INTERNATIONAL JOURNAL
OF SOFT COMPUTING

SINCE 2000
MEDWELL PUBLICATIONS

Performance Evaluation of Support Vector Machines (SVM) and Convolution Neural Networks (CNN) for Video Tampering Classification

[1]S.K. Komal, [2]Puneeth Chandrashekar, [1]B.S. Rekha and [1]G.N. Srinivasan
[1]*Department of ISE, RV College of Engineering (RVCE), Bengaluru, India*
[2]*PRAC_Platforms, Mindtree, Bengaluru, India*

Key words: Support vector machiness, convolutional neural networks, tensorflow, surveillance activities, performance

Corresponding Author:
S.K. Komal
Department of ISE, RV College of Engineering (RVCE), Bengaluru, India

Page No.: 53-60
Volume: 14, Issue 3, 2019
ISSN: 1816-9503
International Journal of Soft Computing
Copy Right: Medwell Publications

Abstract: Intelligent video surveillance system are extensively used in each and every sector of business. Ranging from small shops to safety systems, surveillance has become an integral part. In these fielded systems, a variety of factors can cause camera obstructions and persistent view change. The view change may adversely affect their performance. Examples include intentional blockage, noise, frame freeze, etc. which might warrant alarms. Considering the fact that the intelligent surveillance system is with very less human intervention, it is important to efficiently classify the tampered video. Analysis of the tampered videos helps in further scene investigation. The goal of the project is to use Support Vector Machines (SVM) a machine learning technique which classifies the real-time videos based on features extracted. The features selected are histogram gradients, HSV (Hue Saturation Value) and RGB (Red Blue Green) for the color based classification and edges (edge weight and direction) for the texture based classification. Further improvements are done using a deep learning technique such as CNN. Convolution neural networks make use of large amount of training data and use tensorflow framework for classification. The system accepts video inputs in mp3 or avi format. The output is the classification of tampered videos and alarm generation. Comparison between the two methodologies is done. Support vector machines gives an accuracy of 75% and convolutional neural networks give accuracy of 93%. The system is very useful to monitor all the surveillance activities.

INTRODUCTION

This new evolving world is enclosed with a huge amount of visual and virtual information. For analysis of all the huge data analysis techniques are really important. Image processing is one such field of science that is used to analyze and organize such data. Image processing is used in many fields of knowledge because of its wide application. Analyzing and understanding the surveillance video is gaining importance, especially in the context of citizen security, safety. Machine learning and deep learning algorithms happen to be the best fit technique for the classification of videos. Machine learning and deep learning algorithms such as SVM and CNN are used for classification. The first step in SVM method is to extract features, next is to create training matrix. The matrix is given for training and results are got. In convolutional neural networks the first step is calculating the weights and next is predicting the results. Support vector machines (SVMs, also support vector network) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to a single category, an SVM training algorithm builds a model that assigns new examples to one category of the category. Linear SVM is made use to categorize the tampered images into the given respective classes such as blocked, noised, etc. The features of each class are fed to SVM, training is done and the results are got.

Convolution neural network is a class of deep, feed forward artificial neural network. CNN has successfully been applied for analyzing visual imagery. CNN require very minimal pre-processing because they are bases on multi-layer perception's design. Tensorflow is a form of CNN framework used for classification.

Scope of the video classification system:

- Analysis of video tampering
- Tampering classification further helps in analysis of the situation in case of any criminal activity

Literature review: Karpathy *et al*. (2014) "Large-scale video classification with convolutional neural networks". In the field of image recognition, convolution neural networks are considered to be a very powerful class. Utilizing this powerful class of machine learning techniques the researchers of the study have provided an extensive CNN evaluation technique. This technique is implemented on a dataset of 1 million YouTube videos, a large scale video evaluation with 487 classes. The researchers study number of approaches for extension of connectivity of CNN. The CNN architecture takes advantage of local spatio-temporal information and it suggests a multi-resolution method for speeding up the training process.

Burney and Syed (2016) "Crowd video classification using convolutional neural networks." CNN is one of the deep learning technique that is used extensively for image analysis and interpretation tasks. Deep learning methods tend to become more expensive if they are used for video analysis as they require memory and additional temporal information. Recently crowd video analysis has come to limelight because of its extensive use in all retail places. In the given study, it shown that to classify the videos by 3-clannel image map, 2D CNN have been used. For every video given spatial and temporal values are calculated. The calculation of such reduces space complexity and time complexity, it is very less when compared to a 3D CNN which is usually used in video processing. The model is tested with the dataset and there is no additional requirements to improve upon the accuracy.

