VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2026, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 14, Number 1, January-March 2026 pp:19-35

Hybrid Model for Real-Time Mobile Snatching Detection in Video Surveillance Using Time-Distributed CNN and Attention-Based LSTM

Faisal Khan[1], **Irshad Ahmad**[1], **Muhammad Zubair**[1*], **Yasir Saleem Afridi**[2]

[1]Department of Computer Science, Islamia College Peshawar, Khyber Pakhtunkhwa, 25000, Pakistan; [2]Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar, Pakistan

Keywords: *Mobile Snatching Detection, Attention-Based LSTM Neural Network, Crime Prevention, deep learning, human activity recognition.*

Journal Info:
Submitted:
November 06, 2025
Accepted:
February 08, 2026
Published:
February 14, 2026

Abstract We propose a Hybrid approach that consolidates Time Distributed CNNs with Attention-Embedded LSTM network model for identifying mobile theft activities from video surveillance. Gadget snatching incidents seems to be increasing a little too rapidly globally, and another step has been taken by the police in Pakistan as they now possess around 1,700 mobile phone data in the effort of halting this. We propose a model to tackle this challenge by combining temporal relation modeling ability of LSTMs and the spatial feature extraction power of CNNs.An attention mechanism that directs focus to salient cues in video sequences enhances its effectiveness. The system was trained and tested with a real-life dataset of snatching events that were reported on social media. The results of the test show that our method works because it is 96.45% accurate.The research presented here highlights the potential of social media platforms as effective instruments for crime prevention and identification, thereby advancing the field of artificial intelligence-driven crime detection. We want to make the algorithm's source code and dataset public so that more people can use it and do more research in this area.

***Correspondence author email address:** zubair@icp.edu.pk
DOI: 10.21015/vtse.v14i1.2279

1 Introduction

In various computer vision applications, video surveillance has increasingly attracted the attention of researchers, and mobile snatching has emerged as an important problem within this domain. The theft of mobile phones has become a significant social issue in many cities around the world, including Pakistan This type of street crime [1, 2] has shown a noticeable increase in recent years, particularly in major Pakistani cities such as Peshawar, Islamabad, and Karachi. As mobile devices become more common, their high value and ease of resale make them a frequent target for criminals.

Mobile snatching is often premeditated or occurs when offenders identify an opportunity ,typically taking place when the activity spaces of the victim and the criminal intersect. Snatching involves the illegal act of assaulting a victim to steal a valuable item, while snatch theft refers to the unlawful taking of property usually valuables carried by pedestrians, such as handbags, mobile phones, or jewelry and is categorized under felony and misdemeanor crimes [3]. Such incidents may also involve two-person operations, for example, when one individual uses a motorcycle to intimidate or overpower

the victim. Pedestrian theft raises significant concerns, as it can lead to accidents and intensify feelings of trauma, fear, and anxiety among the public.

For instance, individuals frequently check their mobile phones while walking, which can distract them and make them more vulnerable to such crimes. Snatching incidents captured in surveillance videos can occur in multiple ways, resulting in different victim reactions. Table 1 and Figure 1 illustrate several phone-snatching scenarios identified in the observation videos used for this study. Although much of the existing research focuses on detecting general abnormal or suspicious activities, relatively little work specifically addresses the detection of mobile snatching. Computer vision.

Computer vision [4] is a branch of artificial intelligence that studies how well computer systems and software can understand digital or visual information in the form of pictures or videos. Artificial neural networks (ANNs) are used by computer vision to process and analyze visual data. They find and interpret information [5]. These are complicated computer models that try to copy how biological systems work and look . Deep learning neural networks, such as Convolutional Neural Networks (CNNs), are some of the current technologies that have become common in the development of HAR [6].

CNNs are the best at feature extraction and classification because they learn hierarchical representations directly from raw visual data. This lets them automatically find complex patterns and details in images and videos. This has had a big impact on computer vision. [7], Combining the feature extraction and classification stages makes it faster and better. Researchers have utilized various CNN architectures, including 2D-CNN [8], ,3DCNN [9], and multimodal input networks, for Human Action Recognition (HAR). The ImageNet dataset [10] showed that CNNs were successful in 2012. This led to the creation of frameworks like VGG16 [11], VGG19 [12],InceptionV3 [13], and MobileNetV2 [14].

2D-CNNs have been used to discover and categorize features, however they have trouble acquiring information about time. 3D-CNNs, on the other hand, have been shown to be useful for HAR because they can acquire both spatial and temporal data from video sequences at the same time. Many studies that try to understand behaviors like snatch snatching use data gathered in controlled environments without crowds or other distractions so that the activity can be readily viewed. A variety of various ways have been found to catch snatch thieves. People like CNNs in computer vision because they can find complex patterns on their own, without any help. That's why they are so brilliant at recognizing stuff like movies and photographs. They use both movies and pictures, which shows that research is making progress in many areas. CNNs will probably be employed in more and more places and for more and more things as computers get smarter and big data becomes more common. These algorithms for learning could keep getting better and better.

Many people are discussing about aberrant activity recognition (AAR), which analyzes camera footage to detect anomalous actions such as theft or violence. New approaches have made it possible for CNNs to operate with movies and pictures. 3D-CNNs, RNNs, and LSTM-CNNs are some of the design choices. In AAR, CNNs manage changes in lighting, background, camera angle, clothing, and body shape. Researchers combine transfer learning with LSTM networks to improve how models handle sequential data. A new framework uses pre-trained models integrated with LSTM networks to improve their accuracy and usefulness. [15].This hybrid approach [16] can, however, make things more complicated for computers. Ahmed et al. [17] recently created a deep learning detection system for self-driving cars that focuses on detecting snatching in real time in busy, changing environments. Their work offers a solid framework for crime detection through the use of CNNs and LSTM networks, but it doesn't focus on mobile phone snatching.

Their research concentrates on snatch theft detected via video, analyzed by scrutinizing events across successive frames. LSTM networks are used to keep frames information so that they can understand what is happening in a video. Researchers are interested in action recognition because it can be used in many different fields, including surveillance, robotics, human-computer interaction, sports analytics, gaming, and managing online videos. Recognizing actions in video takes a lot of computer power because the frame rates are high and every frame is important. Some suggested methods are motion tracking, space-time characteristics, and

Figure 1. Snatching Types and victim reaction overview.

path analysis. Our method directly deals with phone snatching by using an LSTM network to look at changes in action from frame to frame.

