Automatic Fight Detection In Surveillance Videos

By : Shady Abdelaziz El Ghareb Sakr

Supervised by :
Dr. Heba Ali Elkhobby
(Assistant Professor)

Electronics and Communication Engineering Department at Tanta University

ABSTRACT : Violence rates however have been brought down about 57% during the span of the past 4 decades yet it doesn't change the way that the demonstration of violence actually happens, unseen by the law. Violence can be mass controlled sometimes by higher authorities, however to hold everything in line one must "Micro Govern" over each movement occurring in every road of each square. To address the butterfly effects impact in our setting, We present an efficient method for detecting fight scenes in videos. Recent applications of convolution neural networks have shown promises of convolution layers for object detection and recognition, especially in images. However, convolution neural networks are supervised and require labels as learning signals.The proposed method is a deep learning based automatic detection approach that uses Convolutional Neural Network to detect violence present in a video. But, the disadvantage of using just CNN is that it requires a lot of time for computation and is less accurate. Hence, a pre-trained model, MobileNet, which provides higher accuracy and acts as a starting point for the building of the entire model. We propose an architecture for fight detection in videos including crowded scenes. Our project helps detect the fight scenes in videos with high accuracy, thus saving time for organizations and individuals who would have to go through the entire footage instead.

Keywords

activity recognition, deep learning, MobileNetV2, supervised, surveillance, violence detection.

1. Introduction

Violent behavior in public places is an issue that has to be addressed. Communities are also eroded by violence, which reduces productivity,lowers property values, and disrupts social services. Across the world, violence is a severe public health issue. It affects people at various phases of life, from infants to the elderly.

Recognizing violence is challenging since it must be done on real-time videos captured by a large number of surveillance cameras at any time and in any location. It should be able to make reliable real-time detection and alert corresponding authorities as soon as violent activities occur.

Public video surveillance systems are widespread around the world and can provide accurate and complete information in many security applications. However, having to watch videos for hours reduces your ability to make quick decisions. Video surveillance is essential to prevent crime and violence. In this regard, several studies have been published on the automatic detection of scenes of violence in video. This is so that authorities do not have to watch videos for hours to identify events that only last a few seconds. Recent studies have highlighted the accuracy of deep learning approaches to violence detection.

In this work we will be discussing the implementation of an Automatic Fight Detection in Surveillance Videos system using MobileNetv2.

2. Technical Background

2.1 Computer Vision :

Computer vision is the process of using machines to understand and analysing imagery (both photos and videos).
Computer vision is the broad parent name for any computations involving visual content – that means images, videos, icons, and anything else with pixels involved. But within this parent idea, there are a few specific tasks that are core building blocks:

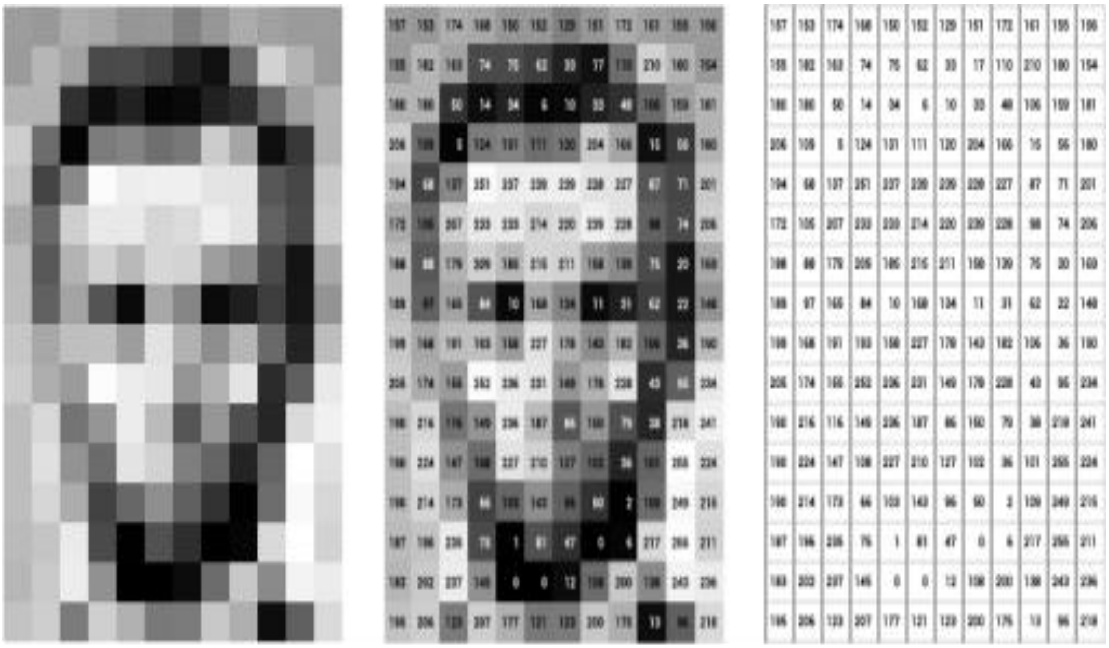
- In object classification, you train a model on a dataset of specific objects, and the model classifies new objects as belonging to one or more of your training categories.
- For object identification, your model will recognize a specific instance of an object

Outside of just recognition, other methods of analysis include:

- Video motion analysis uses computer vision to estimate the velocity of objects in a video, or the camera itself.
- Scene reconstruction creates a 3D model of a scene input through images or video.
- In image restoration, noise such as blurring is removed from photos using Machine Learning based filters.
- Any other application that involves understanding pixels through software can safely be labeled as computer vision.

