MAJOR PROJECT REPORT

(Semester-22231)

Cigarette Smoking Detection using Deep Neural Networks

UNDER THE GUIDANCE OF:

PROF. POONAM SAINI

PROF. ARUN SINGH

TEAM MEMBERS (GROUP17):

SHUBHAM ARYA (18103070)

KARANVEER (18103107)

OM BINDAL (18103114)

VIPUL ANAND (18103122)

Department of Computer Science and Engineering,
Punjab Engineering College (Deemed to be University), Chandigarh

DECLARATION

We hereby declare that this project work entitled “Cigarette Smoking Detection using Deep Neural Networks” submitted to Punjab Engineering College, Chandigarh, is a record of an original work done by us under the guidance of Professor Poonam Saini and Assistant Professor Arun Singh Pundir, Department of Computer Science and Engineering, Punjab Engineering College, Chandigarh. This project work is submitted in the partial fulfillment of the requirements of the work of the degree, Bachelors in Technology in Computer science and Engineering. We also declare that this project has not been submitted to any other university for the award of any degree.

Date: 19th December, 2022 Signature:

CERTIFICATE

This is to certify that the Major Project entitled, "Cigarette Smoking Detection using Deep Neural Networks" being submitted by Shubham Arya, Karanveer, Om Bindal and Vipul Anand for the award of the degree of Bachelors in Technology in Computer science and Engineering, Punjab Engineering College, Chandigarh, is a record of bonafide research work carried out by them under my supervision and guidance. They have worked for one semester on the above problem at the Department of Computer Science and Engineering, Punjab Engineering College, Chandigarh and this has reached the standard fulfilling the requirements and the regulation relating to the degree. The contents of this thesis, in full or part, have not been submitted to any other university or institution for the award of any degree or diploma.

Date: 19th December, 2022

Prof. Poonam Saini

(Computer Science and Engineering,

Punjab Engineering College, Chandigarh)

Assistant Prof. Arun Singh

(Computer Science and Engineering,

Punjab Engineering College, Chandigarh)

ACKNOWLEDGEMENT

While bringing out this thesis to its final form, we came across a number of people whose contributions in various ways helped our field of research and they deserve special thanks. It is a pleasure to convey our gratitude to all of them.

First and foremost, we would like to express our deep sense of gratitude and indebtedness to our mentors Prof. Poonam Saini and Assistant Arun Singh for their invaluable support, encouragement and suggestions from an early stage of this research and providing us extraordinary experiences throughout the work. Above all, their priceless and meticulous mentoring at each and every phase of work inspired us in innumerable ways.

We specially acknowledge them for their advice, supervision, and the vital contribution as and when required during this research. Their involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come.

We would like to thank our mentors Prof. Poonam Saini and Assistant Arun Singh and Punjab Engineering College, Chandigarh for providing us an opportunity to work on such a knowledgeable project.

Shubham Arya (19103070)

Karanveer (19103107)

Om Bindal (19103114)

Vipul Anand (19103122)

ABSTRACT

Convolutional neural networks (CNNs) have been extensively applied for image recognition problems giving state of-the-art results on recognition, detection, segmentation and retrieval. In this work we propose and evaluate several deep neural network architectures to combine image information across a video over longer time periods. We propose two methods capable of handling full length videos. The first method explores various convolutional temporal feature pooling architectures, examining the various design choices which need to be made when adapting a CNN for this task. The second proposed method explicitly models the video as an ordered sequence of frames. For this purpose we employ a recurrent neural network that uses Long Short-Term Memory (LSTM) cells which are connected to the output of the underlying CNN.

This Project Titled “Cigarette Smoking Detection using Deep Learning Networks” is a research project based on the smoking detection in videos using deep learning models, enhancing them and then using them for validation. We have 2 datasets, UCF-101 and HMDB-51.

