used for this part. Despite the recommended name, each dataset class contains additional instances such as a person, car, bus, and motorcycle.

5.1.2 mean Average Precision

The primary evaluation for the object detection task is measuring the model performance when finding objects in digital media (e.g., images and videos). Nevertheless, in some circumstances, the term Average Precision (AP) reveals beneficial information about the detector's performance concerning certain classes. Moreover, calculating additional metrics is essential in order to acquire the *AP* and *mAP* measures when these metrics are *Precision* & *Recall*. Following that, *TP* and *FP* are also required for the precision and recall estimation.

$$\text{AP} \quad = \quad \frac{1}{N} * \sum^{N} peak_i \tag{6}$$

The calculation of average precision is described in the equation above when N denotes the number of interpolated points. For example, eleven points are the most commonly used when $peak_i \geq 0$ represents the peak values at N interpolated point in the precision versus recall plot. Afterwards, the mean average precision is obtained by the following equation, where C is the class name(s) [69].

$$\text{mAP} \quad = \quad \frac{AP_{person} + AP_{car} + \ldots + AP_C}{\textit{Total number of classes}} \tag{7}$$

However, to facilitate the datasets evaluation, this research introduced an object detection metric tool suggested by [70], which allows the user to upload the following requirements: (1) ground truth files, (2) image files, (3) class names file, (4) predictions files and (5) several options to choose coordinates style (e.g. COCO, Yolo and VOC), metrics to be measured (e.g. mAP, AP per class, $AP_{(50/70)}$ when IoU is equal to 0.5 or 0.7, $AP_{(large/medium/small)}$ for evaluating large, medium or small objects only). For this work, the options considered are: IOU@0.5, $AP_{(50)}$ for person, car, motorcycle and bus, mAP and Yolo coordinates sorted as $ClassId$, $Confidence$, X_{center}, Y_{center}, $Width$ & $Height$.

5.1.3 Results

All measured metrics related to object detection before and after applying low-light image enhancement techniques are presented in this subsection. The testing and evaluation occurred in an environment with an Intel(R) Xeon Gold 6148 CPU @ 2.10GHz (x86_64-bit) and NVIDIA Tesla P100 with CUDA 11.5. The results of the *"Detectron2"* baseline model on the original datasets without undergoing any enhancements are displayed in Table 2. In addition, the average detection speed on both datasets is calculated as images are maintained at their default size and scale. Further, after the enhancement process is carried out, the metrics above are computed for each enhancement technique. In brief, the tables below demonstrate the implementation of best practices techniques in different aspects.

Table 2: Baseline metrics (Before Enhancement).

Dataset Class	Num. of Predictions	Person (AP %)	Car (AP %)	Bus (AP %)	Motorcycle (AP %)	mAP (%)	Time (msec)	Num. of GT
Car	1704	0.115	0.438	0	0	0.1386	0.12	1700
Person	2240	0.1885	0.1415	0.25	0	0.145	0.12	2073

Table 3: The fastest and slowest enhancement model (sec).

Car		Person	
Slow	**Fast**	**Slow**	**Fast**
RetinexNet: 7.57 ElightenGAN: 6.38	**Zero-DCE++: 0.0013** SLiteCSDNet_LOL: 0.0024	RetinexNet: 7.13 TBEFN: 5.77	**Zero-DCE++: 0.0012** Zero_DCE: 0.0023

Table 4: The high and low "mAP" (%).

Car		Person	
Low	**High**	**Low**	**High**
RetinexNet: 0.300 CSDGAN: 0.320	**MBLLEN: 0.486** SLiteCSDNet_UPE: 0.486	Baseline: 0.340 CSDGAN: 0.370	**Zero-DCE++: 0.590** MBLLEN: 0.587

Table 5: The high/low AP for "Person" class (%).

Car		Person	
Low	**High**	**Low**	**High**
CSDGAN: 0.355 RetinexNet: 0.374	**CSDNet_UPE: 0.696** Baseline: 0.577	CSDGAN: 0.524 RetinexNet: 0.569	**Zero-DCE++: 0.778** RUAS: 0.760

Table 6: The high/low AP for "Car" class (%).

Car		Person	
Low	**High**	**Low**	**High**
RetinexNet: 0.428 LiteCSDNet_LOL: 0.504	**RUAS: 0.725**	RetinexNet: 0.421 CSDGAN: 0.475	**ElightenGAN: 0.639** MBLLEN: 0.633

Table 7: The high/low AP for "Bus" class (%).

Car		Person	
Low	**High**	**Low**	**High**
All Models: 0	All Models: 0	Baseline: 0	**Zero-DCE++: 0.717**

Table 8: The high/low AP for "Motorcycle" class (%).

Car		Person	
Low	**High**	**Low**	**High**
KinD: 0.029 ElightenGAN: 0.052	**LiteCSDNet_LOL: 0.225** SLiteCSDNet_UPE: 0.194	CSDGAN: 0 Baseline: 0.0277	**Zero-DCE++: 0.277** MBLLEN: 0.24786

Table 3 presents the average time to enhance a single image using a particular method, including the slowest and fastest times. The images were maintained at their original dimensions, which ranged from (220 and 293) to (2906 and 4372) in width and height. The superior method *"Zero-DCE++"* is highlighted in green, while the second-best approach and underperformed methods for both datasets are also reported. Moreover, Table 4 shows that *"Zero-DCE++"* outperformed other techniques again with a mean average precision of approximately 60% on the person dataset. In contrast, *"MBLLEN"* achieved an AP of approximately 50%. Moreover, *"RUAS"* obtained more predictions than the others. Additionally, as demonstrated in Tables 5, 6, 7, and 8.

Several techniques performed well for specific classes, highlighting the importance of selecting a suitable method for conditions such as light intensities and object types. These results suggest that no single technique is universally superior and that combining different approaches is necessary for achieving optimal results for different scenarios.