Our research aims to tackle these challenges by creating a hybrid model that merges Time-Distributed CNNs with attention-oriented LSTM networks, resulting in a thorough and effective system for identifying mobile snatching. One important thing this study did was make a new dataset called Mobile Snatching. We put together this dataset from videos we found on different online platforms. It provides us a big and strong set of data to use to train our model. The attention layer is another important part of it. It uses a probability distribution to give weights to important cues in the sequence, which makes it easier to find snatches. These videos show people stealing things, which makes the dataset very useful and important for this research.

The Table 2 report shows clearly how many cell phones were stolen in different cities in Pakistan over the years.This work has made the following contributions:

- For automatic snatching detection, we present a "mobile snatching dataset." The dataset includes videos taken from social media sites like YouTube, TikTok, Google, Facebook, and Twitter (X).
- We create a new model that combines a Time-Distributed CNN with an attention-based LSTM neural network to find mobile snatching. This new method makes it easier to understand and identify complicated activities in the real world, especially in videos of snatch-theft events.

2 Related Work

Thefts and crimes on the street are big problems for society. Petty crimes, especially thefts, are becoming more and more dangerous to public safety. This trend makes it necessary to have smart surveillance and preventive systems that can find and stop crimes as they happen. The increase in these brazen crimes shows how important it is for police to have automated video-based crime detection systems to keep the public safe and keep the community safe. In response, studies employ computer vision and deep learning to identify, categorize, and comprehend human actions and criminal activities in surveillance footage.

2.1 Videos Dataset

The necessity to create systems that can comprehend and recognize intricate activities in real-world environments has resulted in the establishment of standardized datasets for the training and assessment of algorithms. However, existing benchmarks have problems in a few important areas: the number of activity categories included, the number of samples available for each category, the length of each sample, the variety of video capture conditions and environments, and the range of category taxonomy diversity. We look at a few important action datasets to get a better idea of the resources that are available and how they affect

Table 1. Snatching Incident Types and Associated Victim Responses

Snatch Type	Victim Reaction
Distraction and grab	The victim fights back
Grab and run	The victim yells for help
Fake accident and grab	The victim calls the police
Group snatching	The victim tries to fight back
Snatching from a vehicle	Victim sounds alarm

Table 2. Mobile Phone Snatching Statistics Across Major Cities

Year	City	Snatched	Source
2022	Karachi	Over 19,000	Dawn News [18]
2021	Karachi	12,000 (Jan–Jul)	The News Int. [19]
2020	Karachi	2,510 (Sep)	Geo News [20]
2023	Karachi	29,536 (Jan–Oct)	Express Tribune
2019–23	Lahore	14,000	The Nation [21]
2023	Islamabad	Over 680	ARY News [20]
2023	Peshawar	Over 1750	Dunya News [22]

things.Early examples, such as KTH [23] and Weizmann, were mostly about simple human actions. Based on this, datasets like UCF101 [24] and THUMOS, which came from online videos, became important benchmarks for video classification.

After that, Kinetics [25] and YouTube-8M [26] used YouTube videos that were already available to add more types of events. The micro-videos dataset [27] used social media content to look at a wider range of video vocabulary. ActivityNet [28]focuses on identifying activities in videos, while AVA[29] focuses on accurately recognizing and locating actions. Datasets like Something-Something [30] and Charades used crowd-sourced data collection [31] records everyday human activities in spatiotemporal settings in the real-world.

EPIC-KITCHENS-100 is one of the most recent datasets. [32], which focuses on recognizing objects and actions in kitchen settings; Moments in Time for understanding actions; HVU [33] for for human-vehicle interaction; and Ego4D [34] as a big source of information about egocentric video. Innovations keep happening, like HVP [35] for watching people, victims, and predators; AV-Lanes for self-driving and lane detection; UCCS Da- taste [36] for finding crimes in surveillance footage; Street Scene Datase [37] for street crime detection, and RECOLOR [38],CRIME [39],and VSD [40] for finding or classifying violent or criminal events, like snatching and robbery, in different video contexts.

2.2 Videos Classification

Several studies have utilized transfer learning and LSTM networks for sequential data analysis. For instance, Yesinia et al. [41] showed how to change pre-trained CNN models to work well for different tasks. They got good results in recognizing different activities in video footage. In the same way, Al-Selwyn, S. Met al. [42] used LSTMs to look at time sequences in video data, focusing on how well they can handle long-term dependencies. One drawback of this method is the possibility of domain mismatch, since pre-trained models may not always work well with new tasks or datasets. If the source and target domains are very different, this can lead to worse performance.

U. M. Butt al.[43] They discussed about using CNNs with the VGG19 model to find large images or videos of snatch-steal crimes. They used 21 movies as a database and found snatch stealing with 81% accuracy. The authors encountered limitations in selecting features or objects, commonly experienced during snatching, utilizing MATLAB Image Labeler software, and had to take into account the quantity, velocity, and number of moving objects that replicate the act depicted in the snatch video. Wang et al. [44] proposed a hybrid model that uses both transfer learning and LSTM networks to predict the stock market. This model adjusts pre-trained CNN models on large image datasets and combines the features learned with LSTM layers to make predictions about time series.

A possible drawback is the restricted transferability of features acquired from image datasets to financial time series, potentially leading to suboptimal performance or overfitting. S. A. et al. [45] We researched how to use transfer learning with LSTM networks to identify human activities using data from wearable sensors. We also looked into how to improve pre-trained CNN models on big image datasets and then use LSTM networks to handle data from sensors that comes in in order. One problem is that it might not work.between the source and target domains, since features learned from images may not provide critical information for the analysis of sensor data. Andrade et al. [46] Utilized optical flow features derived from crowd dynamics and Hidden Markov Models to detect anomalous occurrences such as exits and falls. focusing on things that don't happen very often without knowing a lot about specific activities. Ihaddadene and Djeraba are not the same. [47] looked at how people moved in real time by guessing when there were sudden changes at certain points of interest. These methods show how optical flow features can help find unusual behavior in crowds, which is useful for security and surveillance.

Kocher Goya et al. [48] established a video surveillance system and a security framework comprising four stages: data collection, object detection, feature extraction, and scene classification. Tracking uses methods like movement, background removal, and optical flow. The system can only see objects from a longitudinal angle, so it can only measure the object's speed at the moment of the snatch. It takes longer to detect objects that are far from the intended location, which suggests that snatch-stealing incidents may not happen in the same place. As we said before, snatch theft is when someone steals valuables like purses, jewelry, cell phones, and other expensive items from people who are walking by. This is often done using run-and-rob strategies. In most cases of snatch stealing, two people work together, with one person driving a motorcycle and the other person getting close to or grabbing the victim.

