Developing A Smart Integrated Violence Detection System In Video Datasets By Leveraging The RESNET-50 Architecture[1]

Janhvi Garg, Mrs. Surinder Kaur
Department of Information Technology
Bharati Vidyapeeth's College of Engineering, New Delhi, India

Received: 15 March 2023; Accepted: 27 April 2023; Published: 06 May 2023

ABSTRACT

Although rarely utilized to prevent crime or respond to it in real-time, CCTV is a significant piece of evidence in court. Despite having accuracy rates that are close to 85%, present approaches lack precision and memory requirements. For the majority of security systems, automatic violence detection is still an unresolved problem.

Due to the increase in violent crime, thorough surveillance systems are now required to spot potentially dangerous situations and deal with violent encounters. Modern methods for detecting violence in surveillance recordings include the extraction of auditory features, spatiotemporal analysis, or the construction of optical flow motion vectors. The feature extraction techniques and object detection techniques are implemented in a dataset for detection of violent activities in the proposed dataset.

Keywords: *Convolutional Neural Network; Violence Detection; Residual Net50.*

INTRODUCTION

In order to maintain high security levels, our society significantly relies on the use of CCTV cameras. As we typically use CCTV evidence just hours or even days after the act has already occurred, such a strategy is, however, very contentious. Although rarely utilized to stop crime or respond to it immediately, it offers crucial evidence in court. Due to the fact that just a small group of security personnel are primarily responsible for monitoring vast amounts of CCTV footage, there is such inefficiency.

Businesses, the government, and law enforcement agencies were driven to utilize thorough surveillance systems to identify unsafe surroundings and successfully handle violent confrontations due to the development of illegal activities, their unpredictability, and the extent of destruction they caused. Still,

Modern methods for detecting violence in surveillance recordings include the extraction of auditory features, spatiotemporal analysis, and the construction of optical flow motion vectors. Despite the encouraging results with near 85% accuracy rates, current methods are still imprecise, memory-intensive, and expensive computationally, which prevents them from being used for practical purposes, especially surveillance, where high accuracy must be accompanied by timely results.

[1] *How to cite the article:* Garg J., Kaur S., (May, 2023); Developing A Smart Integrated Violence Detection System In Video Datasets By Leveraging The RESNET-50 Architecture, *International Journal of Advances in Engineering Research*, May 2023, Vol 25, Issue 5, 1-9

RELATED WORK

S.NO.	PAPER TITLE	YEAR	TOOL USED	DATASET USED	DESCRIPTION
1	Detection of violent behavior using neural networks and pose estimation	2022	3D-CNN	Kranok-NV.	Using 10-fold cross validation, 98% accuracy is obtained on the Kranok NV dataset for classification.
2	Low-cost CNN for automatic violence recognition on embedded systems	2022	MOBILE CNN	Violent-Flow1, UCF-1012, HMDB3	A prototype of an intelligent monitoring system was developed on a Raspberry Pi embedded platform, able to run a mobile CNN model and accuracy of 92.05% was obtained.
3	Motion-shape-based deep learning approach for divergence behaviour detection in high-density crowd	2021	CNN	UCF, UMN, NGSIM, PETS2009	Cnn is trained on motion shape images and is used to detect crowd divergence behaviour.
4	Deepanomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes	2018	FCN	UCSD, subway	This paper presents a new FCN architecture for generating and describing abnormal regions for videos. Furthermore, the proposed FCN is a combination of a pre-trained CNN (an AlexNet version) and a new convolutional layer where kernels are trained with respect to the chosen training video.
5	Learning to detect anomaly occurrences in crowd scenes from synthetic data	2021	3D-CNN	SHADE	3D CNN is designed to detect abnormalities in the video dataset and further C3DGAN is used for domain adaption to reduce domain gap.

Table 1: Reference paper and dataset set used

DATASET

This study created a complicated violence dataset with several films by combining two datasets, Real-Life Violence Situations (RLVS) and RWF-2000. (4000 videos). 2000 samples of these videos were violent, while 2000 samples were not violent.

