IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 July 2024, accepted 9 August 2024, date of publication 15 August 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444604

RESEARCH ARTICLE

SafeguardNet: Enhancing Corporate Safety via Tailored Deep Transfer Learning for Threat Recognition

NUSRAT JAHAN[1], MOHAMMAD SAYEM CHOWDHURY[1], (Associate Member, IEEE), TOFAYET SULTAN[1], M. F. MRIDHA[1], (Senior Member, IEEE), AND MD SADDAM HOSSAIN MUKTA[2], (Member, IEEE)

[1]Computer Science and Engineering, American International University-Bangladesh (AIUB), Dhaka 1229, Bangladesh
[2]LUT School of Engineering Sciences, LUT University, 53850 Lappeenranta, Finland

Corresponding authors: M. F. Mridha (firoz.mridha@aiub.edu) and Md Saddam Hossain Mukta (saddam.mukta@lut.fi)

ABSTRACT In today's rapidly evolving corporate environments, ensuring comprehensive security measures is paramount. This paper presents SafeguardNet, a deep transfer learning-based model designed to enhance corporate safety through effective multiclass threat detection. Recognizing the limitations of existing binary threat detection systems, our approach introduces a diverse dataset encompassing a wide array of threat categories, including knives, guns, fires, and normal scenarios. This diversity in threat classes significantly improves the model's ability to accurately distinguish between various types of security risks, leading to enhanced robustness and reliability in real-world applications. Utilizing the Xception architecture, SafeguardNet achieves an overall accuracy of 94.5%, precision of 92.3%, recall of 93.8%, and an F1 score of 93.0%., the model demonstrates exceptional capability with individual F1 scores of 96% for guns and fires, 95% for Additionallyknives, and 89% for normal scenarios, reflecting its proficiency in handling diverse threat types. The integration of a varied dataset plays a critical role in enhancing these performance metrics by providing the model with a comprehensive range of scenarios for training. This diversity ensures that SafeguardNet can robustly and accurately detect and classify multiple security threats, offering a reliable and comprehensive solution for corporate security needs.

INDEX TERMS Building safety, weapon detection, fire detection, deep learning, explainable AI, automated home.

I. INTRODUCTION

In the landscape of corporate security, buildings today face a broad spectrum of threats that endanger the safety of employees, visitors, and assets. The stakes are high, as these threats ranging from armed violence to fires and structural accidents not only pose physical dangers but also threaten substantial financial, legal, and reputational damages. The ramifications extend beyond immediate physical harm, potentially resulting in severe brand damage, costly litigation, and operational disruptions. The psychological impact on employee morale and productivity further amplifies the urgency for robust and proactive security measures. Recognizing the multifaceted nature of these threats, it is imperative that corporations enhance their safety protocols to foster a secure and positive working environment [1].

Traditional security measures in organizations often integrate manual surveillance, physical security personnel, and rule-based technologies. These measures, while providing baseline security, are hampered by significant drawbacks. Manual surveillance, for instance, demands high labor costs and suffers from human fatigue, which can lead to lapses in monitoring and delayed responses to security breaches [2], [3]. Physical security staff, although essential for on-the-ground responses, cannot monitor all areas effectively at all times, creating gaps in coverage. Rule-based technologies,

The associate editor coordinating the review of this manuscript and approving it for publication was Sukhdev Roy.

on the other hand, operate under fixed algorithms that lack the flexibility to adapt to novel or sophisticated threats, often resulting in either oversensitivity, leading to false positives, or under-sensitivity, which may miss genuine threats.

In response to these limitations, the security industry is increasingly turning to advanced technologies such as deep learning (DL), renowned for its capacity to significantly enhance threat detection capabilities [4], [5], [6]. Deep learning models, especially those employing convolutional neural networks (CNNs), are adept at discerning intricate patterns in visual data, facilitating the real-time detection of diverse threats with high precision. Unlike traditional methods, these models learn from vast amounts of data, allowing them to improve over time and adapt to new, previously unseen scenarios. This capability is pivotal for maintaining robust security measures in dynamic environments where threat vectors continuously evolve.

Recognizing the constraints of typical binary threat detection systems [7], [8], [9], which often rely on datasets lacking in variability and scope, our research introduces a comprehensive new dataset. This dataset merges data from two distinct sources: publicly available security footage and a custom-collected set of images and videos specifically designed for threat detection in corporate environments. The selection criteria for these data sources were based on the diversity and representativeness of various threat scenarios, including knives, guns, fires, and normal scenarios, ensuring a wide coverage of potential security threats. The threat classes were chosen to encompass the most common and critical threats faced by corporate security systems, thereby maximizing the practical applicability of our model. By integrating such a varied dataset, we enable the development of a sophisticated deep learning model based on the Xception architecture, which is specifically optimized for multi-class threat detection within corporate environments. This approach not only broadens the scope of detectable threats but also enhances the accuracy and reliability of the system in distinguishing between different types of security risks.

The contributions of this research are multifaceted:

1) We propose an optimized real-time threat detection architecture that utilizes a deep learning model based on the Xception framework, capable of handling a diverse classification of security threats.
2) A comprehensive evaluation of current methodologies is presented, demonstrating the enhanced performance of our model through the integration of advanced neural network features such as global average pooling, dense layers, batch normalization, dropout, and ReLU activation functions.
3) We conduct a detailed comparative analysis between our model and existing baseline models, showcasing significant improvements in threat detection accuracy and reliability.
4) The application of advanced visualization techniques such as Grad-CAM++ and SmoothGrad enhances the interpretability of our model, providing deeper insights into its decision-making processes.

This paper is structured into seven sections, starting with this introduction. Section II reviews related literature, Section III details the dataset and describes the methodology, Section IV discusses the results, Section V analyzes these findings, and Section VI concludes the study and outlines future work directions.

II. LITERATURE REVIEW

A. BACKGROUND STUDY

Corporate security concerns have reached unprecedented heights and, require prompt attention and creative solutions. Fires and weapons such as knives and guns endanger corporate personnel, visitors, and assets. These dangers can cause more than just physical harm. Financial losses, legal liabilities, reputational damage, and business continuity disruptions can occur. Traditional security systems that use human monitoring or outdated detection methods often fail to identify and address dynamic security issues. Thus, sophisticated and adaptable monitoring systems that can identify and respond to threats in real-time must be prioritized. Given these challenges, a detailed literature study is needed to evaluate the current methodologies, identify research gaps, and propose alternative business risk recognition strategies.

B. WEAPON DETECTION

The increasing number of guns in public spaces has prompted the development of advanced detection devices to reduce risks. There has been a notable increase in scholarly attention to the use of cutting-edge technology, such as deep learning and computer vision, to create automated weapon detection systems that can accurately and efficiently recognize knives and weapons. In this study, we discuss the latest weapon detection technologies for knives and weapons in workplace environments.

According to Salido et al. [10], surveillance film can automatically detect handguns. Their technique reduced false positives by incorporating firearm pose data from training dataset photos. According to the study, RetinaNet with the unfrozen ResNet-50 backbone had the highest average precision (96.36%) and recall (97.23%), whereas YOLOv3 with posture information had the highest precision (96.23%) and F1 score values (93.36%. The last design increased by approximately 2% when pose information was explicitly employed during training. The models may misidentify smartphones, wallets, and books as firearms, causing false positives. Future studies should improve models to differentiate between these objects and improve the detection accuracy. Kaya et al. investigated automatic firearm detection in surveillance film [11]. VGGNet was used to classify assault rifles, bazookas, grenades, hunting rifles, knives, handguns, and revolvers. This paradigm requires constant picture data, which are vital. This reliance may not cover dynamic weapons concealment and use. The

proposed model outperformed established models such as VGG-16, ResNet-50, and ResNet-101 with a success rate of 98.40%. The model struggles to adapt to surveillance settings and weapon concealing strategies. Examine and improve future studies. Globally, human violence causes 7.9 deaths per 10,000 individuals year [12]. Human violence usually occurs suddenly or in distant areas. Without timely information, it is difficult to prevent these behaviors. This study addressed this issue through detection. CCTV cameras help investigate crimes in cities. This study examined violent CCTV footage. The Inception v3 and Yolo v5 models recognize violence, perpetrators, and weapons. The analysis showed 74% accuracy for the proposed model. Automated violence detection systems have 74% accuracy; however, human behavior and changing monitoring settings may produce ambiguity and false positives.

Another group used Efficient-Net for real-time firearm detection in surveillance cameras [13]. A tested and promising Efficient-Net was also built with this effort. With more epochs than the other methods, the Efficient-Net technique achieved 98.12% accuracy. The number of training epochs may overfit and prevent the generalization of the unknown data. More research is needed to evaluate the technique in real-world settings to solve scalability, computing efficiency, and model interpretability challenges. Investigating weapon detection and tracking in crime forecasting can aid detectives in understanding event chronologies [14]. A professional carefully annotated each photo for identification and classification. Validating such data requires object detection and classification. Weapon detection uses SSD, YOLO, and a Faster RCNN. The Mediapipe library calculates weapon-human relationships from human body data. Faster RCNN with the Mediapipe library achieved 97% accuracy. This study examined fire and pistol detection in camera-monitored areas [15]. Wildfires, industrial explosions, and residential fires have affected the environment. In this study, they developed a YOLOv3 deep learning model. This model detects video anomalies and alerts authorities frame-by-frame. Finally, the model validation loss was 0.2864 and the detection rate was 45 fps. In the IMFDB, UGR, and FireNet tests, the model had 89.3%, 82.6%, and 86.5% accuracy. Despite the low precision of the sensitive automated systems.