3 Proposed Methodology

Spatio-temporal modeling captures changes in visual tempo, and spatial information is very important for recognizing actions [49]. When analyzing videos with little change in appearance, video networks must rely heavily on differences in time and space to find relevant cues. In spatio-temporal modeling, temporal information pertains to variations over time, whereas spatial information concerns the distribution and configuration of features in space.

Conventional temporal modeling methods employing 3D CNNs strive to encapsulate visual tempo at the frame level. [50].However, difficulties remain in effectively modeling varied temporal patterns while concurrently tackling spatial complexities within frames. To solve these problems, we suggest a new hybrid model for video classification that combines the best features of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks with an attention mechanism. The combined model has a Time-Distributed CNN and an LSTM that is based on attention. Where:

- The Time-Distributed CNN layer of the combined model pulls out spatial features from video frames. The combined model's LSTM layer that is based on attention does the same thing.
- The attention-based LSTM layer finds temporal relationships and the importance of features across frames, which lets the model focus on the most important parts of the video.

3.1 Data Preprocessing

Data preparation is crucial for how well a model works because it makes the data better, which makes it easier to extract and understand features. The way data is pre-

processed before it is put into a model has a big impact on how well the model works.

3.1.1 Frame Extraction

Frame extraction is a key step in getting our data ready. We make videos 64 frames long for our hybrid `TD_CNN-LSTM` model. The videos in our dataset are different lengths because the actions in them are different lengths. If a video has fewer than 64 frames, the first and last frames are added to it. If a video has more than 160 frames, every third frame is chosen.This method gets rid of frames that aren't needed, which makes the computer do less work and lowers the chance of overfitting.

3.1.2 Frame Normalization

Frame normalization makes sure that all frames are the same and can be compared. This makes the model more reliable. We utilize the formula presented in Equation (1) [51].

$$I_{\text{normalized}}(x,y) = \frac{I(x,y) - I_{\min}}{I_{\max} - I_{\min}} \tag{1}$$

Equation (1) djusts each pixel's brightness by using the lowest ($I_{\min}$) and highest ($I_{\max}$) brightness values found in each frame. By fitting all of the pixel values into a common range, this procedure prepares them for additional analysis. The frame data is further improved by cropping after normalization.Utilizing the lowest.

3.1.3 Frame Cropping

We resized each frame to 256×256 pixels before adding it to the model as part of the preprocessing step. So, we were able to get specific areas of interest from a larger frame, which had many benefits. First, cropping helps the model focus on the most important details, which lets it put the most important information first for the task. Also, it makes the data smaller, which makes processing it later more efficient and lowers the amount of computing power needed. [52]. The cropping process can be represented mathematically as shown in Equation (2):

$$C = I[X_1, X_2 : Y_1, Y_2] \tag{2}$$

In the above equation C denotes the cropped frame, which is a subsection of the original image I. The coordinates X_1 and Y_1 represent the top-left corner of the crop, while X_2 and Y_2 indicate the bottom-right corner of the crop area.

3.2 Proposed Model

This section explains how the proposeded TD_CNN-LSTM classifier model is put together. It has two main parts:

- **Time-Distributed CNN layer:** This layer processes data in order and is very important for getting spatial features from each frame.
- **LSTM layer:** The LSTM layer processes the sequential data even more after the Time-Distributed CNN layer. It does this to find long-term temporal dependencies in the features that the CNN found.

The proposed model, shown in Figure 2, has the first three layers of the Time-Distributed 3D CNN, followed by the LSTM layers. This setup makes it easy to model both time and space properties.

3.2.1 Features Extraction with Time Distributed CNN

The first part of our proposed model is made up of three layers of Time-Distributed CNNs [53]. The Time-Distributed CNN extracts spatial features from frames, but it doesn't explicitly capture temporal dependencies because it processes each frame separately. To get around this problem and better model sequential relationships, temporal convolutional layers are added. These layers make standard convolutional layers better by letting them handle sequential information more quickly.

This block takes in a tensor that is $10 \times 240 \times 240 \times 3$, where 10 is the number of timestamps (or frames) in the sequence 240×240 is the height and width of each frame, and 3 is the number of color channels. After each layer in this block, normalization is done, and then a ReLU activation layer is added to make it non-linear. The outputs from each layer are activated by ReLU, and the FC8 layer creates a feature vector with 1,000 dimensions that has shown good results. In initial experiments, the feature of a specific frame is regarded as a distinct segment and inputted into each RNN iteration. We process six of the thirty frames that were sampled over a period of T_s, rewhich is one second, with a stride of six frames in the video sequence. The RNN processes these features in six

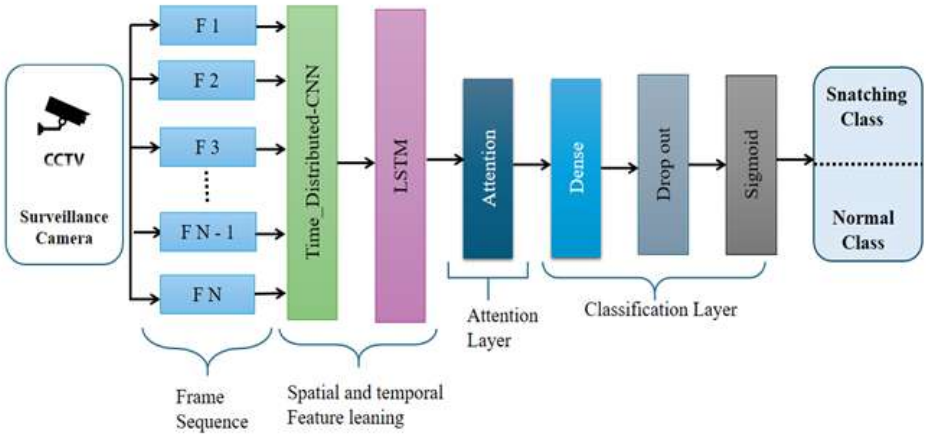


Figure 2. Overall Structure of the Proposed TD_CNN-LSTM Classifier Model

steps, one for each frame that was chosen. At each time step, the RNN's last hidden state is combined with the others to make the final recognition decision.

After the block of convolution, We get feature maps that are $30 \times 30 \times 32$ pixels in size. Then, these feature maps are turned into a vector with 28800×1 elements and sent to a fully connected layer. The stride values for the pooling and convolutional layers, as well as the number of kernel maps, are carefully chosen to make sure that the features that come out have the right size. More information about the RNN can be found in the next sections. This setup lets the model pick up on both spatial and temporal features..

