future internet

MDPI

Article

Enhancing Smart City Safety and Utilizing AI Expert Systems for Violence Detection

Pradeep Kumar, Guo-Liang Shih, Bo-Lin Guo, Siva Kumar Nagi, Yibeltal Chanie Manie, Cheng-Kai Yao, Michael Augustine Arockiyadoss and Peng-Chun Peng *

Department of Electro-Optical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan; t111999408@ntut.org.tw (P.K.); t109650037@ntut.org.tw (G.-L.S.); t108650021@ntut.org.tw (B.-L.G.); t112998403@ntut.org.tw (S.K.N.); yibeshmamaru@gmail.com (Y.C.M.); t109658093@ntut.org.tw (C.-K.Y.); pushpamichaeldoss@gmail.com (M.A.A.)

* Correspondence: pcpeng@ntut.edu.tw; Tel.: +886-2-2771-2171 (ext. 4671)

Abstract: Violent attacks have been one of the hot issues in recent years. In the presence of closed-circuit televisions (CCTVs) in smart cities, there is an emerging challenge in apprehending criminals, leading to a need for innovative solutions. In this paper, the propose a model aimed at enhancing real-time emergency response capabilities and swiftly identifying criminals. This initiative aims to foster a safer environment and better manage criminal activity within smart cities. The proposed architecture combines an image-to-image stable diffusion model with violence detection and pose estimation approaches. The diffusion model generates synthetic data while the object detection approach uses YOLO v7 to identify violent objects like baseball bats, knives, and pistols, complemented by MediaPipe for action detection. Further, a long short-term memory (LSTM) network classifies the action attacks involving violent objects. Subsequently, an ensemble consisting of an edge device and the entire proposed model is deployed onto the edge device for real-time data testing using a dash camera. Thus, this study can handle violent attacks and send alerts in emergencies. As a result, our proposed YOLO model achieves a mean average precision (MAP) of 89.5% for violent attack detection, and the LSTM classifier model achieves an accuracy of 88.33% for violent action classification. The results highlight the model's enhanced capability to accurately detect violent objects, particularly in effectively identifying violence through the implemented artificial intelligence system.

Keywords: expert system; smart city; artificial intelligence; real-time application; violence detection; image-to-image stable diffusion; edge computing; MediaPipe; YOLO v7; LSTM

Citation: Kumar, P.; Shih, G.-L.; Guo, B.-L.; Nagi, S.K.; Manie, Y.C.; Yao, C.-K.; Arockiyadoss, M.A.; Peng, P.-C. Enhancing Smart City Safety and Utilizing AI Expert Systems for Violence Detection. *Future Internet* **2024**, *16*, 50. https://doi.org/10.3390/fi16020050

Academic Editor: Guan Gui

Received: 4 January 2024
Revised: 26 January 2024
Accepted: 29 January 2024
Published: 31 January 2024

1. Introduction

In recent years, challenges have emerged in the security aspect; public and personal protection have been always considerable priorities and crucial problems for people and the citizenry in smart cities have emerged [1,2]. Many violent attacks are occurring in these cities and citizens are being injured, assaulted, subjected to unknown murder activity, and robbed where CCTVs are not available, due to which police or cops cannot reach on time. However, due to the limitations of certain area ranges of closed-circuit television cameras, violent attacks cannot be detected outside these ranges. Several techniques [1–6] are being implemented to detect day-by-day violent attacks based on the data from CCTVs. Recently, many researchers have proposed attack- or violence-detecting systems using object detection, pose estimation and image generation methods. These implement systems are artificial-intelligence-based systems. However, these AI-based systems face limitations due to the availability of violence data, resulting in lower performance when detecting accurate targets. In circumstances that have needed combined violent objects, the pose estimation model and image generation method have been considered as solutions. The main advantages of such a system are that it is a system that is both fixed and portable at minimal cost and with high accuracy [2,4–6].

Using previous detection methods, datasets are collected through different sources. However, these datasets are not sufficient to train a real-time expert system [6]. These insufficient datasets constitute a challenge for violence detection and are not of high enough quantity to develop a robust detection system [7]. To address this challenge of creating a large number of data, recently, the image generation method has been the most popular database resource for creating synthetic data to resolve this issue. Image generation is a widely used approach to create synthetic data, using models such as the diffuser model and generative adversarial network (GAN) model [8,9]. Diffusers constitute a distinctive initiated and discussed approach for data generation; numerous models are proposed for different types of utilization and requisition, such as the synthesis of images using old images and prompt text, and some of the implementations have also introduced text-to-image generation [10,11]. Moreover, this image generation technique is also quite novel and is the latest implementation for this system [12,13].

A traditional detection system is mainly based on deep learning and machine learning when it comes to the classification and detection techniques used [6,14,15]. In previous studies, especially those involving complex violence detection scenes, small images, constant continuous monitoring, and the lack of real-time processing were challenges and limitations. This lack within models created a struggle in training, leading to false results. So, this was the main disadvantage of these previous systems. Then, in further studies on object detection ideas, different algorithms are being implemented like YOLO v3, YOLO v4, and YOLO v5 [16–18]. YOLO v3 is the improved version of the previous model YOLO; it is a powerful algorithm for real-time object detection. It uses Darknet-53 as the backbone and has 53 convolution layers in its architecture. One benefit is that it uses multiple scales for detecting objects of different sizes. The limitation in YOLO v3 is that it struggles with small object detection and has a slower processing speed. To overcome the limitations of this algorithm, YOLO v4 was proposed in 2020 to improve the performance of YOLO v3 [18–20]. YOLO v4 improves the feature representation and detection of the objects using advanced techniques like a pseudo-attention network (PAN), path aggregation network (PANet), and spatial attention module (SAM), which improve both accuracy and the speed of detection ability [18,21]. However, the limitation when working with small objects has not been overcome. In the same year, 2020, a new object detection algorithm was again introduced with the name of YOLO v5 and provided a lightweight architecture in comparison to the previously discussed version. It utilizes a single-stage detector based on a modified version of the EfficientNet network, attaining a balance between accuracy and speed and providing compatibility with numerous real-time applications. But this model faces the limitation of low accuracy in small object detection because of a lack of information. The performance has not proven sufficient due to semantic information [22,23]. Further, another model, which is YOLO v7, which overcomes all previous model limitations such as those related to small object detection and slow processing speed and provides a lighter model architecture, is being studied. These features enhance the performance of YOLO v7. YOLO v7 has the benefit of taking input in red–green–blue (RGB) image format with an augmentation technique for better training. The backbone layer architecture of YOLO v7 has a robust feature extraction layer network that is the same as the previous versions such as the residual network (ResNet) and EfficientNet, with an additional robust-feature hierarchical structure. The neck is the intermediate layer network that enhances the feature extraction of the backbone network by using the attention mechanism or context aggregation module and feature fusion module. The final network of prediction has multiple detection heads to generate the bounding boxes for small violent object detection with highly accurate performance. The detailed architecture is discussed in a further subsection on the objection detection method to understand the mechanism.