This work suggests binarization methods to enhance the robustness, accuracy, and reliability [16]. The gun, knife, phone, bill, wallet, and card are logged. The empirical investigation reveals that the proposed technique lowers the number of false positives compared to multiclass detection. Binarization may not capture the details of small items in complex real-world contexts, which limits this research. This may cause misclassifications or missing detections. This study used PGGAN and superimposition to simulate X-ray images [17]. Training advanced detection models such as YOLO, SSD, and RetinaNet. The technique recognizes weapons, knives, razor blades, and shuriken with high mean average precision (mAP) on real X-ray images. YOLOv3 performed the best among the tested methods. This work may be constrained by simulated X-ray photographs, which may not accurately depict scanning details. However, this methodology may not be applicable to real-world deployments.

This study used CNNs with pre-trained VGG-16 network weights for real-time weapon monitoring and detection [18]. This study suggests ways to create and improve unique images. The approach achieves 98.07% accuracy with isolated photos and 98.42% accuracy with handled images, demonstrating its efficiency. More accurate preprocessed photographs support the suitability of the algorithm and exceed comparable research by 7%. The improved results demonstrate that the proposed method can improve real-world weapon detection. Another study sought to create a deep learning-based system to detect hidden firearms in thermal pictures [19]. Thermal video and public data train two deep learning models to locate firearms. A well-adjusted VGG19 model had an F1 score of 0.84, and a Yolo-V3 model had 0.95 mean average precision of 10 milliseconds. These findings demonstrate that deep learning and infrared imaging can improve real-time surveillance. Effective detection of weapons and knives in audio recordings is crucial for public safety [20]. Deep Learning (DL) has improved object identification; however, problems persist. Weapon size against camera range and quick reaction are limitations, especially with inexpensive edge devices. This issue can be addressed using a two-step deep learning technique that utilizes a CNN to distinguish humans from anothers to detect firearms. The technique's COCO Average Precision (AP) of 79.30 and FPS of 5.10 suggest economical, widely available automated video surveillance systems.

Recent advancements in deep learning have improved detection accuracy, but often require substantial computational resources. This research fills the gap by introducing a lightweight, pose-based approach, leveraging hand pose pattern analysis and novel techniques like Fuzzy Discernible Feature Selection (FDFS) to efficiently and accurately detect small firearms from visual media. Cao et al. introduces a novel approach that focuses solely on refining PAFs, leading to significant improvements in both runtime performance and accuracy [21]. Additionally, we present the first combined body and foot keypoint detector, which reduces inference time without compromising accuracy. This comprehensive and efficient solution is embodied in OpenPose, the first open-source realtime system capable of detecting keypoints for body, foot, hand, and facial features in multi-person scenarios. Deep learning methods have enhanced detection capabilities but typically focus solely on the visual appearance of weapons [22]. Another work bridges this gap by incorporating human pose analysis, utilizing pose keypoints to better detect handguns, especially when they are not fully visible. This combined approach significantly enhances detection accuracy, outperforming previous methods. Existing pose estimation techniques lack the precision needed for detecting weapon operation activities, which is crucial for effective surveillance [23]. Our research utilizes human body skeleton graphs and integrates an LSTM-RNN network

with a Kalman filter, achieving 89.09% accuracy in real-time pose identification and significantly enhancing the detection of suspicious activities amidst regular movements.Though direct object detection provides immediate and actionable intelligence for security personnel, which is crucial in real-time scenarios. Incorporating pose detection can add complexity and computational overhead, potentially increasing false positives in dynamic settings.

C. FIRE DETECTION

Fire detection is essential to save lives and property. The Internet of Things and Deep Neural Networks can automate fire detection. System design, data quality, and computational resources determine the effectiveness of these systems.

Jiao 2020 introduces YOLOv3 for classifying objects in UAV fire images. They proposed boosting the quantity and variety of training data. A WSN and deep learning system have been introduced for early forest fire detection [24]. According to the experiments, the GRU model identified forest fires with 99.89% precision and 0.0088 loss function. The authors advise improving their approach by improving the network design, sensor node capabilities, and deep learning. The efficacy and adaptability of the proposed system to forest fire scenarios must be demonstrated using a small dataset. Another study proposed a UAV-AI forest firefighting system for detection and monitoring. The researchers compiled and organized a unique dataset (DeepFire) comprising visual representations of multiple real-world forests, with and without fire, to aid future research. Comparison of machine learning methodologies. The authors recommend VGG19-based transfer learning to improve prediction accuracy. The simulated results show that the proposeded approach achieves 95% classification accuracy, 95.7% precision, and 94.2% recall rates. An early fire detection model (EFDM) is developed using computer vision and CCTV monitoring [25]. EFDM detects flames faster than smoke or heat detectors depending on the material. A study proposed a CNN based YOLOv4 fire detection system for low-power devices [26]. Deep detection networks were trained with less important convolutional filters removed. This method reduces the network computational load without affecting the performance. Various pruning algorithms can eliminate 83.88 percent of network parameters, reduce computational expense (BFLOPs) by 83.60 percent, and reduce the Raspberry Pi 4 detection time by 83.73 percent while maintaining the network performance. We recommend the use of several CNN architectures for fire detection. Pre-trained four ResNet architectures were used: ResNet18, ResNet50, ResNet101, and InceptionResNetV2 [27]. SVM-classified Ensemble ResNet models with 10-fold cross-validation had 98.91-99.15 percent classification accuracy. A major drawback of the proposed model is the limited photo usage during development. A new voting approach merges YOLO and CNN architectures, combining two weights [28]. The F1 classification model accuracy, sensitivity, and score were 0.95, 0.99, and 0.98e. The model was classified using transfer learning. The detector model's 0.85 and 0.76 smoke and combination mAP scores were good. The smoke detector model scored 0.93 F1.

Similarly, smoke is the initial stage of fire thats why many tried smoke detection methods which often fail in diverse surveillance environments, particularly under hazy conditions [29]. This research overcomes these limitations by introducing a CNN-based framework using EfficientNet for smoke detection and DeepLabv3+ for segmentation, achieving up to 3% higher accuracy and significantly improved performance metrics, making it highly suitable for varied and challenging surveillance conditions. Vision-based detection systems, utilizing surveillance cameras, offer faster and more robust solutions by providing early warnings from both nearby and distant smoke [30]. Our research introduces a dual deep learning framework that leverages Deep Convolutional Neural Networks (CNNs) for feature extraction from smoke patches and motion-based features, significantly enhancing detection accuracy and robustness in varied and challenging conditions. Closely, Sathishkumar et al. addresses issues by employing transfer learning on pre-trained models such as VGG16, InceptionV3, and Xception, and incorporates learning without forgetting (LwF) to retain original classification abilities while adapting to new tasks [31]. This approach significantly enhances detection accuracy and robustness, outperforming state-of-the-art methods and ensuring reliable performance on both existing and novel datasets.

D. OVERALL FINDINGS

Our literature review reveals a critical gap in existing threat detection systems, which predominantly employ binary classification approaches that are insufficient for the complexities of real-world scenarios in corporate environments. To address this, our research focuses on three pivotal threat categories: fire, guns, and knives. These classes were chosen due to their high prevalence and significant impact on corporate safety. Fire detection is vital for early emergency response to prevent widespread damage and loss of life. Gun detection is crucial for averting armed assaults, while knife detection mitigates the risk of close-quarters attacks. Existing typical deep learning-based models often fail to achieve optimal accuracy in these areas due to the limitations of non-diverse datasets. By integrating a comprehensive and varied dataset that encompasses these specific threats, our proposed model significantly enhances detection accuracy and robustness. This approach not only bridges the performance gaps identified in current systems but also provides a more reliable and holistic security solution tailored to the multifaceted nature of corporate security threats.

III. METHODOLOGY

A. DATASETS

The datasets utilized in this study were meticulously selected to ensure a comprehensive and robust evaluation of the

proposed models. The primary datasets include various types of images classified into multiple categories, each aimed at addressing specific detection tasks.

The Weapon Detection YOLOv7 dataset [32] comprises 5000 labeled images, divided into Knife and Gun classes. This dataset includes 4245 images in the training set and 755 images in the test set, all annotated in YOLOv7 format, facilitating efficient training and validation of object detection models. The diversity and quantity of this dataset provide a solid foundation for the development and testing of weapon detection algorithms (Table 3).

Another crucial dataset is the Fire dataset [33], which includes 999 images categorized into Fire and Nonfire classes. This binary classification dataset is pivotal for training models to distinguish between fire and non-fire images, essential for developing reliable fire detection systems. The dataset's balanced nature ensures that the model learns to identify fire instances accurately without being biased towards non-fire images (Table 4).

Additionally, the Fire detection dataset [34] contains 651 images labeled as 0 for non-fire images and 1 for fire images. This dataset further supports the binary classification task, with 541 non-fire images and 110 fire images, enabling the model to learn from a varied set of instances (Table 5). The Fire-Flame-Dataset [35] expands the classification task by introducing three classes: Fire, Smoke, and Neutral. With 3900 images in total, this dataset allows the model to distinguish between fire, smoke, and normal scenarios, enhancing its applicability in real-world situations (Table 5).