3.2.2 Attention Based LSTM spatial Feature Extraction

Hochreiter and Schmidhuber proposed the Long Short-Term Memory (LSTM) network [54], a variant of Recurrent Neural Networks (RNNs) designed to store and retrieve information over long sequences. Unlike conventional RNNs, LSTMs introduce nonlinear multiplicative gates and a memory cell that regulate the flow of information. These gates input, output, and forget control how information is written to, read from, and removed from the memory cell. All spatial characteristics are obtained from the final max-pooling layer of the TDCNN, which produces feature maps of dimension $1 \times 1 \times 28800$.

As illustrated in Figure 3, the proposed model generates a sequence of feature vectors of dimension $m \times n$, where m denotes the number of features and n is the number of frames in a video. After that, the LSTM gets these feature vectors, which show how the TDCNN-extracted sequence changes over time. In mobile snatching detection, it can be hard to find and predict snatching events from video frames, especially when the differences between frames are small. Also not having enough training data can make the model work poorly and cause it to make mistakes. To deal with these problems, we use a Time-Distributed CNN–LSTM architecture that is based on attention. A TimeDistributed wrapper is used to apply the CNN to each individual frame, enabling extraction of spatial features. These features are subsequently processed by the LSTM network to model temporal relationships and detect anomalies within the sequence.

Mathematically, for a sequence of feature representations $(X_1, X_2, \ldots, X_T)$, the LSTM maps the inputs to an output sequence $(Y_1, Y_2, \ldots, Y_T)$ by iteratively updating its internal states from $t = 1$ to $t = T$ using the equations given in (3).

$$\begin{aligned} i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right), \\ f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right), \\ c_t &= f_t\, c_{t-1} + i_t\, \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right), \\ o_t &= \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right), \\ h_t &= o_t\, \tanh\left(c_t\right). \end{aligned} \tag{3}$$

In this context, x_t and h_t denote the input and hidden

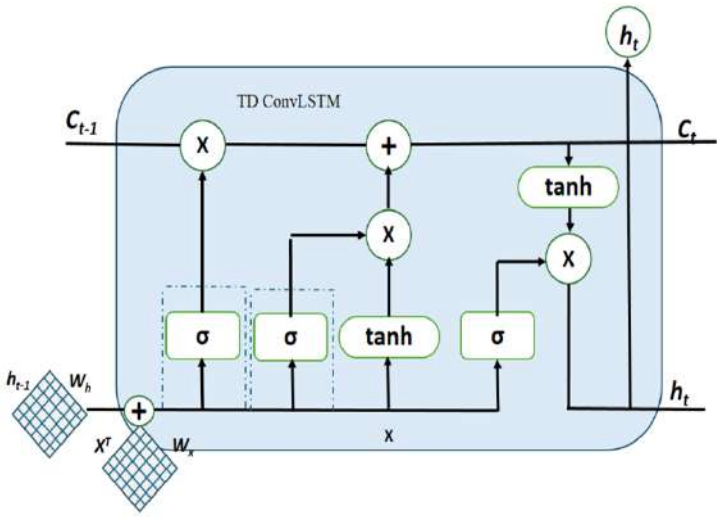


Figure 3. Attention-Driven Temporal Modeling using the Time-Distributed CNN–LSTM Network

state at time t, respectively. The vectors i_t, f_t, c_t, and o_t represent the activations of the input gate, for- get gate, cell state, and output gate, respectively. The matrices $W_{\alpha\beta}$ represent the weights connecting different components; for example, W_{xi} is the weight matrix connecting the input x_t to the input gate i_t. The terms b_α denote the bias vectors of the respective gates, and σ is the sigmoid activation function, defined in below Equation (4).

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

The core concept of the LSTM model, shown in Figure 4, relies on an internal memory cell that retains information over long durations to address sequence dependencies. Nonlinear gating units regulate the flow of information into and out of this cell. As de- scribed in the equations above, the current input x_t and the previous hidden state h_{t-1} serve as inputs to four components at the t-th time step. The memory cell receives information from two sources: the previous cell state C_{t-1}, modulated by the forget gate activation f_t, and the transformed input, controlled by the input gate activation i_t.

This design enables the LSTM to selectively discard irrelevant past information or incorporate new relevant data. The output gate o_t determines how much information to extract from the updated cell state C_t to produce the next hidden state h_t. These gating mechanisms form memory layers that can be flexibly adjusted. Due to these functional gates, LSTMs effectively maintain long-range temporal coherence for sequences of varying lengths. Consequently, the LSTM model can operate as a deep neural network by using the hidden states of one layer as the inputs to the next.

Consider a model with K layers. At the t-th time step, the feature vector x_t enters the first LSTM layer along with the hidden state h_{t-1}^1 from the previous time step. This process produces an updated hidden state h_t^1, which is then passed as input to the next layer. Let f_W denote the function that maps inputs to hidden states. The transition from layer $L-1$ to layer L is defined as:

$$\text{For } L = 1: \quad h_t^1 = f_W(x_t, h_{t-1}^1) \tag{6}$$

$$\text{For } L > 1: \quad h_t^L = f_W(h_t^{L-1}, h_{t-1}^L) \tag{7}$$

To predict the scores for a total of C classes at a given time step t, , the output from the last LSTM layer is passed through a sigmoid layer to estimate class probabilities:

$$\text{prob}_c = \sigma\left(W_c^T h_t^K + b_c\right), \quad c = 1, 2, \ldots, C \tag{8}$$

where prob_c is the predicted probability for class c, and W_c and b_c denote the weight vector and bias term for the c-th class, respectively. The attention weights α_t are computed using a feed-forward architecture with a

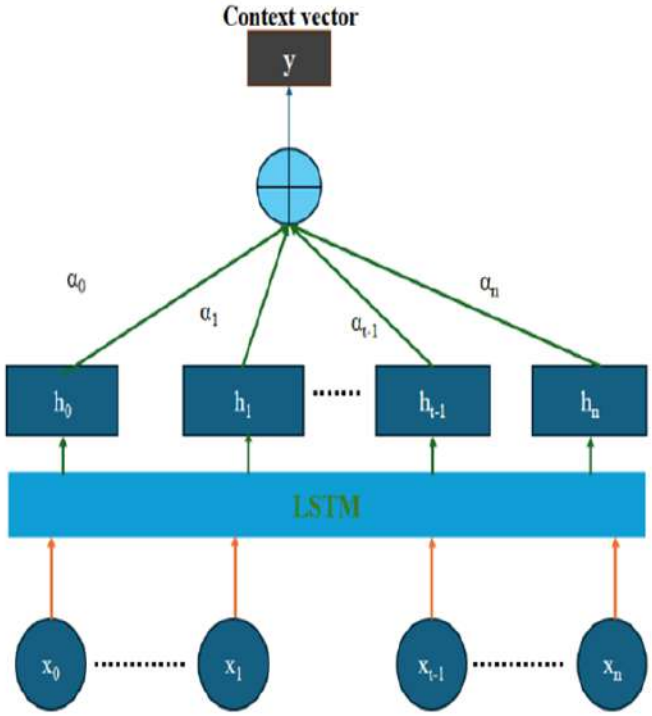


Figure 4. Overview of the Attention-Based LSTM for Spatiotemporal Feature Learning.