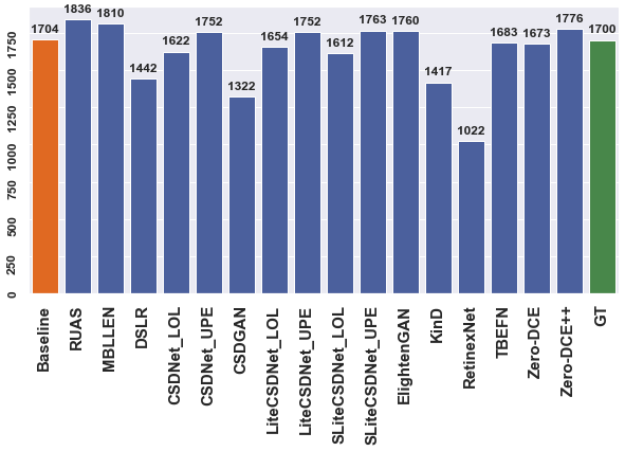


Figure 3: The number of ROIs, *"Car"* dataset.

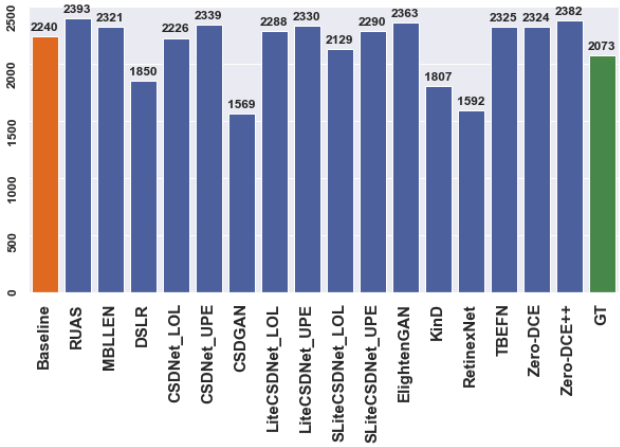


Figure 4: The number of ROIs, "*Person"*

Figures 3 and 4 illustrate the number of predictions for the person and car datasets before and after applying enhancement techniques to the original data for the detection task. The actual regions of interest (ROIs) provided by the dataset serve as the ground truth. For example, figure 3 shows that the detector can identify more objects as the ground truth using multiple tested models compared to the baseline model. Similarly, specific models exhibit improved performance for the car dataset. However, these additional detections may represent false predictions or accurate instances the annotators excluded during the annotation process. Nevertheless, the mean average precision (mAP) ensures that correct ROIs align with ground truth. The evaluation metric only compares the position and size of anticipated ROIs with those already present in the ground truth. Thus, incorrect predictions may result in overestimation or underestimation, affecting the model's accuracy.

5.2 Classification

5.2.1 Data Preparation & Model Selection

The study employed a total of 600 images, which were made up of 200 bright images and 200 dark images for each of the selected techniques, namely, *RUAS* and *Zero-DCE++*, since they yielded the most significant number of unique images. The augmentation involved upsampling the dark images to match the majority class (bright), intending to balance the data by increasing the number of images features from 113 and 102 to 200, as presented in Table 9. Moreover, the data has been split into train and test sets representing 90% for training and 10% for validation with a random state equal to

20. Additionally, to facilitate speedier processing and ensure that every image is treated fairly, the dimensions of the inputted images are scaled down to *128* by *128* pixels, followed by pixel normalization. Indeed, a series of processes, including enhancement and detection, are performed on images, resulting in outputs with different accuracies. First, the detector's performance on an individual input was evaluated using the average mean precision (mAP). Afterwards, the output samples with an mAP value ≥ 0.9 were included and categorized into a distinct folder representing a specific enhancement technique. At the same time, those with lower accuracies were disregarded. Because the same data samples are used for each enhancement model, unique images (outperformed by a specific enhancement technique) were isolated and evaluated solely, and duplicates were discarded; see Table 9 showing unique images with the highest mAP using a particular enhancer.

For example, the three techniques that contribute the largest number of unique samples are "Zero-DCE++," "RUAS," and "SLiteCSDNet_UPE." However, due to the limited availability of ExDark samples (7k) and a constrained number of unique outputs, only two techniques, "Zero-DCE++" and "RUAS," were chosen to be included in the proposed edge design. In general, "Zero-DCE++" is particularly favoured and showed promising results due to its speed, overall accuracy, and accuracy in a specific class. Additionally, "RUAS" achieves the highest number of predictions compared to the ground truth and the baseline method, performing better on the car class where additional predictions may reveal objects concealed by dark pixels.

Table 9: Unique samples with mAP ≥ 0.9 after applying image enhancement techniques and detection on the whole 7k ExDark dataset.

Model Name	Number of unique images	Time (sec)
CSDNet_UPE	33	0.005
LiteCSDNet_UPE	23	0.0035
LiteCSDNet_LOL	52	0.0032
RUAS	**113**	**0.12**
SLiteCSDNet_UPE	59	0.002
Zero_DCE++	**102**	**0.0012**

5.2.2 Feature Extraction & Training

As mentioned before, the *"RUAS"* and *"Zero-DCE++"* strategies were chosen to contribute to the system that was implemented at the edge for the enhancement phase. These techniques are represented by the labels E_1 and E_2, respectively. In addition, the value E_0 indicates that no enhancement is required for inputs deemed to be bright. Due to the small ExDark dataset, the number of genuinely unique samples is restricted. As a result, features were extracted using the whole datasets to represent the *"Zero-DCE++"* and *"RUAS"* by their labels, respectively.

During the extraction phase, several filters were used, although not all contributed to the model's ability to generate satisfactory predictions during the training phase. These filters are (1) The Gabor filter bank, (2) The Sobel, Scharr, Laplacian, and Prewitt's operators for edge detection, (3) The Gaussian blur, (4) The Median filtering, (5) The Variance filter, (6) Sharpen filter, and, finally, the most crucial feature, (7) Original pixels. It is important to note that while multiple filters are suitable for data sources that use RGB and grayscale colour spaces, only filters applicable to RGB colour were introduced and applied to the collected data since RGB colour space provides more comprehensive information for describing the entire image in various aspects. On the other hand, grayscale representation, which ranges from 0 to 255, only describes areas through black, white, or shades of grey. Therefore, for the Gabor filter configuration, it was defined with specific values, σ equal to 1 & 3, θ equal to 0 & 0.785, γ equal to 0.05 & 0.5, λ equal to 0 & 0.0785 & 1.57 & 2.356, Ksize equal to 9, and ϕ equal to 1, enabling the creation of different Gabor kernels (e.g., $G_1, G_2 ... G_N$). On the other hand, the Sobel operator was utilised in its default configuration besides the original pixel values for each trial conducted.