UCF101 is an action recognition data set of realistic action videos, collected from YouTube, having 101 action categories. This data set is an extension of UCF50 dataset which has 50 action categories. With 13320 videos from 101 action categories, UCF101 gives the largest diversity in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc, it is the most challenging data set to date. The HMDB-51 dataset contains 6849 clips divided into 51 action categories, each containing a minimum of 101 clips.We have filtered out relevant categories for smoke estimation and detection hence made a custom dataset for training and testing of various models.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS 7

CHAPTER 1: INTRODUCTION 8

CHAPTER 2: BACKGROUND 10

CHAPTER 3: PROPOSED WORK 12

CHAPTER 4: IMPLEMENTATION 14

MILESTONE 1 : Exploring and Understanding the Dataset 15

MILESTONE 2 : Pre-Processing The Datasets 15

MILESTONE 3: Converting the Videos into Frames 15

MILESTONE 4: Implementation of 3-D CNN 16

MILESTONE 5: Apply and Train LSTM Model over Smoke Dataset 17

MILESTONE 6: Applying GoogleNet Architecture 17

MILESTONE 7: Validation of Smoke in Video using TensorFlow 2.2 19

CHAPTER 5: RESULTS AND DISCUSSIONS 21

CHAPTER 6: CONCLUSION AND FUTURE WORK 25

CONCLUSION 26

FUTURE WORK 26

REFERENCES 27

LIST OF FIGURES

Figure 1 Workflow of 3D CNN 16

Figure 2 Workflow of CNN+LSTM 17

Figure 3 Architectural Detail of GoogLeNet 19

Figure 4 EfficientDets Model 20

Figure 5 Different frames of different categories 22

Figure 6 3D-CNN Train Accuracy and Test Accuracy 23

Figure 7 3D-CNN Train Loss and Test Loss 23

Figure 8 LSTM Train Accuracy and Test Accuracy 24

Figure 9 LSTM Train Loss and Validation Loss 24

LIST OF ABBREVIATIONS

1. CNN: Convolutional Neural Network
2. LSTM: Long Short Term Memory
3. UCF: University of Florida
4. HMDB: Human Metabolome Database
5. LDA: Latent Dirichlet Allocation
6. NLP: Natural Language Processing
7. PCA: Principal Component Analysis
8. PDF: Portable Document Format
9. RDF: Resource Description Framework
10. SGD: Stochastic Gradient Descent
11. TF-IDF: Term Frequency–Inverse Document Frequency

1. INTRODUCTION

CHAPTER 1: INTRODUCTION

Smoking in public places not only causes harm to the health of oneself and others, but also has a great safety risk. Many fire incidents are caused by smoking in sensitive areas. Therefore, more and more public places are beginning to detect and control smoking behavior. Airports, high-speed trains, gas stations, flammable and explosive warehouses and other smoke-free areas need to be equipped with equipment that can accurately and efficiently monitor smoking behavior to ensure that firefighters and site management personnel can detect fire hazards in a timely manner.

With the development of science and technology, the detection of smoking has been improved. Traditional detection methods mostly use various smoke detectors, but in open areas such as airports, gas stations and shopping malls, the smoke concentration will decrease rapidly, and the smoke sensing equipment cannot be triggered, so it is difficult to achieve the effect of monitoring and warning. Some researchers have also designed wearable detection devices, but they need to be worn by everyone. The production cost is high and the service life of the devices is short.

In addition to physical detection equipment, image processing technology is also gradually playing an important role in smoking detection. Smoking detection based on video recognition is roughly divided into detection for hand movement which comes under Human Activity Recognition and detection for smoke. Compared with smoke sensors, smoke detection methods can predict smoking in a large range and over a long distance, and the detection effect is much better. However, when the background light is weak and smoke is thin, the detection accuracy might be low.

The cigarette detection method solves the influence of smoke concentration on the detection accuracy, but the detection accuracy is still not ideal due to the small cigarette target in the image captured by the surveillance camera and the overlap of occlusion.

Several countries like Indonesia and Singapore have tried to implement cigarette smoking detection system but have not been able to achieve success in this domain of work. Hence our work will prove to be a good milestone in the development and implementation of smoke detection system, further enhancing and enriching the already existing models and producing state-of-art results.