Another is the pose estimation model. In the traditional method, 3-dimensional convolutional neural networks (3D CNNs) are commonly used for action recognition and are bound to 3-dimensional (3D) inputs. This model is employed for video analysis. When using a 3D CNN, spatial and temporal dimensionality can be removed from the

model [24,25]. The computational complexity, memory resources, huge number of data requirements, and preprocessing are limitations or challenges associated with 3D CNNs. However, the development of technologies also provides many estimation models such as alphapose, the high-resolution network (HRNet), and MediaPipe. In alphapose, complex problems can be solved, and one can compute 2D and 3D poses together. This model can achieve both multimodal fusion as well as enhancement in 3D pose estimation with the advanced techniques of attention mechanisms [26]. In the HRNnet, complex problems and multi-person poses can be estimated with the advancement method of attention mechanisms. This model also has the feature of adaptive resolution [27]. These model advancements and methods are quite impressive, but here, the MediaPipe model has more features and is easy to use due to the familiar interface and real-time performance.

In this paper, we propose a novel architecture for violence detection and classification to better manage criminal activity within smart cities and enhance smart city safety. The proposed architecture combines an image-to-image stable diffusion model with violence detection, pose estimation, and a long short-term memory (LSTM) network as shown in Figure 1. The diffusion model generates synthetic data while the object detection approach uses YOLO v7 to identify violent objects like baseball bats, knives, and pistols. YOLO v7, which is the latest version of the YOLO family, provides high-speed and accurate prediction results in comparison to the previously developed model. The YOLO v7 model structure is quite similar to that of YOLO v5, including features such as an FPN, backbone, PAN, and head scales of different kinds [22,23]. The structure has ELAN, convolutional layers, two-stride downsampling, and max-pooling. Moreover, MediaPipe is utilized for violent action pose detection. The pose estimated using MediaPipe has every possible key point to detect the pose. Further, a long short-term memory (LSTM) network classifies the action attacks involving violent objects by using the recorded key points as the input. Subsequently, the LSTM network undertakes classification using these obtained body key points or features. Finally, the trained model is deployed into the edge device for testing the performance of the proposed model using unseen test data.

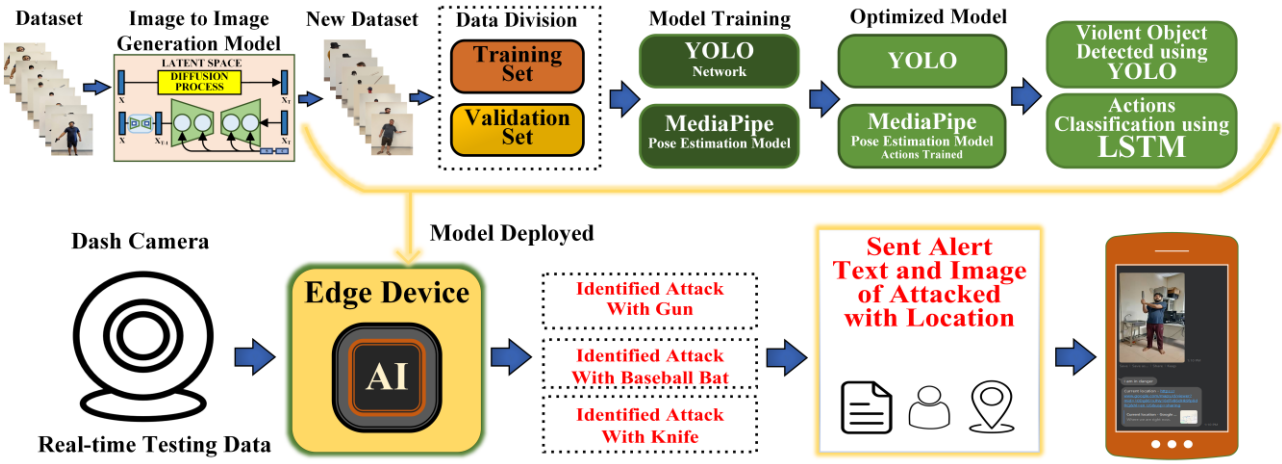

Figure 1. The proposed model of attack alerting system—setup involving an image-to-image generation model, object detection model (YOLO v7), pose estimation model (MediaPipe), and action classification using the LSTM model.

In general, the main contributions of this proposed work are mentioned here as follows:

1. Our proposed model can handle the small dataset problem using the stable diffusion image generative method, in which new image samples can be generated using previous images to increase the number of images for the object detection model to enhance the performance.

2. Our model architecture combines violent object detection (YOLO v7) and pose estimation models (MediaPipe) and an LSTM classifier to improve the performance of the violent attack detection system.
3. An edge computing device is implemented and the whole model is deployed in the computing device to test the model using violent-attack testing data in the city.
4. A commercial social media API is implemented here for sending the violent object and criminal clip as an alert to the registered number.

Further, the rest of this paper is explained in the following sections: Section 2 describes the methodology part that has a brief explanation of the proposed model architecture, involving steps such as data collection, data labeling, the image-to-image data synthesis method, the object detection model, the pose estimation model, the LSTM classification method, the edge computing device, and the attack alerting method. Section 3 presents the results of the paper and Section 4 presents the conclusion part of the research work.

2. Methodology

2.1. Dataset

In this study, data collection was the first part, in which the data were collected from numerous resources, with around 735 images collected from the internet source [28] and around 1365 images collected from Kaggle [29,30]. However, the data we gathered from the above sources were not efficient for training our model, so we proposed a novel stable diffusion approach for generating more image data as shown in the top-left part of Figure 1. The next subsection explains image-to-image diffusion model to generate synthetic images.

Image-to-Image Stable Diffusion Pipeline Method

The image-to-image stable diffusion model [31] was first developed by StabilityAI, CompVis, runway, and LAION. This model generates synthetic or new images using text prompts and clips. Model idea addition associates the noise with the images and follows the Markov chain, continuing to do this step-by-step. After the lapse of time steps, images are converted totally into noise and the noise approaching [31,32] method is halted. This is known as noising or forward diffusion. Now, another diffusion concept is introduced here, and in this process, a complete noise image is used to generate a new image using the reverse method. Here, time steps are subtracted gradually from the image. This process is known as denoising or reverse diffusion [33].

1. Forward diffusion (noising)

$$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots x_T$$

Take a data distribution $x_0 \sim p(x)$ and turn it into noise using the diffusion $x_T \sim N\ (0, \sigma^2 I)$.

