SKU JOURNAL OF ENGINEERING RESEARCH

Vol-2 Issue-1

(NATIONAL PEER REVIEWED E-RESEARCH JOURNAL)

(January – March 2025)

E-ISSN 3048-930X

Innovative Spatial Filtering Integration for Improved Deep Learning-Based Public Activity Detection in Videos

Onkar Tiwari
Assistant Professor
(Computer Science and Applications)
Shri Krishna University, Chhatarpur (M.P.)

ABSTRACT

A new age in surveillance, security, and safety monitoring has been ushered in by the development of advanced technology and the extensive use of digital media. The identification of crucial public behaviors in video material has become a crucial problem in this changing environment, with important ramifications for internet security, law enforcement, and public safety. With an emphasis on pertained models, this paper explores the use of cutting edge machine learning techniques for the automatic identification of important public actions in video. Computer vision and artificial intelligence may be used to solve significant societal issues *via* public activity monitoring. There is an urgent need for techniques that are not only scalable and effective but also able to precisely detect crucial information because video material is becoming more and more prevalent across platforms and domains.

The motivation behind this study stems from the growing awareness of the potential of artificial intelligence and computer vision to alleviate critical societal issues around the surveillance of public activities. There is an urgent need for techniques that are not only scalable and effective but also able to precisely identify crucial actions and reduce related risks, as video material is becoming more and more common across platforms. In this paper, the potential of pertained models in particular, convolution neural networks pertained on extensive video datasets to detect spatial characteristics and patterns suggestive of important public actions is investigated. This project aims to improve our knowledge of public activity identification in video and provide useful solutions with real world applications through a rigorous combination of theoretical analysis, algorithmic development, and empirical assessment.

KEYWORDS

Artificial Intelligence, Deep Learning, Technology, Algorithmic Development.

INTRODUCTION

Customized Feature based Among the earliest methods that use manually created characteristics to identify important behaviors in video recordings are techniques for identifying key public activities. These techniques identify activities using parameters like optical flow, motion vectors, and texture descriptors using conventional machine learning algorithms like Support Vector Machines (SVMs) or Hidden Markov Models (HMMs).

Despite their widespread usage in the past, these techniques frequently fail to capture intricate spatiotemporal patterns and may not transfer well in different situations. However, they have offered fundamental understandings of the nature of important public operations. These techniques make it easier to record intricate spatiotemporal patterns and variations in many situations. They mostly depend on the caliber of manually created features, which could not adequately convey the depth of visual data in videos.

Consequently, they might not be able to generalize across many datasets and might not function effectively in real world situations with dynamic and varied surroundings. Handcrafted Feature-Based Methods are nevertheless useful despite their drawbacks, especially when combined with deep learning strategies. In order to capitalize on the advantages of both paradigms, future research may investigate hybrid techniques that combine handmade features with deep learning architectures. The efficiency of these approaches for identifying important public actions may be increased by advancements in feature design, selection, and fusion techniques.

The suggested method for detecting critical public activities

The three stages of the suggested model are feature extraction, classification, and dataset collection and preprocessing. Normalization, frame scaling, and video-to-frame conversion are all part of the acquisition step. To categorize frames into two categories—0 for normal activity and 1 for suspicious behavior-the top layer is swapped out, and the model uses a pre-trained VGG19 for feature extraction. The block diagram for the suggested model is shown in Figure 8. A proposed algorithm to evaluate video frames and categorize them as regular or suspicious activity types which are given below-

Step 1:

Extraction of Video to Frame- Extract individual frames from a video with a sequence of frames by using the formula Ft= {f1, f2,…….ft}, where T is the total number of frames recovered from the video (v).

Step 2:

Utilizing VGG19 for Feature Extraction- To extract feature maps, for each frame $ft \in F$ $t$runs the frame through a pre-trained VGG19 network (without the fully connected layers): Xt=VGG19(ft), where Xt is a function of both the dimensionality of the VGG19 output feature map and R dis the feature vector for fragment. Let Ft′={X1, X2,..., XT} stand for these feature vectors that were taken out of every frame.