The Fire dataset by Spacewalk01 [36] comprises 3677 images exclusively of fire instances. This dataset enriches the training data with numerous examples of fire, ensuring that the model can generalize well to different fire scenarios. The inclusion of this dataset is critical for fine-tuning the model's ability to detect fire accurately and promptly (Table 6).

To enhance the robustness of our models, these datasets were merged into a single, enriched dataset for training, validation, and testing purposes. This merged dataset ensures a balanced representation of all classes, with 85% of the data allocated for training (further divided into 70% for training and 15% for validation) and 15% for testing. Specifically, the merged dataset includes 2092 Knife images, 2908 Gun images, 1904 Fire images, and 999 Normal images (Table 7). This comprehensive and balanced dataset allows for the development of models that can reliably detect various objects and scenarios, thereby improving their performance and generalizability.

Figure 1 presents sample images from each class in the merged dataset, providing visual insights into the dataset's composition. These images illustrate the diversity and quality of the data, which are crucial for training robust and accurate models. By utilizing such a diverse and well-balanced dataset, we ensure that our models are trained on a wide range of examples, thereby enhancing their ability to perform well in real-world applications.

FIGURE 1. Sample images of each class from the merged dataset.

The datasets utilized in this study were carefully selected and processed to ensure robust evaluation and training of our models. It is important to note that no cropping or segmentation was applied to the images in our dataset. Instead, images were resized to standard resolutions (71×71, 128×128, 224×224, 256×256, and 512×512 pixels) as part of our experimental setup to assess the impact of resolution on model performance. This resizing approach maintains the integrity of the entire image content while uniformly adjusting dimensions, ensuring consistency in our evaluation across various image scales. Moreover, to augment our dataset for improved model generalization, we applied techniques such as rescaling pixel values, random shifts, rotations, zooms, and vertical flips during training. While these augmentations increase the effective dataset size seen by the model during training, the original dataset remains unchanged in terms of stored images. This approach enhances model robustness without altering dataset size or introducing bias, facilitating a comprehensive evaluation across different classification tasks.

Here web-based image scraping was not utilized due to concerns regarding data quality and legal constraints. Images sourced from the web often lack consistent labeling and may introduce noise into the dataset, potentially degrading model performance. Additionally, adhering to copyright laws and ensuring ethical data usage guided our decision to rely on curated datasets from reliable sources. The careful selection and amalgamation of these datasets provide a diverse and representative sample of the scenarios the models aim to address. By integrating datasets with varying complexities and classes, the models are trained on a wide array of images, which enhances their ability to generalize and perform well in real-world applications. Each dataset contributes unique characteristics: the Weapon Detection YOLOv7 dataset brings diversity in weapon types and scenes, while the multiple fire datasets introduce various fire scenarios, from simple fire detection to distinguishing between fire and normal conditions.

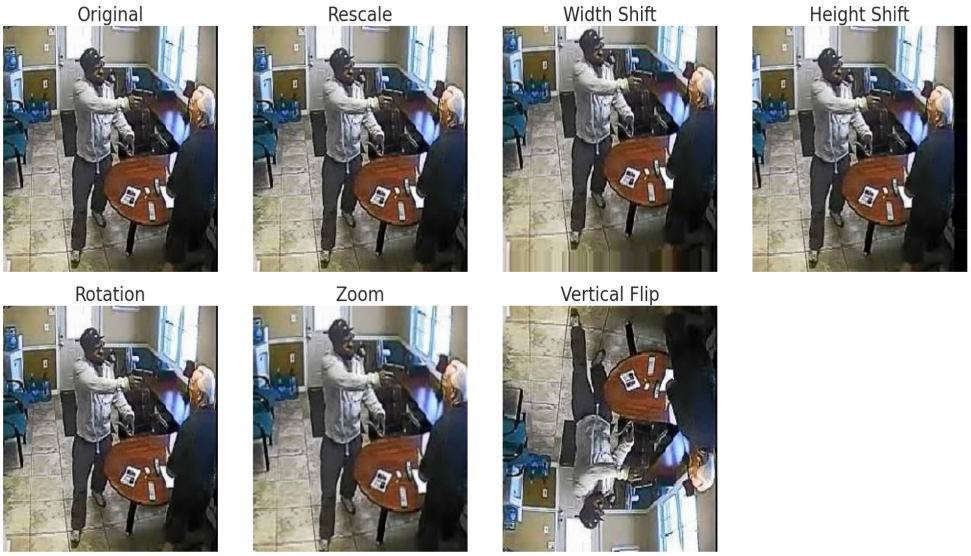

FIGURE 2. Different phases of data augmentation.

TABLE 1. Collected dataset details.

Name	Classes	Short Details of Dataset	No. of Images
Weapon Detection YOLOv7 [32]	Knife, Gun	In total, about 4245 labeled images in train set, and 755 labeled images in test set. All labels in YOLOv7 format.	5000
Fire dataset [33]	Fire-images, Nonfire-images	Data was collected to train a model to distinguish between the images that contain fire (fire images) and regular images (non-fire images), so the whole problem was binary classification.	999
Fire detection dataset [34]	0, 1	non-fire and fire images labeled as 0 and 1. 0 has 541 files and 1 has 110 files	651
Fire-Flame-Dataset [35]	Fire, Smoke, Neutral	This dataset collected in order to recognize Fire, smoke, and neutral(images without fire or smoke)	3900
Fire dataset [36]	Fire images	Only fire images	3677

TABLE 2. Weapon detection YOLOv7 [32].

Class Name	Samples
Knife	2092
Gun	2908

TABLE 3. Fire dataset [33] we have used non-fire images as normal class.

Class Name	Samples
Fire-images	755
Nonfire-images(Normal)	244

TABLE 4. Fire detection dataset [34] we have used non-fire images as normal class.

Class Name	Samples
Fire-images	110
Nonfire-images(Normal)	541

TABLE 5. Fire-Flame-Dataset dataset [35] we have used non-fire images as normal class.

Class Name	Samples
Nonfire-images(Normal)	214

TABLE 6. Fire dataset dataset [36].

Class Name	Samples
Fire-images	1039

B. IMAGE PIPELINE

1) DATA AUGMENTATION

Data augmentation plays a crucial role in enhancing the model's ability to generalize from the training data, particularly for image classification tasks. We employ various augmentation techniques on the fire detection dataset to simulate diverse environmental conditions that the model may encounter in real-world scenarios.The effects of these augmentations are illustrated in Figure 2, and the augmentation parameters are as follows:

- Normalizing pixel values to a standardized range of 0 to 1, which aids in model convergence during training.

TABLE 7. Merged both of the datasets to train, validate, and test with a more enriched one.

Merged Dataset				
Class Name	**Train (85%)**		**Test (15%)**	**Total**
	Train (70%)	**Validation (15%)**		
Knife	1464	313	315	2092
Gun	2037	436	435	2908
Fire	1336	284	284	1904
Normal	699	150	150	999

- Introducing spatial variability of fire and smoke in images through random width and height shifts of up to 10
- Adjusting image rotations within a narrow range of 2 degrees to mimic slight angular differences in camera positioning.
- Applying a zoom range of 10
- Applying vertical flips to represent different orientations of the captured scenes.

By incorporating these augmentations, the training set becomes more varied and closely mirrors real-world conditions. This approach improves the model's ability to perform under different scenarios and enhances its overall robustness.

2) PREPROCESSING AND GENERATORS

The fire detection dataset was divided into separate sets for training, validation, and testing. We used the TensorFlow Keras ImageDataGenerator to streamline image handling and preprocessing for optimal efficiency. This generator is a valuable tool for providing real-time data to a neural network without the need to store augmented images in memory.

3) TRAINING SET

The training set was configured with the data augmentation parameters mentioned earlier to enhance model generalization. It performs the following operations:

- Imports images from the designated training directory.
- The image dimensions are adjusted to 224 × 224 pixels to ensure a uniform input size. This dimension is a standard in deep learning, especially for models trained on the ImageNet dataset, as it strikes a balance between preserving enough detail and keeping computational demands reasonable. The decision to use this size is a compromise between detail preservation and computational efficiency, which will be empirically confirmed in the results section.
- Applies the specified augmentation transformations in a randomized manner to each batch.
- Randomly shuffles the dataset to ensure that each training batch has a diverse image distribution.

The batch size was set to 16, balancing the speed of training with memory usage.

4) VALIDATION AND TEST SETS

For the validation and test sets, the ImageDataGenerator applies minimal processing to preserve the originality of the data:

- Normalizes the pixel values by rescaling them.
- Resizes and loads images to a dimension of 224 × 224 pixels. Consistency in image size between the training and evaluation phases prevents potential biases and allows for an accurate assessment of model performance.
- Avoids any augmentation transformations to ensure that the model's evaluation is based on unaltered data.

Shuffling is disabled for these sets to maintain consistency in the evaluation metrics and testing outcomes.

5) CATEGORIES AND CLASS COUNT OF THE DATASET

The dataset was divided into four classes: 'Fire', 'Gun', 'Knife', and 'Normal'. These categories play a vital role in enabling the model to accurately detect and distinguish between different fire and smoke scenarios, as well as normal conditions without fire. Understanding the classes identified by the training generator is crucial for setting up the softmax output layer of a neural network. The dataset consisted of four classes that represented specific categories.