After applying the convolution kernels on the input images, features are extracted and reshaped into a 1D vector to be fed into the classifier for the training phase. The number of trees in the random forest classifier was set to *n_estimators=48*, which was determined as the optimal value using the GridSearchCV technique for hyperparameters with a *random_state=42* for all trials. Figure 5 illustrates that the mean test score of 93% can be attained by applying 48 trees.

Figure 5: The plot represents the n_estimators in a range of 10 to 80 with the mean test scores. It can be noted that the best score is achieved when trees=48, obtaining a score of 93%

5.2.3 Results

Table 10 displays the accuracies of the classifiers utilized in each experiment. Features were achieved by arbitrarily utilizing an assortment of filters, applying up-sampling for imbalanced data and finding the optimal number of trees using the *GirdSearchCV* method. The accuracy was calculated using the "*metrics.accuracy_score*", which compares the predicted labels on the test set with the actual labels. As a result, the best outcomes are achieved when combining the Gabor filter (G_1, G_2, G_3 and G_4), original pixel values and the Sobel operator. The experiment (5) in Table 10 shows the outperformed approach surpassing high-order kernels for the Gabor filter as well as the modern pre-trained weights Convolutional Neural Networks (e.g. VGG-16) as a feature extractor. These features were generated using six produced kernels and then reshaped into a 1D vector. The findings demonstrated that the classifier achieved the extraction and label prediction for a single input, regardless of its random size, within *0.2 msec* on the Raspberry Pi. This performance adheres to real-time processing demands, ensuring a low response time at the edge with an accuracy on the test set of 85.24%, suggesting that an appropriate technique for the enhancing phase may distinguish certain low-light features as well as differentiate them from bright ones. However, it should be noted that this conclusion was reached through a single experiment on a small portion of the data. Moreover, the confusion matrix depicted in Figure 6a indicates the classifier's ability to differentiate between bright and dark images in general and between the two chosen techniques.

Table 10: Details of the conducted experiments with different convolution kernels and VGG-16 CNN (as a feature extractor), where **OP**, **GB**, **US** and **HP** stand for Original Pixels, Gabor Bank Filter, Up-Sampling and Hyperparameters, respectively.

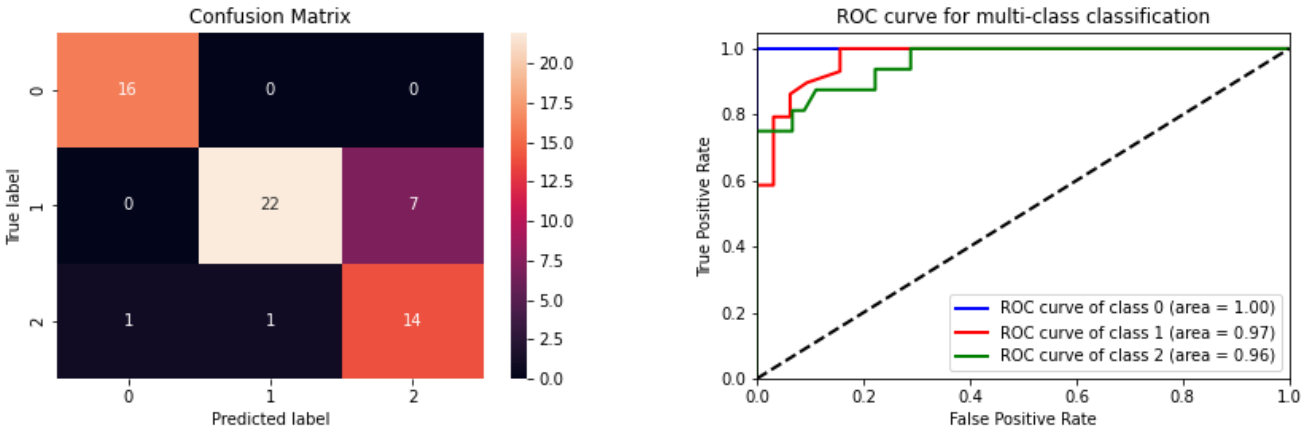
Exp ID	Filters/ Model	US	HP	Accuracy
1	OP, GBF and Sobel	✗	✓	85%
2	OP, GBF and Sobel	✓	✗	81%
3	VGG-16	✓	✗	75%
4	VGG-16	✓	✓	78%
5	**OP, GBF and Sobel**	✓	✓	**85.24%**

Nevertheless, the classifier algorithm encounters challenges in discriminating images exhibiting *"RUAS"* or *"Zero-DCE++"* characteristics. Figure 6b displays the Receiver Operating Characteristic (ROC) curve mainly used to evaluate the performance of binary classification models. The ROC curve plots the True Positive Rate (TPR) against False Positive Rate (FPR) at different threshold values for classification. The TPR represents the proportion of valid positive

instances correctly identified as positive. In contrast, the FPR represents the proportion of valid negative instances incorrectly classified as positive [71]. The classifier obtained a ROC accuracy of 96.6%, indicating that the model can discriminate between positive and negative instances, which implies that the classifier can correctly classify positive instances while minimizing false positives.

5.3 Edge Computing paradigm

In this section, the best practice enhancement models are evaluated and tested for their suitability in an edge environment. The selection of these models is based on the findings from previous benchmarking conducted on the cloud-based platform. Table 11 showcases the performance of the large model, *"Detectron2"* on both paradigms. The results from both datasets indicate that the application of the enhancement model leads to improved accuracy, which remains consistent even in the edge paradigm. The only notable difference is the time required to detect instances in a single image. It is important to note that batch processing of images at the edge is performed rapidly, at approximately 0.11 msec per image.



(a) (b)

Figure 6: (a) Confusion matrix, "0" for *Bright*, "1" for the *RUAS* model and "2" for the *Zero-DCE++* model. (b) ROC curve for the multi-classification.

Consequently, "*Yolov5-tiny*" object detection was used on edge in conjunction with the enhancement models for the suggested design, as mentioned before. The test bed consisted of 40 images taken randomly from both datasets with various brightness levels (e.g. dark, semi-dark, bright, etc.) and resized into 500 x 500 pixels. The tiny detector was evaluated using the same evaluation methods of the large model *"Detectron2"*. The results showed an improvement in both *"mAP"* and *"AP"* for both dataset classes in contrast to the baseline model when using the *"RUAS"* and *"Zero-DCE++"* techniques. Sample outputs comparing results before and after enhancement models are applied, as is shown in Figure 6.