2. Reverse diffusion (denoising)

$$x_T \rightarrow x_{T-1} \rightarrow x_2 \rightarrow \cdots x_0$$

Take a data distribution $x_T \sim N\ (0, \sigma^2 I)$ and turn it into noise using the diffusion $x_0 \sim p(x)$.

Figure 2 shows the sample images generated using stable diffusion to resolve the dataset limitation issue in the object detection system. The total dataset for training our proposed model that was generated using stable diffusion and includes real data collected through a self-generated source, from the internet, and from Kaggle is shown in Table 1. Further, the dataset was annotated using labelImg. labelImg is an annotation tool that is used to annotate or label the data to train the YOLO model. For each class, 700 images were real and 300 images were generated images. Then, the final dataset was divided into ratios for training at about 70% and validation sets at about 30%.

Figure 2. Generated synthetic image samples of image-to-image stable diffusion.

Table 1. Dataset.

Class Label	Number of Images (Real + Generated)
Baseball bat	700 + 300
Gun	700 + 300
Knife	700 + 300
Total	3000
Training size	2100 (70%)
Validation size	900 (30%)

2.2. Violence Object Detection Model (YOLO v7)

In the above section, we showed how synthetic images and previous images are associated and how we created new datasets that were used in the violence detection model (YOLO v7) as shown in Figure 1. This is a one-stage detector abbreviated as "You Only Look Once". YOLO v7 [34,35] is an object detection method in computer vision designed to address various challenges in real-time image processing. This algorithm maintains high accuracy during the real-time processing speed of 5 to 160 frames per second (FPS) and is the latest detection technique of this era that is highly in demand for using various kinds of solutions [36,37]. Here, we are solving the problem of citizen safety in cities. In this research work, YOLO v7 is used to detect three kinds of objects: baseball bats, pistols, and knives, which are used here as violent object detection model training. The architecture of YOLO v7 is mainly divided into four subsections as follows:

- **Input:** This is the initial stage of this model in which input comprising violent images is provided to an algorithm with the images' corresponding annotations; the size of each input image is 416 × 416 and the images are RGB images that provide their output to the next backbone layer architecture.
- **Backbone:** The backbone layer networks are processed after input images and mainly comprise three subsections of these modules: MPI module, E-ELAN, and CBS. The MPI model is a combination of CBS processes and MaxPool, with bottom and top branches. The MaxPool model is at the top branches and is utilized to decrease the image's size in bisection, in both length and width. A CBS process with 128 channel outputs is also utilized to minimize the channel of image sum by fifty per cent and conversely CBS process with a stride and 1 × 1 kernel divides the channels in half numbers. Afterwards, another 2 × 2 stride and 3 × 3 kernel CBS process divide the image dimension in half. Concatenation (Cat) is employed to incorporate the extracted features from that pair of branches. CBS handles the collection of the data from small-scale areas and MaxPool collects from localized locations. The integration techniques of the network raise the capacity to extract useful features from input images.

- **Neck:** This section of YOLO layer architecture consists of FPN structure (stands for feature pyramid network structure) that employs PAN design structure. The network is composed of many convolutional networks, SiLU activation (CBS Block), and Batch normalization along with spatial pyramid pooling (SPP) and the convolutional spatial pyramid (CSP) that improves outcomes of layers, and this network structure extends Maxpool2 (MP2) and efficient layer aggregation network (ELAN). The number of output channels is always the same in both the MP blocks—the output of this neck layer network transfers to the next prediction module.
- **Prediction:** The prediction stage is the final stage of this detection algorithm and has a couple of rep structures. The confidence, anchor, and category are evaluated or predicted using a 1×1 convolutional layer. The inspiration for this kind of rep structure is VGG or Darknet, which decreases the model complexity without reducing its prediction performance.

For training this model, the main specifications are mentioned as follows: CPU was Intel(R) Core (TM) i7-9700k speed @ 3.60 Hz, installed RAM was 16 GB, and graphics card was NVIDIA GeForce RTX 2080Ti.

2.3. Hyperparameter of Model

Table 2 plays a curious role in obtaining the best performance of YOLO v7. It supports the search for optimal outcomes for the model network.

Table 2. Experimental model specification.

Parameters	Value
Learning rate	1×10^{-5}
Momentum	0.98
Weight decay	0.001
Batch size	16
Optimizer	Adam
Dimensions	416×416
Epochs	200

After the setup, the procedure was initialized by conscientiously arranging the pattern of the model and then proceeding to train the model from these pre-decided hyperparameters.

2.4. Violent Pose Estimation Model

The pose estimation model detects the action or movement of the body. MediaPipe is implemented here to achieve violent action as shown in Figure 1. The MediaPipe open-source framework, developed by Google, is used in this context. This provides a platform for creating real-time pipelines of multimedia processes [38]. A total of 33 key points are being covered by the MediaPipe of human bodies in a video or image [39]. These key points are the different locations on the body that analyze the position and help in movement categorization. The three types of action are trained here using this model and each action is trained in hundreds of parameters to achieve the accurate target pose. These actions are trained for violent objects. The input of images accepts the data types in the following form: images, video frame (which should be decoded) form, and real-time video feed. The output of the task provides image coordinates and 3-dimensional coordinates of the world [40].

2.5. Violent Pose Classification Model

In the previous section (i.e., Section 2.4), we explained how violent poses or actions are detected; the next step is the classification of these actions, which is achieved using the LSTM classifier. LSTM network is a kind of recurrence neural network (RNN) architecture also known as RNN [41–43]. This is the architecture used to solve sequential data tasks and mainly focuses on the problem of vanishing gradient. Here, architecture has three dense layers and three LSTM layers or hidden layers; each layer is shown below in Figure 3.

The input shape is taken (30,132), in which 30 is the sequence length or time steps and 132 denotes the number of dimensions or features. Subsequently, the first hidden layer is the lstm_input layer, which has an input of 1024 and 512 units as output, and the second hidden layer is lstm_1, which has an input of 512 and 256 units as output; the return sequence is true and the activation function is relu. The third layer is lstm_2, which has an input of 256 and 128 units as output; the return sequence is false and the activation function is relu. Now, the dense layers are fully connected, and the first layer and second dense layer have 64 and 32 units with relu activation function, respectively. The final dense layer has an equal number of units and actions provided in the dataset. This dense layer activation function is softmax, which can also be used for multiclass classification problems. The learning rate is 0.000005 and the optimizer is Adam. The loss function was categorical cross entropy at the time of the training process of the model. These are the suitable parameters that provide the best performance to classify the violent action pose.