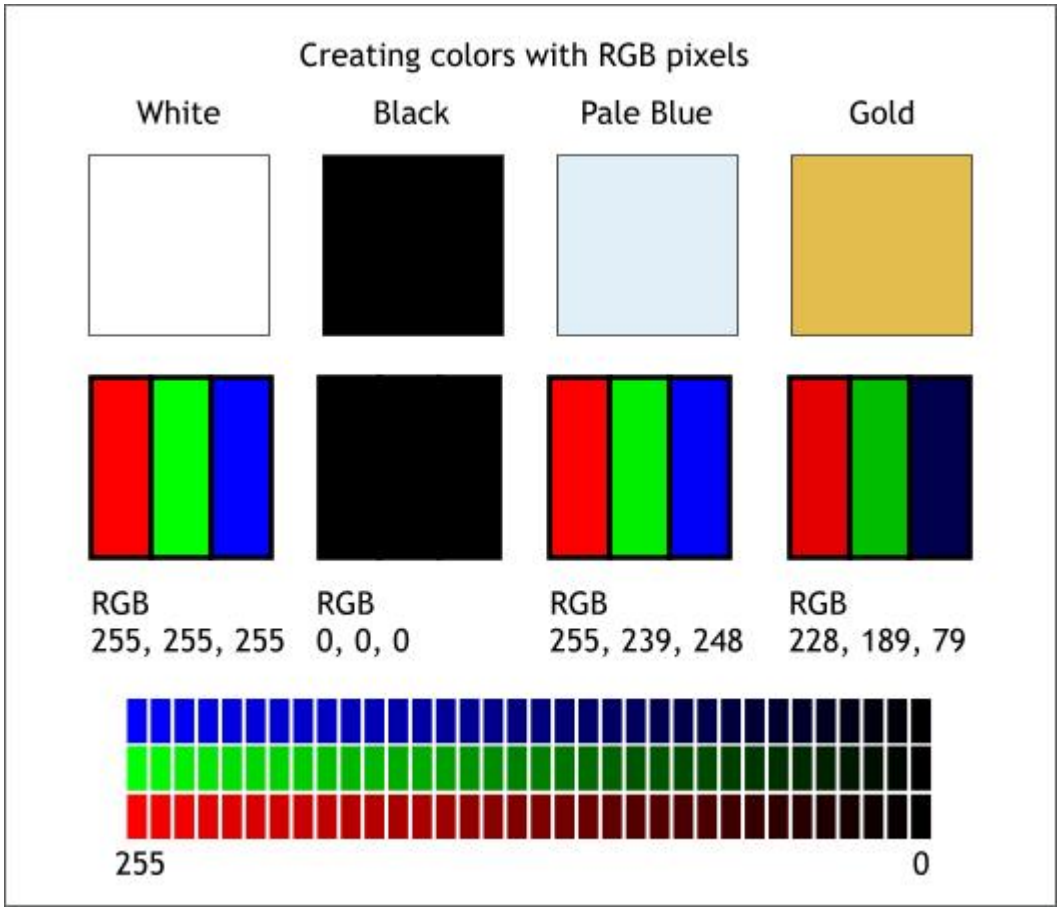
Machines interpret images very simply: as a series of pixels, each with their own set of color values.

Consider the simplified image below,and how grayscale values are converted into a simple array of numbers:



Think of an image as a giant grid of different squares, or pixels (this image is a very simplified version of what looks like either Abraham Lincoln or a Dementor).

Each pixel in an image can be represented by a number, usually from 0 – 255.
The series of numbers on the right is what software sees when you input an image.
For our image, there are 12 columns and 16 rows, which means there are 192 input values for this image.
When we start to add in color, things get more complicated.
Computers usually read color as a series of 3 values – red, green, and blue (RGB) – on that same 0 – 255 scale.



Now, each pixel actually has 3 values for the computer to store in addition to its position.

If we were to colorize President Lincoln (or Harry Potter's worst fear), that would lead to 12 x 16 x 3 values, or 576 numbers.

For some perspective on how computationally expensive this is,consider this tree:

- Each color value is stored in 8 bits.
- 8 bits x 3 colors per pixel = 24 bits per pixel.

A normal sized 1024 x 768 image x 24 bits per pixel = almost 19M bits, or about 2.36 megabytes.

That's a lot of memory to require for one image, and a lot of pixels for an algorithm to iterate over.

But to train a model with meaningful accuracy – especially when you're talking about Deep Learning – you'd usually need tens of thousands of images, and the more the merrier.

2.2 Convolutional neural networks :

Much of the progress made in computer vision accuracy over the past few years is due in part to a special type of algorithm. Convolutional Neural Networks are a subset of Deep Learning with a few extra added operations, and they've been shown to achieve impressive accuracy on image-associated tasks.

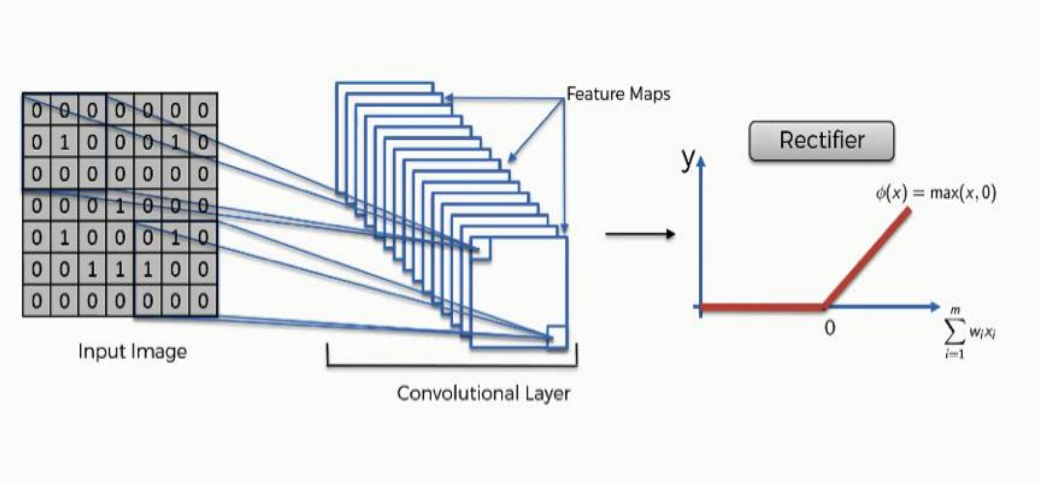
Convolutional Neural Networks (CNNs or ConvNets) utilize the same major concepts of Neural Networks, but add in some steps before the normal architecture. These steps are focused on feature extraction, or finding the best version possible of our input that will yield the greatest level of understanding for our model. Ideally, these features will be less redundant and more informative than the original input , The CNN uses three sorts of filters for feature extraction.