2. BACKGROUND

CHAPTER 2: BACKGROUND

Smoke detectors play a vital role in the safety and security of spaces such as industrial spaces, public places, residential buildings, and commercial spaces. Installation of smoke detectors in such places helps to avoid casualties and deaths occurring due to fire incidents caused by burning cigarettes. To prevent such fatal accidents, various countries' governments have enforced strict laws and regulations for fire and safety purposes. The smoke detector must be hardwired or battery-powered, and the detector must be in the operating stage when tested.

There are numerous factors and variables that influence the specific smoke alarm needs of a residence, especially the social and economic status of the region in which it is located. The prevailing lifestyle, peoples habits, among other things, must be taken into consideration when recommending or requiring a specific type of smoke alarm for a residence. For example, in many developing countries people are allowed to smoke outside, which is likely to trigger false alarms if a smoke alarm is present. This presents a problem because nuisance (false) alarms, which most often occur, have been shown to be one of the primary reasons for people to disable smoke alarms in public places, rendering them useless. In addition, functionality is often less of a concern to the average buyer than affordability and cost, since most buyers are unaware of the different technologies available.

With the rapid development of smoke detector technology and the loT and big data technologies, the concept of smart smoke detectors has come to the market, leading to the expansion of the smoke detectors market. Smart smoke detectors are programmable and can be easily connected to Wi-Fi networks in homes and commercial spaces. These detectors have features such as smoke detection alerts, easy control through mobile devices. These devices can send notifications or fire alerts through text messages or email and facilitate color-coded alerts in case hard of hearing people are using the detectors. Smart smoke detectors are also capable of self-monitoring.

3. PROPOSED WORK

CHAPTER 3: PROPOSED WORK

Through this research project we propose to implement various deep learning network models to identify cigarette smoking in public places using real-time monitoring through cameras, enhancing the results by implementing a double verification system which includes hand movement recognition as well as smoke detection techniques in videos.

The UCF-101 dataset would be our main base for hand movement detection techniques. Various deep learning networks can be trained out of the categories contained in this dataset, by combining the categories like applying make-up, brushing etc. into non-smoking categories and categorizing these categories to validate the smoking action. Additionally we are using

We perform experiments combining state-of-the-art architectures like LSTM, GoogleNet, AlexNet, ResNet to cross train our dataset and increase the accuracy. Hence, we aim to develop an automatic pipeline based on NLP and IE which can be used for RDF triple generation irrespective of the domain. Further, these hand movement recognition models can be integrated with smoke detection models to validate cigarette smoking in public areas

After developing both the models i.e. human activity recognition for smoking and smoke detection, we aim to publish a comprehensive detail comparing the various methodologies and various cross model implementations.

Resulting would be a vast capability of interlinking different entities from varied domains. The entity alignment would require extensive research as it is a less explored area. Finally we compare the results of our best model against previous state-of-art architectures and the results of the best model from cross-model implementation which performs several layers of 3-D convolutions on our dataset.

4. IMPLEMENTATION DETAILS

CHAPTER 4: IMPLEMENTATION

After gathering the useful information about cigarette-smoking detection using deep networks and thoroughly studying the already done research work in this domain, we wire-framed our project into the following pipeline:

MILESTONE 1 : Exploring and Understanding the Dataset

The very first milestone achieved was to explore the datasets which can be used to classify hand movement. The main aim was to fetch a dataset which is relevant to real-world scenarios, is large enough for a classifier model to be built and trained over it and is well structured. Next task was to understand the structure of the datasets to be able to use the desired information through various attributes. The relevant datasets found for human activity recognition include UCF-101 dataset and HMDB-51 dataset.

MILESTONE 2 : Pre-Processing The Datasets

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. The challenge of extracting relevant information from bodies, was to find the right category of human actions that contains the relevant information out of the provided categories. To solve this challenge we applied pre-processing on the UCF-101 dataset and HMDB-51 dataset. In UCF-101 dataset and HMDB-51 dataset preprocessing we:

1. explored categories relevant for hand movements related to smoking.
2. Identified and grouped the non-smoking categories.
3. This was done to ensure and filter out completely out of context videos related to smoking. The dataset thus filtered out contains a robust size of 706 videos differentiated in 5 categories for initial training of models.