Ashwin *et al*. (2016) "Video affective content analysis based on multimodal features using a novel hybrid SVM-RBM classifier".

Content analysis for videos is a very pre-dominant area of study. Nowadays all communication media is inspired through video streaming. Hence, in the recent times video content analysis has become a very important area of research and plays a very important role in communication systems. The users use acoustic or visual features for their immediate state of use and hence, the existing system given by the authors is mainly focused on content analysis. The study gives a novel method which is a hybrid of SVM and RBM classifier. The combined method detect the emotions of videos which are taken while live streaming and normal stored video dataset. Thus the system recognizes the user's current state of mood by the facial emotion descriptors that are mentioned. Test experiments given in the paper are conducted on live streaming data such which are captured from the devices such as normal web cam and Microsoft Kinect. Further to validate the system which was proposed, the researchers used HUMANE and SAVEE datasets. The researchers use both SVM (Support Vector Machine) and RBM (Restricted Boltzmann Machine) for the classification. It is seen that the hybrid classifier performs better than RBM and SVM for annotated datasets.

Li Wang, Member and Dennis SNG "Deep learning algorithms with applications to video analytics for a smart city: A survey" (Wang and Sng, 2015). In recent times deep learning algorithms have given promising output in a verity of areas including speech detection, NPL and computer version. It user deep architectural models to learn the representation is a hierarchical form. When a smart city is considered there is a lot of data that needs to be processed and analyzed (e.g.,: Information captured from different sensors). In the study the researchers examine different deep learning methods for video processing in smart cities. Various research topics include: object tracking, object identification, face recognition, image processing and scene labeling.

MATERIALS AND METHODS

The system makes use of majorly two algorithms SVM and CNN but before that there are common pre-processing steps that are done. The system architecture is shown in Fig. 1.

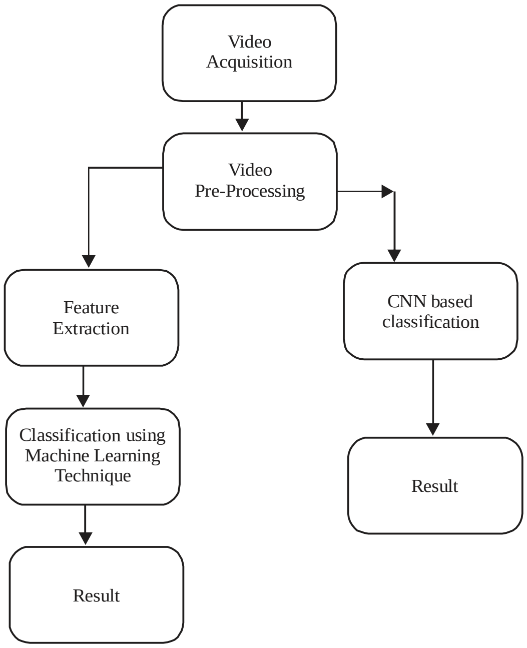


Fig. 1: System architecture

Video sharpening: For sharpening Laplacian method is used. The sharpening process is done by highlighting the sharp intensity changes in the image. This method is often used for edge detection. The input to the method is a gray scale image and output is another gray scale image with different levels of intensity. There are different order of derivatives for intensities the first level is given in Eq. 1. Here, I (x, y) gives the intensity and L(x, y) is Laplacian of an image:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (1)$$

Video to frame conversion: The frame extraction is a very important step role a lot of video processing applications like content based video retrieval, shot detection, segmentation, CC cameras, etc. The video to frame conversion can be done in many ways. To begin user needs how many frames he/she needs per second, so which indicates that there will be a chances of missing the frames on which they are concentrating more, normally, the number frames per second will be different for the different cameras. Figure 2 shows the example of video to frame conversion.