Convolution

During the convolution process (perhaps why it's called a CNN) the input image pixels are modified by a filter. This is just a matrix (smaller than the original pixel matrix) that we multiply different pieces of the input image by. The output – often called a Feature Map – will usually be smaller than the original image, and theoretically be more informative.

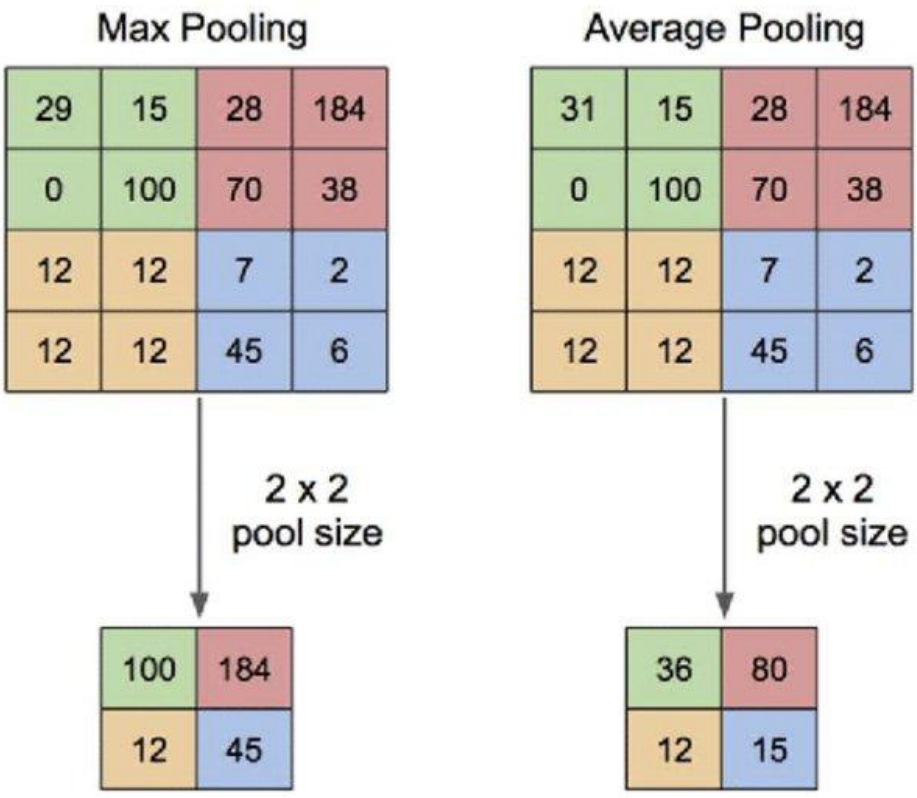
ReLU

This futuristic sounding acronym stands for Rectified Linear Unit, which is an easy function to introduce non-linearity into the feature map. All negative values are simply changed to zero, removing all black from the image. The formal function is $y = \max(0, x)$.



Pooling

In pooling, the image is scanned over by a set width of pixels, and either the max, sum, or average of those pixels is taken as a representation of that portion of the image. This process further reduces the size of the feature map(s) by a factor of whatever size is pooled.



3. Related Works

1.1 Related Survey Papers :

There are a few works on detecting violence more or less via different methods as stated with modifications in the model used. A. Datta , M. Shah et al(2002)[1] proposed a system that infers violence from motion trajectory of the limbs being traced by pose estimation models and then feeding it into an LSTM to get the inference. Tao Zhang, Zhijie Yang, et al (2016)[2] proposed a Gaussian Model of Optical Flows that when passed to the linear classifier gives regions where violence is inferred.

3.2 LITERATURE REVIEW :

Recently proposed methods for violence detection can be roughly classified into three categories visual based approach, audio-based approach and hybrid approach :

1. Visual Based Approach : Visual information is retrieved and represented as rele vant features in this approach. Local features and global features are two types of fea tures. Position, velocity, form, and color are examples of local features, while average speed, region occupancy, relative positional fluctuations, and the interactions between objects and backdrop are examples of global features.

2. Audio Based Approach : Audio data is used to classify violence in this approach.

It uses a hierarchical technique based on Gaussian mixture models and Hidden Markov

models to distinguish gunshots, explosions, and automobile braking in audio.

3. Hybrid Approach : The emphasis in the hybrid method is on merging visual and audio characteristics. Some techniques recognize violent incidents in videos utilizing flame and blood detection and recording the degree of motion, as well as the typ ical sounds of violent occurrences. The CASSANDRA system, detects aggression in surveillance videos using motion features associated with articulation in video and
scream-like cues in audio.

4. DATASET

The dataset contains 3500 video clips which belong to two classes,violence and non-violence respectively. The average duration of the video clips is 5 seconds and majority of those videos are from CCTV footage. For training, 350 videos each from the violent and non-violent classes are taken at each epoch. To test our methodology, we work with these three datasets,Hockey Fight Dataset , Movies Dataset and Real Life Violence Dataset . the 3 datasets captured from closed-circuited-TV,Phone or high-resolution recorder, the quality, number of pixels and length varies between dataset.

1. Hockey fights :
Dataset composed of equal numbers of violence and nonviolence action during hockey professional matches, usually Two players participating in close body interaction.

2. Movies :

This dataset consists of fight sequences collected from movies, for the non-violence label - videos of general action activity gathered from movies. The dataset is made up of an equal number of violent movie clips and non-violent movie clips. Unlike the Hockey dataset, this dataset varies profoundly between samples.