RWF DATASET

RWF-2000 is an open massive video library that contains 2,000 videos that were recorded by various security cameras in actual circumstances. There are 1000 violent films in it, along with 1000 nonviolent ones. One of the key features of RWF-2000 is its diversity, as the videos were recorded under different lighting conditions, camera angles, and environmental settings. This can help improve the generalizability of models trained on the dataset, as they will have been exposed to a wide range of scenarios and variations.

RLVS DATASET

There are 2000 films in Real Life Violence Situations (RLVS) Dataset 1000 of which are violent and 1000 non-violent. It includes YouTube recordings street fights with people of different racial backgrounds, ages, and genders in various settings. Additionally, it has non-violence films depicting various human activities, like games, eating, exercise etc.

Datasets	**Usage**	**Number of Samples**
RWF-2000 + Real life violence Situations	Training , Validation and Testing	2400(Training),800 (Validation),800(testing)

EXPERIMENTAL SETUP

Before being given to our model, the videos in the training dataset were transformed into a series of pictures. The movies in the test dataset were then categorised using the trained model for the purpose of detecting violence.

A. DATA PREPROCESSING

The videos from the combined RWF-2000 and RLVS dataset are converted into frames.

On an average each video contained 150 frames. As the dataset contained videos from the internet, they were limited to 30 frames per second. From each video, 10 frames were taken in a periodic interval which were then resized to 112 × 112. In the CNN model BGR was changed to RGB.

ResNet-50 requires a special type of preprocessing in which the RGB needs to be changed to BGR. Each of the three channels was normalized, which will zero-center each color channel with respect to the ImageNet dataset.

Extracting Frames:

```
def create_data(input_dir):
 X = []
 Y = []

 classes_list = os.listdir(input_dir)
 for c in classes_list:
     print(c)
     files_list = os.listdir(os.path.join(input_dir, c))
     for f in tqdm(files_list):
         frames = frames_extraction(os.path.join(os.path.join(input_dir, c), f))
         if len(frames) == seq_len:
             X.append(frames)
             y = [0]*len(classes)
             y[classes.index(c)] = 1
             Y.append(y)
 print("Frames per second:",fps)
 print("Average number of frames in a video: ",frame_count/4000)
 X = np.asarray(X)
 Y = np.asarray(Y)
 return X, Y

X, Y = create_data(data_directory)
```

B. DATA AUGMENTATION

Data augmentation is a crucial technique in deep learning that helps increase the variability and diversity of training data. By applying various transformations to the input data, data augmentation aims to prevent the model from overfitting and enables it to learn more generalized and robust patterns. In the context of the described scenario, the following data augmentation techniques were applied:

Rescaling pixel intensities: The pixel intensities of each frame were rescaled by a factor of 1/255. This rescaling operation ensures that the pixel values are within the range of 0 to 1, which is a common practice in deep learning models. Rescaling helps normalize the data and ensures consistent input ranges across different samples.

Horizontal and vertical flipping: The rescaled images in the dataset were flipped both horizontally and vertically. This transformation creates new samples by reflecting the original images along the horizontal and vertical axes. By introducing these flipped versions, the model learns to recognize patterns and objects from different orientations, enhancing its ability to generalize and perform well on unseen data.

Random rotation: The rescaled images were randomly rotated during data augmentation. This rotation introduces variations in the orientation of objects, allowing the model to learn to detect and classify objects regardless of their rotation angle. By including randomly rotated images, the model becomes more robust to changes in object orientation and improves its performance under different viewing angles.

Random magnification: The images in the dataset were randomly magnified as part of the data augmentation process. This augmentation technique involves scaling the images by different factors, either increasing or decreasing their size. By applying random magnification, the model learns to handle variations in object size and scale, making it more adaptable to objects of different sizes within the input data.

C. TRAINING

The neural network train-validate-test process is a technique used to reduce model overfitting. The technique is also called early stopping. Although the train-validate-test isn't conceptually difficult, the process is a bit difficult to explain because there are several interrelated ideas involved.