The model is primed to handle a diverse range of datasets by employing a meticulous approach to data augmentation and preprocessing. This, in turn, bolsters the ability to effectively classify images in real-world fire detection scenarios.

C. ARCHITECTURE OF THE MODEL

Model architecture plays a crucial role in deep learning systems. It consists of a network of interconnected layers that work together to process the input data, create meaningful representations, and make precise predictions. In this study, we present a new approach that utilizes the Xception model as the core framework, enhanced with custom layers to meet the unique demands of our task. Every aspect of architecture is carefully designed to strike a perfect balance between complexity, expressiveness, and computational efficiency.

1) XCEPTION BASE MODEL

The Xception architecture, introduced by Chollet in 2017, is a significant advancement in the design of convolutional neural networks [37]. It is constructed based on the principles of depthwise separable convolutions, allowing it to achieve cutting-edge performance while using far fewer parameters than traditional architectures. The model is composed of 71 convolutional layers organized into 14 modules. Each module includes depthwise separable convolutions, batch normalization, and rectified linear unit (ReLU) activation.

This design promotes the reuse of features and allows for smooth training, resulting in quicker convergence and better generalization abilities.

2) CUSTOM LAYERS

- **Global Average Pooling 2D Layer:** When adapting the Xception model for our unique classification task, we enhanced it with custom layers specifically designed to suit our dataset and goals. The model begins with the base Xception architecture, which acts as a robust feature extractor. We added custom layers to further enhance the extracted features and improve classification accuracy. After the feature extraction stage, we used a Global Average Pooling 2D layer to decrease the spatial dimensions of the feature maps while retaining crucial spatial information [38]. This pooling operation combines feature maps across spatial dimensions to create a condensed representation that can better handle spatial translations and distortions.

$$\text{GAP}(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

- **Dense Layer:** We incorporate dense layers with rectified linear unit (ReLU) activation functions and L2 regularization to facilitate feature learning and reduce overfitting. The ReLU activation function enables the model to grasp intricate patterns and connections within the data by introducing non-linearity [39]. Additionally, L2 regularization discourages the model from assigning excessive importance to certain weight values, promoting the learning of more straightforward and widely applicable representations:

$$f(x) = \sigma(\mathbf{W}x + b) + \lambda \sum_{i=1}^{n} w_i^2 \tag{2}$$

- **Batch Normalization Layer:** To enhance the model's resilience and adaptability, we included batch normalization layers [40]. Batch normalization normalizes the activations within each mini-batch, reducing the effect of internal covariate shift and speeding up convergence during training. This normalization technique enhances the stability and efficiency of the optimization process, allowing the model to learn more effectively from the data.

$$\text{BN}(x) = \gamma\left(\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}\right) + \beta \tag{3}$$

  where γ and β are learnable parameters, μ is the mean, σ is the standard deviation, and ϵ is a small constant used to avoid division by zero.
- **Dropout Layer:** To address overfitting, we utilize dropout regularization [41]. During training, dropout randomly masks a fraction of the units, helping the model learn redundant representations and reducing its dependence on specific features. This regularization technique makes the model more resilient and avoids memorizing irrelevant noise during training, enhancing its ability to perform well on new and unseen samples. The dropout operation is defined as follows:

$$\text{Dropout}(x) = x \odot \mathbf{M} \tag{4}$$

- **L2 Regularization:** Regularization is used in the dense layers to discourage large weights and avoid overfitting. It incorporates a penalty term into the loss function, directly proportional to the square of the weights:

$$\lambda \sum_{i=1}^{n} w_i^2 \tag{5}$$

  Here, λ is a coefficient used for regularization, and w_i represents the weights assigned to the model parameters.
- **Softmax Output Layer:** A Softmax Output Layer is integrated to generate class probabilities for multiclass classification [42]. The softmax function normalizes the output scores across different classes, ensuring that the model's predictions adhere to a valid probability distribution:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{6}$$

The resulting architecture showcases a seamless blend of cutting-edge techniques and customized adjustments, perfectly suited to address the unique challenges and demands of our classification task. By leveraging the Xception model and adding custom layers, we aim to maximize performance and improve generalization on our dataset.

3) SUMMARY OF THE MODEL

This detailed explanation offers a comprehensive overview of the model architecture, explaining the reasoning behind each component and how it contributes to the overall system. The citations supporting each technique highlight the scientific rigor and importance of the design decisions made in the study.

- **Xception base model:** An architecture for deep convolutional neural networks proposed by Chollet in 2017 [37], known for its efficiency and impressive performance in tasks involving feature extraction.
- **Specialized Layers:** Custom layers are added to enhance feature representations, introduce non-linearity, and avoid overfitting. These layers include Global Average Pooling 2D, dense, batch normalization, dropout, and softmax.
- **Applying L2 Regularization:** Regularization is used on dense layers to discourage large weights and encourage generalization [39].

D. TRAINING APPROACH

For optimal model convergence and efficiency, we adopted a training strategy utilizing the Adam optimizer with a learning rate of 1×10^{-4} and categorical cross-entropy as the loss

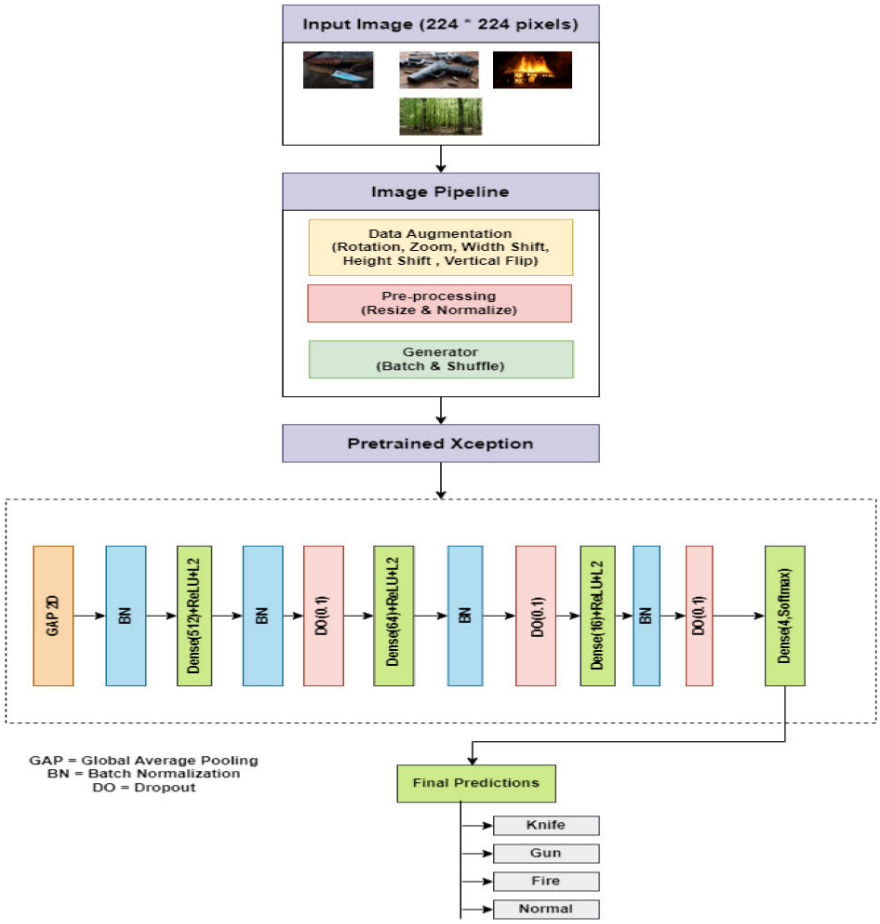

FIGURE 3. The diagram depicts the proposed method for identifying fire and weapon, encompassing preprocessing, an Xception model, and several custom layers.

function. The dataset included both original and augmented images to enhance robustness. Through a series of initial experiments, the learning rate of 1×10^{-4} was chosen for its balance between convergence speed and model performance.

To determine the optimal learning rate, we conducted a series of initial experiments where we tested various learning rates, ranging from 1×10^{-6} to 1×10^{-2}. We evaluated each learning rate based on model performance metrics such as accuracy, loss convergence, and stability during training. The learning rate of 1×10^{-4} consistently provided a good balance, ensuring quick convergence while maintaining high performance.

To ensure training stability, we maintained a batch size of 16, which was optimal given our hardware capabilities, specifically the free version of Google Colab. We implemented early stopping with a patience of three epochs to mitigate overfitting and employed model checkpoints to save the best-performing model based on validation loss. The training specifications, detailed in Table 8, leveraged cutting-edge models optimized for performance on an NVIDIA Tesla T4 GPU.

Our training approach was designed for efficiency and effectiveness, incorporating early stopping and the power of the NVIDIA Tesla T4 GPU on Google Colab to expedite the process. Multiple workers were utilized to further accelerate training.

To train the baseline models in Google Colab, we used the Keras and TensorFlow libraries. The datasets had varied input sizes, epoch batch sizes, augmentation parameters, learning rates, and optimizer activation functions. We ensured that each algorithm was trained under the same conditions to allow for a fair comparison. By maintaining consistent learning rates across different models, we ensured that the comparative results reflected the inherent performance differences of the models rather than variations due to different learning rates.

The framework of our proposed strategy is outlined below, as shown in Figure 3.