Table 11: Evaluation of the large model "Detectron2" on the Cloud and Edge computing paradigms, **w**: with enhancement & **wo**: without enhancement.

Env	mAP%	P (Person)	AP (Car) %	Speed (in sec)	Size
Cloud **(wo)**	0.42103767	0.625056609	0.638056402	**0.09**	500 x 500
Cloud **(w)**	**0.425116397**	0.573534606	**0.701814585**		
Edge **(wo)**	0.42103767	0.625056609	0.638056402	**4.4**	500 x 500
Edge **(w)**	**0.425116397**	0.573534606	**0.701814585**		

(a) Yolo (**Only**). (b) RUAS **with** Yolo (c) Zero-DCE++ **with** Yolo.

Figure 7: Detection outputs on the constrained devices using the state-of-art Yolov5-tiny object detection algorithm and the chosen enhancement techniques.

Furthermore, the computational resources metricise for the contributed devices, also known as *"Nodes"* or *"Servers"*, were taken into consideration throughout this work. A test of three images per category, bright, dark for E_1 and dark for E_2, was carried out in a separate experiment to evaluate device computation for a unique scenario. Table 12 shows different measurements simultaneously by running an input image for each class when resources such as RAM usage, temperature readings, CPU & GPU usages are also gauged and considered for each test. The device's measurements were obtained by creating a dashboard for a single device using a visualisation web application called *"Grafana"* [72]. These dashboards were then used to collect the device's measurements and report values by matching the timestamp during a particular process. The Grafana works along with the *"Node-Exporter"* to extract the desired computation resource readings along with the *"Prometheus"* to establish the connection with the dashboard and publish the metrics for visualisation [73]. For instance, the *"Ideal Mode"* indicates device stability, and no processing occurs across all the nodes. On the other hand, a bright pixels image was used as the input for "*Test 1*"; thus, since no processing is needed, the detection job is immediately executed on the same node. In addition, it is essential to note that the CPU utilisation and temperature of "*Node 3*" rise due to the execution of the classification and detection tasks only. Following the same principle, *"Test 2"* & *"Test 3"* with low-light images as inputs that required to have their brightness improved through *"Node 1"* & *"Node 2*" prior to the detection task. Similarly to "*Node 3*", resources increased dramatically since classification and detection are involved. For the GPU utilisation, both servers reached 99 % of usage since these enhancement models rely upon during processing.

Furthermore, an alternative design proposal was considered in the study since the evaluation results revealed a high accuracy in detecting instances of *"People"* when employing the *"RUAS"* technique, achieving an average precision (AP) of approximately 0.778. Conversely, the *"Car"* class exhibited superior performance with an AP of approximately 0.725 when utilizing *"Zero-DCE++"*. Therefore, both techniques can be utilized concurrently and enhance inputs in parallel since each process occurs in a separate node. Subsequently, the detection for *"People"* will be conducted on images enhanced by *"RUAS"*, while *"Car"* instances will be detected using *"Zero-DCE++"*, and the results will be combined into a single message. It is worth noting that in this situation, the same detector *"Detectron2"*, is recommended based on the previous evaluation.

6 Discussion

This research addressed the problem of recognizing objects in environments with low-light settings at the edge. First, existing techniques for low-light image enhancement were studied along with the state-of-art object detection algorithm by evaluating the detection before, after enhancement and among multiple enhancement networks. Afterwards, best practice techniques were utilized to create a lightweight dynamic classifier that assigns labels based on image features and the amount of light intensity accumulated to choose the optimal technique for the enhancement task. Moreover, the model selection relied on the mean average precision for a single image with a value greater than 0.9, in addition

Table 12: Nodes computational resource metrics, where **N/A**: Not applicable, **N/U**: Not used in processing, **C**: Classification task, **OD**: Object Detection task, **IE**: Image Enhancement task, **Node 1**: Enhancement model (1), **Node 2**: Enhancement model (2) & **RPi**: Raspberry Pi.

Test ID	CPU Usage			GPU Usage		RAM Usage			Temp Usage					Task Involved		
	Rpi (Node 3)	Jetsons		Jetsons		Rpi (Node 3)	Jetsons		Rpi (Node 3)	Jetsons				Rpi (Node 3)	Jetsons	
		Node 1	Node2	Node 1	Node2		Node 1	Node2		Node 1		Node2			Node 1	Node 2
										CPU	GPU	CPU	GPU			
Idle Mode	**1.47%**	**7.23%**	**14.50%**	**0%**	**0%**	**800 MB**	**1.23 GB**	**1.28 GB**	**41 C°**	**29 C°**	**27.5 C°**	**27 C°**	**25.5 C°**	**N/U**	**N/U**	
1	15.00%	N/A		N/A		1.15 GB	N/U		49.9 C°	N/U		N/U		C + OD	N/U	
2	14.70%	N/U	65.60%	N/U	13%	1.19 GB	N/U	3.16 GB	50 C°	N/U		31 C°	29.5 C°	C + OD	N/U	IE
3	17.20%	31%	N/U	98%	N/U	1.13 GB	2.68 GB	N/U	49.2C°	36C°	33.5 C°	N/U		C + OD	IE	N/U

to several metrics and computational resources, including processing speed, number of predictions and specific class accuracy. Finally, the end-to-end proposed system was implemented on constrained-resources devices.

According to the findings, the *"Zero-DCE++"* method is the optimum approach for the vast majority of situations, regarding more rapid inference on several different input dimensions, around *0.001 seconds*. The highest mAP was accomplished using the following techniques: *"Zero-DCE++"* and *"MBLLEN"* on the person and car datasets, respectively. Whereas the baseline model produced an *"mAP"* value of *34 % & 45%*. Both models achieved a higher *"mAP"* value, equivalent to *59% & 48%*, than the baseline model. Again, when assessing a single class *"AP"*, the *"Zero-DCE++"* outperforms other approaches for the person, bus, and motorcycle classes with accuracies of *77%, 71%, & 27%*, respectively. However, the baseline results for several classes had extremely few advanced placements or almost none. The *"EnlightenGAN"*, on the other hand, did quite well in the car class, obtaining an *"AP"* score of *64%*. On the other hand, the *"RUAS"* was the only model that produced more significant prediction after enhancement than the others did for both datasets, about 2393 and 1836 bounding boxes, respectively, as well as compared to the actual predictions, 2073 and 1700. In addition, it is an optimal technique when it comes to detecting the *"Car"* classes with an AP *72%*. Indeed, the results demonstrate that at least seven enhancement techniques can identify more correct objects than direct detection (without enhancement), as well as the ground truth.