hyperbolic tangent activation function, as shown in Equation (9). In this equation, , v^t and w_a represent the attention weight vector and weight matrix, respectively:

$$\alpha_t = v^t \tanh(w_a[x_t, h_i]) \tag{9}$$

The unnormalized attention scores are then normalized through the softmax function as:

$$\alpha_t = \frac{\exp(\text{score}(x_t, h_i))}{\sum_{i=1}^{t} \exp(\text{score}(x_t, h_i))} \tag{10}$$

The context vector at time t is calculated as the weighted sum of hidden states from the input sequence.

$$C_t = \sum_{i=1}^{t} \alpha_{ti} h_i \tag{11}$$

In this expression, α_{ti} indicates the attention weight that corresponds to the significance of the i-th hidden state in deciding what the final representation of the sequence will be. The LSTM extracted spatiotemporal snatching feature vector is then sent to a dense layer with 128 units. After that, a dropout layer with a dropout rate of 0.7 is used to stop overfitting during training. Lastly, a fully connected layer with as many neurons as there are classes in the dataset and a sigmoid activation function is used to predict both snatching events and normal frames. Algorithm 1 shows how the TD_CNN-LSTM model works.

4 Experimental Setup and Results

This section discusses into extensive detail about how the suggested method was put into action and tested. It was made with TensorFlow as the backend and Keras as the deep learning library. All tests were done with Python 3.6, TensorFlow v0.1.11, and Keras v3.0.2. We used a TD-CNN model in Keras to extract features. We trained the model on a desktop computer with an Intel i7-9700F processor, 16 GB of RAM, and an NVIDIA RTX 2080 Super GPU. This setup was chosen to speed up training and handle high-dimensional visual data well.

4.1 Dataset Collection

The mobile snatching dataset has 200 videos that show how quickly people can steal phones. In most cases,

Algorithm 1. Time-Distributed CNN and Attention-Based LSTM Neural Network

Input: Video frame sequence $\{f_1, f_2, \ldots, f_T\}$

Output: Predicted class label

Step 1: Spatial Feature Extraction

for each frame f_t in the video sequence **do**

Extract spatial features using TimeDistributed CNN:

$$\mathbf{s}_t = \text{TimeDistributed(CNN)}(f_t)$$

end for

Step 2: Temporal Feature Extraction

for t = 1 to T **do**

Compute hidden state using LSTM Equations (4)–(11):

$$\mathbf{h}_t = \text{LSTM}(\mathbf{s}_t)$$

Append $\mathbf{h}_t$ to the hidden state sequence

end for

Step 3: Attention Mechanism

for each hidden state $\mathbf{h}_t$ **do**

Compute attention weight α_t and context vector $\mathbf{c}_t$:

$$\alpha_t, \mathbf{c}_t = \text{Attention}(\mathbf{h}_t)$$

Apply Dense layer with ReLU activation:

$$\mathbf{d}_t = \text{Dense}(128, \text{ReLU})(\mathbf{c}_t)$$

Apply Dropout layer with rate 0.7:

$$\mathbf{z}_t = \text{Dropout}(0.7)(\mathbf{d}_t)$$

Generate final classification output using sigmoid activation:

$$\hat{y}_t = \text{Dense}(1, \sigma)(\mathbf{z}_t)$$

end for

thieves on bikes, in cars, or on foot are targeting people who are not paying attention, like people sitting in cafes or on the street. The videos were taken from a number of social media sites, such as YouTube, TikTok, Twitter, Facebook, and Google. Also, publicly available datasets and old surveillance footage were carefully looked at to find useful samples. After a lot of work collecting and sorting through the videos, 100 were found to be mobile snatching incidents, and 100 were found to be normal (non-snatching) events, as shown in Figure 5 and 6.

This balanced dataset made it possible to train and test the proposed models well.It only takes 4 to 5 seconds for most snatch thefts to happen. We divided the dataset into three groups: 70% for training, 10% for testing, and 20% for validation. We made sure that both groups were equally represented. The videos were all resized to 240× pixels by 240 pixels and had a sequence length of 10 frames to keep the samples the same. This systematic and thorough preprocessing workflow set the stage for training and testing the TD-CNN-LSTM model.

4.2 Performance Indicators

We use standard evaluation metrics like accuracy, recall, and precision to judge how well the proposed model works. These metrics together give us an idea of how well the model can tell the difference between frames that are snatching and frames that are not.

Accuracy is the percentage of samples that are correctly classified out of all samples. It is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

The letters *TP*, *TN*, *FP*, and *FN* stand for true positives, true negatives, false positives, and false negatives, respectively.

Recall checks how well the model can find all the real positive cases. A higher recall means that fewer positive detections were missed, as shown mathematically below: Equation (6).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

In the same way, precision measures the percentage of correctly predicted positive samples out of all predicted positive samples. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

Figure 5. Representative Normal Frames Depicting Non-Snatching Scenarios.

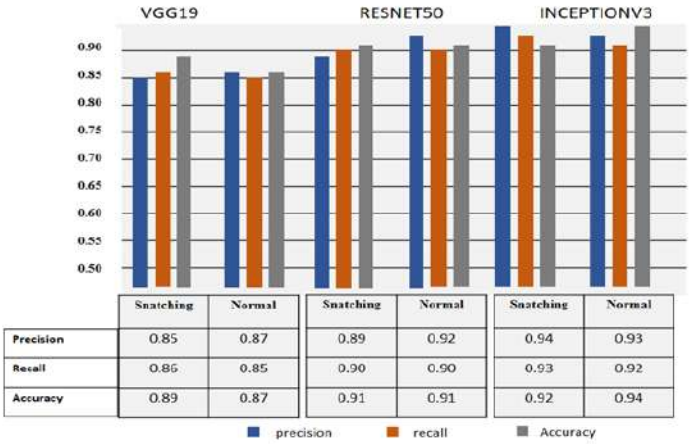
Figure 6. Representative Frames Depicting Mobile Snatching Incidents.

