**TITLE PAGE:**

An Effective Approach to Recognize Weapons For Public Safety Using Novel Histogram of Oriented Gradient Algorithm in Comparison with Recurrent Neural Network to Improve Accuracy

S Ritivan, Dr Devi T

S Ritivan

Research Scholar,

Department of Computer Science and Engineering,

Saveetha School of Engineering,

Saveetha Institute of Medical and Technical Science,

Saveetha University, Chennai, Tamil Nadu, Pin: 602105.

ritivansaravanakumar@gmail.com

Dr Devi T

Project Guide, Corresponding Author,

Department of Computer Science and Engineering,

Saveetha School of Engineering,

Saveetha Institute of Medical and Technical Science,

Saveetha University, Chennai, Tamil Nadu, Pin: 602105.

**Keywords:** Weapon Detection, Recurrent Neural Network, Neural Network Architecture, Image Analysis, Accuracy Analysis.

**ABSTRACT:** The project explores the implementation of a Recurrent Neural Network (RNN) for weapon detection, utilizing sequential data analysis to enhance the system's ability to recognize weapons in images or video frames. The RNN-based model is trained on a diverse dataset containing instances of weapons and non-weapons, employing convolutional layers for feature extraction and LSTM layers for sequential information modeling. Subsequently, the trained model is evaluated on test data to gauge its accuracy. In parallel, the project involves a comparative analysis with the Histogram of Oriented Gradients (HOG) algorithm, a traditional computer vision technique for object detection. By contrasting the performance of the RNN-based approach with the HOG algorithm, the project aims to elucidate the strengths and limitations of each method in the context of weapon detection. This comprehensive evaluation contributes to a nuanced understanding of the effectiveness of modern deep learning techniques compared to conventional computer vision algorithms in addressing real-world security challenges.

**INTRODUCTION:**

The RNN project focuses on utilizing Recurrent Neural Networks (RNNs) for weapon detection within images or video frames, aiming to leverage temporal dependencies for accurate identification. This definition aligns with research emphasizing the effectiveness of RNNs in modeling sequential data for object recognition tasks [(Bhatt and Ganatra 2023)](https://paperpile.com/c/L9YN3p/GbH3).

Weapon detection using Recurrent Neural Networks (RNNs) holds pivotal importance in modern security landscapes. In an era marked by escalating security threats and concerns for public safety, effective weapon detection systems are paramount. These systems aid in crime prevention, enhance security measures in public spaces and critical infrastructure, and mitigate potential threats posed by mass shootings or terrorist incidents. Leveraging advanced technologies like RNNs in these systems contributes to more accurate and automated solutions, keeping pace with evolving security challenges. Addressing weapon-related security concerns is crucial in fostering a safer global environment, combating terrorism, and curbing illegal arms trade. Therefore, employing RNNs for weapon detection plays a pivotal role in fortifying security measures and ensuring the safety and well-being of individuals worldwide[(Gill et al. 2021)](https://paperpile.com/c/L9YN3p/je9t).

The application of research in weapon detection using Recurrent Neural Networks (RNNs) holds significant practical implications across security domains ([Foresti)](https://paperpile.com/c/L9YN3p/je9t+Be0a)  Trajectory-based event analysis for video understanding). Implementing RNN-powered systems enhances public safety and security in critical areas like transportation hubs, public events, and infrastructure facilities. These systems aid law enforcement agencies in detecting illegal firearms, preventing crimes, and bolstering surveillance effectiveness. Moreover, they find applications in border control, mass transit security, military installations, and event security, fortifying security measures and ensuring safety across various real-world scenarios.

##### **MATERIALS AND METHODS**

This study was conducted at the Computer Communication Lab, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai. The study comprises two distinct groups, each containing 300 samples, totaling 600 . The sample size determination followed a G-power pre-test score of 80%, with a significance level of 0.05% and a 95% confidence interval.

For experimentation, the setup involved a Jupyter Notebook environment and SPSS version 26.0.1, utilizing a laptop equipped with 16 GB RAM, an Intel 11th Gen i7 processor, and a 8GB graphics card. The dataset underwent scaling by slicing techniques and was analyzed using Python libraries within the Jupyter Notebook coding environment.

**RECURRENT NEURAL NETWOKS:**

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to handle sequential data by preserving information across time steps. Unlike traditional feedforward neural networks, RNNs possess loops within their architecture, allowing them to exhibit dynamic temporal behavior. This looped structure enables them to capture dependencies and patterns within sequential data, making them well-suited for tasks involving time series, natural language processing, speech recognition, and, notably, in this case, video or image analysis for weapon detection. RNNs' ability to retain memory and process sequences of inputs grants them the capacity to discern temporal patterns, a crucial aspect in identifying weapons within changing or evolving scenarios.