Further, despite the limited samples in the ExDark dataset used to build the classifier, an accuracy of 85.24% was obtained on the test set, indicating the ability to differentiate between bright scenes from low-light ones and among various low-light settings for an appropriate enhancement method. Indeed, the findings demonstrated rivalry amongst all approaches in some facets. For example, model E_1 performed better than model E_2 on $Image_1$ or vice versa. Therefore, the capabilities of the various models permit a combination and integration of several models for handling various input conditions and directing it to the most confident approach based on particular features within the image, making it feasible by the fact that the models can work together rather than relying on a single method and improving the objects identification task.

Moreover, encouraging findings were observed in the computational resource performance of the resource-constrained devices, indicating their potential to carry out all processing activities while ensuring quality and rapid response times. As an illustration, the cumulative duration required to process an individual image from its origin to its destination was recorded as *0.43 second*, with respective time intervals of *0.4, 0.001, & 0.03 seconds* for classification, enhancement (utilizing Zero-DCE++), and detection (employing Yolov5-tiny), indicating that other processes could be incorporated into the suggested framework, either on the same nodes or different ones. In addition, the inter-node transmission duration is under *one second*, as all nodes are part of a shared local network.

Furthermore, since the ExDark dataset was not intended to address low-light image enhancement tasks but rather emphasised high-level tasks such as object detection and classification, particular challenges were encountered when applying enhancement techniques to low-light images. These challenges include (1) Increased coverage and obscured of objects by dark pixels, (2) Distant and small objects, and (3) Partially displayed objects, wherein only a portion of an object is visible in the image (e.g., the rear part of a car). All these factors refer to objects not labelled and included in the ground truth. As a result, the newly discovered and missing objects considerably impact the object identification performance, leading to inadequate overall accuracy and accuracy for a specific class. Henceforth, it is suggested that the dataset be relabeled using suggested effacement techniques identified during the current research for future work, allowing other researchers to evaluate new proposed methodologies in the same field and obtain accurate outcomes.

Regarding the classification task, the number of unique images that can outperform for single enhancement model is limited, especially during the feature extraction phase. Because of this, increasing the number of images by utilizing additional datasets rather than relying solely on a single dataset for the training phase is recommended. Indeed, additional datasets besides the ExDark might boost the likelihood of obtaining more unique samples from which to extract relevant characteristics.

Furthermore, the collected metrics regards the resource-constrained devices across all stages pointed to a significant orientation toward consolidating more jobs and computations onto a single device. Several low-light enhancement networks may be carried out in a single *"Node"* when the appropriate enhancer is selected depending on the acquired input features and attributes. Further, moving techniques related to other problems to separate nodes to the design allows for handling different challenges, such as blur and weather conditions. Thus, an input may be pre-processed and directed to numerous nodes before the detection occurs when inputs are represented with multi-labels describing the scene content.

7 Conclusions

In this research, a technique was developed to improve the detection accuracy of objects by enhancing images captured under low-light conditions. Initially, the input images were subjected to feature extraction. Subsequently, a distinctive label was assigned to identify the most effective model from multiple enhancement networks. Each task was carried out using a separate resource-constrained device. The findings illustrated an improvement in accuracy when enhancement techniques occur before the detection phase by boosting object detection accuracy by more than *20%*. Moreover, each technique was superior in particular conditions, object class, images captured with noise accumulated, weather conditions and resolutions, indicating that diverse capabilities might be obtained from a single technique to handle a specific situation. Furthermore, the proof-of-concept showcased the practicability of deploying the entire system exclusively with the edge computing paradigm without depending on external computational resources achieving *1 second* from source to destination, adhering to real-time requirements and encouraging to add more tasks on the same node as well as additional nodes to address further challenges.

Acknowledgments

This work was a collaborative effort involving the Welsh Government and various organizations such as Cisco, BT, Cardiff University, UtterBerry, Jam Creative, and AppyWay. Monmouthshire County Council and Blaenau Gwent County Borough Council also contributed to the initiative.

Appendix

The following samples presented results when detection occurred before and after enhancement to understand better how image enhancement methods as a prior stage help achieve higher detection accuracy. It is worth mentioning that only best practice methods were selected and considered. Figures 8, 9, 10 & 11 demonstrate randomly selected samples to examine the approach's functionality and capabilities. After more investigation, the optimal technique is shown by the green box surrounding the model names, which provides extra correct predictions compared to ground truth and direct detection; see the figures caption. Moreover, the total number of predictions after applying each method are presented in figures 12, 13, 14 & 15, when the "Green" bar represents the ground truth (GT), "Orange" represents detection before enhancement (Direct or Baseline) and the "Blue" ones for low-light image enhancement methods. The number on the top bars indicates the total number of predictions.