4.3 Transfer Learning-based Performance Evaluation

This section clarifies about how we used pre-trained models on our benchmark dataset for snatching mobile phones. Transfer learning is a very important part of machine learning. It uses knowledge gained in one area to make things better in another. For instance, CNNs' convolutional layers are great at finding features in images, even when they are trained on data sets other than ImageNet. Transfer learning allows us to use what we learned from ImageNet-trained models on other tasks, which helps us get around the problem of not having enough training data for big CNN models. We trained the models for 50 epochs and tested a few of the best CNN architectures, including VGG19, ResNet50, and Inception V3, to find the one that worked best for our task. ImageNet was used to train all of the models used in this study. The CNN models have set parameters, and the last fully connected layer has been taken out. At first, features are taken from the layer before the fully connected layer that was removed. After that, these features go into a new fully connected layer, and then a sigmoid output layer for classification. LSTM layers were added to the model to capture temporal dependencies. These layers took sequential features from the CNN features and made the model better at finding temporal patterns in the data. The different models' accuracies are shown in Figure 7, VGG19, ResNet50, and Inception V3 got scores of 87.27%, 91.29%, and 94.45%, respectively.

4.4 Result with hybrid TD_CNN-LSTM

We propose a new architecture for video classification that effectively captures both spatial and temporal features in video data. The model starts with a Time Distributed wrapper that applies several convolutional layers to each frame of the input video. These layers take spatial features from each frame. We used three groups of convolutional layers with increasing filter sizes (16, 32, and 64) and ReLU activation functions to add non-linearity. Batch normalization layers were added after each convolutional layer to make training more stable. These layers normalized activations and made the



	Snatching	Normal	Snatching	Normal	Snatching	Normal
Precision	0.85	0.87	0.89	0.92	0.94	0.93
Recall	0.86	0.85	0.90	0.90	0.93	0.92
Accuracy	0.89	0.87	0.91	0.91	0.92	0.94

Figure 7. Performance Comparison of Transfer Learning Models.

model as a whole more robust. We used max-pooling layers with the right kernel sizes to downsample the spatial dimensions, which made the computation less complex. To stop overfitting, dropout layers were added after each max-pooling layer.During training, dropout was used to make the model better at generalizing by randomly turning off neurons. The feature maps were flattened after the convolutional layers and sent to an LSTM layer with 10 units to find patterns between frames over time. The Mobile Snatching Dataset was used to train the CNN-LSTM model for 50 epochs. The Adam optimizer was used with a learning rate of 0.0001 and a batch size of 16. The benchmark dataset showed that it was 96.45% correct. We used a different test set to see how well the model worked.

Figure 8. shows the training and validation loss curves over 50 epochs. In the beginning, the loss goes up from epoch 0 to epoch 20 (loss measures how well the model's predictions match the true labels; lower loss means better predictions). After that, the loss keeps going down, and by epoch 50, it is much lower, which means the model is working better. Over 50 epochs, we also looked at precision and recall metrics. Precision measures the percentage of correct positive predictions, while recall measures the percentage of actual positives that were correctly identified. The dataset was divided so that 70% was used for training, , 20% for validation, and 10% for testing. Figure 8shows the results. The learning curves show that the model fits well because the training and validation curves are very close to each other, which means that the model is working at its best.

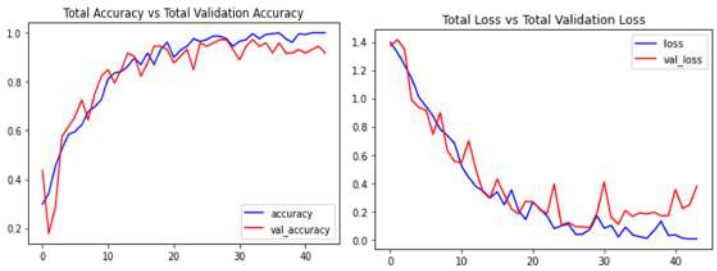


Figure 8. Proposed method training and validation loss and accuracy over 50 epochs.

Figure 9 shows the confusion matrix, which gives a detailed look at how well the model did on the test set by showing how many predictions were right and wrong for each class. The matrix shows that the model correctly identified 115 cases of "Snatching" and 118 cases of "Normal" behavior. However, it also shows some mistakes in classification. For example, 5 "Snatching" cases were wrongly labeled as "Normal," and 2 "Normal" cases were wrongly labeled as "Snatching." This analysis shows that the model is very good at telling the difference between snatching and normal behavior. It also points out areas where it could be better to cut down on false positives and false negatives.

4.5 Comparison with Existing Techniques

"Mobile/Phone Snatching" dataset to compare the proposed scheme's performance to that of other frame-

Figure 9. Confusion Matrix for TD-CNN-LSTM Model on Mobile Snatching Dataset.

Table 3. Comparison of Different Models on the Mobile/Phone Snatching Dataset

Model Architecture	Configuration	Epochs	Average Time (s)	Accuracy (%)
LanHAR [55] (2024)	RGB	100	0.41	72.10
Full Transformer Network [56] (2022)	RGB	100	0.0521	85.72
SPOTER [57] (2022)	Pose	100	0.034	73.53
MIPA-ResGCN [58] (2023)	Pose	150	0.0491	85.43
SIGNGRAPH [59] (2023)	Pose	150	0.0521	84.01
Hybrid CNN LSTM Model [60] (2024)	RGB	100	0.0307	88.32
Our TD-CNN-LSTM with Attention	RGB	50	0.0371	96.45

works that use RGB and pose modalities. The results are shown in Table 3. The authors in [55] put forth a technique for approximate alignment of distinct actions, utilizing temporal attention mechanisms to identify instances of mobile snatching. This method focuses on recognizing actions by looking at how they relate to each other over time, which makes it easier to find the right sequence of events. But it doesn't deal with problems that come up when sensor locations and activity patterns are different in different datasets. LanHAR, on the other hand, uses Large Language Models (LLMs) to create semantic representations of sensor data and activity labels. This solves these problems and gives a better basis for cross-dataset Human Activity Recognition (HAR) and finding new activities. A full transformer architecture is used for action recognition. The Swin Transformer serves as the spatial encoder, while a masked future transformer replaces the conventional masked self-attention mechanism for action detection [56]. We also examined pose-based approaches such as the Sign Pose-based Transformer (SPOTER) and the Graph Convolutional Network (GCN) integrated with BERT. It was observed that relying exclusively on pose data significantly reduced action detection accuracy compared to the use of complete RGB data.The drop is because pose or skeleton data doesn't have enough fine-grained details, like how hands move and how people interact with objects, that are needed to find mobile snatching events. Our hybrid CNN-LSTM network with attention made predictions that were 96.45% accurate, which was better than other methods.