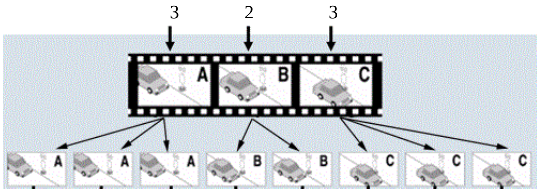


Fig. 2: Video to frame conversion

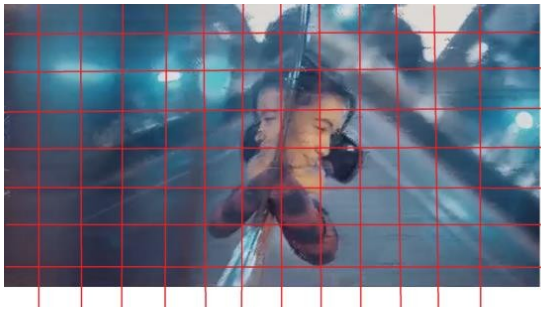


Fig. 3: Dividing image into blocks

Support vector machines: Based on the decision plane concept and the method that separates two classes by the plane SVM is built. The plane that separates two classes is called hyperplane. The classes on either side of the plane are called member classes. The separation of member functions is based on the features that are extracted from the class members. All the class members are given separate labels that define them. The labels are either numerals or any other representation. Most of the machine learning algorithms have feature extraction at the base.

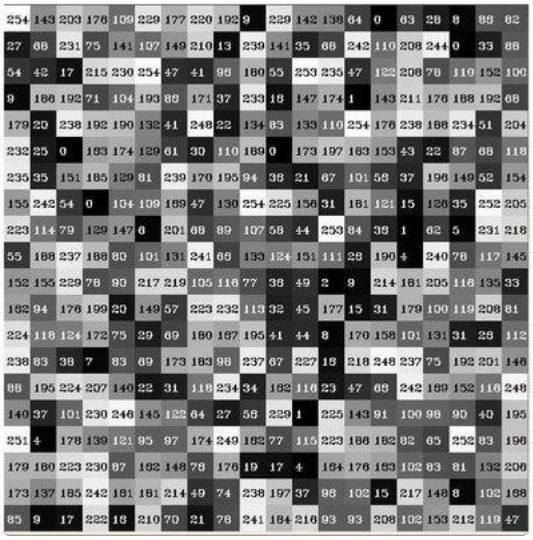
Feature extraction

Histogram gradients: To calculate accurate histograms of any image the whole image is divided into number of blocks. In this method the image is divided into 4×4 cells. For each cell a feature descriptor is calculated. The feature descriptors are mean and standard deviation. A feature descriptor is selected because it gives an exact and compact representation of the image block. Figure 3 shows the division of image into blocks. For each block histogram bins are calculated. The data organized in pre-defined counts are called as histogram bins. The bin values are modified according to the classifier to give best results. The standard value is 50 which is considered for tampered frames. Taking the figure one as example a matrix of bins can be constructed. Imagine that a matrix contains information of an image (i.e., intensity in the range 0-255) as shown in Fig. 4.

Since, it is known that the range of information value for this case is 256 values, the values can be segmented in subparts (called bins) like:

$$[0,255] = [0,15] \cup [16,31] \cup ,..., \cup [240,255]$$
$$\text{range} = \text{bin}_1 \cup \text{bin}_2, ..., \text{bin}_{n=15}$$

The pixels that fall into the bin range can be kept count bin_i.



254	143	203	176	109	229	177	220	192	9	229	142	138	64	0	63	28	8	88	82
27	68	231	75	141	107	149	210	13	239	141	35	68	242	110	208	244	0	33	88
54	42	17	215	230	254	47	41	98	180	55	253	235	47	122	208	76	110	152	100
9	186	192	71	104	193	88	171	37	233	18	147	174	1	143	211	176	188	192	68
179	20	238	192	190	132	41	248	22	134	83	133	110	254	176	238	188	234	51	204
232	25	0	183	174	129	61	30	110	189	0	173	197	183	153	43	22	87	68	118
235	35	151	185	129	81	239	170	195	94	38	21	67	101	58	37	196	149	52	154
155	242	54	0	104	109	169	47	130	254	225	156	31	161	121	15	126	35	252	205
223	114	79	129	147	6	201	68	89	107	58	44	253	84	38	1	62	5	231	218
55	188	237	188	80	101	131	241	66	133	124	151	111	28	190	4	240	76	117	145
152	155	229	76	90	217	219	105	116	77	38	49	2	9	214	161	205	116	135	33
182	94	176	199	20	149	57	223	232	113	32	45	177	15	31	179	100	119	208	81
224	118	124	172	75	29	69	180	187	195	41	44	8	170	158	101	131	31	28	112
238	83	38	7	83	69	173	183	98	237	67	227	18	218	248	237	75	192	201	146
88	195	224	207	140	22	31	118	234	34	182	116	23	47	68	242	169	152	116	248
140	37	101	230	246	145	122	64	27	58	229	1	225	143	91	100	98	90	40	195
251	4	178	139	121	95	97	174	249	182	77	115	223	186	182	82	65	252	83	196
179	180	223	230	87	182	148	78	176	19	17	4	184	176	183	102	83	81	132	206
173	137	185	242	181	181	214	49	74	238	197	37	98	102	15	217	148	8	102	168
85	9	17	222	18	210	70	21	78	241	184	216	93	93	208	102	153	212	119	47