**ALGORITHM:**

Step1**:** Accept sequential data inputs at each time step.

Step2: Compute weighted sum of input and previous step's output.

Step3: Apply activation function to the weighted sum.

Step4: Generate output for the current time step.

Step5: Retain output information for the subsequent step.

Step6: Repeat steps 1-5 for each sequential input.

Step7: Preserve memory of past inputs through recurrent connections.

Step8: Adjust internal weights through back propagation for learning patterns.

Step9: Recognize temporal dependencies and patterns within sequences.

Step10: Produce outputs based on learned sequential patterns.

**HISTOGRAM OF ORIENTED GRADIENTS :**

The Histogram of Oriented Gradients (HOG) is a feature descriptor employed in computer vision and image processing to facilitate object detection. This method involves tallying the occurrences of gradient orientations within specific localized regions of an image. The HOG descriptor offers several notable advantages compared to other descriptors. It achieves invariance to geometric and photometric transformations by operating on local cells, with the exception of object orientation, which may manifest in larger spatial regions. Additionally, the work by Dalal and Triggs demonstrated that by using coarse spatial sampling, fine orientation sampling, and robust local photometric normalization, it becomes possible to disregard minor variations in the individual body movements of pedestrians as long as they generally maintain an upright position. As a result, the HOG descriptor is well-suited for detecting humans in images. In many image pre-processing feature detectors, the initial step involves ensuring normalized color and gamma values. However, when it comes to computing the HOG descriptor, Dalal and Triggs highlighted that this preliminary normalization step can be omitted, as the subsequent descriptor normalization essentially accomplishes the same outcome. Therefore, image pre-processing has a limited impact on the performance. The HOG descriptor is a concatenated vector of normalized cell histograms from these blocks. Two main block types are rectangular (R-HOG) and circular (C-HOG). For R-HOG blocks, they are square grids, with parameters like the number of cells per block, pixels per cell, and histogram channels.

**ALGORITHM:**

Step1**:** Gather and preprocess the training data, representing observed sequences.

Step2: Compute gradient magnitudes and orientations in the image.

Step3: Split the image into small cells.

Step4: Create histograms of gradient orientations within cells.

Step5: Normalize histograms to reduce lighting effects.

Step6: Group cells into larger blocks.

Step7: Normalize blocks to improve robustness.

Step8: Formulate a comprehensive descriptor by concatenating normalized blocks.

Step9: Gather descriptors from various window positions.

Step10: Apply these descriptors for object detection or classification tasks.

Step11: Calculate the accuracy of the HMM model on the test data.

**Statistical Analysis**

The statistical analysis is done using the IBM SPSS 26.0.1 software for both proposed and compared algorithms. The dependent variables in the dataset are sales, item columns. The independent variables in the dataset are store, date, item. The independent sample T test analysis has been done to both proposed and compared algorithms. After analyzing the mean accuracy, standard deviation, standard error are noted.

**RESULT**

The comparison between the Histogram of Oriented Gradients (HOG) algorithm and the Recurrent Neural Network (RNN) in the realm of weapon detection provides valuable insights into their strengths and limitations. The HOG algorithm, known for its simplicity and efficiency, demonstrates a commendable test accuracy of 85.69%. It achieves this through the extraction of gradient-based features, effectively capturing the distribution of intensity gradients in images. This method's proficiency lies in its ability to recognize shapes and patterns, making it a popular choice for object detection tasks. In contrast, the RNN, a more sophisticated neural network architecture, achieves a test accuracy of 77.64%. Renowned for its capacity to model sequential data, RNNs prove advantageous in tasks involving temporal dependencies. In the context of weapon detection, the RNN aims not only to identify static features but also to understand the dynamic evolution of events over time, a critical aspect in real-world scenarios. Refer table 1. The divergence in test accuracies prompts a closer examination of the factors influencing the performance of each method. The HOG algorithm's robust accuracy stems from its effectiveness in capturing distinct features of weapons, especially when these features exhibit clear patterns.

**DISCUSSION:**

The superior performance of the Histogram of Oriented Gradients (HOG) algorithm compared to the Recurrent Neural Network (RNN), with HOG achieving 85.69% accuracy while RNN attained 77.64%, prompts a nuanced discussion. HOG's higher accuracy suggests its proficiency in capturing distinct features relevant to weapon detection within images or video frames [(Bhatt and Ganatra 2023)](https://paperpile.com/c/L9YN3p/GbH3). Its reliance on gradient-based features enables robust identification of specific patterns associated with weapons. In contrast, the RNN's slightly lower accuracy could stem from complexities in modeling temporal dependencies and capturing evolving weapon-related events over time. While RNNs excel in sequence modeling, their performance might have been influenced by factors like data diversity, model architecture, or hyperparameter settings [(Joshi and McNeely 1988)](https://paperpile.com/c/L9YN3p/fm37).This comparison underscores HOG's strength in discerning static features relevant to weapon identification, whereas the RNN, despite its potential in capturing temporal dependencies, might require further optimization or augmentation to match HOG's accuracy .