Step 3:

Extractions of Temporal Features- There are two ways to model temporal interdependence between frames: 3D CNNs, or 3D Convolutional Neural Networks. Let {ft,ft+1,...,ft+n} be a block of successive frames. To capture temporal dynamics, use a 3D convolution in both spatial and temporal dimensions: Where Yt: t+n is the result of the 3D convolution operation spanning nnn frames, Yt: t+n=Conv3D ({Xt, Xt+1,..., Xt+n}). Network with Long-Short-Term Memory (LSTM) Alternatively, to capture long-term dependencies, interpret the feature vectors Ft′ as time series and run these sequences through an LSTM layer: ht=LSTM(Xt,ht−1), where ht∈Rh is the LSTM's hidden state at time ttt, and he is the LSTM's hidden unit count. The output is a sequence of hidden states: H= {h1,h2,...,hT}.

Step 4:
Completely Connected Layers for Categorization. The final hidden state or pooled feature vector should be sent through many fully connected layers after temporal features have been extracted using either 3DCNN or LSTM. ReLU activation for a fully linked layer: z(l+1) = ReLU(W(l)z(l)+b(l)) where ReLU is the rectified linear activation function, z(l) is the input to the layer, and W(l) and b(l) are the weights and biases of the layer: ReLU(x)=max(0,x) After every completely linked layer with a dropout rate, apply dropout regularization. p: z(l+1)=Dropout(z(l+1),p) Repeat this procedure for a number of completely linked layers, such as 1024 neurons per layer.

Step 5:
Binary Classification Output Layer for binary classification (crime vs. non-crime), use a final fully linked layer with a sigmoid activation function: y^=σ(W(out)z(L)+b(out)), where y^∈[0,1] is the projected probability of a frame falling into the crime class and σ(x)=1/1+e−x is the sigmoid function.

Step 6:
Optimization and Loss Function for binary classification, use binary cross-entropy as the loss function: Where y∈{0,1} is the true label for a frame (0 for non-crime, 1 for crime), and y^ is the anticipated probability, L=−(ylog (y^)+(1−y)log (1−y^)). Use an optimizer like Adam with learning rate η\etaη to optimize the model: θ=θ−η∇θL, where θ stands for the model parameters.

Step 7:
Instructions use the labeled dataset of video sequences to train the model. Minimizetheloss function by updating the model parameters over several epochs, using mini-batches of data. Optionally, apply early stopping if the validation loss stops improving.

Step 8:
Assessment After training, use metrics like accuracy, precision, recall, and F1 score to assess the model on a test set: (TP+TN)/(TP +TN+FP+FN) is the accuracy. Accuracy = TP/(TP+FP) Where TP, TN, FP, and FN stand for true positives, false positives, and false negatives, respectively, recall is equal to TP/(TP + FN).

Step 9:
Adjusting the Hyper parameter to maximize performance, adjust hyper parameters (such as learning rate η, dropout rate p, number of hidden units h, number of fully linked layers etc.).

1. **Video to Frame Conversion and Labeling Objective:**
   This step's objective is to turn the video into a collection of still images, or frames, that may be examined separately. The continuous motion is converted into discrete data points by segmenting the movie into frames.
   Operation: To extract individual frames, the video clip is processed, usually at a predetermined frames-per-second (FPS) rate. The movie is captured at a certain instant in time by each frame. Following extraction, each frame is annotated with pertinent information, such as whether it depicts "normal activity" (daily, non-violent behavior) or "critical public activity" (violent or possibly dangerous acts). Either automated tools or manual labor can be used for this tagging.

2. **Pre-processing of frames:**
   Pre-processing is essential to preparing the extracted frames for accurate and efficient analysis. It guarantees that every frame is high-quality and standardized, which enhances the efficiency of the feature extraction and classification phases that follow.

3. **Functioning:**
   This step entails a number of actions:
   Resizing: To guarantee a consistent input size for the neural network model, frames are downsized to a standard dimension (e.g., 224x224 pixels). Normalization: To preserve uniform lighting and color balance across frames, pixel values are normalized to a standard scale, usually between 0 and 1. Noise reduction: To eliminate extraneous noise and highlight significant aspects of the photos, methods such as Gaussian blur or median filtering may be used. Augmentation (optional): To improve the model's resilience and diversify the training data, random transformations like flipping, zooming, or rotation can be used.