Fig. 4: Matrix of bins

Edge feature: Sobel operator is based on the concept of discrete differentiation. It computes an approximation of the gradient of an image intensity function. Sobel method is a combination of differentiation and Gaussian smoothing. There are two derivatives that are calculated from sobel method. The derivatives are min and max or X and Y, respectively. Considering the image to be operated as I the values of horizontal and vertical changes are calculated as:

Horizontal changes: The value is calculated by convolving an odd sized kernel with I. Consider a kernel of size 3 then, Gx would be computed as:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times I$$

Vertical changes: The value is calculated by convolving an odd sized kernel with I. Consider a kernel of size 3 then, Gx would be computed as:

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \times I$$

The final equation used is:

$$G = |G_x| + |G_y|$$

SVM architecture: In all the SVM methodologies the classification is done in majorly two steps. The two steps are training and testing. The training phase consists of feature extraction and training of SVM with SVM parameters. The features considered, here are histogram gradients and sobel deravatives. The values got from these features are put into a matrix. The matrix is usually a training mat that is fed into the SVM. The matrix consists of rows of feature values and every row is ending with the label. This matrix is then fed to the SVM. The paramentrs considered are: Kernal = Linear, Matrix= Training mat, Gamma function = 3. Figure 5 shows the training and testing phase of SVM.

Convolutional neural network

Convolution neural network based on tensorflow: Tensorflow is a framework released by Google which is a neural network framework. The dataset for this CNN consists of 4 classes, those classes are blank frame, blocked, noised and normal frame. The images from all the four classes are fed into the convolutional neural network. These form the base and the next two layers. After the layers are stacked upon each other flattening is done. At the end two fully connected layers are got. The second fully connected convolution layer has the probability of the image belonging to one of the four classes. The whole method is described in Fig. 6.

Reading inputs: Typically, the dataset is divided into 3 parts:

Training data: Use 80%.

Validation data: About 20% pictures will be utilized for validation. These pictures are chosen from training dataset to ascertain accuracy independently during the training process.

Test set: The test set consists of data that are different from the training set. In this case it consists of data from different cameras. The test data must also consist of all the classes that are considered for the classification. The very common problem faced while testing is overfitting. Overfitting is the method where the validation set gets good results but the test results areinaccurate.

Creating network layers

Creating the convolution layer in tensorflow: There are mainly three inputs that are considered for building of the network layer and those are:

Input: The second layer of the network which consists of the input from the first layer. This should be a 4-D tensor

Filter: trainable variables defining the filter.

Strides: Describes how much the sliding window moves while the convolution is done.