5 Conclusion

In conclusion, The suggested hybrid framework, which mixes attention-based LSTM networks with Time-Distributed CNNs, is very good at finding cases of mobile snatching. The attention mechanism helps the model find important time-based cues in video sequences, which makes detection more accurate. But there are still some limits.We haven't fully tested how well the model works in different city settings, and it might not work as well in situations with a lot of occlusion or in low-quality footage. Also, it might be hard to use in real time because it needs a lot of processing power.Future research can tackle these issues by developing scalable methodologies for varied urban settings, integrating edge computing to enhance processing efficiency, and optimizing datasets to bolster model resilience.It might also help to add more advanced neural architectures and test the model on different types of crimes. More

work in this area could make public safety systems much better overall.

Data Availability Statement

The dataset and source code used in this study is publicly available and can be accessed at: https://drive.google.com/drive/folders/130rbkDPgf-ixJFfOGgDxIdhRraT3fKfZ?usp=sharing.

Author Contributions

Writing-Original draft preparation, Conceptualization, Methodology, Data curation, Software implementation, Visualization, Writing and Editing. **Faisal Khan**. **Irshad Ahmad**:Conceptualization, Methodology, Investigation, Supervision, Software Validation, Writing- Reviewing and Editing, but not limited to the above keywords. **Muhammad Zubair**: Conceptualization, Methodology, Investigation, Supervision, Software Validation, Writing-Reviewing and Editing, but not limited to the above keywords. **Yasir Saleem Afridi**: Investigation, Software Validation, Writing- Reviewing and Editing, but not limited to the above keywords.

Compliance with Ethical Standards

The authors declare no conflicts of interest.

Funding Information

This research received no external funding.

References

[1] K. Walby, "Open-street camera surveillance and governance in Canada," *Canadian Journal of Criminology and Criminal Justice*, vol. 47, no. 4, pp. 655-684, 2005.

[2] A. Isnard and T. C. Council, "Can surveillance cameras be successful in preventing crime and controlling anti-social behaviours," *The Character, Impact and Prevention of Crime in Regional Australia Conference*, pp. 1-3, Aug. 2001.

[3] N. F. M. Zamri, N. M. Tahir, M. S. M. Ali, N. Dalila, K. Ashar, and A. Abd Almisreb, "Real time snatch theft detection using deep learning networks," *Institute for Big Data Analytics and Artificial Intelligence (IBDAAI)*, vol. 2, no. 3, p. 4, 2023.

[4] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, no. 1, p. 7068349, 2018.

[5] J. Zou, Y. Han, and S. S. So, "Overview of artificial neural networks," in *Artificial Neural Networks: Methods and Applications*, pp. 14–22, 2008.

[6] S. R. Ke, H. L. U. Thuc, Y. J. Lee, J. N. Hwang, J. H. Yoo, and K. H. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, 2013.

[7] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2021.

[8] Y. Liu, Y. Song, Y. Zhang, and Z. Liao, "WT-2DCNN: A convolutional neural network traffic flow prediction model based on wavelet reconstruction," *Physica A: Statistical Mechanics and its Applications*, vol. 603, p. 127817, 2022.

[9] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3200–3225, 2022.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[11] M. Humayun, R. Sujatha, S. N. Almuayqil, and N. Z. Jhanjhi, "A transfer learning approach with a convolutional neural network for the classification of lung carcinoma," *Healthcare*, vol. 10, no. 6, p. 1058, 2022.

[12] T. H. Nguyen, T. N. Nguyen, and B. V. Ngo, "A VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease," *AgriEngineering*, vol. 4, no. 4, pp. 871–887, 2022.

[13] M. A. Mamun, M. S. Kabir, M. Akter, and M. S. Uddin, "Recognition of human skin diseases using inception-V3 with transfer learning," *International Journal of Information Technology*, vol. 14, no. 6, pp. 3145–3154, 2022.

[14] X. Yin, W. Li, Z. Li, and L. Yi, "Recognition of grape leaf diseases using MobileNetV3 and deep transfer learning," *International Journal of Agricultural and Biological Engineering*, vol. 15, no. 3, pp. 184–194, 2022.

[15] D. Dresvyanskiy, E. Ryumina, H. Kaya, M. Markitantov, A. Karpov, and W. Minker, "End-to-end modeling and transfer learning for audiovisual emotion recognition in-the-wild," *Multimodal Technologies and Interaction*, vol. 6, no. 2, p. 11, 2022.

[16] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Hybrid deep neural networks for detection of non-technical losses in electricity smart meters," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1254–1263, 2019.

[17] M. Wasim, I. Ahmed, J. Ahmad, and M. M. Hassan, "A novel deep learning based automated academic activities recognition in cyber-physical systems," *IEEE Access*, vol. 9, pp. 63718–63728, 2021.

[18] M. Jan, "Print media on coverage of political parties in Pakistan: treatment of opinion pages of the 'Dawn' and 'News'," *Gomal University Journal of Research*, vol. 29, no. 1, pp. 118–128, 2013.

[19] S. Hess, *International News and Foreign Correspondents*, vol. 5, Brookings Institution Press, 1996.

[20] K. Sultan, A. Iqbal, Z. Khalid, and S. Ali, "Media ethics and responsibility: analysis of GEO news and ARY news' coverage on Hamid Mir's issue," *Journal of Social Sciences*, pp. 225–249, 2016.

[21] T. A. Van Dijk, *News Analysis: Case Studies of International and National News in the Press*, Routledge, 2013.

[22] R. B. Sohail, "Analyzing the portrayal of political leadership in leading Pakistani news channels: A critical analysis," *Journalism, Politics and Society*, vol. 2, no. 1, pp. 69–86, 2024.

[23] S. M. Kang and R. P. Wildes, "Review of action recognition and detection methods," arXiv preprint arXiv:1610.06906, 2016.

[24] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human action classes from videos in the wild," arXiv preprint arXiv:1212.0402, 2012.

[25] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics human action video dataset," arXiv preprint arXiv:1705.06950, 2017.

[26] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8M: A large-scale video classification benchmark," arXiv preprint arXiv:1609.08675, 2016.

[27] P. X. Nguyen, Y. Wang, M. S. Ryoo, A. Shah, and L. Davis, "The open world of micro-videos," arXiv preprint arXiv:1603.09439, 2016.