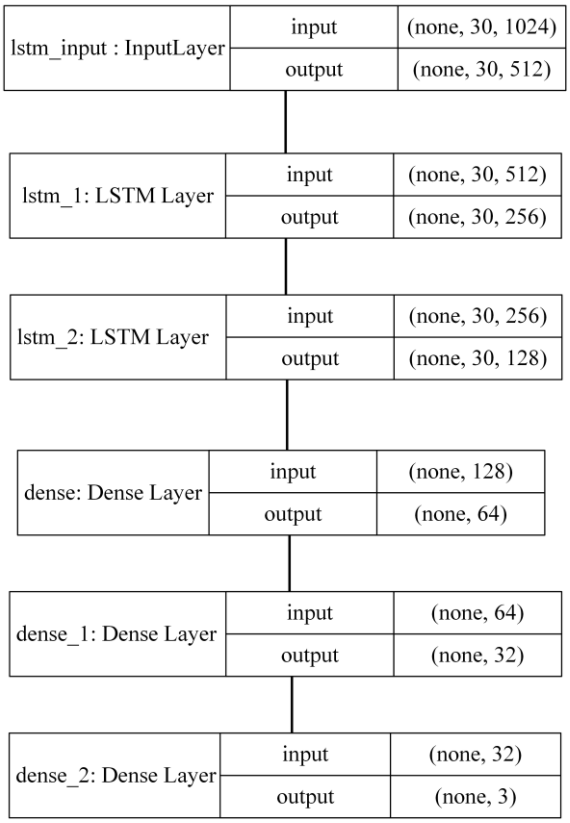

Figure 3. LSTM model architecture.

2.6. Edge Computing Device and Attack Alerting Method

In this paper, we consider a commercial edge computing device, and we deployed the whole model involving YOLO, MediPipe, and LSTM models in this device for testing the proposed model using real unseen violent attack testing data in the city as shown in Figure 1. Edge devices [2] are computing devices that constitute specific hardware platforms designed for implementing and accomplishing computing tasks. These devices can be either fixed or portable, serving computing purposes. They enable local processing, thereby improving efficiency and reducing latency, storage, and data analysis [44,45]. This series of devices enhances performance by leveraging powerful GPU computing capabilities. Furthermore, after detecting violent attacks and attack activity using the proposed model in the edge device, we used commercial social media and integrated it into the edge device for sending an alert message to the mobile of the victim. The attack alert service provided by commercial social media apps allows for the delivery of notifications like image messages and sends notifications to social media app users [46] without using or creating full-fledged

bots. In this study, we used and obtained the first commercial social media app notification access token and replaced the variable value of the 'token' along with the commercial social media app notification access token about which we provided information in the previous process step. The message can also be modified by replacing the value of the variable 'message' that the text author or victim wants to deliver to the registered commercial social media app account along with images of criminals with crime spots and violent objects.

3. Results

In this section of the paper, we study the detection of the proposed YOLO v7 model and pose estimation model with defined tuning parameters that provide the outcomes on the validation dataset; the further subsection outlines the performance matrices of the model as follows.

3.1. Detection of YOLO v7 Model and Pose Estimation Model

This subsection shows the detection result of the YOLO v7 model and the pose estimation model's action for violent objects. Violent objects include baseball bats, knives, and pistols. A violent target object is detected using the different-color bounding box and the pose is estimated through MediaPipe key points. Each key point is estimated according to the body posture. The below figures show the prediction result using an object detection algorithm.

Figure 4 shows the real-time prediction or test results of the object detection with bounding boxes on the objects in different colors using the unseen dataset from a dash camera. In the prediction result, different kinds of the condition are targeted. In Figure 4a,d, the green-color bounding boxing shows the target images that show the violent object, a baseball bat; Figure 4b shows the violent object, a gun, in the red bounding box; and the other figures, Figure 4c,e, also indicate that a knife was detected in yellow-color bounding boxes in the image frames with a different standing view. These results cover possible different conditions to check the performance of the proposed model. The further figures below display the pose estimation prediction results.

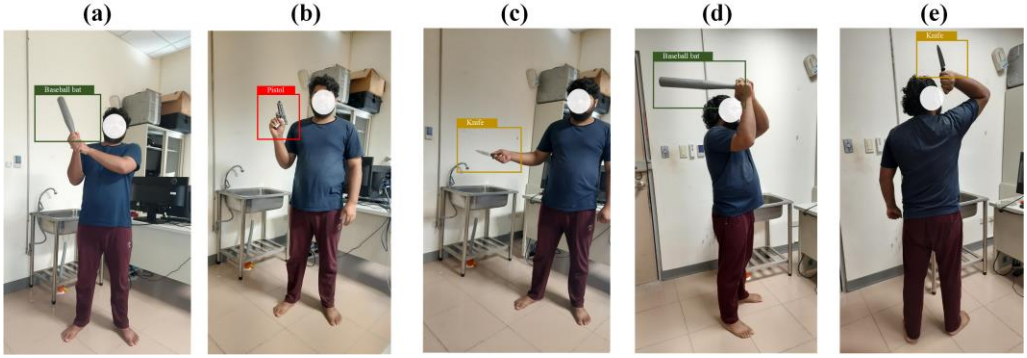

Figure 4. The proposed YOLO v7 model—real-time predicted results in different attack-type conditions: (**a**,**d**) show a detected violent object, a baseball bat; (**b**) shows a detected violent object, a gun; (**c**,**e**) show a detected violent object, a knife.

In Figure 5, the pose estimation of the main three types of action is covered at the time of the development of MediaPipe. These actions are related to the object detected in the YOLO model. These are the types of actions: in Figure 5a, the first action shows that criminals are trying to attack with a baseball bat from the right side of the victim; in Figure 5b, the second action shows that a criminal is willing to attack with a knife in his left hand; Figure 5c shows the third action, which provides the action with a gun; Figure 5d shows that the criminal can be detected even from a flipped-back position; and Figure 5e shows the side portion of the criminal who is attacking. These are different scenarios in which a criminal is using violence to attack a victim. These test predictions were predicted in real time using real test data from a dash camera. All actions were trained

in the various possible kinds of positions and postures for highly accurate detection. Here, we snapped 100 videos of each action. Every position has been stored in the dataset of MediaPipe. Further, the LSTM classification method is used to classify the actions. In the next subsequent section of the model, evaluation metrics show the performance of YOLO v7 and the pose estimation model along with the LSTM performance.