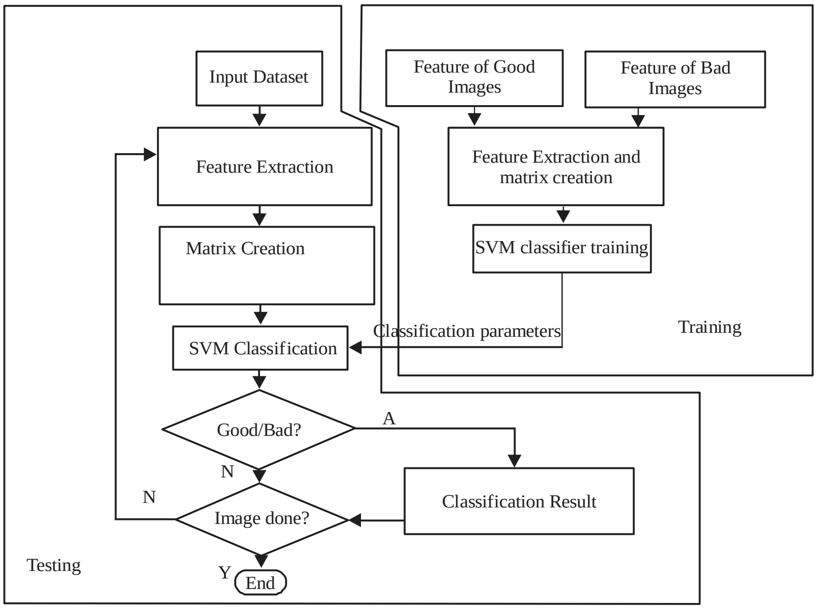


Fig. 5: SVM architecture

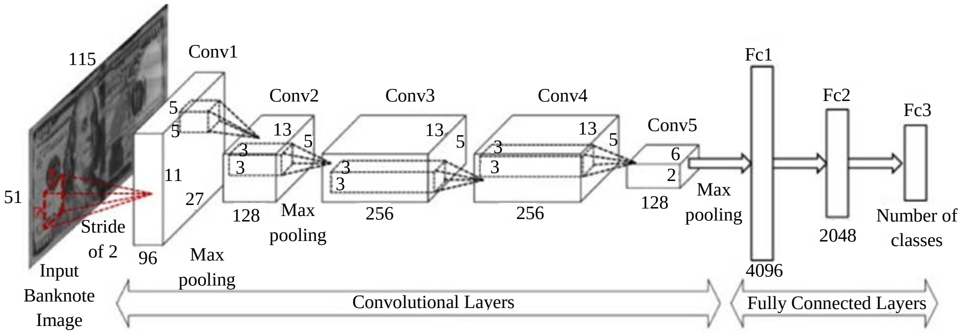


Fig. 6: Tensorflow based CNN

Padding: This is an option where there is consideration of overfitted data. The usage of SAME function means that user shall 0 pad the input such a way that output x,y dimensions are same as that of input.

After the first convolution step is done, there is an option to add biases on the neurons (this is done when there are new images that are added to the system). These biases can be learnable or trainable again. The next step is to learn the new values and this is done by random normal distribution. Finally, there is application of the max- pooling method which is very similar to the conv2d method. In this step all the operations are completed in a single layer. This final step is to creation of a function to define a complete convolutional layer.

Flattening layer: This is the second step after layer stacking, output of which is a multi-dimensional tensor. Naturally this is converted to a single dimensional for complexity reduction. The method used is flattening. The flatten method uses a simple reshape layer function for conversion.

Fully connected layer: After flattening is done to all the data a fully connected layer has to be formed. For this layer there is declaration of weights and biases. The weights and biases form the random normal distribution. In this layer all the inputs are considered and the slandered operation of z = wx+b is performed.

Place holders and input: The further step is to define a placeholder that carries all the training images. The input dataset is read and all the variable sized images are resized to a single size. The input placeholder is then created. The first dimension being none means the users can pass any number of images to it. For this program, user shall pass images in the batch of 16, i.e., shape will

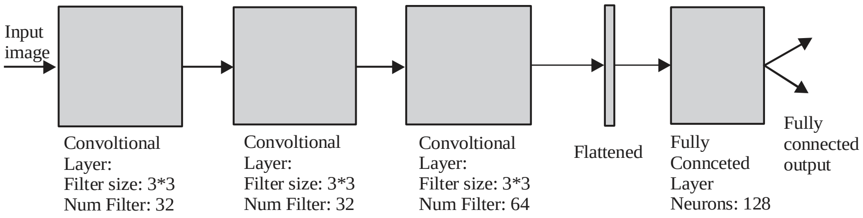

Input image
Convoltional Layer: Filter size: 3*3 Num Filter: 32
Convoltional Layer: Filter size: 3*3 Num Filter: 32
Convoltional Layer: Filter size: 3*3 Num Filter: 64
Flattened
Fully Connceted Layer Neurons: 128
Fully connected output