3. Real Life Violence Dataset :

This is a crowd violence dataset. Most of the crowd violence seen in this dataset are random clips

5. Methodology

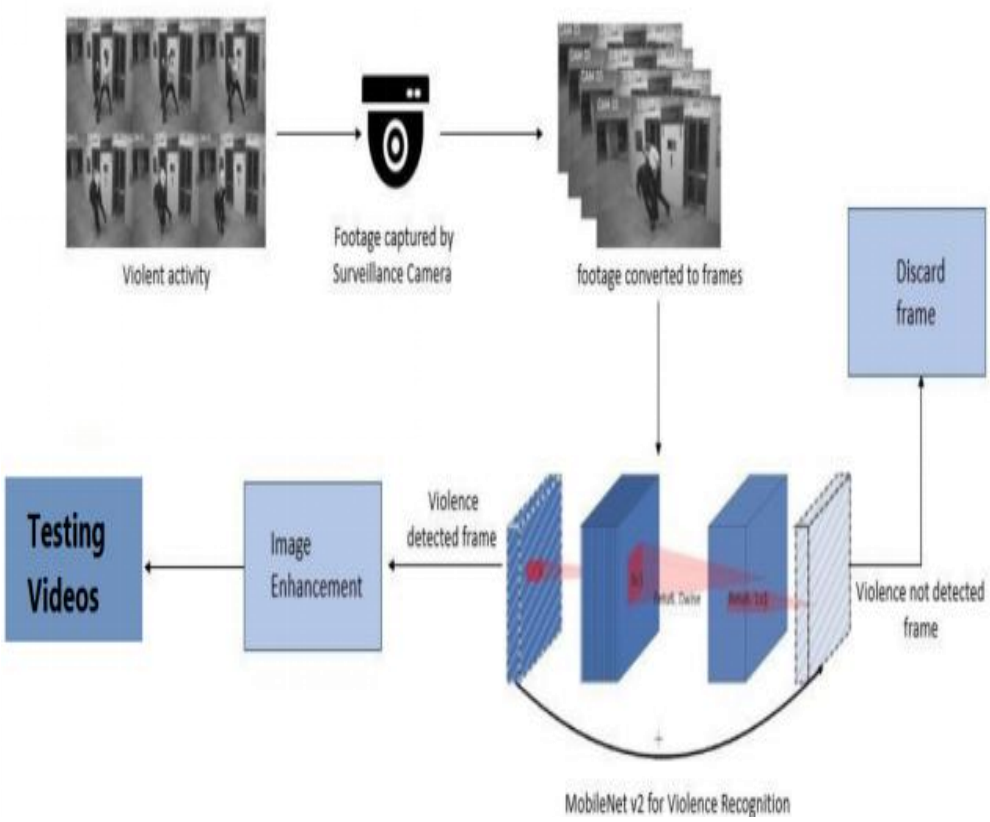
The method described here is based on the principle that when a fight occurs in the Surveillance Videos, the fighting frames of video will be significantly different than the Non-fighting frames.

we train a model that consists of a feature extractor and learns the temporal patterns of the input volume of frames.

The model is trained with video volumes consisting of normal and fighting scenes.

Footage from the surveillance camera is broken down into frames , The frames are given as input to MobileNet v2 classifier for detecting violent activities in the given sequence of input frames.

If no violent activity is recognized the respective frames are discarded. The violence detected frame is obtainedand it is enhanced for better clarity



5.1 Pre-Processing :

The pre-processing step includes importing the video frames and making it ready for training. It also involves feature extraction which is the input to the training algorithm. Initially the video is converted to frames. Each video will generate a set of frames which approximately denotes the number of seconds. This process of converting video to frame using python libraries is illustrated in the Figure .

The next task of this stage is to convert raw data to an acceptable input for the model. Each frame is extracted from the raw videos are modified by rotating , flipping , zooming and random brightness.