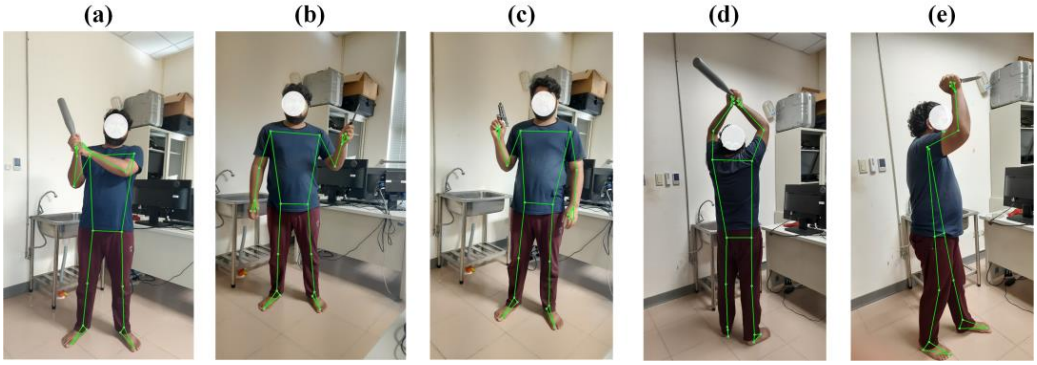

Figure 5. Violent attack pose estimation using MediaPipe: (**a**,**d**) show an attacking action with a baseball bat; (**b**,**e**) show an attacking action with a knife; (**c**) show an attacking action with a gun.

3.2. Performance Metrics of Model

The performance evaluation of the proposed algorithm for this system involves the use of various metrics to assess its accuracy and effectiveness in detecting violent objects. The metrics employed include precision, recall, intersection over union (IoU) at a threshold value of 0.5, mean average precision (mAP), average precision (AP), and mean [20]. The IoU metric measures the degree of overlap between the predicted bounding boxes and the ground-truth bounding boxes in the actual data. This threshold is a critical parameter in evaluating the model's ability to accurately delineate object boundaries, contributing to the robustness of its predictions. It assesses the performance of the output of the algorithm aligned with the actual data. Precision is the identification proportion of correct positive predictions while recall is the correct identification of actual positive predictions. These metrics ensure that the model's predictions are both valid and accurate in classifying violent objects. The crucial metric is mean average precision (mAP), which is extensively employed in target detection. This involves the maximum computation of average precision across multiple classes, denoted as 'k' [22]. The process begins by calculating the average precision for each class, and subsequently, the mean of all these average precisions is computed, providing a comprehensive measure of the mean average precision [35,37].

This section discusses the proposed model evaluation performance for violent objects. In Figure 6a, the three classes are accurately classified using YOLO v7, and each class's separate accuracy is in the confusion matrix. Each class outcome is presented as follows: baseball bat, knife, and pistol, along with false positive and false negative background detection results. In addition to that, in Figure 6b, a precision recall curve evaluates the detection algorithm performance correctly to identify the positive detection with fewer false positives. In the evaluation result, each class performance is shown with separate results and along with final average precision. The curve also raises the focus on the ratio of true and actual positives and the area under the curve (AUC) shows the performance of the evaluated model. The precision recall curve for the YOLO v7 model demonstrates exceptional performance in distinguishing between three classes: baseball bats, knives, and pistols, achieving individual accuracies of 84.5%, 94.7%, and 89.2%, respectively. This curve also shows the relation between and performance in terms of recall and precision for each class separately. The mean average precision (mAP), calculated as the mean average precision across all classes, is reported at 89.5% as shown in Figure 6b. The evaluation of the model's performance shows an outstanding level of accuracy, boosting the overall model performance to enhance the violence detection system.

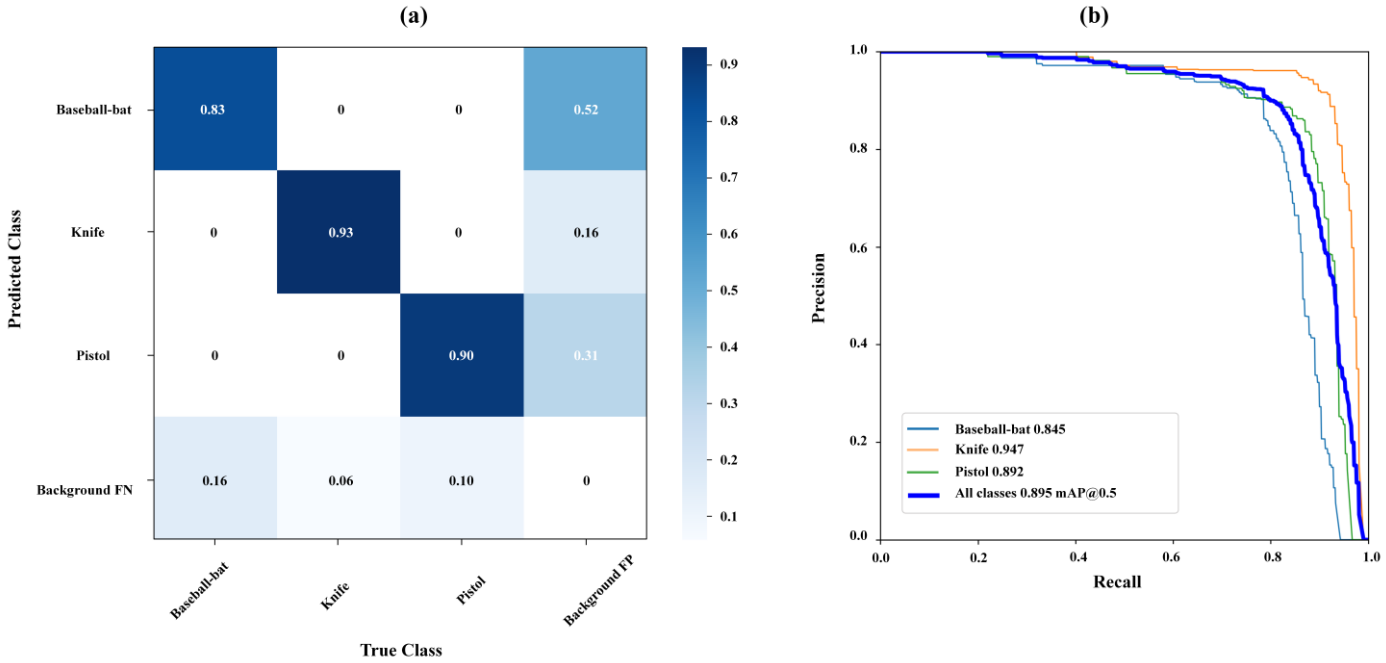

Figure 6. (**a**) The confusion matrix; (**b**) the precision recall curve for attack types or divergent classes.

In Figure 7, the overall performance of the object detection model during model training and validation loss flow in a single graph plot reflects the outcomes of the training loss of the box with the validation loss of the box concerning the increasing number of epochs, objectness training with the validation loss of objectness, and classification loss curve with the validation classification loss curve. This plotted graph also shows the separate precision with the mean average precision curve with the threshold along with different per cent ratios and recall for understanding the model performance. The loss is a consistent decrease in loss throughout training and validation with defined tuning parameters. The training loss for bounding boxes stabilized around 0.02, while the validation loss for bounding boxes settled at 0.04. For objectness, the training loss was 0.04, with a validation loss of 0.055. Classification loss remained at 0.001 during training and dropped to 0.0028 during validation. On the other hand, the individual precision, recall, and mean average precision (mAP) at thresholds of 0.5 and 0.95 were evaluated with respect to the increasing number of epochs. These graphs depict the performance at each epoch. The graph depicting precision and recall illustrates an increasing accuracy concerning the number of epochs. The average precision graph at a threshold of 0.5 initially showed instability, but after 50 epochs, precision accuracy steadily increased, maintaining stable performance. Similarly, at the threshold of 0.5:95, a similar pattern emerged but stabilized around 30 epochs. These graphs collectively depict the model's accurate performance at individual points.