E. EXPLAINABLE AI

In this study, we introduce an Xception-based convolutional neural network model that is, meticulously trained on a comprehensive dataset encompassing a various scenarios involving guns, knives, fires, and normal scenarios. The model's architecture was designed to yield transparent visual

TABLE 8. Training specifications.

Parameter	Value
Optimizer	Adam
Learning Rate	1×10^{-4}
Loss Function	Categorical Cross-Entropy
Batch Size	16
Epoch	100
Early Stopping Patience	3
GPU	NVIDIA Tesla T4
Libraries	Keras, TensorFlow

explanations for its inferential processes, thereby enhancing trust and understanding of its predictive capabilities.

- **Grad-CAM++:** Grad-CAM++ serves as an extension of the original Gradient-weighted Class Activation Mapping (Grad-CAM), augmenting it with the capability to leverage second-order gradient information. This advancement permits a more granular visualization of influential regions within an image that guides the model's predictions. The mathematical representation of Grad-CAM++ is expressed as follows:

$$w_k^c = \sum_i \sum_j \alpha_{ijk}^c \cdot \text{ReLU}\left(\frac{\partial A_{ijk}}{\partial Y_c}\right) \tag{7}$$

$$L_{ij}^C = \sum_k w_k^C \cdot A_{ijk} \tag{8}$$

  where w_k^C symbolizes the importance weights of neurons, A_{ijk} delineates the activation maps for the kth feature map, α_{ijk}^c represents the pixel-wise gradient weighting coefficients, and $\frac{\partial A_{ijk}}{\partial Y_c}$ denotes the contribution of feature map pixels to the class score. The resultant localization map L_{ij}^C accentuates the spatial locations that are imperative for class c discrimination [43], [44].
- **SmoothGrad:** SmoothGrad is employed in tandem with Grad-CAM++ to refine visual explanations by mitigating noise. This is accomplished by averaging the gradients of the class score of the input image across multiple perturbed instances with Gaussian noise:

$$\hat{M}_c(x) = \frac{1}{n} \sum M_c(x + N(0, \sigma^2)) \tag{9}$$

  where n signifies the number of noisy samples, and $N(0, \sigma^2)$ are Gaussian noises with a standard deviation of σ. This methodology engenders saliency maps with enhanced clarity, facilitating a more coherent interpretation of the critical regions that influence a model's decisions [45].

The initial phase of our methodology involved deploying the pre-trained Xception model and defining the class labels pertinent to the domain of interest. Subsequently, we curated a repository of images for each category, and meticulously selected and processed individual samples to construct a descriptive analysis. By documenting both the true and predicted labels of the images, we ensured the veracity of the provided explanations. The insights gained from the application of Grad-CAM++ and SmoothGrad will be elucidated in the forthcoming sections dedicated to the results and discussion.

Though our decision to use a single architecture for both fire detection and weapon detection is driven by the practical benefits of a unified approach, which leverages the advanced capabilities of the Xception architecture. This architecture excels in extracting fine-grained features from images, allowing it to effectively learn and identify distinct visual patterns associated with both fire and weapons. By using a single model, we reduce computational complexity and resource requirements, streamlining the training and deployment processes. This unified approach ensures consistent and reliable performance across different types of threats, which is crucial for real-time threat detection in corporate environments. Furthermore, integrating both tasks into one model allows for shared feature representations, optimizing overall efficiency and enhancing the system's ability to accurately and promptly identify a wide range of security risks. This approach simplifies system architecture, providing a robust and practical solution for comprehensive threat detection.

IV. RESULTS

Based on our analysis, we propose a unique classification approach using Xception for various threat categorizations. To demonstrate the effectiveness of our approach, we conducted a comparison with existing models. To evaluate our model, we calculated various performance measures, such as accuracy, precision, recall, and F1-score.

A. PERFORMANCE COMPARISON

The top-performing model surpassed Xception in accuracy when data augmentation was applied, achieving an impressive accuracy, precision, recall, and F1-score all at 95%. This significant improvement is crucial for swiftly identifying potentially life-threatening situations such as fires, knives, or guns. The results of this comparison are shown in Table 10.

Various CNN models were evaluated using diverse accuracy metrics, both with and without data augmentation. Without augmentation, models showed varied levels of accuracy, precision, recall, and F1-score. The proposed model achieved the highest overall performance at 94%, with notable performances by Xception and NASNetLarge at 9% accuracy. Introducing data augmentation improved model efficacy across the board. The results are summarized in Table 9 for models without augmentation and Table 9 for models with augmentation. The proposed model reached 95% accuracy with augmentation, enhancing precision, recall, and F1-score to 95%. Similarly, models like DenseNet201 and Xception improved to 93% accuracy. This highlights the effectiveness of data augmentation in enhancing model robustness and generalization across diverse data scenarios.

The Xception network was selected as the foundational model due to its exceptional performance across various image classification tasks. Its depthwise separable convolutions effectively reduce parameter count without sacrificing

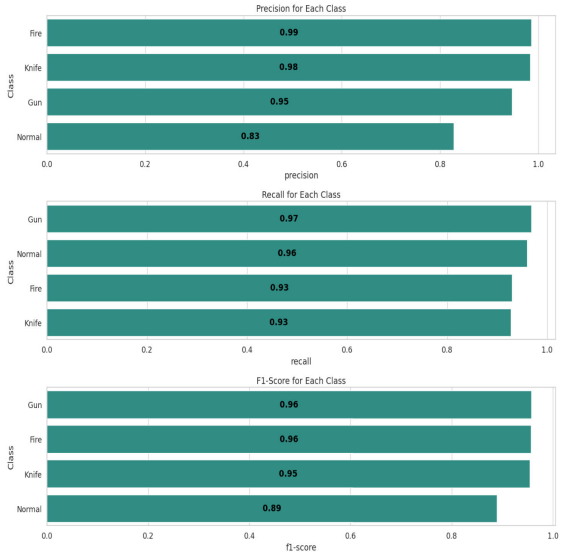

FIGURE 4. The class wise performance of each model's is summarized in this bar chart.

accuracy, making it particularly adept at detecting diverse threats with high precision. Moreover, its established success in transfer learning applications validated its suitability for our model's objectives.

The class-wise performance metrics in Figure 4 for precision, recall, and F1-score reveal distinct patterns in the model's ability to accurately classify different categories. For the 'Fire' class, the model achieved the highest precision at 0.99, indicating an exceptional ability to correctly identify true positives with minimal false positives. Similarly, the 'Knife' class demonstrated high precision at 0.98. However, the 'Normal' class had a notably lower precision of 0.83, suggesting more difficulty in distinguishing this class from others. In terms of recall, the 'Gun' class led with 0.97, showing the model's effectiveness in capturing most of the true positive instances. The 'Normal' class also performed well with a recall of 0.96, indicating that the model correctly identified most normal instances despite its lower precision. The F1-score, which balances precision and recall, was highest for the 'Gun' and 'Fire' classes at 0.96 each, reflecting consistent performance across both metrics. The 'Knife' class followed closely with an F1-score of 0.95. Meanwhile, the 'Normal' class had the lowest F1-score at 0.89, aligning with its lower precision. These results highlight the model's strong overall performance, particularly in identifying hazardous objects, while also pointing out areas for potential improvement in classifying normal scenarios.

B. CONFUSION MATRIX ANALYSIS

We acquired confusion matrices to thoroughly examine the model's performance. The confusion matrices of different models are shown in Figure 5. Through an in-depth examination, it became evident that the proposed model consistently outperformed the others across a wide range of images. The color on the diagonal of the confusion matrix indicates the number of instances in which the model accurately predicted the ground truth value.

C. TRAINING AND VALIDATION METRICS

Figures 6, 7, and 8 show the accuracy, loss, precision, recall, and area under the curve (AUC) of our model. These graphs indicate that our model performs exceptionally well. Our model has made significant progress and has effectively learned from the training data, as evidenced by the substantial improvement in the accuracy ratio between the training and validation curves. The loss graph shows that the validation data consistently had a slightly lower loss value than the training data, suggesting our model performed effectively without overfitting. Examination of the precision and recall graphs indicates that the validation precision and recall values surpassed the training precision and recall values, confirming that our model was not overfitting.

Based on the AUC graph, it is evident that our model performed well, with a score close to 1. A larger AUC value indicated a strong ability to distinguish between multiple classifications. Our proposed model for categorizing data proved to be highly effective.

As shown in Table 10, the results of the performance assessment demonstrate that our model outperformed the others in terms of accuracy (0.95), precision (0.95), recall (0.95), and F1-score (0.95).

D. ABLATION STUDIES

To accurately assess the role and significance of each layer, our ablation studies incorporated layerwise modulations rather than merely inserting or deleting layers. This approach provided a detailed understanding of the contribution of each layer to the overall model performance. Table 11 presents the results of these ablation studies, demonstrating the incremental benefits of each added component:

- **Base Model:** The baseline Xception model achieved an accuracy of 0.93, precision of 0.93, recall of 0.93, and F1 score of 0.93.
- **Base Model + Global Average Pooling:** Adding Global Average Pooling maintained the performance at 0.93 across all metrics, suggesting no immediate impact.
- **Base Model + Global Average Pooling + Dense Layer 1:** Introducing the first custom Dense layer (512 units) improved accuracy and recall to 0.94, while precision and F1 score remained at 93%.
- **Base Model + Global Average Pooling + Dense Layer 1 + Dense Layer 2:** Adding the second custom Dense layer (64 units) resulted in uniform performance improvements, achieving 0.94 in accuracy, precision, recall, and F1 score.
- **Base Model + Global Average Pooling + Dense Layer 1 + Dense Layer 2 + Dense Layer 3:** Adding the third custom Dense layer (16 units) reached the highest performance with 0.95 across all metrics.