4. **Classification Using Fully Connected (FC) Layers with ReLU and Dropout Goal:**
   The fully connected layers serve as a component for decision-making, classifying each frame based on the features that were retrieved. The model can learn intricate associations while avoiding over fitting thanks to the use of ReLU (Rectified Linear Unit) activation and dropout layers.
   First FCLayer with 1024 Units+ReLUActivation is operating: This layer starts integrating the feature maps from VGG-19. More intricate interactions can be captured by the model thanks to the non-linearity introduced by the ReLU activation function.

Layer of Dropout (0.1 Rate): 10% of the neurons are turned off at random during training, according to a dropout rate of 0.1. By keeping the model from becoming overly dependent on certain neurons, this lessens over fitting.

5. **Sigmoid activation for final classification:**
The last layer's job is to produce a categorization score that shows how likely it is that each frame falls into a particular category. Functioning: The final output layer makes use of the sigmoid activation function. It assigns a probability value between 0 and 1 to the final processed characteristics. This number is the probability that a certain frame displays "Normal Activity" or "Critical Public Activity." A standard cutoff point, such as 0.5, is used: A sigmoid output of greater than 0.5 indicates that the frame is displaying "Critical Public Activity." Frames that have an output of 0.5 or less are categorized as displaying "Normal Activity."

An algorithm efficiently breaks down video content into manageable chunks (frames), refines those chunks to highlight key details (pre-processing), learns from those details using a deep learning model (VGG-19), and then uses FC layers and sigmoid activation to make well-informed classifications. This enables the system to identify and classify scenes of interest in real-time, which makes it useful for applications like behavioral analysis, crowd control, and security surveillance. Dropout and regularization are used to make sure the model is reliable and capable of handling new, unseen data, which makes it appropriate for application in real-world scenarios. An algorithm for retrieving frames from video and labeling which are-

```
1.Get started
2. Set Up Video Input: V
3. Use Open(V) to open the video file.
4. Video Properties Extracted: T←Total Count of Frames FPS←Frames Per Second
5. S←Frame Step(Extraction Frequency) is the set Frame Parameter.
6. Frame Extraction Loop: Frame Counter Initializen ←0
7. While
8. Condition for Check Frame Step: IfnmodS≠0gotostep10(Increment Frame Counter)
Otherwise, proceed to step 9.
9. Save Frame and Capture: Save Frame(V,n)  IncreaseFrameCounter: n←n+1
11. Repeat Step 7 Loop
12. Close (V) is the Close Video File and Release Resources
```

CONCLUSION

The given algorithm describes a methodical procedure for removing particular frames at predefined intervals from a video file. First, the video file is opened and the video input V is initialized. After the file has been successfully opened, the algorithm goes ahead and extracts key video properties, such as the frames per second (FPS), which controls the playing speed, and the total frame count T, which indicates the total number of frames in the video. The algorithm determines a frame step SSS, which serves as a frequency parameter for frame extraction, after extracting these properties. In essence, SSS specifies the frequency of frame extraction; for example, if S=10, then every tenth frame will be recorded.

Setting a frame counter n to 0 is the first step in the frame extraction procedure. After entering a loop, the algorithm determines if the current frame counter (n) has surpassed the total number of frames (T). The loop ends when n reaches T since every frame has been processed. The algorithm assesses the condition n mod S otherwise. Whether the current frame number n is a multiple of the frame step S is ascertained by this modulus procedure. The procedure advances the frame counter n by 1 and moves on to the following loop iteration if the result is not zero, which indicates that n is not a multiple of S.