Figure 8 displays the performance metrics of the action classification using the LSTM model based on pose estimation. This performance evaluation reflects the model's ability to classify actions trained using MediaPipe pose estimation outputs. The three actions of the validation results are shown in the graph as follows: baseball bat action, knife action, gun action, and no action. Each separate class and average of action performance is evaluated. The accuracy values of these three action classes are 96%, 82%, and 87% for the baseball bat, knife, and gun violent actions. These performances are quite impressive for this type of complex model as depicted in the figure below.

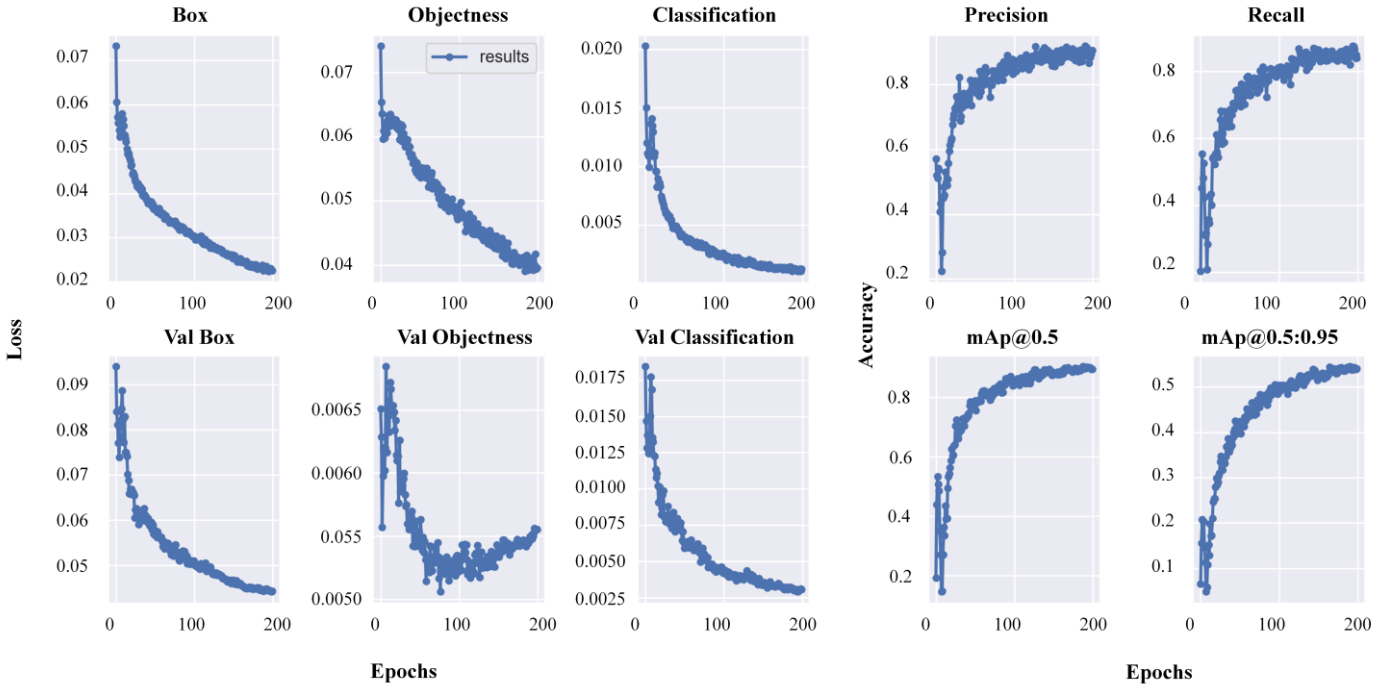

Figure 7. The YOLO v7 main metrics of the training and validation loss and accuracy of the model.

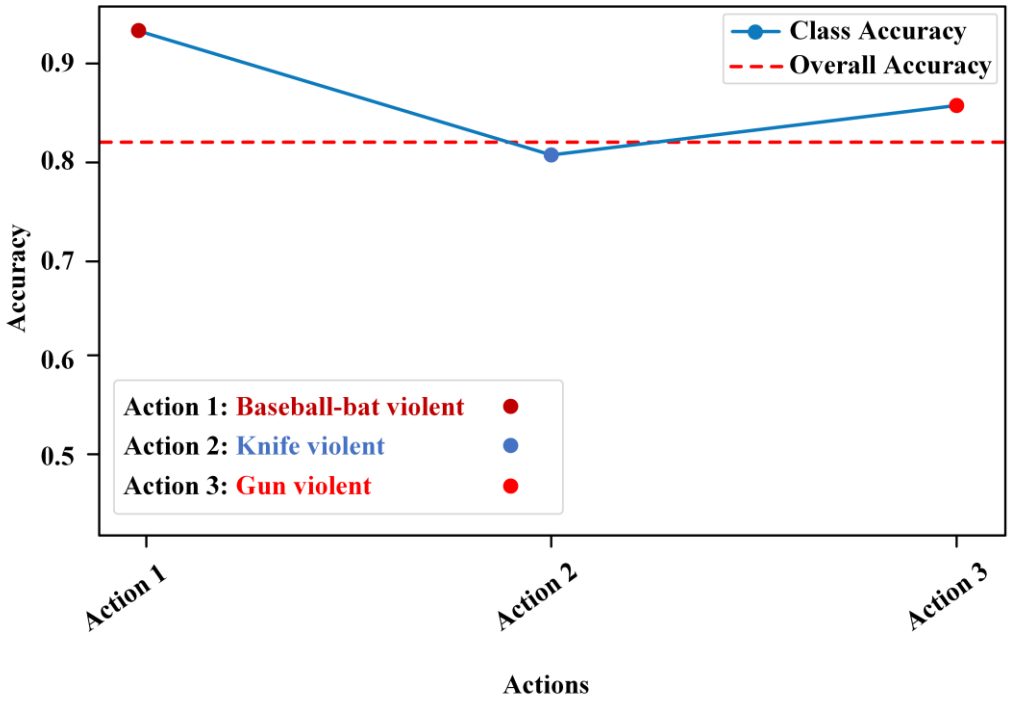

Figure 8. Performance of violence action classification in MediaPipe using LSTM.