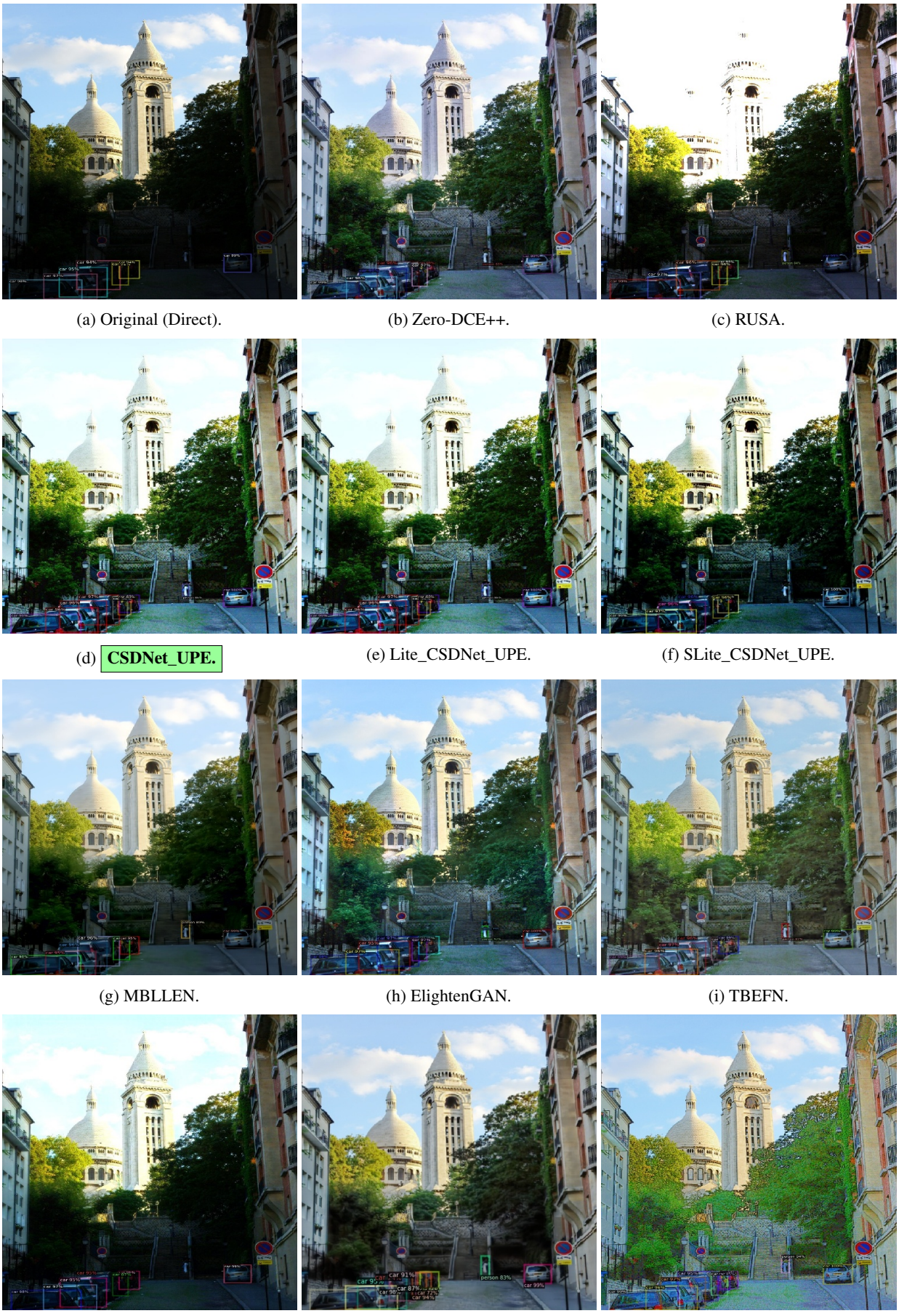
(a) Original (Direct). (b) Zero-DCE++. (c) RUSA.

(d) CSDNet_UPE. (e) Lite_CSDNet_UPE. (f) SLite_CSDNet_UPE.

(g) **MBLLEN.** (h) ElightenGAN. (i) TBEFN.

(j) DSLR. (k) KinD. (l) RetinexNet.

Figure 8: Comparison of model enhancement for the detection stage on sample *"2015_02448.jpg"*.



(a) Original (Direct). (b) Zero-DCE++. (c) RUSA.

(d) **CSDNet_UPE.** (e) Lite_CSDNet_UPE. (f) SLite_CSDNet_UPE.

(g) MBLLEN. (h) ElightenGAN. (i) TBEFN.

(j) DSLR. (k) KinD. (l) RetinexNet.

Figure 9: Comparison of model enhancement for the detection stage on sample *"2015_02926.jpg"*.

(a) Original (Direct). (b) **Zero-DCE++.** (c) RUSA.

(d) CSDNet_UPE. (e) Lite_CSDNet_UPE. (f) SLite_CSDNet_UPE.

(g) MBLLEN. (h) ElightenGAN. (i) TBEFN.

(j) DSLR. (k) KinD. (l) RetinexNet.

Figure 10: Comparison of model enhancement for the detection stage on sample *"2015_06339.jpg"*.

(a) Original (Direct). (b) Zero-DCE++. (c) **RUSA.**

(d) CSDNet_UPE. (e) Lite_CSDNet_UPE. (f) SLite_CSDNet_UPE.

(g) MBLLEN. (h) ElightenGAN. (i) TBEFN.

(j) DSLR. (k) KinD. (l) RetinexNet.

Figure 11: Comparison of model enhancement for the detection stage on sample *"2015_06574.jpg"*.

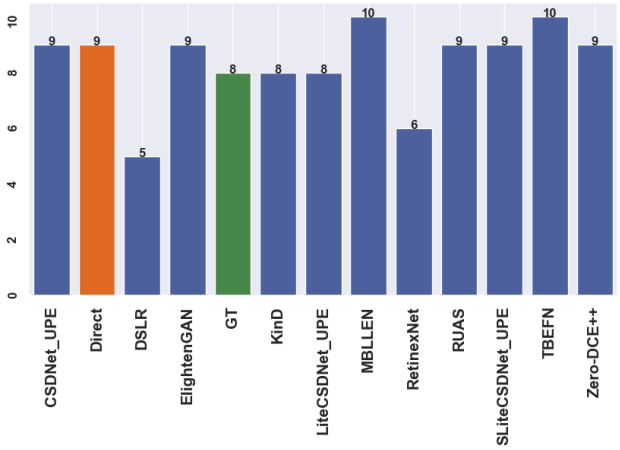


Figure 12: Ground Truth vs Direct vs After Enhancement for sample *"2015_02448.jpg"*.

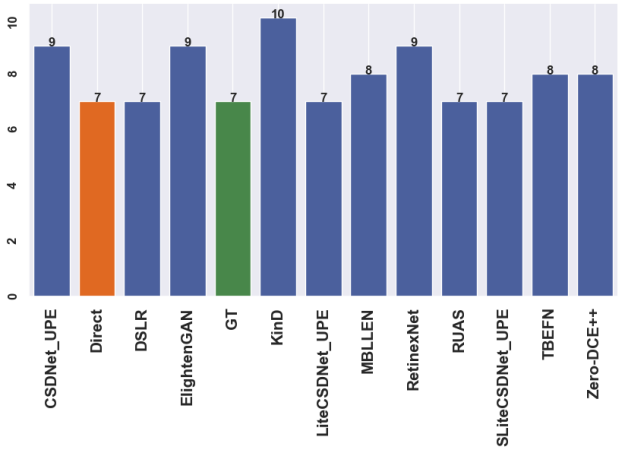


Figure 13: Ground Truth vs Direct vs After Enhancement for sample *"2015_02926.jpg"*.