[28] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[29] N. Murray, L. Marchesotti, and F. Perronnin, "AVA: A large-scale database for aesthetic visual analysis," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, IEEE.

[30] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, J. Haenel, I. Fischer, O. Bousquet, and Y. Bengio, "The 'Something' video database for learning and evaluating visual common sense," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[31] J. Yoon, H. Park, J. Kim, and S. Lee, "D-Vlog: Multimodal Vlog dataset for depression detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[32] D. Damen, T. D. B. Sousa, G. Rogez, S. Escalera, C. Leistner, S. Little, A. Gupta, and H. Kjellström, "EPIC-KITCHENS-100," *International Journal of Computer Vision*, vol. 130, pp. 33–55, 2022.

[33] D. Lin, S. Fidler, and R. Urtasun, "Holistic scene understanding for 3D object detection with RGB-D cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[34] K. Grauman, A. Cartas, D. Damen, C. D. Castillo, A. Farhadi, A. Furnari, H. Koppula, H. Li, M. Rohrbach, and S. Sengupta, "Ego4D: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[35] W.-R. Ko, Y.-T. Chen, C.-H. Chou, S. Liu, and H. Su, "AIR-Act2Act: Human–human interaction dataset for teaching non-verbal social behaviors to robots," *The International Journal of Robotics Research*, vol. 40, no. 4–5, pp. 691–697, 2021.

[36] J. Laufs, H. Borrion, and B. Bradford, "Security and the smart city: A systematic review," *Sustainable Cities and Society*, vol. 55, p. 102023, 2020.

[37] H. Yue, Y. Li, L. Wang, and X. Zhang, "Detecting people on the street and the streetscape physical environment from Baidu Street View images and their effects on community-level street crime in a Chinese city," *ISPRS International Journal of Geo-Information*, vol. 11, no. 3, p. 151, 2022.

[38] D. D. de Paula, D. H. Salvadeo, and D. M. de Araujo, "CamNuvem: A robbery dataset for video anomaly detection," *Sensors*, vol. 22, no. 24, p. 10016, 2022.

[39] P. Kapoor and P. K. Singh, "Multidimensional crime dataset analysis," in *Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018)*, Vellore, India, Dec. 6–8, 2018, vol. 1, Springer, 2020.

[40] B. M. Peixoto, M. A. Pimentel, M. A. T. Figueiredo, and J. L. Oliveira, "Harnessing high-level concepts, visual, and auditory features for violence detection in videos," *Journal of Visual Communication and Image Representation*, vol. 78, p. 103174, 2021.

[41] M. K. Singh and B. Kumar, "Fine tuning the pre-trained convolutional neural network models for hyperspectral image classification using transfer learning," in *Computer Vision and Robotics: Proceedings of CVR 2022*, Springer, 2023, pp. 271–283.

[42] S. M. Al-Selwi, M. H. Ali, A. A. H. Al-Saedi, and M. A. Khan, "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review," *Journal of King Saud University-Computer and Information Sciences*, p. 102068, 2024.

[43] U. M. Butt, S. S. Iqbal, and M. Z. Khan, "Detecting video surveillance using VGG19 convolutional neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, 2020.

[44] T. Wang, T. Liu, and Y. Lu, "A hybrid multi-step storm surge forecasting model using multiple feature selection, deep learning neural network and transfer learning," *Soft Computing*, vol. 27, no. 2, pp. 935–952, 2023.

[45] S. An, J. Lee, M. Kim, and H. Park, "Transfer learning for human activity recognition using representational analysis of neural networks," *ACM Transactions on Computing for Healthcare*, vol. 4, no. 1, pp. 1–21, 2023.

[46] E. L. Andrade, S. Blunsden, and R. B. Fisher, "Hidden Markov models for optical flow analysis in crowds," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006, IEEE.

[47] N. Ihaddadene and C. Djeraba, "Real-time crowd motion analysis," in *2008 19th International Conference on Pattern Recognition*, 2008, IEEE.

[48] K. Goya, T. Higuchi, K. Fujimura, and M. Nakajima, "A method for automatic detection of crimes for public security by using motion analysis," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2009, IEEE.

[49] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, B. Schiele, and C. C. Loy, "AVA: A video dataset of spatio-temporally localized atomic visual actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[50] Y. Lee, J. Kim, H. Kim, S. Park, and K. Sohn, "Diverse temporal aggregation and depthwise spatiotemporal factorization for efficient video classification," *IEEE Access*, vol. 9, pp. 163054–163064, 2021.

[51] I. M. Pires, R. Alves, M. A. Silva, and J. M. R. S. Tavares, "Homogeneous data normalization and deep learning: A case study in human activity classification," *Future Internet*, vol. 12, no. 11, p. 194, 2020.

[52] K. Apostolidis and V. Mezaris, "A fast smart-cropping method and dataset for video retargeting," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, IEEE.

[53] S. Montaha, M. H. Asghar, A. Al-Obaidi, and M. T. Mahmood, "TimeDistributed-CNN-LSTM: A hybrid approach combining CNN and LSTM to classify brain tumor on 3D MRI scans performing ablation study," *IEEE Access*, vol. 10, pp. 60039–60059, 2022.

[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[55] H. Yan, H. Tan, Y. Ding, P. Zhou, V. Namboodiri, and Y. Yang, "Language-centered human activity recognition," arXiv:2410.00003, 2024.

[56] Y. Du, X. Li, H. Zhang, and J. Wang, "Full transformer network with masking future for word-level sign language recognition," *Neurocomputing*, vol. 500, pp. 115–123, 2022.

[57] C. Lazo-Quispe, J. Smith, M. Chen, and A. Kumar, "Impact of pose estimation models for landmark-based sign language recognition," 2024.

[58] N. Naz, M. R. Khan, S. Ahmed, and A. Iqbal, "MIPA-ResGCN: A multi-input part attention enhanced residual graph convolutional framework for sign language recognition," *Computers and Electrical Engineering*, vol. 112, p. 109009, 2023.

[59] N. Naz, M. R. Khan, S. Ahmed, and A. Iqbal, "SignGraph: An efficient and accurate pose-based graph convolution approach toward sign language recognition," *IEEE Access*, vol. 11, pp. 19135–19147, 2023.

[60] H. D. Shoorkand, M. Nourelfath, and A. Hajji, "A hybrid CNN-LSTM model for joint optimization of production and imperfect predictive maintenance planning," *Reliability Engineering System Safety*, vol. 241, p. 109707, 2024.