Fig. 7: Tensorflow CNN architecture

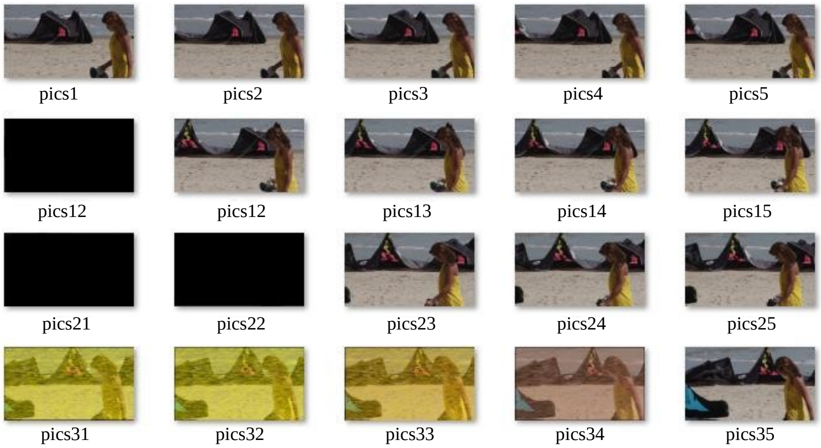


Fig. 8: Input tampered frames

be [16 128 128 3]. Similarly, he/she create a placeholder y_true for storing the predictions. For each image, having two outputs, i.e., probabilities for each class.

Predictions: The result of CNN is in the form of probability percentage. The probability of each class is given by the soft-max layer. The softmax is used to output the last layer. Softmax layer is usually last layer or the prediction layer in the classifier. When softmax are considered the networks are trained under the log-loss regime. The log- loss regime gives a non-linear variant of logistic regression. There are a lot of weights that are obtained after the final layer of the poling. There are costs that are used to describe the best of the weights given. The simple cost that is obtained or the least obtained is taken. This calculation is done using the function softmax_cross_entropy_with_logits that takes the output of last fully connected layer and actual labels to calculate cross_entropy whose average will give them thecost.

System specific CNN architecture: The convolution neural network is based on a tensorflow framework and has the architecture as shown in Fig. 7. Normally the basic CNN structure consists of stacks of convolutional layers. First few layers perform the feature extraction and give the weights and biases. Following all the stacked layers are the pooling layers which calculate the winner weights. The layers get dense and the last dense layer contains the target classes. In the architecture shown in Fig. 7. The convolution is done and the models are formed. The models are then saved. When the testing dataset is applied on to the network the saved model is retrived with the winner weights and the probability prediction is done.

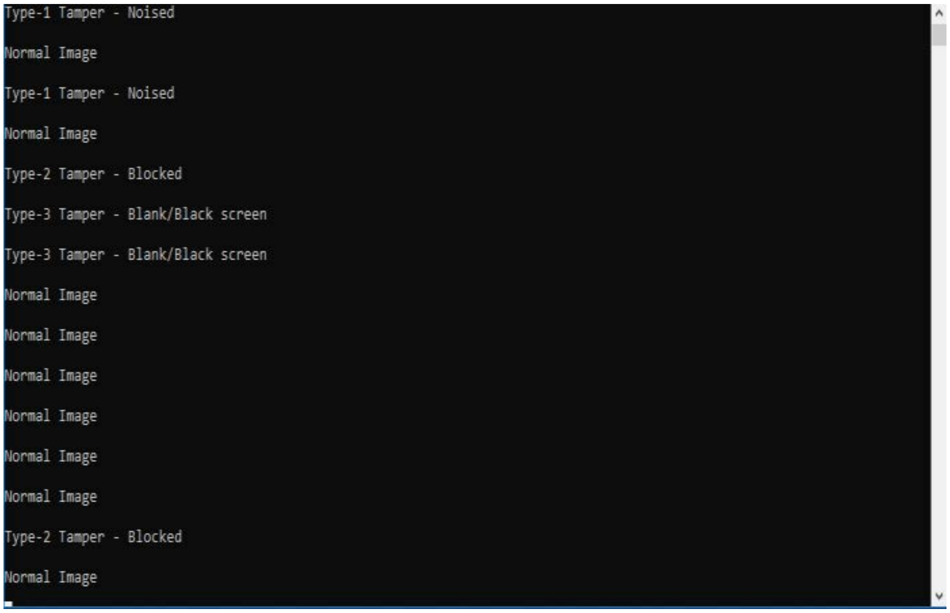
RESULTS AND DISCUSSION

This chapter gives the actual picture of the output and verifies if the system works and verifies the given design for the work. The correctness of the project's result is analyzed here.

Input data: The given input is a video of mp4 or avi format. Figure 8 shows the frames which are obtained from the tampered video that is got. Here, the video frames are obtained from video.