MILESTONE 3 : Converting the Videos into Frames

After getting the required we converted those videos into frames/images so that these could be applied on Neural Network models which work on sequential image data for pattern recognition.

1. Converted Videos into 15 frames/second sequence length.
2. Resized each video frame to specific height and width for further pre-processing.

MILESTONE 4 : Implementation of 3-D CNN

3D convolutions apply a 3 dimensional filter to the dataset and the filter moves 3-direction (x, y, z) to calculate the low level feature representations. Their output shape is a 3 dimensional volume space such as cube or cuboid. They are helpful in event detection in videos, 3D medical images etc. They are not limited to 3D space but can also be applied to 2D space inputs such as images.

Implementation Workflow of Model:

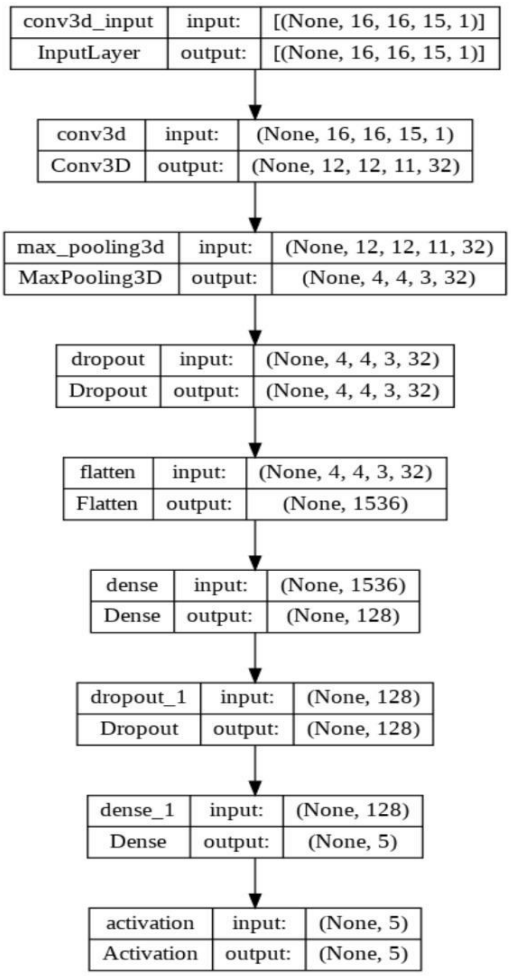

Figure 1 Workflow of 3D CNN

This was done on Colab and the test results were recorded. The model was evaluated on factors like accuracy and loss rate.

MILESTONE 5 : Apply and Train LSTM Model over Smoke Dataset

Long Short Term Memory or LSTM networks are a special kind of RNNs that deal with the long term dependency problem effectively. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn. LSTMs also have this chain-like structure, but the repeating module has a different structure. The repeating module has 4 different neural network layers interacting to deal with the long term dependency problem.

Implementation Workflow of Model:

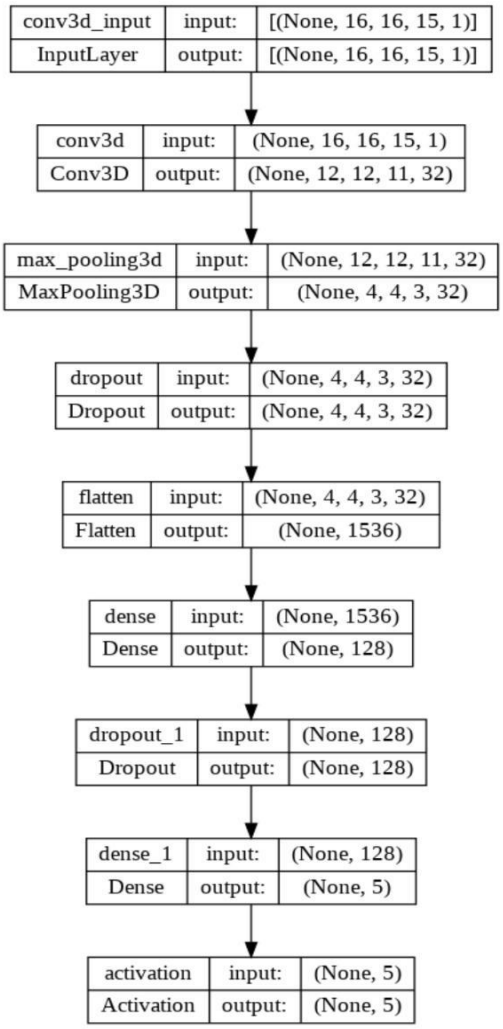

Figure 2 Workflow of CNN+LSTM

MILESTONE 6 : Applying GoogleNet Architecture

Google Net (or Inception V1) was proposed by researchers at Google (with the collaboration of various universities) in 2014 in the research paper titled "**Going Deeper with Convolutions**". This architecture was the winner at the **ILSVRC 2014 image classification challenge**. It has provided a significant decrease in error rate as compared to previous winners AlexNet (Winner of ILSVRC 2012) and ZF-Net (Winner of ILSVRC 2013) and significantly less error rate than VGG (2014 runner up). This architecture uses techniques such as 1×1 convolutions in the middle of the architecture and global average pooling.

The overall architecture is 22 layers deep. The architecture was designed to keep computational efficiency in mind. The idea behind that is that the architecture can be run on individual devices even with low computational resources. The architecture also contains two auxiliary classifier layers connected to the output of Inception (4a) and Inception (4d) layers.

The architectural details of auxiliary classifiers as follows:

- An average pooling layer of filter size 5×5 and stride 3.
- A 1×1 convolution with 128 filters for dimension reduction and ReLU activation.
- A fully connected layer with 1025 outputs and ReLU activation
- Dropout Regularization with dropout ratio = 0.7
- A softmax classifier with 1000 classes output similar to the main softmax classifier.

The figure below depicts the conventional GoogLeNet architecture. Have a quick review of the table before reading more on the table's characteristics and features.

We have implemented the GoogleNet over the dataset that we have built, using the pre-processed data and implementation of the frames from videos, we trained the model and evaluated the model over the test set. The training was done on Google colab. We were not able to start the training process because of computation power limitations.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Figure 3 : Architectural Detail of GoogLeNet

MILESTONE 7 : Validation of Smoke in Video using TensorFlow 2.2

Before implementing the smoke detection directly into the project, we first tried it using Tensorflow Object Detection API. This API helps in object detection by using pre-trained models on our custom datasets, which in our case is for smoke detection in images. Now to train the object detection model, we obviously need to have some images in the dataset. So, for the dataset, we used 713 annotated smoke images. The training, validation and testing dataset is divided in the ratio 7:2:1 i.e. 513 images for training, 147 for validation and 73 images for testing. This dataset was obtained from 'aiformankind/wildfire-smoke-detection-camera' github repository.

We have used Roboflow which is used to label the data, apply image preprocessing, data augmentation, generate TF Records and many other useful techniques in machine learning. We would also like to thank Roboflow for the excellent tutorials.

Now to implement the Tensorflow object detection model, following steps were taken -

1. Install TensorFlow2 Object Detection Dependencies
2. Download Smoke Images Dataset and necessary files

3. Write your own TensorFlow2 Object Detection Training Configuration
4. Train Custom TensorFlow2 Object Detection Model
5. Export Custom TensorFlow2 Object Detection Weights
6. Use Trained TensorFlow2 Object Detector For Inference on Test Images
7. Save your model for future applications

Model :

Our model is trained EfficientDet-D0, which is a state of the art object detection model. You will find EfficientDet useful for real time object detection. EfficientDet has an EfficientNet backbone and a custom detection and classification network. EffcientDet is designed to efficiently scale from the smallest model size. The smallest EfficientDet, EfficientDet-D0 has 4 million weight parameters - which is truly tiny. EfficientDets are developed based on the advanced backbone, a new BiFPN, and a new scaling technique:

- Backbone: we employ EfficientNets as our backbone networks.
- BiFPN: we used BiFPN, a bi-directional feature network enhanced with fast normalization, which enables easy and fast feature fusion.
- Scaling: we use a single compound scaling factor to govern the depth, width, and resolution for all backbone, feature & prediction networks.