Moreover, we compare the detection performance of the proposed YOLO v7 model with other violent attack detection approaches using different data and within different environments as presented in Table 3. In this comparison, we consider previous studies that focused on object detection. As shown in the table, our proposed model's accuracy is higher than those of others. Therefore, the results prove that the proposed model has the capacity to accurately detect violent attacks and violent actions.

Table 3. Comparison table with different models. Our proposed model has higher accuracy.

Model (Backbone)	Accuracy (mAP@0.5)
YOLO v7 + Pose estimation [47]	79.66%
Faster R-CNN (RegNet+) [48]	79.78%
YOLO v5 (CSPDarkNet 53) [48]	77.26%
YOLO v3 (DarkNet53) [49]	84%
YOLO v4 [49]	85%
Our proposed model	**89.5%**

4. Conclusions

This study presented a novel approach to improve violent attack detection systems by integrating advanced artificial intelligence techniques to enhance the safety of smart cities. The proposed model integrates image-to-image stable diffusion, YOLO v7, MediaPipe, and an LSTM classifier with an edge computing device, demonstrating significant advancements in real-time violence detection. The innovative use of image-to-image stable diffusion addresses the challenge of handling small datasets by generating synthetic data, contributing to enhancing the model performance. Meanwhile, the YOLO v7 object detection approach helps to identify violent attack objects like baseball bats, knives, and pistols. Moreover, MediaPipe is utilized for violent action pose detection and a long short-term memory (LSTM) network classifies the action of attacks involving violent objects by using the recorded key points using MediaPipe as the input. Finally, the trained model has been deployed into the edge device for testing the proposed model using unseen test data in the smart city. The object detection accuracy values of the YOLO v7 model for three classes, baseball bat, knife, and pistol, are 84.5%, 94.7%, and 89.2%, respectively. The mean average precision (mAP) of YOLO v7 for all classes is 89.5% at a threshold of 0.5. Moreover, the accuracy of the LSTM model for classifying the baseball bat action is 96%; for the knife action, it is 82%, and for the gun action, it is 87%. Therefore, our proposed integration of YOLO v7, MediaPipe, and the LSTM model improves the violence detection performance and enhances the safety of smart cities.

Author Contributions: Conceptualization, P.K., G.-L.S., B.-L.G. and P.-C.P.; methodology, P.K., G.-L.S., B.-L.G., S.K.N. and P.-C.P.; data preparation, P.K., G.-L.S., B.-L.G., S.K.N., M.A.A. and P.-C.P.; software, P.K., G.-L.S., B.-L.G. and P.-C.P.; model validation, P.K., G.-L.S., B.-L.G. and P.-C.P.; formal analysis, P.K., G.-L.S., S.K.N., C.-K.Y., Y.C.M., M.A.A. and P.-C.P.; investigation, P.K., G.-L.S., B.-L.G., S.K.N., Y.C.M., C.-K.Y., M.A.A. and P.-C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science and Technology Council, Taiwan, under Grant NSTC 112-2221-E-027-076-MY2.

Data Availability Statement: The data presented in this study are available in this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Baba, M.; Gui, V.; Cernazanu, C.; Pescaru, D. A Sensor Network Approach for Violence Detection in Smart Cities Using Deep Learning. *Sensors* **2019**, *19*, 1676. [CrossRef] [PubMed]
2. Bai, T.; Fu, S.; Yang, Q. Privacy-Preserving Object Detection with Secure Convolutional Neural Networks for Vehicular Edge Computing. *Future Internet* **2022**, *14*, 316. [CrossRef]
3. Ali, S.A.; Elsaid, S.A.; Ateya, A.A.; ElAffendi, M.; El-Latif, A.A.A. Enabling Technologies for Next-Generation Smart Cities: A Comprehensive Review and Research Directions. *Future Internet* **2023**, *15*, 398. [CrossRef]
4. Ullah, F.U.M.; Ullah, A.; Muhammad, K.; Haq, I.U.; Baik, S.W. Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network. *Sensors* **2019**, *19*, 2472. [CrossRef] [PubMed]
5. Aremu, T.; Zhiyuan, L.; Alameeri, R.; Khan, M.; Saddik, A.E. SSIVD-Net: A novel salient super image classification & detection technique for weaponized violence. *arXiv* **2022**, arXiv:2207.12850.
6. Jebur, S.A.; Hussein, K.A.; Hoomod, H.K.; Alzubaidi, L. Novel Deep Feature Fusion Framework for Multi-Scenario Violence Detection. *Computers* **2023**, *12*, 175. [CrossRef]