FIGURE 1
FRAME TO FRAME DIFFERENCE

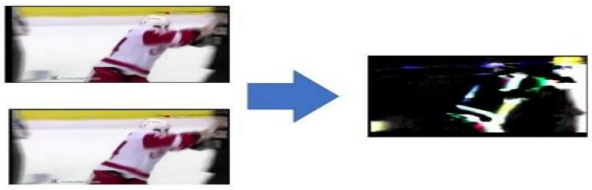


FIGURE 2
DARK EDGES REMOVAL

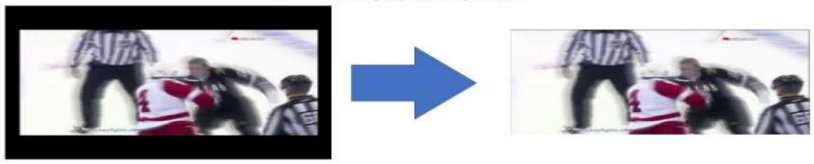


FIGURE 3
IMAGE CROPPING

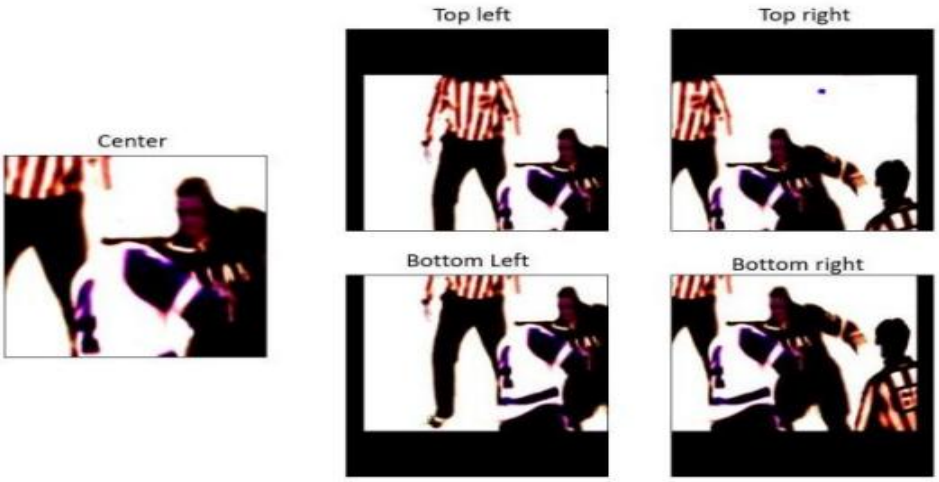
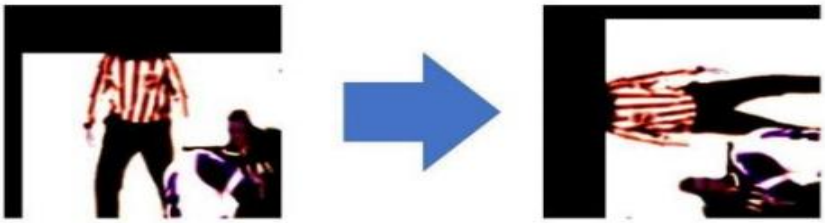
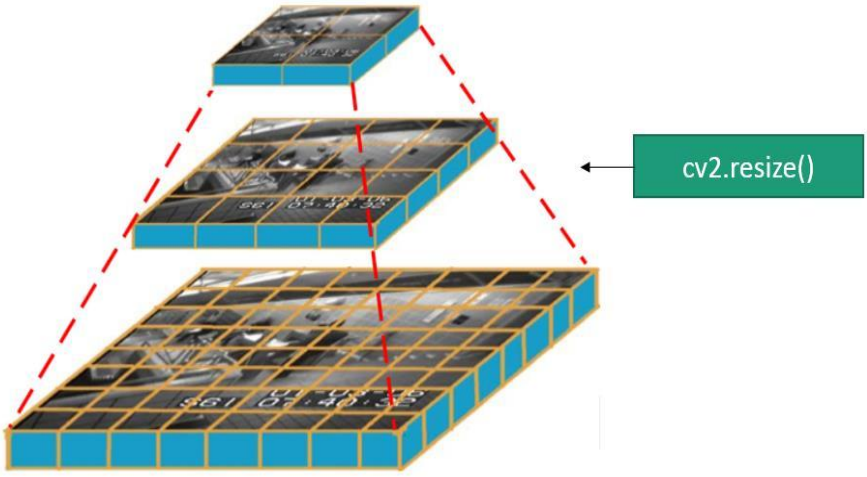


FIGURE 4
IMAGE TRANSPOSITION

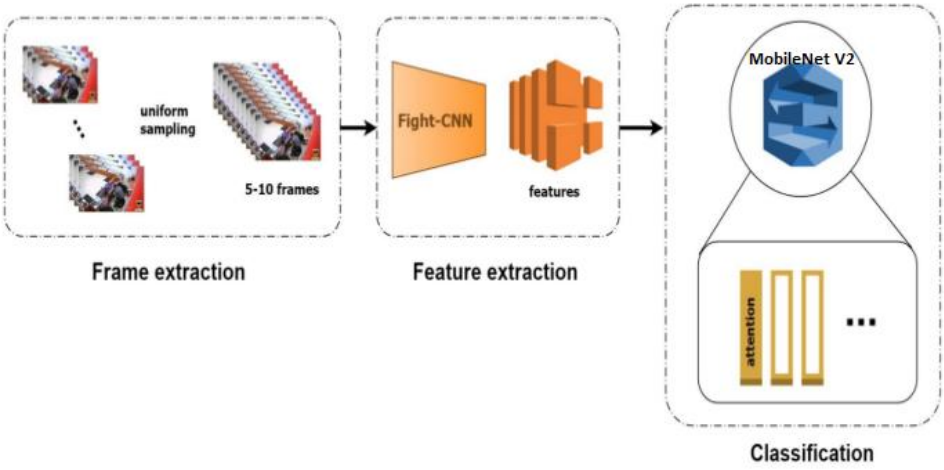


and resized to 128*128. To ensure that the input images are all on the same scale, the pixel values are scaled between 0 and 1 for normalization. Now the input is ready for model training.



5.2 Training

The features extracted in the data pre-processing step are used as an initial input to the training algorithm. We will use MobileNet v2 to make our model , Adam optimization algorithm and Binary_Crossentropy loss .
We use One Cycle Learning Rate with initial learning rate of 0.00001, with maximum learning rate of .0001 and learning rate decay rate of 0.8 per epoch . We use a batch size of 16.

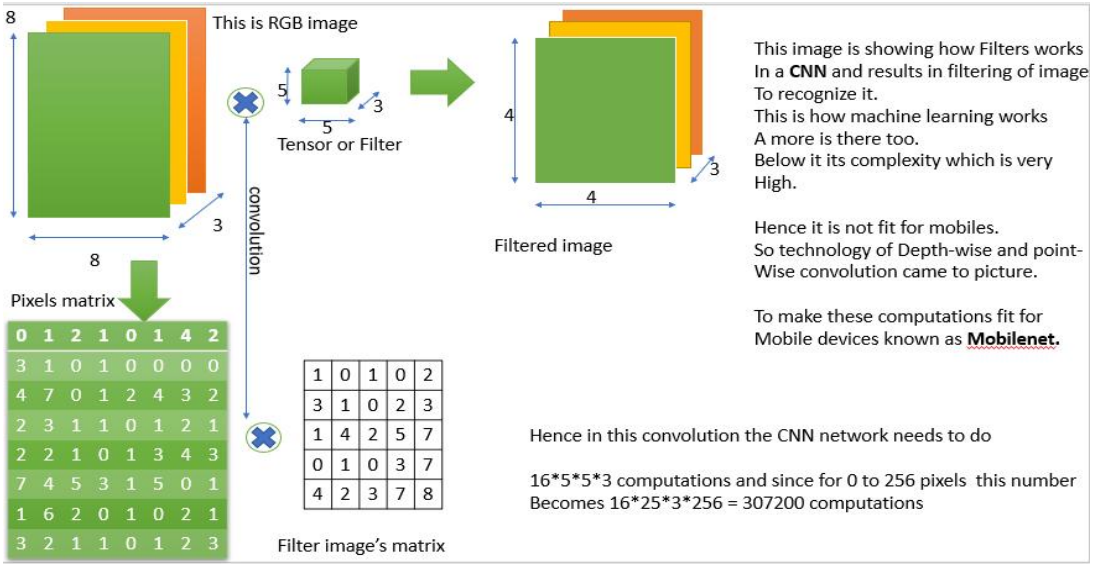


5.2.1 MobileNet v2 :

MobileNet v2 is a convolutional neural network architecture . It is based on an inverted residual structure where the residual connections are between the bottleneck layers .