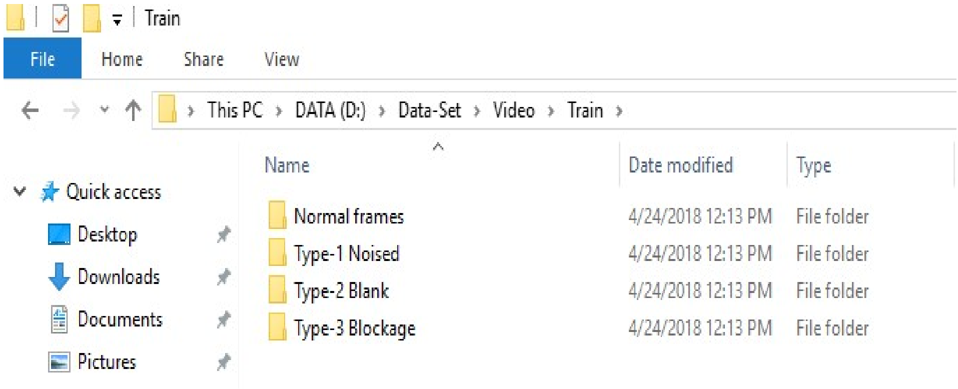
Output for multi-class tampered frames: Figure 8 shows the result given by SVM and the classified frames. The output is the labels for the corresponding classes.

Output of CNN: Figure 9 shows the different folders used for CNN where each folder is considered as the



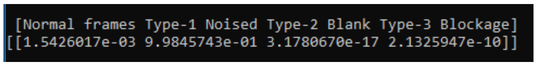
```
Type-1 Tamper - Noised

Normal Image

Type-1 Tamper - Noised

Normal Image

Type-2 Tamper - Blocked

Type-3 Tamper - Blank/Black screen

Type-3 Tamper - Blank/Black screen

Normal Image

Normal Image

Normal Image

Normal Image

Normal Image

Normal Image

Type-2 Tamper - Blocked

Normal Image
```

Fig. 9: The output of SVM



Name	Date modified	Type
Normal frames	4/24/2018 12:13 PM	File folder
Type-1 Noised	4/24/2018 12:13 PM	File folder
Type-2 Blank	4/24/2018 12:13 PM	File folder
Type-3 Blockage	4/24/2018 12:13 PM	File folder

Fig. 10: Folders for CNN



```
 [Normal frames Type-1 Noised Type-2 Blank Type-3 Blockage]
[[1.5426017e-03 9.9845743e-01 3.1780670e-17 2.1325947e-10]]
```

Fig. 11: CNN results

index and the label. Figure 10 shows the execution of CNN where the folders are indexed. Figure 11 shows the predicted results for CNN.

Comparison classification results

SVM classification:

- The training data consisted of 400
- Kernel considered: Linear with size 3
- Testing accuracy: 75%
- Number of images in one test class: 20

CNN classification:

- Training dataset consists of 2000 images
- Training accuracy: 93%
- Number of images per testing class: 10

CONCLUSION

The system depicts: Preprocessing techniques which include Sharpening and video to frame conversion. Features such as histogram gradients and edge weights have been considered for SVM classification. CNN classifier is based on tensorflow framework. The system is useful in analysis of surveillance videos and video tampering classification in case of anymal-practices.

RECOMMENDATION

The classification can be enhanced when more number of cases such as fine ringing and distortion effects are considered. Correction techniques can be applied for the correctly identified cases such as noise and physical camera coverage.

REFERENCES

Ashwin, T.S., S. Saran and G.R.M. Reddy, 2016. Video affective content analysis based on multimodal features using a novel hybrid SVM-RBM classifier. Proceedings of the 2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON'16), December 9-11, 2016, IEEE, Varanasi, India, pp: 416-421.

Burney, A. and T.Q. Syed, 2016. Crowd video classification using convolutional neural networks. Proceedings of the 2016 International Conference on Frontiers of Information Technology (FIT'16), December 19-21, 2016, IEEE, Islamabad, Pakistan, pp: 247-251.

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, 2014. Large-scale video classification with convolutional neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2014), June 2014, IEEE, New York, USA., pp: 1725-1732.

Wang, L. and D. Sng, 2015. Deep learning algorithms with applications to video analytics for a smart city: A survey. J. IEEE., Vol. 1,