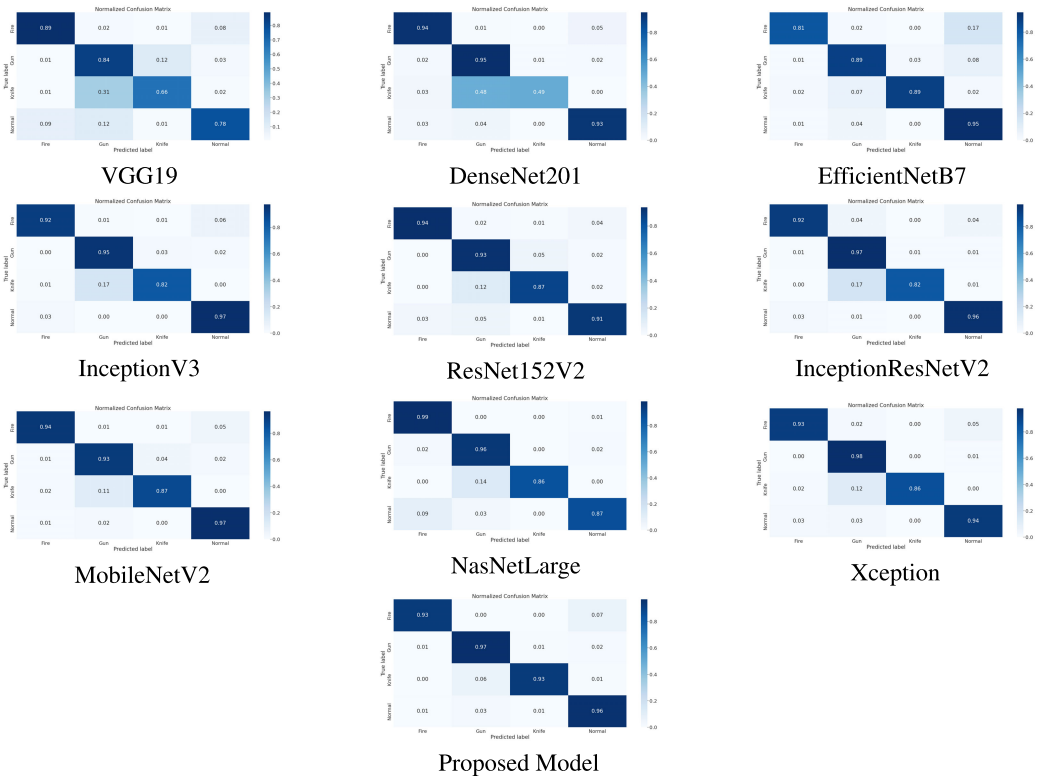

FIGURE 5. Confusion matrix of different models.

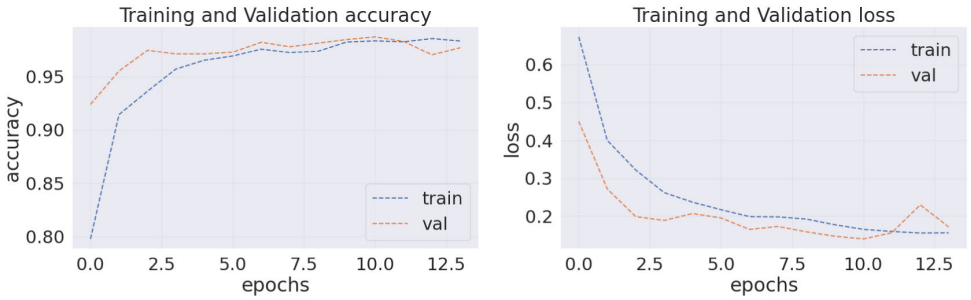

FIGURE 6. Training and validation accuracy (left) and training and validation loss (right).

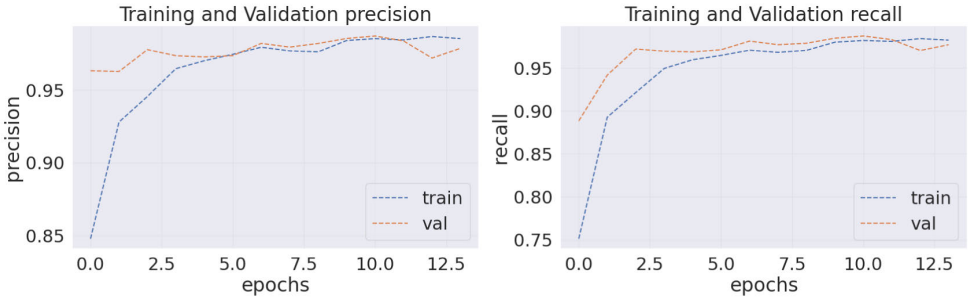

FIGURE 7. Precision for training and validation data (left) and recall for training and validation data (right).

These quantitative results highlight the performance variations of each layer, confirming the significance of our architectural choices and ensuring robust and accurate threat classification.

We evaluated the performance of our proposed model across various image resolutions to determine its efficacy and robustness. The results, as shown in Table 12, demonstrate a clear trend in performance metrics such as Accuracy, Precision, Recall, and F1-Score.

At the lowest resolution of 71 × 71 pixels, the model achieved an accuracy of 0.80, with precision, recall, and F1-score closely aligned at 0.81, 0.79, and 0.80, respectively.

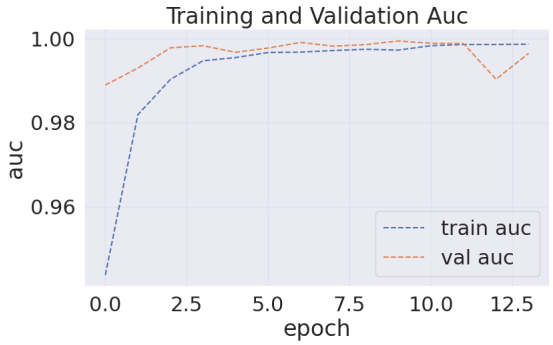

FIGURE 8. Training and validation AUC.

TABLE 9. Evaluation of the performance of applied models using various accuracy parameters without augmentation.

CNN Method Name	Accuracy	Precision	Recall	F1-score
VGG19	0.80	0.80	0.79	0.79
DenseNet201	0.82	0.86	0.83	0.82
EfficientNetB7	0.88	0.87	0.88	0.86
InceptionV3	0.91	0.91	0.91	0.91
ResNet152V2	0.91	0.91	0.92	0.91
InceptionResNetV2	0.92	0.92	0.92	0.92
MobileNetV2	0.92	0.93	0.92	0.92
NASNetLarge	0.93	0.93	0.92	0.92
Xception	0.93	0.93	0.93	0.93
Proposed Model	**0.94**	**0.94**	**0.94**	**0.94**

TABLE 10. Evaluation of the performance of applied models using various accuracy parameters with augmentation.

CNN Method Name	Accuracy	Precision	Recall	F1-score
VGG19	0.86	0.86	0.85	0.85
EfficientNetB7	0.90	0.89	0.90	0.88
InceptionV3	0.90	0.91	0.90	0.90
ResNet152V2	0.89	0.89	0.89	0.89
InceptionResNetV2	0.91	0.92	0.90	0.91
MobileNetV2	0.90	0.91	0.89	0.90
NASNetLarge	0.92	0.93	0.88	0.89
DenseNet201	0.93	0.92	0.93	0.92
Xception	0.93	0.93	0.93	0.93
Proposed Model	**0.95**	**0.95**	**0.95**	**0.95**

As we increased the resolution to 128 × 128 pixels, a significant improvement was observed across all metrics, with the model reaching an accuracy of 0.92, precision of 0.92, recall of 0.91, and an F1-score of 0.91.

The highest performance was recorded at a resolution of 224 × 224 pixels, where the model achieved an impressive accuracy, precision, recall, and F1-score of 0.95 each. This peak performance indicates that this resolution is optimal for our model. Interestingly, at a resolution of 256 × 256 pixels, the performance metrics slightly declined to 0.94 across the board, suggesting a potential saturation point or overfitting at higher resolutions.

Further increasing the resolution to 512 × 512 pixels resulted in a slight decrease in performance, with an accuracy of 0.92, precision of 0.91, recall of 0.93, and an F1-score of 0.91. This decline might indicate that very high resolutions do not necessarily enhance model performance and might introduce additional computational complexity.

In summary, our findings highlight the importance of selecting an appropriate image resolution to balance performance and computational efficiency, with 224 × 224 pixels emerging as the optimal resolution for our proposed model.

E. EXPLAINABLE AI

Our approach to Explainable AI (XAI) in this research involved using advanced visualization techniques, specifically Grad-CAM++ and SmoothGrad, to interpret the predictions made by our Xception-based model. These methods help us understand which parts of an image the model focuses on when making its classification decisions, thereby providing transparency into the model's decision-making process.