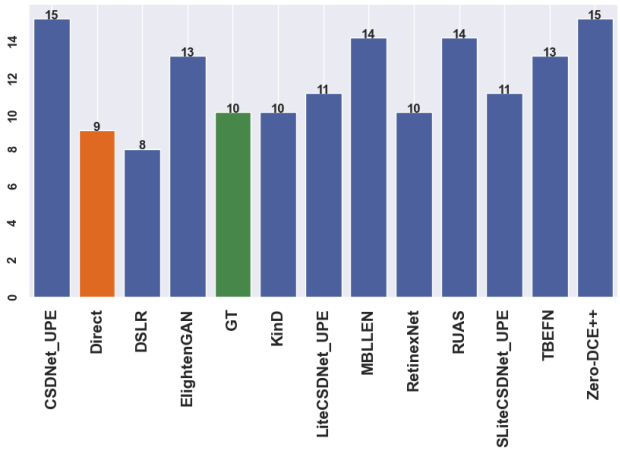


Figure 14: Ground Truth vs Direct vs After Enhancement for sample *"2015_06339.jpg"*.

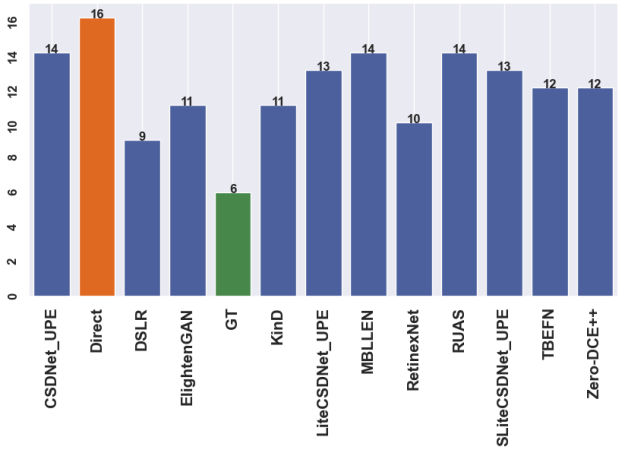


Figure 15: Ground Truth vs Direct vs After Enhancement for sample *"2015_06574.jpg"*.

References

[1] Gaudenz Danuser. Computer vision in cell biology. *Cell*, 147(5):973–978, 2011.

[2] Ning Chen, Yu Chen, Erik Blasch, Haibin Ling, Yang You, and Xinyue Ye. Enabling smart urban surveillance at the edge. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 109–119. IEEE, 2017.

[3] V Cisco. Cisco visual networking index: Forecast and methodology 2016–2021.(2017), 2017.

[4] Hongshan Li, Chenghao Hu, Jingyan Jiang, Zhi Wang, Yonggang Wen, and Wenwu Zhu. Jalad: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 671–678, 2018.

[5] Sonam Srivastava and Sarv Pal Singh. A survey on latency reduction approaches for performance optimization in cloud computing. In *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*, pages 111–115. IEEE, 2016.

[6] Rahee Walambe, Aboli Marathe, Ketan Kotecha, and George Ghinea. Lightweight Object Detection Ensemble Framework for Autonomous Vehicles in Challenging Weather Conditions. *Comput. Intell. Neurosci.*, 2021, 2021.

[7] Vung Pham, Chau Pham, and Tommy Dang. Road damage detection and classification with detectron2 and faster r-cnn. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5592–5601, 2020.

[8] Ju-Wei Que and Ching-Hu Lu. Lightweight and dynamic deblurring for iot-enabled smart cameras. *IEEE Internet of Things Journal*, 9(20):20693–20705, 2022.

[9] Devashree R. Patrikar and Mayur Rajaram Parate. Anomaly detection using edge computing in video surveillance system: review. *Int. J. Multimed. Inf. Retr.*, 2022.

[10] Xudong Zhao, Peng Liu, Jiafeng Liu, and Xianglong Tang. Real-Time Human Objects Tracking for Smart Surveillance at the Edge. *IEEE Int. Conf. Commun.*, 2018-May(May), 2018.

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

[12] B Jagadeesh and Chandrashekar M Patil. Video based action detection and recognition human using optical flow and svm classifier. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 1761–1765. IEEE, 2016.

[13] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014.

[14] Yu Zhao, Yue Yin, and Guan Gui. Lightweight Deep Learning Based Intelligent Edge Surveillance Techniques. *IEEE Trans. Cogn. Commun. Netw.*, 6(4):1146–1154, 2020.

[15] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[16] B. S. Ajay and Madhav Rao. Binary neural network based real time emotion detection on an edge computing device to detect passenger anomaly. *Proc. IEEE Int. Conf. VLSI Des.*, 2021-Febru:175–180, 2021.

[17] Xudong Zhao, Peng Liu, Jiafeng Liu, and Xianglong Tang. Toward Intelligent Surveillance as an Edge Network Service (iSENSE) Using Lightweight Detection and Tracking Algorithms. *IEEE Trans. Serv. Comput.*, 14(6):1624–1637, 2021.

[18] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. *Dl*, 1995.

[19] Massimo Piccardi. Background subtraction techniques: a review. In *2004 IEEE international conference on systems, man and cybernetics (IEEE Cat. No. 04CH37583)*, volume 4, pages 3099–3104. IEEE, 2004.

[20] Evangelos Maltezos, Panagiotis Lioupis, Aris Dadoukis, Lazaros Karagiannidis, Eleftherios Ouzounoglou, Maria Krommyda, and Angelos Amditis. A Video Analytics System for Person Detection Combined with Edge Computing. *Computation*, 10(3):35, 2022.

[21] Glenn Jocher and Ayush Chaurasia. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. *Dl*, feb 2022.

[22] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[23] Node-red. `https://nodered.org/`, n.d. Accessed: 26/01/2023.

[24] Chongyi Li, Chunle Guo, Ling-Hao Han, Jun Jiang, Ming-Ming Cheng, Jinwei Gu, and Chen Change Loy. Low-light image and video enhancement using deep learning: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, DI(01):1–1, 2021.