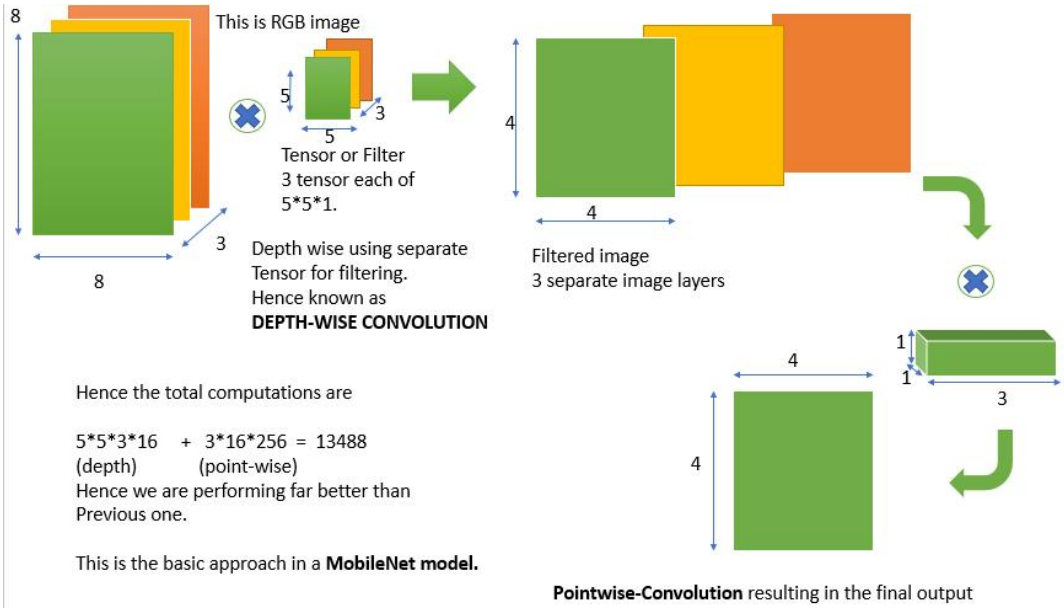
Since the images can be seen as a matrix of pixels and each pixel describes some features of the image, these technologies use filters to filter out a certain set of pixels in the images and results in the formation of output predictions about images.
CNN uses a lot of pre-defined and stored filters and does a convolution (X) of that filter with the pixel matrix of the image.

This results in filtering the image's objects and comparing them with a large set of predefined objects to identify a match between them. Hence in this way these models are able to predict the image.



But these technologies require a high GPU to increase the comparison rate between millions of data which cannot be provided by any mobile device.Hence, here comes in action what is known as MobileNet.

Mobilenet is a model which does the same convolution as done by CNN to filter images but in a different way than those done by the previous CNN. It uses the idea of Depth convolution and point convolution which is different from the normal convolution as done by normal CNNs. This increases the efficiency of CNN to predict images and hence they can be able to compete in the mobile systems as well. Since these ways of convolution reduce the comparison and recognition time a lot, it provides a better response in a very short time and hence we are using them as our image recognition model.



The main strategies introduced in MobileNetV2 were linear bottleneck and inverted residual blocks. In the linear bottleneck layer, the channel dimension of input is expanded to reduce the risk of information loss by nonlinear functions such as ReLU. It comes from the fact that information lost in some channels might be preserved in other channels. The inverted residual block has a ("narrow" -"wide"-""narrow") structure in the channel dimension.

5.2.2 Optimization for training

Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses
and We using **the "Adam" Optimizer :**
Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient

when working with large problems involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm.
Adam optimizer involves a combination of two gradient descent methodologies:

Momentum:
This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the 'exponentially weighted average' of the gradients.

$$w_{t+1} = w_t - \alpha m_t$$

where,

$$m_t = \beta m_{t-1} + (1 - \beta) \left[\frac{\delta L}{\delta w_t} \right]$$

m_t = aggregate of gradients at time t [current] (initially, $m_t = 0$)
m_{t-1} = aggregate of gradients at time t-1 [previous]
W_t = weights at time t
W_{t+1} = weights at time t+1
α_t = learning rate at time t
∂L = derivative of Loss Function
∂W_t = derivative of weights at time t
β = Moving average parameter (const, 0.9)

Root Mean Square Propagation (RMSP):

Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve performance on problems with gradients (e.g. natural language and computer vision problems).

$$w_{t+1} = w_t - \frac{\alpha_t}{(v_t+\varepsilon)^{1/2}} * \left[\frac{\delta L}{\delta w_t}\right]$$

where,

$$v_t = \beta v_{t-1} + (1-\beta) * \left[\frac{\delta L}{\delta w_t}\right]^2$$

W_t = weights at time t
W_{t+1} = weights at time t+1
α_t = learning rate at time t
∂L = derivative of Loss Function
∂W_t = derivative of weights at time t
V_t = sum of square of past gradients. [i.e sum($\partial L/\partial W t-1$)] (initially, $V_t = 0$)
β = Moving average parameter (const, 0.9)
ϵ = A small positive constant (10^{-8})

Mathematical aspect of Adam optimizer :

Taking the formulas used in the above two methods, we get

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta w_t}\right] v_t = \beta_2 v_{t-1} + (1-\beta_2)\left[\frac{\delta L}{\delta w_t}\right]^2$$

Parameters Used :
1. ϵ = a small +ve constant to avoid 'division by 0' error when (v_t -> 0). (10^{-8})
2. β_1 & β_2 = decay rates of average of gradients in the above two methods. (β_1 = 0.9 & β_2 = 0.999)
3. α – Step size parameter / learning rate (0.001)