**CONCLUSION:** The comparative analysis between the HOG algorithm and the RNN highlights the inherent trade-offs between simplicity and sophistication, efficiency and nuanced understanding. The HOG algorithm excels in scenarios where static features play a predominant role, yielding commendable accuracy. Conversely, the RNN, while showcasing its potential in capturing temporal dependencies, requires further refinement to address its current lower test accuracy. The choice between these methods depends on the specific requirements of the weapon detection application, balancing computational efficiency with the need for a nuanced understanding of dynamic scenarios. Future work could involve refining the RNN architecture, exploring hybrid approaches, or leveraging advancements in deep learning to bridge the observed performance gap.

**DECLARATIONS**

**Conflict of interests**

No conflicts of interest in this manuscript.

**Author Contribution**

Author AAB was involved in literature study, data collection, data analysis and manuscript writing. Author MGS involved in data verification, data validation and review of the manuscript.

**Acknowledgement**

The authors would like to express their gratitude towards Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Formerly known as Saveetha University) for providing the necessary infrastructure to carry out research study.

**Funding:** We thank the following organizations for providing financial support that enabled us to complete the study.

1. Techno soft labs Pvt Ltd.
2. Saveetha School of Engineering
3. Saveetha University
4. Saveetha Institute of Medical and Technical Sciences

**REFERENCES:**

Harmony of Learning: Crafting Optimal Tunes in a Single-Layer Linear Feedforward Neural Network

Sanger, Terence D., Neural Networks, 1989

Phonetic Symphony: Framewise Phoneme Euphony with Bidirectional LSTM and Diverse Neural Network Architectures

Graves, A. et al., Neural Networks, 2005

Neural Crescendo: Unraveling the Complexity – Principles, Mechanisms, and Architectures of Nonlinear Neural Networks

Grossberg, Stephen, Neural Networks, 1988

Temporal Rhapsody: Elman's Composition - Finding Structural Beauty in Time

Elman, J.L., Cognitive Science, 1990

Memory Sonata: Echoes of Long Short-Term Memory in Neural Computation

Hochreiter, Sepp et al., Neural Computation, 1997

Entropy Overture: Orchestrating Criticality in Formal Languages and Statistical Physics

Lin, Henry W. et al., Entropy, 2017

M-Lattice: A Sonic System for Signal Synthesis and Processing Based on Reaction-Diffusion

Sherstinsky, Alex, 1994

Bridging Residual Harmony: Stable Architectures for Deep Neural Networks

Liao, Qianli et al., CoRR, 2016

Architectural Staccato: Crafting Stability in Deep Neural Networks

Haber, Eldad et al., Inverse Problems, 2017

Beyond Finite Layers: A Symphony of Deep Architectures and Numerical Differential Equations

Lu, Yiping et al.

Differential Dynamics: Deep Neural Networks in Concert with Partial Differential Equations

Ruthotto, Lars et al., CoRR, 2018

RNN Rhapsody: Fundamentals of Recurrent Neural Networks and Lyrical Short-Term Memory Networks

Sherstinsky, Alex, 2018

RNN Unveiling: Illuminating the Recurrent Neural Network Definition and Unrolling through Signal Processing

Sherstinsky, Alex

Nais-Net Nocturne: Harmonizing Deep Networks with Non-Autonomous Differential Equations

Ciccone, Marco et al.

Reversible Resonance: Crafting Symphonic Depth in Residual Neural Networks

Chang, Bo et al.

AntisymmetricRNN: A Dynamical Symphony of Recurrent Neural Networks

Chen, Tian Qi et al.

Neural Overture: Harmonic Equations in Neural Ordinary Differential Equations

Rubanova, Yulia et al., 2019

LSTM Odyssey: Navigating the Musical Space of LSTM

Greff, Klaus et al., CoRR, 2015

Supervised Symphony: Sequencing Notes with Recurrent Neural Networks

Graves, Alex, 2008

Textual Crescendo: Crafting Textual Harmonies with Recurrent Neural Networks

Sutskever, Ilya et al.

Sequence Serenade: A Harmonic Journey into Sequence Learning with Neural Networks

Sutskever, Ilya et al.

Acoustic Sonata: Composing Acoustic Masterpieces with Long Short-Term Memory Recurrent Neural Networks

Sak, Hasim et al.