[25] Feifan Lv, Feng Lu, Jianhua Wu, and Chongsoon Lim. Mbllen: Low-light image/video enhancement using cnns. *BMVC*, 220(1):4, 2018.

[26] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE Transactions on Image Processing*, 30:2340–2349, 2021.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, v1:234–241, 2015.

[28] Chongyi Li, Chunle Guo, and Change Loy Chen. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[29] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*, 61:650–662, 2017.

[30] Chongyi Li, Jichang Guo, Fatih Porikli, and Yanwei Pang. Lightennet: A convolutional neural network for weakly illuminated image enhancement. *Pattern recognition letters*, 104:15–22, 2018.

[31] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560*, 2018.

[32] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018.

[33] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6849–6857, 2019.

[34] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. *Proceedings of the 27th ACM international conference on multimedia*, pages 1632–1640, 2019.

[35] Yonghua Zhang, Xiaojie Guo, Jiayi Ma, Wei Liu, and Jiawan Zhang. Beyond brightening low-light images. *International Journal of Computer Vision*, 129(4):1013–1037, 2021.

[36] Lin Zhang, Lijun Zhang, Xiao Liu, Ying Shen, Shaoming Zhang, and Shengjie Zhao. Zero-shot restoration of back-lit images using deep internal learning. *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1623–1631, 2019.

[37] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1780–1789, 2020.

[38] Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3063–3072, 2020.

[39] Ke Xu, Xin Yang, Baocai Yin, and Rynson WH Lau. Learning to restore low-light images via decomposition-and-enhancement. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2281–2290, 2020.

[40] Kun Lu and Lihong Zhang. Tbefn: A two-branch exposure-fusion network for low-light image enhancement. *IEEE Transactions on Multimedia*, 23:4093–4105, 2020.

[41] Anqi Zhu, Lin Zhang, Ying Shen, Yong Ma, Shengjie Zhao, and Yicong Zhou. Zero-shot restoration of underexposed images via robust retinex decomposition. *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020.

[42] Seokjae Lim and Wonjun Kim. Dslr: deep stacked laplacian restorer for low-light image enhancement. *IEEE Transactions on Multimedia*, 23:4272–4284, 2020.

[43] Risheng Liu, Long Ma, Jiaao Zhang, Xin Fan, and Zhongxuan Luo. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10561–10570, 2021.

[44] Zunjin Zhao, Bangshu Xiong, Lei Wang, Qiaofeng Ou, Lei Yu, and Fa Kuang. Retinexdip: A unified deep framework for low-light image enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1076–1088, 2021.

[45] Chuanjun Zheng, Daming Shi, and Wentian Shi. Adaptive unfolding total variation network for low-light image enhancement. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021*, pages 4439–4448, 2021.

[46] Long Ma, Risheng Liu, Jiaao Zhang, Xin Fan, and Zhongxuan Luo. Learning deep context-sensitive decomposition for low-light image enhancement. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[47] Mohit Lamba and Kaushik Mitra. Restoring extremely dark images in real time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3487–3497, 2021.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[50] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[51] Nandhini Sivasubramaniam, D. Easwaramoorthy, and R. Abinands. An extensive review on recent evolutions in object detection algorithms. *International Journal of Emerging Trends in Engineering Research*, 8:3766–3776, 07 2020.

[52] Facebook. Github - facebookresearch/detectron2: Detectron2 is a platform for object detection, segmentation and other visual recognition tasks. `https://github.com/facebookresearch/detectron2`, n.d. Accessed: 26/01/2023.

[53] Yolo. Github - ultralytics/yolov5: Yolov5 in pytorch > onnx > coreml > tflite. `https://github.com/ultralytics/yolov5`, n.d. Accessed: 12-12-2022.

[54] Mahbub Hussain, Jordan J Bird, and Diego R Faria. A study on cnn transfer learning for image classification. In *UK Workshop on computational Intelligence*, pages 191–202. Springer, 2018.

[55] N Pizzolato Angelo and V Haertel. On the application of gabor filtering in supervised image classification. *International Journal of Remote Sensing*, 24(10):2167–2189, 2003.

[56] Itzhak Fogel and Dov Sagi. Gabor filters as texture discriminator. *Biological cybernetics*, 61(2):103–113, 1989.

[57] Woods Gonzalez. Digital image processing second edition. *Dl*, Dl(Dl):414–428, 1992.

[58] Mamta Juneja and Parvinder Singh Sandhu. Performance evaluation of edge detection techniques for images in spatial domain. *International journal of computer theory and Engineering*, 1(5):614, 2009.

[59] Mike Gashler, Christophe Giraud-Carrier, and Tony Martinez. Decision tree ensemble: Small heterogeneous is better than large homogeneous. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 900–905. IEEE, 2008.

[60] Sotiris Kotsiantis. Combining bagging, boosting, rotation forest and random subspace methods. *Artificial intelligence review*, 35(3):223–240, 2011.

[61] Sukarna Barua, Md Monirul Islam, Xin Yao, and Kazuyuki Murase. Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on knowledge and data engineering*, 26(2):405–425, 2012.

[62] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[63] Yuen Peng Loh and Chee Seng Chan. Getting to know low-light images with the exclusively dark dataset. *Computer Vision and Image Understanding*, 178:30–42, 2019.

[64] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[65] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[66] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423 vol.2, 2001.

[67] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1253–1260, 2010.

[68] Piotr Dollár. Piotr's Computer Vision Matlab Toolbox (PMT). `https://github.com/pdollar/toolbox`, n.d.

[69] Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science*, 2, 2004.

[70] Rafael Padilla, Wesley L Passos, Thadeu LB Dias, Sergio L Netto, and Eduardo AB Da Silva. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, 10(3):279, 2021.

[71] Jerome Fan, Suneel Upadhye, and Andrew Worster. Understanding receiver operating characteristic (roc) curves. *Canadian Journal of Emergency Medicine*, 8(1):19–20, 2006.

[72] Grafana. grafana/grafana: The open and composable observability and data visualization platform. visualize metrics, logs, and traces from multiple sources like prometheus, loki, elasticsearch, influxdb, postgres and many more. `https://github.com/grafana/grafana`, n.d. Accessed: 01/12/2022.

[73] Prometheus. prometheus/node_exporter: Exporter for machine metrics. `https://github.com/prometheus/node_exporter`, Nov 2022.