Since mt and vt have both initialized as 0 (based on the above methods), it is observed that they gain a tendency to be 'biased towards 0' as both β1 & β2 ≈ 1. This Optimizer fixes this problem by computing 'bias-corrected' mt and vt. This is also done to control the weights while reaching the global minimum to prevent high oscillations when near it. The formulas used are:

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \widehat{v_t} = \frac{v_t}{1-\beta_2^t}$$

We are adapting to the gradient descent after every iteration so that it remains controlled and unbiased throughout the process, hence the name Adam.
Now, instead of our normal weight parameters mt and vt , we take the bias-corrected weight parameters (m_hat)t and (v_hat)t. Putting them into our general equation, we get

$$w_{t+1} = w_t - \widehat{m_t}\left(\frac{\alpha}{\sqrt{\widehat{v_t}}+\varepsilon}\right)$$

5.2.3 Binary Crossentropy loss :

In machine learning lingo, a ‘cost function‘ is used to evaluate the performance of a model.The cost function quantifies the difference between the actual value and the predicted value and stores it as a single-valued real number. The cost function can analogously be called the ‘loss function‘ and we are using here The Cross-Entropy Cost Function

The idea behind Shannon entropies :

The Entropy of a random variable X can be measured as the uncertainty in the variables’ possible outcomes. This means the more the certainty/probability, the lesser is the entropy.
The formula to calculate the entropy can be represented as:

$$H(x) = -\int_x 1.p(x)\log p(x), \text{ if } X \text{ is continuous} \quad (1)$$

$$H(x) = \sum_x p(x)\log p(x), \text{ if } X \text{ is discrete} \quad (2)$$

Binary cross-entropy cost function:

In Binary cross-entropy, there is only one possible output. This output can have discrete values, either 0 or 1. For example, let an input of a particular fruit's image be either that of an apple or that of an orange. Now, let us rewrite this sentence: A fruit is either an apple, or it is not an apple. There are only binary, true-false outputs possible.

Let us assume that the actual output is represented as a variable y:

- ***Cross-entropy(d) = - y*log(p) when y = 1***
- ***Cross-entropy(d) = - (1-y)*log(1-p) when y = 0***

5.2.4 Activation function :

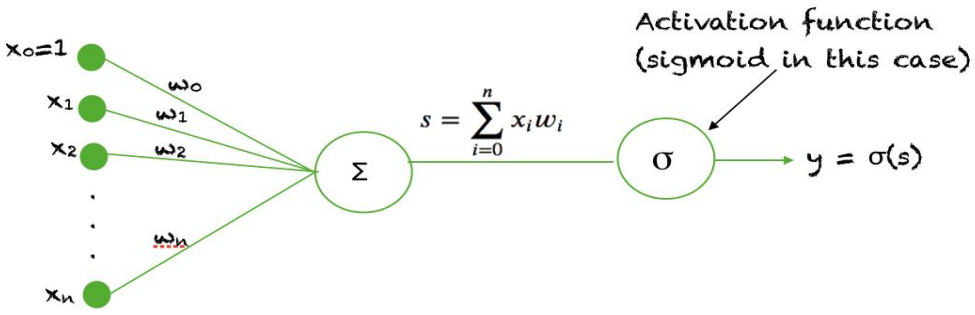


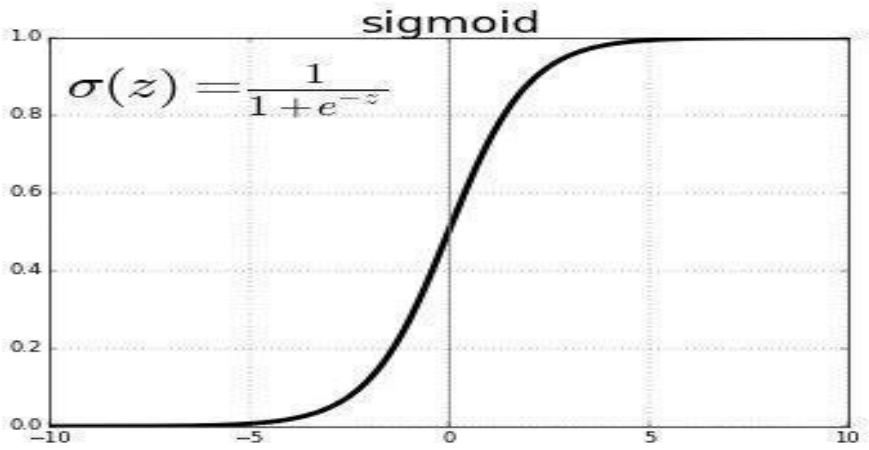
Figure 22 activation function

We use Sigmoid Function and it is a special form of the logistic function and is usually denoted by

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

Sigmoid function

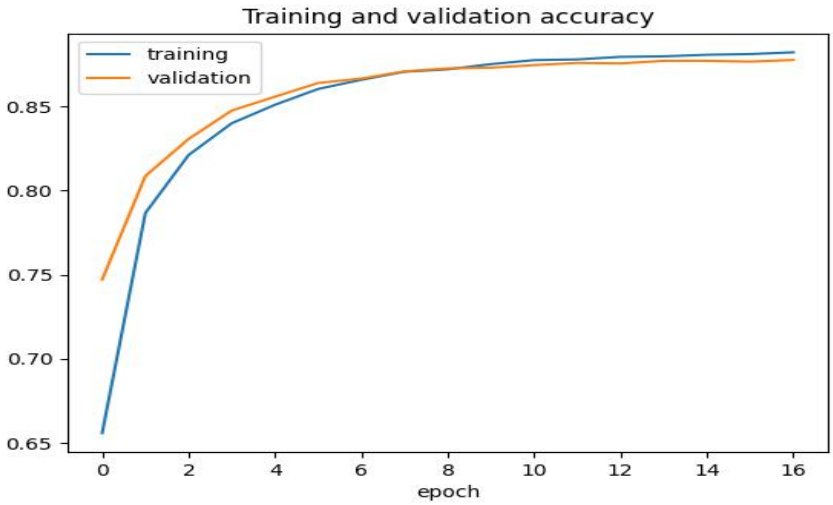
When the activation function for a neuron is a sigmoid function it is a guarantee that the output of this unit will always be between 0 and 1. Also, as the sigmoid is a nonlinear function, the output of this unit would be a nonlinear function of the weighted sum of inputs.