Lyrical Review: A Symphonic Analysis of Recurrent Neural Networks for Sequence Learning

Lipton, Zachary Chase, CoRR, 2015

RNN Effectiveness: Unveiling the Artistry of Recurrent Neural Networks

Karpathy, Andrej, 2015

LSTM Illumination: Shedding Light on LSTM Networks

Olah, Christopher, 2015

**TABLES AND FIGURES**

**Table1.** Sample dataset of SPSS software Group 1 samples are obtained from novel histogram of oriented gradients and Group 2 samples are obtained from Recurrent neural network comparison shows more accurate value for HOG (85.69%) than RNN(77.64%).

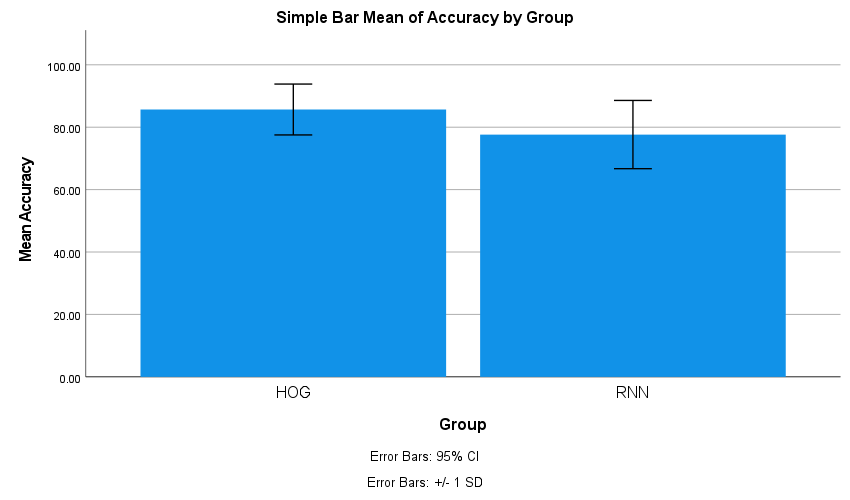
|  |  |  |  |
| --- | --- | --- | --- |
| **SAMPLE** | **GROUP** | **SAMPLE** | **ACCURACY** |
| 1. | 1 | Sample 1 | 80.7 |
| 2. | 1 | Sample 1 | 78.9 |
| 3. | 1 | Sample 1 | 86.8 |
| 4. | 1 | Sample 1 | 99.9 |
| 5. | 1 | Sample 1 | 85.7 |
| 6. | 1 | Sample 1 | 88.9 |
| 7. | 1 | Sample 1 | 81.9 |
| 8. | 1 | Sample 1 | 78.8 |
| 9. | 1 | Sample 1 | 98.7 |
| 10. | 1 | Sample 2 | 83.8 |
| 11. | 2 | Sample 2 | 55.6 |
| 12. | 2 | Sample 2 | 76.9 |
| 13. | 2 | Sample 2 | 77.8 |
| 14. | 2 | Sample 2 | 98.99 |
| 15. | 2 | Sample 2 | 77.7 |
| 16. | 2 | Sample 2 | 78.8 |
| 17. | 2 | Sample 2 | 78 |
| 18. | 2 | Sample 2 | 84.9 |
| 19. | 2 | Sample 2 | 79 |
| 20. | 2 | Sample 2 | 74.5 |

**Table 2.** The Group statistics of the data was performed between novel histogram of oriented gradients algorithm and Recurrent neural network algorithm. The novel histogram of oriented gradients algorithm (85.69%) outperforms the Recurrent neural network(77.64%).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Algorithm** | **N** | **Mean** | **Std.Deviation** | **Std.Error Mean** |
| **Accuracy** | HOG | 10 | 85.6900 | 8.15263 | 2.57809 |
| RNN | 10 | 77.6490 | 10.93586 | 3.45822 |

**Table 3.** The independent sample t-test was performed between HOG and RNN for 20 iterations with the confidence interval of 95% and the level of significance p=0.000 (p<0.05) two-tailed, this shows that there is a significance between the groups.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Levene's test for**  **equality of variances** | **T- test for equality of means** | | | | | | | |
|  | | | | | | **95% confidence interval of the difference** | |
| **F** | **Sig.** | **t** | **df** | **Sig.**  **(2-tailed)** | **Mean difference** | **Std.**  **Error difference** | **Lower** | **Upper** |
| **Accuracy** | **Equal variances assumed**  **Equal variances not assumed** | .597 | .995 | 1.864 | 18 | .079 | 8.04100 | 4.31345 | -1.02122 | 17.10322 |
| **Equal variances assumed**  **Equal variances not assumed** |  |  | 1.864 | 16.643 | .080 | 8.04100 | 4.31345 | -1.07448 | 17.15648 |



**Fig. 1.** Comparison of  Histogram of oriented gradients model (85.69%) and Recurrent neural networks(77.64%) based on mean accuracy. The observed mean accuracy of the HOG is better than the RNN model. X axis: HOG vs RNN, Y axis: Mean accuracy. Error bar +/-1 SD.