To implement these techniques, we loaded our trained Xception-based model and used a set of test images across various classes (Fire, Gun, Knife, and Normal). For each image, we applied both Grad-CAM++ and SmoothGrad to generate visual explanations of the model's predictions. Grad-CAM++ generates heatmaps that highlight important regions in the image for prediction, while SmoothGrad reduces noise in saliency maps to provide clearer visualizations of model attention.

The process included the following steps:

1) **Loading the Model and Images:** We initialized our trained Xception model and prepared a batch of test images representing different classes.
2) **Generating Explanations:** For each test image, we computed Grad-CAM++ heatmaps and SmoothGrad saliency maps. Grad-CAM++ computes gradients of the predicted class score with respect to feature maps from the last convolutional layer, emphasizing regions relevant to the prediction. SmoothGrad enhances these maps by averaging multiple noisy perturbations of the input image.
3) **Visualizing Results:** We visualized the original images alongside their corresponding Grad-CAM++ and SmoothGrad visualizations. This comparison included both the actual class labels and the model's predicted labels, facilitating a direct assessment of model interpretability.
4) **Analyzing Model Behavior:** We analyzed these visualizations to discern which image regions the model deemed significant for classification. This analysis helped evaluate the model's accuracy in identifying distinguishing features across different classes and identify potential areas of misclassification.

In Figure 9, we present a grid of images alongside their explanatory visualizations. This includes the original image, Grad-CAM++, and SmoothGrad outputs, illustrating the model's attention distribution for each sample. For images classified as normal, such as hallways, the model's focus on central regions likely reflects its learning to recognize symmetrical and consistent patterns that are characteristic of normal scenes. This behavior underscores the model's ability

TABLE 11. Ablation study results.

Model Configuration	Accuracy	Precision	Recall	F1 Score
Base Model	0.93	0.93	0.93	0.93
Base Model + Global Average Pooling	0.93	0.93	0.93	0.93
Base Model + Global Average Pooling + Dense Layer 1	0.94	0.93	0.94	0.93
Base Model + Global Average Pooling + Dense Layer 1 + Dense Layer 2	0.94	0.94	0.94	0.94
Base Model + Global Average Pooling + Dense Layer 1 + Dense Layer 2 + Dense Layer 3	0.95	0.95	0.95	0.95

TABLE 12. Our proposed model performance at different resolutions.

Resolution	Accuracy	Precision	Recall	F1-Score
71x71	0.80	0.81	0.79	0.80
128x128	0.92	0.92	0.91	0.91
224x224	0.95	0.95	0.95	0.95
256x256	0.94	0.94	0.94	0.94
512x512	0.92	0.91	0.93	0.91

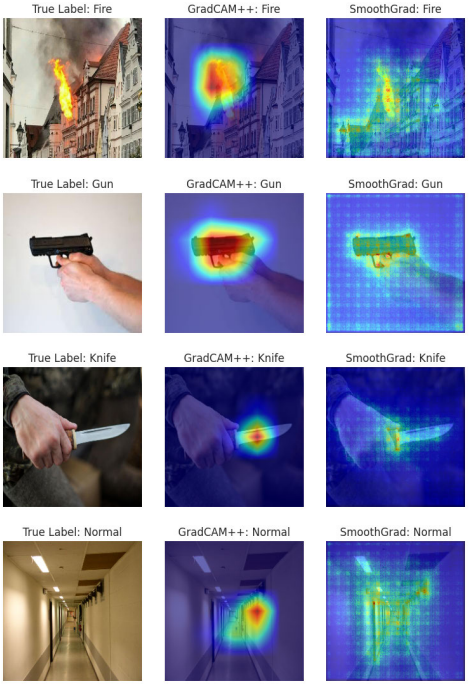

FIGURE 9. Explainable AI for all classes.

to leverage contextual and spatial cues to distinguish normal from abnormal scenarios, as informed by patterns observed during training.

These techniques yield valuable insights into the model's decision-making process, aiding in the assessment and refinement of its performance and interpretability.

V. DISCUSSION

The continual menace of threats such as fires, knives, and guns presents significant risks to public safety. The development of highly efficient and accurate detection systems. In response, our research introduces a sophisticated classification approach utilizing a custom-tailored Xception model, specifically designed for the multiclass categorization of these threats. The choice of the Xception model was predicated on its demonstrated higher accuracy, minimal loss, and exceptional performance in complex and critical scenarios. In terms of the model architecture, the adaptation of the Xception model improved the efficiency and computational speed by using depthwise separable convolutions. Additional custom layers equipped with rectified linear units (ReLUs), batch normalization, and dropout regularization are instrumental in learning intricate representations, curbing overfitting, and bolstering generalization. These modifications have significantly advanced existing models, particularly in adapting to multifaceted urban environments and diverse threat scenarios. This strategy not only diversified the training data but also enhanced the model's ability to decode complex patterns. Our evaluations confirm the model's efficacy in real-life threat detection scenarios, as evidenced by its remarkable accuracy, precision, recall, and F1-score metrics. However, future work could expand the practical applications of the model to residential and industrial settings. Moreover, multiple data sources were used to reflect the complexity of urban environments and the variety of potential threat scenarios. To address these unique challenges further, collecting additional threat-related data, such as from CCTV footage, social media, or other digital media sources, could enrich our model's training dataset. Platforms such as Google Street View, OpenStreetMap, or even proprietary surveillance systems can be leveraged to obtain real-time or near-real-time imagery of targeted public areas for enhanced situational awareness. Despite the availability and accessibility of these data sources by region, custom solutions tailored to specific needs and challenges of threat detection systems are essential. Our proposed model effectively meets these requirements through its customization and the use of explainable AI techniques, proving that the proposed system is a new standard in the field of threat detection and public safety.

VI. CONCLUSION

Given the seriousness of threats such as fire, knife, and gun incidents, it is crucial to have strong detection systems in place to minimize risks and safeguard the well-being of people and communities. We developed a specialized advanced classification approach based on Xception to detect various threat categories, such as fire, knife, and gun incidents. With careful analysis and thorough testing, we proved that the exceptional performance and reliability in accurately detecting potential threats, surpassing other existing models in the field. The results of our research

show the impressive performance of our proposed model, boasting an outstanding accuracy, precision, recall, and F1-score of 0.95 across all metrics. Ensuring precise and timely threat detection is crucial in a wide range of real-world applications, ranging from public safety to security surveillance. Explainable AI techniques such as Grad-CAM++ and SmoothGrad, have provided valuable insights into how the model makes decisions, which improves the understandability and reliability of its predictions. These techniques provide clear visual explanations of the model's reasoning, allowing stakeholders to make informed decisions based on the model's results. Although our model has shown significant progress, there are still some limitations, especially in situations in which contextual cues can impact classification outcomes. To overcome these limitations, additional research is needed to explore contextual awareness learning techniques and broaden the training dataset to include a more diverse range of scenarios. Our study provides a comprehensive and reliable classification approach for detecting threats in images. This approach has significant implications for improving safety and security in different domains,and offers valuable insights. In the future, further research should be conducted to improve the model, tackle any remaining obstacles, and expand its use in various real-life situations. This will ultimately help advance the threat detection technology and ensure the safety of people and communities.

REFERENCES

[1] M. A. Hassanain, M. Al-Harogi, and A. M. Ibrahim, "Fire safety risk assessment of workplace facilities: A case study," *Frontiers Built Environ.*, vol. 8, Mar. 2022, Art. no. 861662.

[2] J.-P.-A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli, "Cyber-physical systems security: Limitations, issues and future trends," *Microprocess. Microsyst.*, vol. 77, Sep. 2020, Art. no. 103201.

[3] V. Mavroeidis, K. Vishi, and A. Jøsang, "A framework for data-driven physical security and insider threat detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2018, pp. 1108–1115.

[4] J. Alves, C. Soares, J. M. Torres, P. Sobral, and R. S. Moreira, "Automatic forest fire detection based on a machine learning and image analysis pipeline," in *New Knowledge in Information Systems and Technologies*, vol. 2. USA: Springer, 2019, pp. 240–251.

[5] M. Grega, A. Matiolanski, P. Guzik, and M. Leszczuk, "Automated detection of firearms and knives in a CCTV image," *Sensors*, vol. 16, no. 1, p. 47, Jan. 2016.

[6] R. Debnath and M. K. Bhowmik, "A comprehensive survey on computer vision based concepts, methodologies, analysis and applications for automatic gun/knife detection," *J. Vis. Commun. Image Represent.*, vol. 78, Jul. 2021, Art. no. 103165.

[7] J. L. Salazar González, C. Zaccaro, J. A. Álvarez-García, L. M. Soria Morillo, and F. Sancho Caparrini, "Real-time gun detection in CCTV: An open problem," *Neural Netw.*, vol. 132, pp. 297–308, Dec. 2020.

[8] P. Barmpoutis, P. Papaioannou, K. Dimitropoulos, and N. Grammalidis, "A review on early forest fire detection systems using optical remote sensing," *Sensors*, vol. 20, no. 22, p. 6442, Nov. 2020.

[9] D. A. Noever and S. E. M. Noever, "Knife and threat detectors," 2020, *arXiv:2004.03366*.

[10] J. Salido, V. Lomas, J. Ruiz-Santaquiteria, and O. Deniz, "Automatic handgun detection with deep learning in video surveillance images," *Appl. Sci.*, vol. 11, no. 13, p. 6085, Jun. 2021.