7. Vosta, S.; Yow, K.-C.A. CNN-RNN Combined Structure for Real-World Violence Detection in Surveillance Cameras. *Appl. Sci.* **2022**, *12*, 1021. [CrossRef]
8. Alrashedy, H.H.N.; Almansour, A.F.; Ibrahim, D.M.; Hammoudeh, M.A.A. BrainGAN: Brain MRI Image Generation and Classification Framework Using GAN Architectures and CNN Models. *Sensors* **2022**, *22*, 4297. [CrossRef]
9. Nichol, A.Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; Mcgrew, B.; Sutskever, I.; Chen, M. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, ML, USA, 17–23 July 2022; pp. 16784–16804.
10. Kawar, B.; Zada, S.; Lang, O.; Tov, O.; Chang, H.; Dekel, T.; Mosseri, I.; Irani, M. Imagic: Text-based real image editing with diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 6007–6017.
11. Avrahami, O.; Lischinski, D.; Fried, O. Blended diffusion for text-driven editing of natural images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 18208–18218.
12. Borji, A. Generated faces in the wild: Quantitative comparison of stable diffusion, mid-journey and dall-e 2. *arXiv* **2022**, arXiv:2204.06125.
13. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
14. Xin, Y.; Kong, L.; Liu, Z.; Chen, Y.; Li, Y.; Zhu, H.; Gao, M.; Hou, H.; Wang, C. Machine learning and deep learning methods for cybersecurity. *IEEE Access* **2018**, *6*, 35365–35381. [CrossRef]
15. Khan, S.U.; Haq, I.U.; Rho, S.; Baik, S.W.; Lee, M.Y. Cover the Violence: A Novel Deep-Learning-Based Approach Towards Violence-Detection in Movies. *Appl. Sci.* **2019**, *9*, 4963. [CrossRef]
16. Maity, M.; Banerjee, S.; Sinha, C.S. Faster R-CNN and YOLO based Vehicle detection: A Survey. In Proceedings of the 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 1442–1447.
17. Liu, K.; Tang, H.; He, S.; Yu, Q.; Xiong, Y.; Wang, N. Performance validation of YOLO variants for object detection. In Proceedings of the 2021 International Conference on Bioinformatics and Intelligent Computing, Harbin, China, 22–24 January 2021; pp. 239–243.
18. Hussain, M. YOLO-v1 to YOLO-v8: The Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines* **2023**, *7*, 677. [CrossRef]
19. Chen, D.; Ju, Y. SAR ship detection based on improved YOLOv3. In Proceedings of the IET International Radar Conference (IET IRC 2020), Online, 4–6 November 2020; pp. 929–934.
20. Li, Y.; Zhao, Z.; Luo, Y.; Qiu, Z. Real-Time Pattern-Recognition of GPR Images with YOLO v3 Implemented by Tensorflow. *Sensors* **2020**, *20*, 6476. [CrossRef] [PubMed]
21. Wahyutama, A.B.; Hwang, M. YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins. *Electronics* **2022**, *11*, 1323. [CrossRef]
22. Zhou, F.; Deng, H.; Xu, Q.; Lan, X. CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote Sensing Images. *Electronics* **2023**, *12*, 2671. [CrossRef]
23. Xiao, Y.; Chang, A.; Wang, Y.; Huang, Y.; Yu, J.; Huo, L. Real-time Object Detection for Substation Security Early-warning with Deep Neural Network based on YOLO-V5. In Proceedings of the IEEE IAS Global Conference on Emerging Technologies (GlobConET), Arad, Romania, 20–22 May 2022; pp. 45–50.
24. Fan, L.; Rao, H.; Yang, W. 3D Hand Pose Estimation Based on Five-Layer Ensemble CNN. *Sensors* **2021**, *21*, 649. [CrossRef]
25. Luvizon, D.C.; Picard, D.; Tabia, H. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5137–5146.
26. Fang, H.S.; Li, J.; Tang, H.; Xu, C.; Zhu, H.; Xiu, Y.; Li, Y.L.; Lu, C. AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 7157–7173. [CrossRef]
27. Yu, C.; Xiao, B.; Gao, C.; Yuan, L.; Zhang, L.; Sang, N.; Wang, J. Lite-HRNet: A Lightweight High-Resolution Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10440–10450.
28. Guns-Knives Object Detection Dataset. Available online: https://www.kaggle.com/datasets/iqmansingh/guns-knives-object-detection (accessed on 14 June 2023).
29. Baseball Bat Dataset. Available online: https://images.cv/dataset/baseball-bat-image-classification-dataset (accessed on 14 June 2023).
30. Narejo., S.; Pandey, B.; Esenarro Vargas, D.; Rodriguez, C.; Anjum, M.R. Weapon Detection Using YOLO V3 for Smart Surveillance System. *Math. Probl. Eng.* **2021**, *2021*, 9975700. [CrossRef]
31. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2022; pp. 26–30.
32. Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; Guo, B. Vector quantized diffusion model for text-to-image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10696–10706.

33. Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*; Red Hook Inc.: Brooklyn, NY, USA, 2022; Volume 35, pp. 36479–36494.
34. Hemmatirad, K.; Babaie, M.; Afshari, M.; Maleki, D.; Saiadi, M.; Tizhoosh, H.R. Quality Control of Whole Slide Images using the YOLO Concept. In Proceedings of the IEEE 10th International Conference on Healthcare Informatics (ICHI), Rochester, MN, USA, 11–14 June 2022; pp. 282–287.
35. Wang, Y.; Wang, H.; Xin, Z. Efficient Detection Model of Steel Strip Surface Defects Based on YOLO-V7. *IEEE Access* **2022**, *10*, 133936–133944. [CrossRef]
36. Kaiyue, L.; Qi, S.; Daming, S.; Lin, P.; Mengduo, Y.; Nizhuan, W. Underwater Target Detection Based on Improved YOLOv7. *J. Mar. Sci. Eng.* **2023**, *3*, 677.
37. Kumar, P.; Shih, G.-L.; Yao, C.-K.; Hayle, S.T.; Manie, Y.C.; Peng, P.-C. Intelligent Vibration Monitoring System for Smart Industry Utilizing Optical Fiber Sensor Combined with Machine Learning. *Electronics* **2023**, *12*, 4302. [CrossRef]
38. Chen, K.-Y.; Shin, J.; Hasan, M.A.M.; Liaw, J.-J.; Yuichi, O.; Tomioka, Y. Fitness Movement Types and Completeness Detection Using a Transfer-Learning-Based Deep Neural Network. *Sensors* **2022**, *22*, 5700. [CrossRef]
39. MediaPipe: Pose Landmark Detection Guide. Available online: https://developers.google.com/mediapipe (accessed on 14 November 2023).
40. Zeng, Y.; Ye, W.; Stutheit-Zhao, E.Y.; Han, M.; Bratman, S.V.; Pugh, T.J.; He, H.H. MEDIPIPE: An automated and comprehensive pipeline for cfMeDIP-seq data quality control and analysis. *Bioinformatics* **2023**, *39*, btad423. [CrossRef]
41. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A Tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv* **2019**, arXiv:1909.09586.
42. Zhou, C.; Sun, C.; Liu, Z.; Lau, F.C.M. A C-LSTM Neural Network for Text Classification. *arXiv* **2015**, arXiv:1511.08630.
43. Ghourabi, A.; Mahmood, M.A.; Alzubi, Q.M. A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages. *Future Internet* **2020**, *12*, 156. [CrossRef]
44. Mittal, S. A Survey on Optimized Implementation of Deep Learning Models on the NVIDIA Jetson Platform. *J. Syst. Archit.* **2019**, *97*, 428–442. [CrossRef]
45. Shi, Z. Optimized Yolov3 Deployment on Jetson TX2 with Pruning and Quantization. In Proceedings of the 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC), Greenville, SC, USA, 12–14 November 2021; pp. 62–65.
46. Chumuang, N.; Hiranchan, S.; Ketcham, M.; Yimyam, W.; Pramkeaw, P.; Tangwannawit, S. Developed Credit Card Fraud Detection Alert Systems via the Notification of LINE Application. In Proceedings of the 2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Bangkok, Thailand, 18–20 November 2020; pp. 1–6.
47. Kumar, P.; Li, C.-Y.; Guo, B.-L.; Manie, Y.C.; Yao, C.-K.; Peng, P.-C. Detection of Acrimonious Attacks using Deep Learning Techniques and Edge Computing Devices. In Proceedings of the 2023 International Conference on Consumer Electronics—Taiwan (ICCE-Taiwan), Pingtung, Taiwan, 9–11 July 2023; pp. 407–408.
48. Tang, Y.; Chen, Y.; Sharifuzzaman, S.A.; Li, T. An automatic fine-grained violence detection system for animation based on modified faster R-CNN. *Expert Syst. Appl.* **2024**, *237*, 121691. [CrossRef]
49. Tufail, H.; Nazeef, U.H.; Muhammad, F.; Muhammad, S. Application of Deep Learning for Weapons Detection in Surveillance Videos. In Proceedings of the 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), Islamabad, Pakistan, 20–21 May 2021; pp. 1–6.