The approach uses the function Save Frame(V,n) to capture and save the frame at the then-th position if the condition n mod S=0 is true. This procedure guarantees that only Frames that satisfy the extraction requirements (as determined by step S) are stored. After saving the frame, the algorithm goes back to the start of the loop to process the subsequent frame, incrementing the frame counter n. The video file is closed and the resources used for the video processing are released once this loop is completed and all desired frames have been removed. After that, the algorithm stops. This technique is helpful for tasks like video summarizing or sampling since it efficiently extracts essential frames without processing each and every frame of the video by utilizing the modulus operation and frame step S.

Resizing the Frame

A crucial preprocessing step in getting datasets ready for deep learning models is frame scaling, especially when analyzing important public behavior. There are several factors that contribute to its-significance:

Efficiency of Memory

Input tensors of a specified size are frequently used for deep learning models. We guarantee consistency throughout the dataset by shrinking frames to a uniform dimension, which lowers the memory footprint and enables more effective memory use for both training and inference.

Efficiency of Computation

The neural network has to execute fewer operations on each frame when frames are resized to a smaller size. Faster forward and backward passes through the network as a result of this decrease in computing load speed up the training and inference procedures.

Compatibility of Models

Certain input size restrictions are frequently present in pre-trained deep learning models. Resizing frames to adhere to these guidelines enables these models to be applied seamlessly without requiring extra changes. Additionally, it guarantees interoperability with various deep learning systems and frameworks, each of which may have distinct requirements for input size.

Normalization

Normalization is commonly used in frame resizing to standardize pixel values within a specific range (e.g., [0, 1] or [-1, 1]). By stabilizing the input data, this phase improves the neural network's ability to identify significant characteristics.

Generalization

Frame resizing can improve a model's capacity for generalization by decreasing input sizes, hence lowering the possibility of overfitting. The model is encouraged to concentrate on robust traits that hold true regardless of size and aspect ratio changes when the input dimensions are smaller.

Augmenting Data

Normalization of Frames

In order to ensure that all aspects have equal weight in analysis and do not unduly affect conclusions because of their magnitude, normalization is the process of converting data features to a standard scale. It is crucial for many data analysis and machine learning activities, especially for comparing and evaluating frames or data points. The most popular method for normalization is feature scaling, which involves rescaling each feature to fall into a predetermined range. Standardization, min-max scaling, and statistical normalization with mean and standard deviation are some of the techniques. By reducing features to a range of 0 to 1, min-max scaling preserves the proportionality between values.
Because normalization enhances the performance and interpretability of analytical and machine learning methods, it is particularly important when working with features that have multiple units, scales, or distributions. To normalize frames with pixel values between 0 and 255, divide each pixel by 255, as seen in Normalized Frame=Resized Frame/255. The datasets' sample photos, after being resized and normalized.

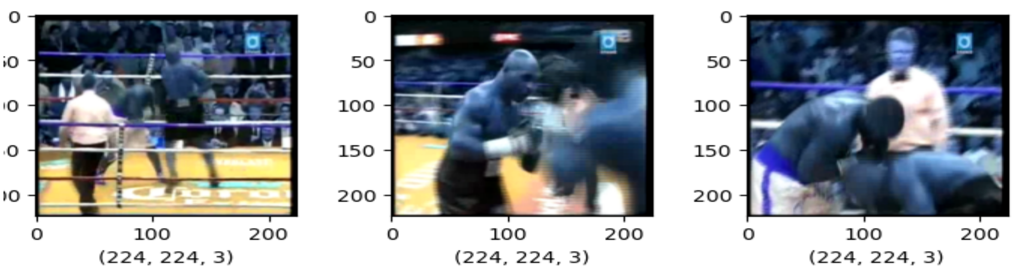
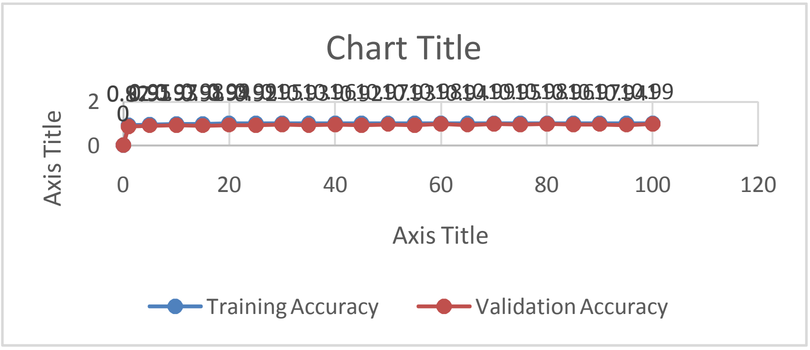