Training a neural network is the process of finding the values for the weights and biases. In most scenarios, training is accomplished using what can be described as a train-test technique. The available data, which has known input and output values, is split into a training set (typically 80 percent of the data) and a test set (the remaining 20 percent).

The training data set is used to train the neural network. Various values of the weights and biases are checked to find the set of values so that the computed output values most closely match the known, correct output values. Or, put slightly differently, training is the process of finding values for the weights and biases so that error is minimized. There are many training algorithms, notably back-propagation, and particle swarm optimization.

During training, the test data is not used at all. After training completes, the accuracy of the resulting neural network model's weights and biases are applied just once to the test data. The accuracy of the model on the test data gives you a very rough estimate of how accurate the model will be when presented with new, previously unseen data.

One of the major challenges when working with neural networks is a phenomenon called overfitting. Model overfitting occurs when the training algorithm runs too long. The result is a set of values for the weights and biases that generate outputs that almost perfectly match the training data, but when those weights and bias values are used to make predictions on new data, the model has very poor accuracy.

The train-validate-test process is designed to help identify when model overfitting starts to occur, so that training can be stopped. Instead of splitting the available data into two sets, train and test, the data is split into three sets: a training set (typically 60 percent of the data), a validation set (20 percent) and a test set (20 percent).

The pre-processed and augmented frames are then fed through all the layers of the ResNet-50 architecture, known for its state-of-the-art performance. This architecture utilizes residual connections to alleviate the vanishing gradient problem and facilitate training of deep networks.

Following the ResNet-50 layers, batch normalization is applied to the obtained feature maps. This normalization technique helps in stabilizing and accelerating the training process by reducing internal covariate shift.

To reduce the number of parameters and expedite computation, average pooling is performed on the feature maps obtained from the ResNet-50 layers. Average pooling downsamples the spatial dimensions of the feature maps while retaining important information, thus enhancing computational efficiency.

In parallel, a separate CNN model with 8 layers is utilized. This CNN model consists of consecutive convolutional layers, with the number of filters progressively increasing from 64 to 128, 256, and 512, respectively. ReLU activation functions are applied after each convolutional layer to introduce non-linearity.

After every two convolutional layers in the CNN model, batch normalization and pooling are applied. These techniques help to mitigate overfitting by regularizing the network and reducing spatial dimensions, respectively.

Finally, to process the sequence of feature maps from both the CNN model and the ResNet-50 model in a chronological manner, an LSTM layer is employed. Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that effectively filters out relevant information from sequential input data, allowing the model to capture temporal dependencies and make predictions based on the context of previous feature maps.

Batch normalization is a popular technique used in deep learning models, primarily in convolutional neural networks (CNNs), to improve training stability and accelerate convergence. It operates by normalizing the activations of each layer across a mini-batch of training examples. Here's an explanation of batch normalization and its benefits:

During the training process, the input to a layer in a neural network can vary in scale and distribution, which can make training difficult. Batch normalization addresses this issue by ensuring that the mean activation of each feature map is close to zero and has a unit variance. This is achieved by normalizing the activations using the mean and variance computed over the mini-batch.

The Adam optimizer is an optimization algorithm commonly used in training deep learning models. It combines the concepts of adaptive learning rates and momentum to efficiently update the model's parameters during the training process.

The following is the description of the parameters given above:

learning_rate: The learning rate to use in the algorithm. It defaults to a value of 0.001.
beta_1: The value for the exponential decay rate for the 1st-moment estimates. It has a default value of 0.9.
beta_2: The value for the exponential decay rate for the 1st-moment estimates. It has a default value of 0.999.
epsilon: A small constant for numerical stability. It defaults to 1e-7.
amsgrad: It is a boolean that specifies whether to apply the AMSGrad variant of this algorithm from the paper "On the Convergence of Adam and beyond". It has a default value of False.
name: The Optional name for the operations created when applying gradients. Defaults to "Adam".