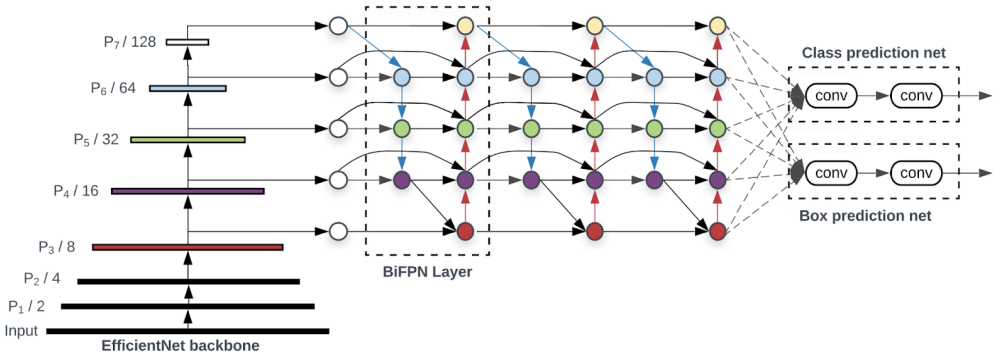

Figure 3: EfficientDets Model

5. RESULT And DISCUSSIONS

CHAPTER 5: RESULTS AND DISCUSSIONS

We created a dataset using UCF101 and HMDB51 which included activities like Applying Makeup, Applying Lipstick, Brushing Teeth, Playing Flute and Smoking, and converted those videos into frames for better classification into different categories and tried implementing different approaches like 3D CNN, LSTM(Long Short Term Memory), LSTM+GoogLeNet.

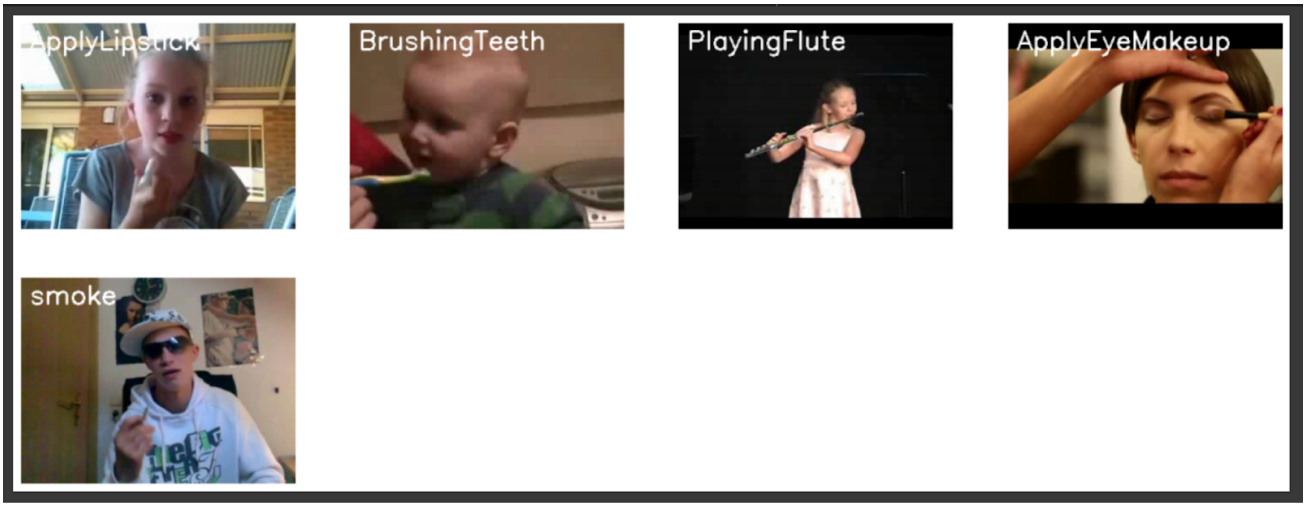

Figure 4: Different frames of different categories

We implemented 3D Convolutional Neural Network, which is a type of deep neural network that is specifically designed to work with image data and excels when it comes to analyzing the images and making predictions on them, where the temporal and spatial information are merged slowly throughout the whole network. We took 5 categories Brushing Teeth, Applying Eye Makeup, Applying Lipstick, Playing Flute and Smoking which comprised of 615 total videos and converted it into 5 categories, relu was used as the activation function and categorical crossentropy was used as the loss function, which had a total of 201,413 trainable parameters and using these we implemented 3D CNN on them and the resulted accuracy is 86.52%, validation accuracy is 80.28%, the resultant loss is 40.10% and validation loss is 76.91%. The following are the results of implementation of 3D CNN on 50 epochs.