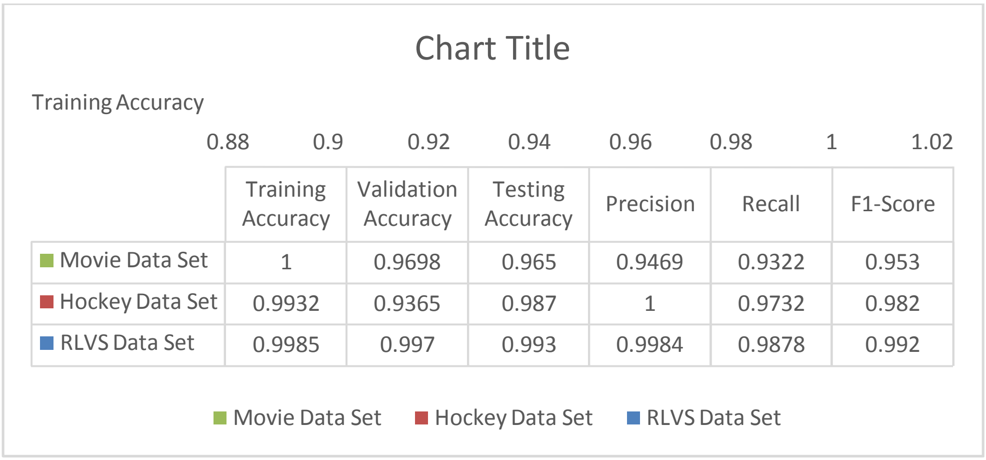
Fig.1: Sample images of datasets (a) Feature extraction in computer vision: a scholarly approach to movie fight (b) hockey fight A key technique in computer vision is feature extraction using convolution neural networks (CNNs), which allows important representations to be derived from images. These features that have been retrieved are used as inputs for tasks like object classification and picture classification.

Detection as well as picture retrieval. An academic summary of CNN-based feature extraction is provided below:
A specific subclass of deep neural networks called CNNs is made to handle grid-like data, most notably photographs. Convolution layers, pooling layers, fully connected layers, and activation functions (like ReLU) are some of the layers that make up these networks. CNNs are particularly good at using convolution filters and nonlinear activation functions to learn hierarchical representations of visual data.

The Feature Extraction Method
CNNs use the output of an intermediate layer, usually the final convolution layer, as feature vectors in feature extraction, which excludes fully connected layers (the classifier) from a pre-trained CNN model. Across several levels of abstraction, CNNs' convolution layers identify patterns and characteristics at various spatial scales, including edges, textures, and object components. High-dimensional feature vectors that capture the visual characteristics of input images are obtained by extracting features from these convolution layers.

Learning Transfer
By applying the information acquired from a source job—such as image classification using a sizable dataset like ImageNet—to a target goal—such as a new classification task with a smaller dataset—feature extraction using pre-trained CNNs is a type of transfer learning. Even with little labeled data, transfer learning enables us to leverage the generalization potential of previously trained CNN models and modify them for novel tasks. Without requiring rigorous training from scratch, a variety of computer vision problems can be successfully addressed by optimizing the pre-trained CNN model or using existing characteristics as inputs for a new classifier.





	Training Accuracy	Validation Accuracy	Testing Accuracy	Precision	Recall	F1-Score
🟩 Movie Data Set	1	0.9698	0.965	0.9469	0.9322	0.953
🟥 Hockey Data Set	0.9932	0.9365	0.987	1	0.9732	0.982
🟦 RLVS Data Set	0.9985	0.997	0.993	0.9984	0.9878	0.992

Fig.2: Training, validation, testing, precision, recall accuracy and F1-score of the proposed model.