Batch size - We took batch size of 8.
refers to the number of training examples utilized in one iteration. The batch size can be one of three options:
batch mode: where the batch size is equal to the total dataset thus making the iteration and epoch values equivalent
mini-batch mode: where the batch size is greater than one but less than the total dataset size. Usually, a number that can be divided into the total dataset size.
stochastic mode: where the batch size is equal to one. Therefore the gradient and the neural network parameters are updated after each sample

Learning rate - Learning rate used in the project is 10^-4.
The rate of learning or speed at which the model learns is controlled by the hyperparameter
Smaller learning rates necessitate more training epochs because of the fewer changes. On the other hand, larger learning rates result in faster changes.

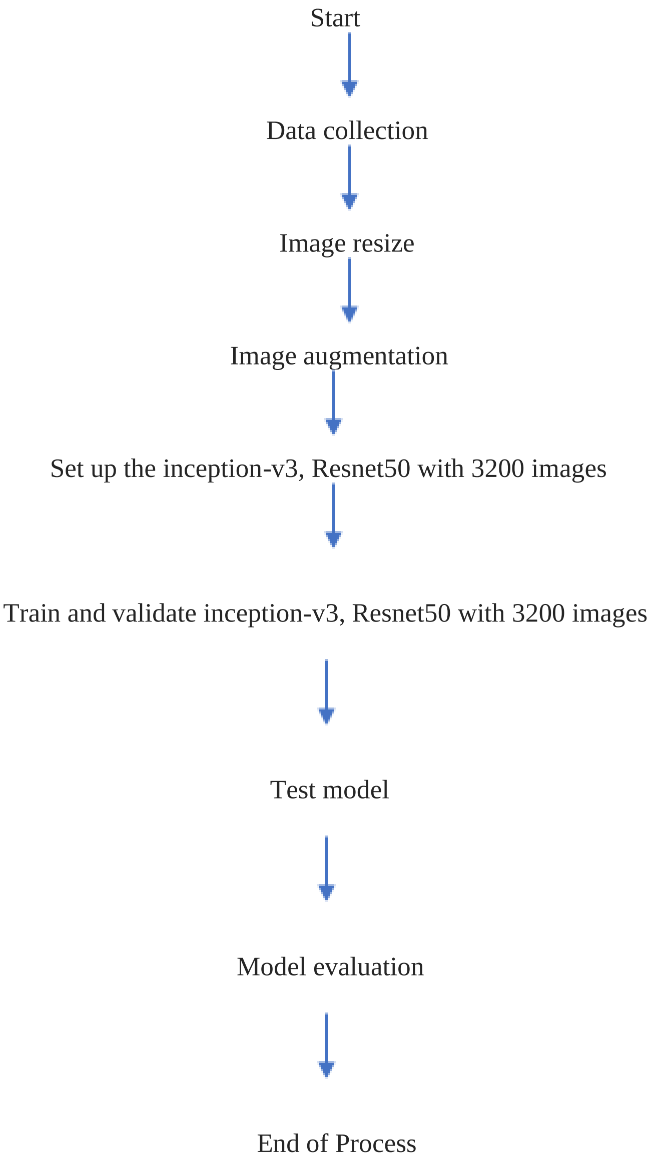
Epoch- Indicates the number of passes of the entire training dataset the machine learning algorithm has completed. Datasets are usually grouped into batches (especially when the amount of data is very large)

Confusion Matrix- This matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known.

ROC or Receiver Operating Characteristic curve represents a probability graph to show the performance of a classification model at different threshold levels. The curve is plotted between two parameters, which are:
True Positive Rate or TPR
False Positive Rate or FPR

AUC(area under curve) computes the performance of the binary classifier across different thresholds and provides an aggregate measure. The value of AUC ranges from 0 to 1, which means an excellent model will have AUC near 1, and hence it will show a good measure of Separability.

FLOW CHART


Start
Data collection
Image resize
Image augmentation
Set up the inception-v3, Resnet50 with 3200 images
Train and validate inception-v3, Resnet50 with 3200 images
Test model
Model evaluation
End of Process

RESULTS

The performance of the model was verified using a total of six performance measures. Recall, Accuracy, Precision, F1 score, FPR, and FNR. Below is a brief explanation of each of the phrases mentioned above:

a) Accuracy -
The proportion of correctly labeled subjects to the entire group of subjects is the measure of accuracy.