[11] V. Kaya, S. Tuncer, and A. Baran, "Detection and classification of different weapon types using deep learning," *Appl. Sci.*, vol. 11, no. 16, p. 7535, Aug. 2021.

[12] S. A. A. Akash, R. S. S. Moorthy, K. Esha, and N. Nathiya, "Human violence detection using deep learning techniques," *J. Phys., Conf.*, vol. 2318, no. 1, Aug. 2022, Art. no. 012003.

[13] E. Arif, S. K. Shahzad, M. W. Iqbal, M. A. Jaffar, A. S. Alshahrani, and A. Alghamdi, "Automatic detection of weapons in surveillance cameras using efficient-net," *Comput., Mater. Continua*, vol. 72, no. 3, pp. 4615–4630, 2022.

[14] T. Ruprah and H. Shrivastav, "Crime prediction based on person-weapons relation using deep learning techniques," *Eur. Chem. Bull.*, vol. 12, pp. 984–994, Jun. 2023.

[15] P. Mehta, A. Kumar, and S. Bhattacharjee, "Fire and gun violence based anomaly detection system using deep neural networks," in *Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Jul. 2020, pp. 199–204.

[16] F. Pérez-Hernández, S. Tabik, A. Lamas, R. Olmos, H. Fujita, and F. Herrera, "Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance," *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105590.

[17] D. Saavedra, S. Banerjee, and D. Mery, "Detection of threat objects in baggage inspection with X-ray images using deep learning," *Neural Comput. Appl.*, vol. 33, no. 13, pp. 7803–7819, Jul. 2021.

[18] N. Dwivedi, D. K. Singh, and D. S. Kushwaha, "Employing data generation for visual weapon identification using convolutional neural networks," *Multimedia Syst.*, vol. 28, no. 1, pp. 347–360, Feb. 2022.

[19] O. Veranyurt and C. O. Sakar, "Concealed pistol detection from thermal images with deep neural networks," *Multimedia Tools Appl.*, vol. 82, no. 28, pp. 44259–44275, Nov. 2023.

[20] D. Berardini, L. Migliorelli, A. Galdelli, E. Frontoni, A. Mancini, and S. Moccia, "A deep-learning framework running on edge devices for handgun and knife detection from indoor video-surveillance cameras," *Multimedia Tools Appl.*, vol. 83, no. 7, pp. 19109–19127, Jul. 2023.

[21] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1302–1310.

[22] J. Ruiz-Santaquiteria, A. Velasco-Mata, N. Vallez, G. Bueno, J. A. Álvarez-García, and O. Deniz, "Handgun detection using combined human pose and weapon appearance," *IEEE Access*, vol. 9, pp. 123815–123826, 2021.

[23] A. Bhatt and A. Ganatra, "Weapon operating pose detection and suspicious human activity classification using skeleton graphs," *Math. Biosciences Eng.*, vol. 20, no. 2, pp. 2669–2690, 2022.

[24] W. Benzekri, A. El, O. Moussaoui, and M. Berrajaa, "Early forest fire detection system using wireless sensor network and deep learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 5, pp. 496–503, 2020.

[25] Y. Ahn, H. Choi, and B. S. Kim, "Development of early fire detection model for buildings using computer vision-based CCTV," *J. Building Eng.*, vol. 65, Apr. 2023, Art. no. 105647.

[26] P. V. A. B. de Venâncio, A. C. Lisboa, and A. V. Barbosa, "An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices," *Neural Comput. Appl.*, vol. 34, no. 18, pp. 15349–15368, Sep. 2022.

[27] S. Dogan, P. Datta Barua, H. Kutlu, M. Baygin, H. Fujita, T. Tuncer, and U. R. Acharya, "Automated accurate fire detection system using ensemble pretrained residual network," *Exp. Syst. Appl.*, vol. 203, Oct. 2022, Art. no. 117407.

[28] C. Bahhar, A. Ksibi, M. Ayadi, M. M. Jamjoom, Z. Ullah, B. O. Soufiene, and H. Sakli, "Wildfire and smoke detection using staged YOLO model and ensemble CNN," *Electronics*, vol. 12, no. 1, p. 228, Jan. 2023.

[29] S. Khan, K. Muhammad, T. Hussain, J. D. Ser, F. Cuzzolin, S. Bhattacharyya, Z. Akhtar, and V. H. C. de Albuquerque, "DeepSmoke: Deep learning model for smoke detection and segmentation in outdoor environments," *Exp. Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115125.

[30] A. S. Pundir and B. Raman, "Dual deep learning model for image based smoke detection," *Fire Technol.*, vol. 55, no. 6, pp. 2419–2442, Nov. 2019.

[31] V. E. Sathishkumar, J. Cho, M. Subramanian, and O. S. Naren, "Forest fire and smoke detection using deep learning-based learning without forgetting," *Fire Ecol.*, vol. 19, no. 1, p. 9, Feb. 2023.

[32] M. Zahrawi, "Weapon detection YOLOv7," IEEE, USA, 2022, doi: 10.21227/3akm-cb29.

[33] A. Saied. (2018). *Fire Dataset*. [Online]. Available: https://www.kaggle.com/phylake1337/fire-dataset

[34] A. Kumar. (2019). *Fire Detection Dataset*. [Online]. Available: https://www.kaggle.com/atulyakumar98/test-dataset

[35] O. Abimbola. (2019). *Fire-Smoke-dataset*. [Online]. Available: https://github.com/DeepQuestAI/Fire-Smoke-Dataset
[36] Spacewalk01. (2022). *YOLOv5-Fire-Detection: Datasets*. [Online]. Available: https://github.com/spacewalk01/yolov5-fire-detection/tree/main/datasets
[37] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016, *arXiv:1610.02357*.
[38] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.
[41] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jan. 2014.
[42] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *Genet Program Evolvable*, vol. 22, no. 4, pp. 351–354, Oct. 2016.
[43] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.
[44] Y. Zhang, D. Hong, D. McClement, O. Oladosu, G. Pridham, and G. Slaney, "Grad-CAM helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging," *J. Neurosci. Methods*, vol. 353, Apr. 2021, Art. no. 109098.
[45] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," 2017, *arXiv:1706.03825*.

NUSRAT JAHAN received the Bachelor of Science degree in computer science and engineering (CSE) with a major in software engineering from American International University-Bangladesh (AIUB), in 2022, where she is currently pursuing the master's degree. Her research interests include machine learning, web interaction design, and natural language processing.

MOHAMMAD SAYEM CHOWDHURY (Associate Member, IEEE) received the M.Sc. degree in computer science and engineering from American International University-Bangladesh (AIUB), Dhaka, Bangladesh, in 2023. His major field of study is intelligent systems. He is currently a Researcher, focusing on machine learning, deep learning, and natural language processing. He has contributed to various open-source projects and participated in numerous data science competitions on platforms, such as Kaggle and Tableau Public. He has authored or co-authored several research articles on AI and data science, with his work, including developing predictive models and implementing deep learning techniques. His current research interests include deep learning, natural language processing, and the ethical implications of AI technologies.

TOFAYET SULTAN is currently a Lecturer with the Department of Computer Science, Faculty of Science and Technology, American International University-Bangladesh (AIUB). Previously, he has been a Lecturer with the Department of Computer Science and Engineering, Faculty of Science and Engineering, Uttara University. He has worked for a number of research projects in different domains. His research interests include agent-based modeling, human–computer interaction, machine learning, natural language processing, and deep learning.

M. F. MRIDHA (Senior Member, IEEE) received the Ph.D. degree in AI/ML from Jahangirnagar University, in 2017. He is currently an Associate Professor with the Department of Computer Science, American International University-Bangladesh (AIUB). Before that, he was an Associate Professor and the Chairperson of the Department of CSE, Bangladesh University of Business and Technology. He was also a CSE Department Faculty Member with the University of Asia-Pacific and as the Graduate Head, from 2012 to 2019. His research experience, within both academia and industry, results in over 120 journals and conference publications. His research work contributed to the reputed *Scientific Reports*, *Nature*, *Knowledge-Based Systems*, *Artificial Intelligence Review*, IEEE Access, *Sensors*, *Cancers*, and *Applied Sciences*. His research interests include artificial intelligence (AI), machine learning, deep learning, natural language processing (NLP), and big data analysis. For more than ten years, he has been with the master's and undergraduate students as a supervisor of their thesis work. He has served as a program committee member for several international conferences/workshops. He served as an Associate Editor for several journals, including *PLOS One* journal. He has served as a Reviewer of reputed journals and IEEE conferences, such as HONET, ICIEV, ICCIT, IJCCI, ICAEE, ICCAIE, ICSIPA, SCORED, ISIEA, APACE, ICOS, ISCAIE, BEIAC, ISWTA, IC3e, ISWTA, CoAST, icIVPR, ICSCT, 3ICT, and DATA21.

MD SADDAM HOSSAIN MUKTA (Member, IEEE) received the Ph.D. degree from the BUET Data Science and Engineering Research Laboratory (DataLaboratory), Bangladesh, in 2018. He is currently a Postdoctoral Researcher with the LUT School of Engineering Sciences, LUT University, Finland. He has a number of quality publications in both national and international conferences and journals. His research interests include machine learning, social network analysis and mining, social computing, and data mining.