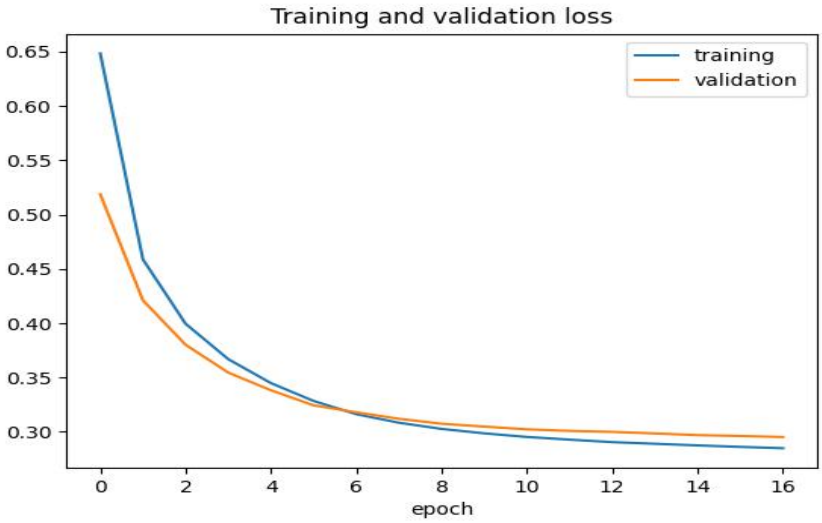


6. Results and Discussion

In this section testing and training accuracy are displayed in the below given graphical representation. displaying the training and testing accuracy and loss for the MobileNet v2 model when a dataset containing 3500 videos of average duration 5 seconds is given as input. For each epoch 1750 videos from the fight class and 1750 videos from the non-fight are trained , the accuracy and loss comes to a constant level of increment and

decrement after approximately 6 epochs. 90% accuracy was obtained on training.





For the output : A video with violence is given as input to the system. It shows one frame in the video that was labeled to have violent activity. Another video clip without violent activity was given as input. It shows one frame of that video which is rightly labeled as false or non-violence.

7. CONCLUSION

Violence scene detection in real-time is a challenging problem due to the diverse content and large variations in quality. In this research, we use the MobileNet v2 model to offer an innovative and efficient technique for identifying violent events in real-time surveillance footage. The proposed network has a good recognition accuracy in typical benchmark datasets, indicating that it can learn discriminative motion saliency maps

successfully. I implemented deep learning model to predict violence in video data, I found our implementation to deal well with this task even though our GPU power was relatively low. The potential of deep learning models is high and can be used easily by law enforcements officers to identifying violence in the streets or in high schools. I found the smart data preprocessing of the video's frames play an important factor as well as some of the training parameters such as: CNN network, learning rate and data augmentation. Looking forward to more complex violence scenarios and appliances it will take researchers to find creative solutions for data collection, advance generalization techniques and real-time optimizations.

REFERENCES :

[1] A. Datta, M. Shah and N. Da Vitoria Lobo, "Person-on-person violence detection in video data," Object recognition supported by user interaction for service robots, 2002, pp. 433-438 vol.1, doi: 10.1109/ICPR.2002.1044748.

[2] Yu Zhao and Rennong Yang and Guillaume Chevalier and Maoguo Gong (2017). Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors. CoRR, abs/1708.08989.

[3] Sudhakaran, Swathikiran, and Oswald Lanz. "Learning to detect violent videos using convolution long short-term memory." IAdvanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on, pp. 1-6. IEEE, 2017.

[4] Nievas, Enrique Bermejo and Suarez, Oscar Deniz and Garcia, Gloria Bueno and Sukthankar, Rahul, "Hockey Fight Detection Dataset", 2016, hosted on bittorrent.

[5] T. Hassner, Y. Itcher, and O. Kliper-Gross, Violent Flows: Real-Time Detection of Violent Crowd Behavior, 3rd IEEE International Workshop on Socially Intelligent Surveillance and Monitoring (SISM) at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Rhode Island, June 2012 .
[6] Nievas, E., Suarez, O., Garcia, G., & Sukthankar, R. (2011). Movies Fight Detection Dataset. In Computer Analysis of Images and Patterns (pp. 332–339).
[7] E. Acar, F. Hopfgartner and S. Albayrak, "Breaking down violence detection: Combining divide-et-impera and coarse-to-fine strategies", Neurocomputing, vol. 208, pp. 225-237, 2016.
[8] L.-H. Chen, H.-W. Hsu, L.-Y. Wang and C.-W. Su, "Violence detection in movies", International Conference on Computer Graphics Imaging and Visualization (CGIV), 2011.
[9] T. Giannakopoulos, A. Pikrakis and S. Theodoridis, "A multi-class audio classification method with respect to violent content in movies using bayesian networks", IEEE Workshop on Multimedia Signal Processing (MMSP), 2007.
[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun (2015). Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385.
[11] Christian Szegedy and Vincent Vanhoucke and Sergey Ioffe and Jonathon Shlens and Zbigniew Wojna (2015). Rethinking the Inception Architecture for Computer Vision. CoRR, abs/1512.00567.
[12] Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.
[13] Zhang, T., Yang, Z., Jia, W. et al. A new method for violence detection in surveillance scenes. Multimed Tools Appl 75, 7327–7349 (2016).
[14] Abdarahmane Traoré, Moulay A. Akhloufi, "Violence Detection in Videos using Deep Recurrent and Convolutional Neural Networks", Systems Man and Cybernetics (SMC) 2020 IEEE International Conference on, pp. 154-159, 2020.
[15] Tal Hassner, Yossi Itcher, and Orit Kliper-Gross. Violent flows: Real Time detection of violent crowd behavior. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pages 1–6, IEEE, 2012.