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

b) Precision -
The proportion of accurately labeled "Violent Videos" to all videos that have been given that classification..

$$Precision\ = \frac{TP}{TP+FP}$$

c) Recall -
The proportion of "Violent Videos" that are genuinely violent to those that are appropriately labeled as such.

$$Recall\ = \frac{TP}{TP+FN}$$

d) F1- score the harmonic mean of Precision and Recall.

$$F1\ score\ = \frac{2*(Recall*Precision)}{(Recall\ +\ Precision)}$$

e) False Positive Rate -
The ratio of videos incorrectly labeled as violent to all "Non-Violent Videos" is known as the false positive rate.

$$FPR\ = \frac{FP}{TN+FP}$$

f) False Negative Rate -
The ratio of the number of videos incorrectly labeled as "non-violent" to the real number of "Violent Videos" is known as the "False Negative Rate."

$$FNR = \frac{FN}{TP+FN}$$

PERFORMANCE OF RESNET 50 MODEL

	Accuracy	F1-Score	Recall	Precision	FPR	FNR
Fold 1	0.83125	0.827145	0.8075	0.847769	0.145	0.1925
Fold 2	0.81625	0.818294	0.8275	0.809291	0.195	0.1725
Fold 3	0.83875	0.835249	0.8175	0.853786	0.14	0.1825
Fold 4	0.8225	0.824691	0.835	0.814634	0.19	0.165
Fold 5	0.85375	0.856089	0.87	0.842615	0.1625	0.13
Average	0.8325	0.832294	0.8315	0.833619	0.1665	0.1685
Standard Deviation	0.0130862	0.0130784	0.0213659	0.0181115	0.022561	0.0213659