3D CNN Model Implementation Results:

When 3D CNN was implemented following results were observed through graphs using tensorboard.

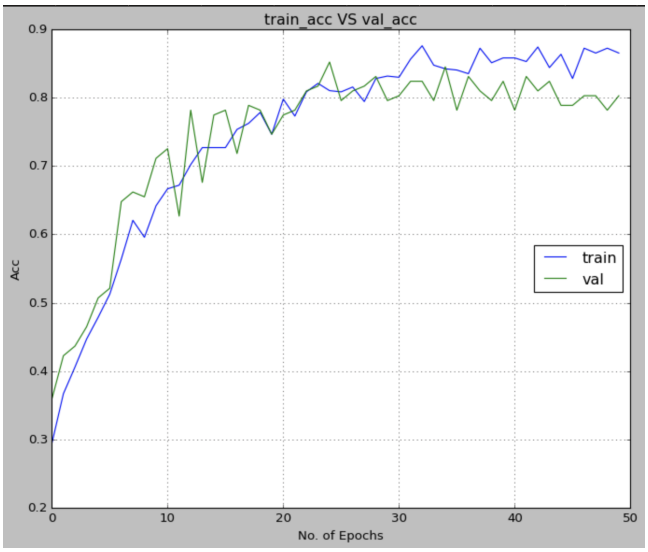

Figure 5: 3D-CNN Train Accuracy and Test Accuracy

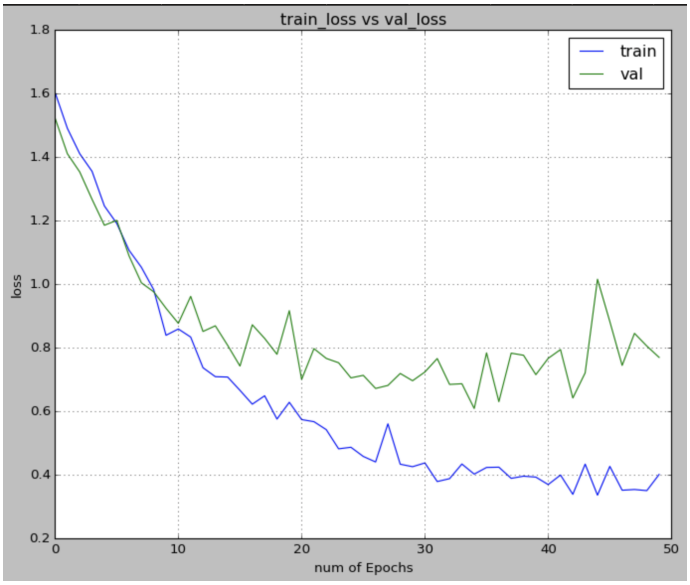

Figure 6: 3-D CNN Train Loss and Test Loss

CNN + LSTM Model Implementation Results:

An LSTM network is specifically designed to work with a data sequence as it takes into consideration all of the previous inputs while generating an output. LSTMs are actually a type of neural network called Recurrent Neural Network. We took 5 categories Applying Eye Makeup, Applying Lipstick, Brushing Teeth, Smoking, Playing Flute which comprised of 615 total videos and converted it into 5 categories, relu was used as the activation function and categorical crossentropy was used as the loss function, which had a total of 43,805 trainable parameters and using these we implemented CNN+LSTM on them and the resulted accuracy is 93.38%, validation accuracy is 78.86%, the resultant loss is 29.82% and validation loss is 67.76%. The following are the results of implementation of CNN+LSTM in 50 epochs.