Using a variety of parameters, including training accuracy, validation accuracy, testing accuracy, precision, recall, and F1-score, the table provides a thorough performance evaluation of a machine learning model applied to three datasets: RLVS, Hockey, and Movie. With training accuracy of 99.85%, validation accuracy of 99.7%, and testing accuracy of 99.3%, the RLVS Data Set exhibits nearly flawless performance. With an amazing F1-score of 0.992, which indicates an excellent balance between precision and recall, the model's precision of 99.84% and recall of 98.78% demonstrate that it accurately identifies genuine positives with very few false positives and false negatives. With training accuracy of 99.32%, validation accuracy of 93.65%, and testing accuracy of 98.7%, the Hockey Data Set likewise exhibits impressive results.

The model's high accuracy of 100% indicates that it produces perfect positive predictions, and its recall of 97.32% indicates that it successfully captures the majority of actual positives, even while a noticeable discrepancy between training and validation accuracies suggests overfitting. The F1-score of 0.982 emphasizes this balance even more, demonstrating the model's good reliability for this dataset. The model's precision of 94.69% and recall of 93.22% are marginally lower than those of the other datasets, suggesting a larger rate of false positives and some missing genuine positives. The Movie Data Set achieves a training accuracy of 100%, validation accuracy of 96.98%, and testing accuracy of 96.5%. The F1-score of 0.953, which is strong but still suggests space for improvement in balancing the detection of true and false positives, reflects this.

Final results

The research results unequivocally show the suggested model's improved performance and stability across a variety of datasets, underscoring its potential for real-world uses in crucial activity identification. The model demonstrates its effectiveness in managing real-life important scenarios with low errors, as evidenced by its near-perfect precision, recall, and F1-

scores on the RLVS dataset and its remarkable accuracy of 99.43% on the RLCS dataset. While the performance of the movie dataset points to areas that need improvement to better balance precision and recall, the findings from the hockey dataset further highlight the model's great generalization ability. The suggested method demonstrated its adaptability and efficacy by constantly outperforming or closely matching the competition across all datasets when compared to existing models. It stands out as a useful solution in crucial activity detection systems due to its notable generalization characteristics, which make it a dependable tool for a variety of video-based jobs. These findings demonstrate the model's applicability and make it a solid contender for further study and implementation in actual applications.

REFERENCES

1. Akhtar, M.S., Ekbal, A., Narayan,S.,Singh, V.,& Cambria, E. (2018). No, that never happened!! Investigating rumors on Twitter. IEEE Intelligent Systems, 33(5), 8-15.
2. Alharbi, A. S. M.,& de Doncker, E. (2019). Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. Cognitive Systems Research, 54, 50-61.
3. Alistair, K., & Diana, I. (2005). Sentiment classification of movie and productreviews using contextual valence shifters. In Proceedings of FINEXIN.
4. Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F.,& Manicardi, S. (2016). A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter. In KDWeb.
5. Anjaria, M.,& Guddeti, R. M. R. (2014). A novel sentiment analysis of social networks using supervised learning. Social Network Analysis and Mining, 4(1), 181.
6. Appel, O., Chiclana, F., Carter, J.,& Fujita, H. (2016). A hybrid approach to the sentiment analysis problem at the sentence-level. Knowledge-Based Systems, 108,110-124.
7. Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F.,& Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Systems with Applications, 77, 236-246.
8. [13] In May 2010, Sebastiani, F., Esuli, A., and Baccianella, S. An improved lexical resource for sentiment analysis and opinion mining is Sentiwordnet 3.0. Vol. 10, No. 2010, pages. 2200-2204, in Lrec.
9. Bakilwal, A., Foster, J., O'Brien, R., van der Puil, J., Tounsi, L., & Hughes, M. (2013, June). Analyzing political tweet sentiment: Moving toward a precise classification system. Computational Linguistics Association.
10. Evaluation and Interpretation of Product Review Sentiment Analysis through Enhanced Machine Learning Methods," International Journal on Recent and Innovative Trends in Computing and Communication, Vol. 11, Issue 10, Pages 2321-8169, Dr. Mukesh Kumar, Pawan Kumar