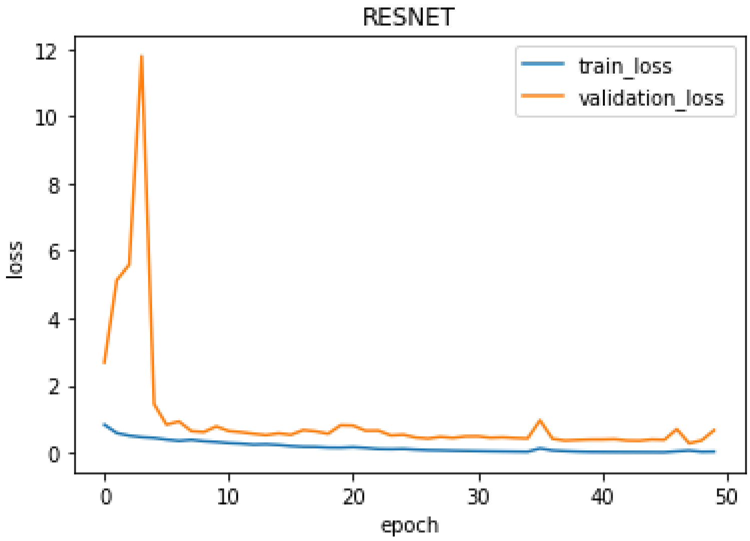


Figure 5: Loss of Training and Validation

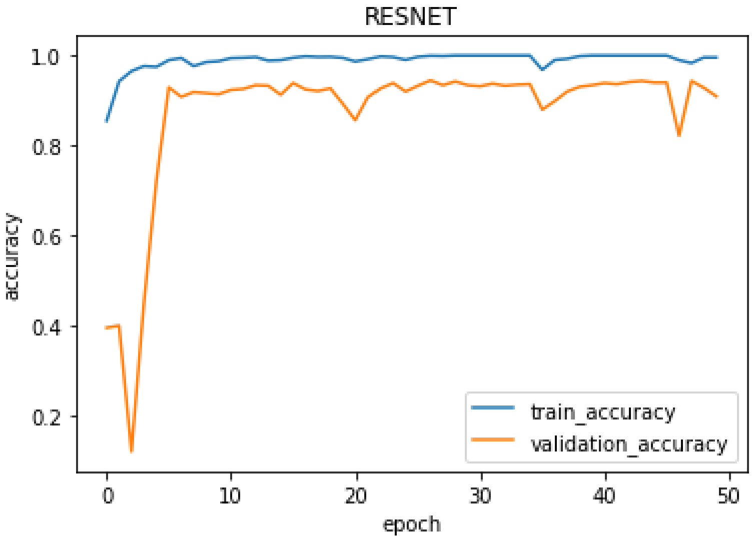


Figure 6: Accuracy of Training and Validation

CONCLUSION

In conclusion, violence detection using the ResNet-50 architecture has proven to be an effective approach. By leveraging the powerful capabilities of deep learning and the ResNet-50 model, it is possible to detect and classify violent content in images or video frames with a high degree of accuracy.

ResNet-50, with its deep layers and residual connections, allows for the extraction of complex visual features and enables the model to learn intricate patterns associated with violence. The architecture's ability to capture both low-level and high-level features contributes to its success in violence detection tasks

FUTURE SCOPE

This work can be extended in future to incorporate other techniques based on machine learning and deep learning.

REFERENCES

[1]"Violence Detection for Smart Surveillance Systems," Abto Software, online at: https://www.abtosoftware.com/blog/violence-detection, reached on May 26, 2021

[2]RWF-2000: An Open Large Scale Video Database for Violence Detection, M. Cheng, K. Cai, and M. Li, 2021, doi: 10.1109/icpr48806.2021.9412502.

[3]Violence Detection and Localization in Surveillance Video, D. G. C. Roman and G. C. Chavez, 2020, doi: 10.1109/SIBGRAPI51738.2020.00041.

[4]https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/

[5]Real-time violence detection for football stadium using big data analysis and deep learning via bidirectional LSTM, S. R. Dinesh Jackson et al., Comput. Networks, vol. 151, pp. 191-200, 2019.

[6]U.M. Butt, S. Letchmunan, F. H. Hassan, S. Zia and A. Baqir, "Detecting video surveillance using VGG19 convolutional neural networks", Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 2, pp. 674-682, 2020.

[7]Learning to detect anomaly occurrences in crowd scenes from synthetic data, W. Lin, Gao, Q. Wang, and X. Li. May 2021, Neurocomputing, volume 436, pages 248–259.

[8]M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, ‘‘Deepanomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes,’’ Comput. Vis. Image Understand., vol. 172, pp. 88–97, Jul. 2018

[9]Hu J, Liao X, Wang W, Qin Z (2022) Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network. In: IEEE Transactions on Circuits and Systems for Video Technology

[10]Jalal A, Mahmood M, Hasan AS (2019) Multi-features descriptors for human activity tracking and recognition in Indoor-outdoor environments. In: 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)

[11]M. U. Farooq, M. N. M. Saad, and S. D. Khan, ‘‘Motion-shape-based deep learning approach for divergence behaviour detection in high-density crowd,’’ Vis. Comput., p. 25, Feb. 2021

[12]Ajani OS, El-Husseiny H (2019) An ANFIS-based Human Activity Recognition using IMU sensor Fusion. In: 2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES)

[13]Afza F, Khan MA, Sharif M, Kadry S, Manogaran G, Saba T et al (2021) A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection

[14]X. Hu, J. Dai, Y. Huang, H. Yang, L. Zhang, W. Chen, G. Yang, and D. Zhang, ‘‘A weakly supervised framework for abnormal behaviour detection and localization in crowded scenes,’’ Neurocomputing, vol. 383, pp. 270–281, Mar. 2020.

[15]J. C. Vieira, A. Sartori, S. F. Stefenon, F. L. Perez, G. S. De Jesus, and V. R. Q. Leithardt, ‘‘Low-cost CNN for automatic violence recognition on embedded systems,’’ IEEE Access, vol. 10, pp. 25190–25202, 2022.

[16]K. B. Kwan-Loo, J. C. Ortiz-Bayliss, S. E. Conant-Pablos, H. Terashima-Marin, and P. Rad, "Detection of violent behavior using neural networks and pose estimation," IEEE Access, vol. 10, pp. 86339-86352, 2022.