CNN+LSTM Results Accuracy Graph:

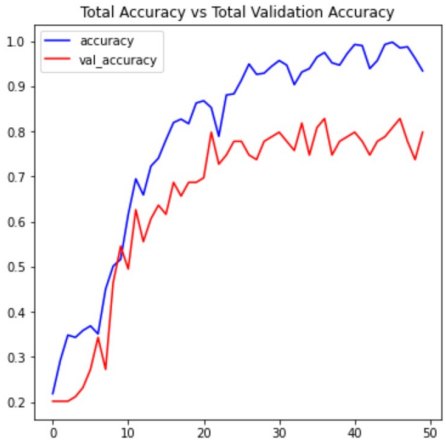

Figure 7: LSTM Train Accuracy and Test Accuracy

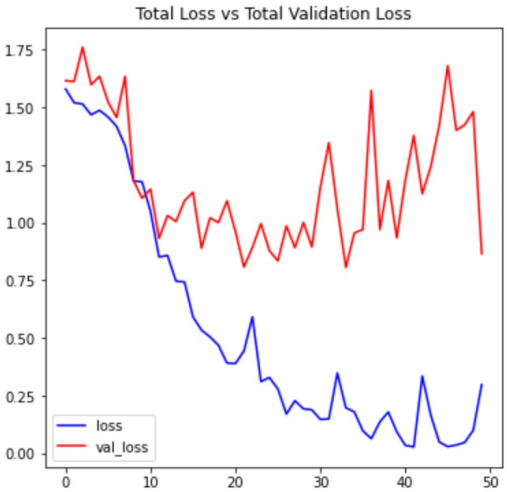

Figure 8: LSTM Train Loss and Validation Loss

6. FUTURE SCOPE

CHAPTER 6: CONCLUSION AND FUTURE WORK

CONCLUSION

The results of models based on deep learning neural networks on UCF-101 and HMDB-51 dataset have been clearly understood and well analyzed. Through this research project we tend to contribute towards a more accurate, enriched, and fast validation system. This model not just detects the hand movements for smoking but also detects the smoke in videos but proves to be a double validation for cigarette smoking detection, hence will prove to be more accurate the existing architectures. but also recommends relevant information related to the query. A trained implementation has been developed which can be used for further cross-model implementation for better accuracy.

FUTURE WORK

1. Implementation of GoogleNet over a large dataset with increased computational power to produce better results and gain insight over the loss rate and the accuracy achieved.
2. Getting more accuracy over smoke detection system with implementation
3. Implementation of other deep learning networks like ResNet, MobileNet over the dataset to produce results and compare it with the existing architectures.
4. Experiments to perform Cross-model Implementation over the dataset so as to get better accuracy and less loss rate.

7. REFERENCES

REFERENCES

Wang, X., He, X., Cao, Y., Liu, M. and Chua, T.S., 2019, July. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 950-958).

Dessì, D., Osborne, F., Recupero, D.R., Buscaldi, D., Motta, E. and Sack, H., 2020, November. Ai-kg: an automatically generated knowledge graph of artificial intelligence. In International Semantic Web Conference (pp. 127-143). Springer, Cham.

Wang, Q., Mao, Z., Wang, B. and Guo, L., 2017. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering, 29(12), pp.2724-2743.

Sun, Z., Hu, W., Zhang, Q. and Qu, Y., 2018, July. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In IJCAI (Vol. 18, pp. 4396-4402).

Wang, Z., Zhang, J., Feng, J. and Chen, Z., 2014, June. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 28, No. 1).

Xu, K., Wang, L., Yu, M., Feng, Y., Song, Y., Wang, Z. and Yu, D., 2019. Cross-lingual knowledge graph alignment via graph matching neural network. arXiv preprint arXiv:1905.11605.

Iglesias, E., Jozashoori, S., Chaves-Fraga, D., Collarana, D. and Vidal, M.E., 2020, October. SDM-RDFizer: An RML interpreter for the efficient creation of rdf knowledge graphs. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (pp. 3039-3046).

Wadden, D., Wennberg, U., Luan, Y., & Hajishirzi, H. (2019). Entity, Relation, and Event Extraction with Contextualized Span Representations. *ArXiv, abs/1909.03546*.