**Abstract:**

With the development of Internet technology and the improvement of network quality, online videos have become increasingly popular. In particular, online live broadcast has become a hotspot in recent years, and smoking behavior in these broadcasts is harmful to smokers and the surrounding environment. Therefore, it is necessary to detect and thereby effectively control smoking behaviors in video content. Traditionally, smoking images are detected based on the detection algorithms of cigarette smoke. Given the limited resolution of live broadcast videos, cigarette smoke is not visually apparent in the video content. This paper proposes a smoking image detection model based on a convolutional neural network, referred to as SmokingNet, which automatically detects smoking behaviors in video content through images. This method can detect smoking images by utilizing only the information of human smoking gestures and cigarette image characteristics without requiring the detection of cigarette smoke, showing high accuracy and superior performance for real-time monitoring.

**Existing System:**

Online live broadcast has become a hotspot in recent years. Social apps such as Twitter and Facebook and mobile personal livecast (MPL) services have emerged and received much attention. With such social apps as Periscope and Facebook Live in the U.S. and Inke1 in China, numerous geo-distributed amateur broadcasters can broadcast their video content live to viewers around the world. It is well known that smoking behaviors are harmful to both smokers and the surrounding environment. Therefore, it is necessary to use images to automatically detect whether there are smoking behaviors in video content.

In recent years, some researchers have proposed smoking image detection methods based on image recognition technology. Inoue et al. assigned eigenvectors to low dimensional spaces using subspace theory, and thereafter used feature clustering to classify cigarette smoke. Although this method can achieve smoking identification through smoke classification, the threshold in the algorithm is empirically set and its value will change with the background, leading to high false detection rates and poor applicability

**Proposed System:**

The CNNs in deep learning have been widely used in image detection. The features to be extracted through CNNs for image recognition no longer need to be defined manually, and the feature extraction is achieved via automatic fitting through training. Each convolution operation can be regarded as a process of feature extraction, in which the weights of the convolution kernels are not preset but are continuously updated through training until the model converges, when the weights constitute the optimal feature extraction scheme. SmokingNet, a detection model based on CNNs, optimizes the characteristics of smoking images based on GoogLeNet and enhances the ability of feature extraction of the target images using non-square convolution kernels. This model is pre-trained with a super-large data set similar to target images prior to model training, and the trained model is used to detect smoking images.

Convolutional neural networks (CNNs) are a deep learning model. Here, “deep” indicates that, compared with shallow learning models, deep learning models involve neural networks with more hidden layers, and thus, the neural networks used for deep learning are called deep neural networks (DNNs). With the deepening of the research and the progress of computer hardware conditions, the number of layers of the deep learning models has increased from the initial value of 6 to more than 100 nowadays In this study, we design a CNN-based model called SmokingNet, which can automatically detect smoking behaviors in video content through images. Based on GoogLeNet, the model optimizes the characteristics of smoking images. With nonsquare convolution kernels, the model enhances the ability of feature extraction of the target images. Before model training, a super-large data set similar to the target image is used for pre-training the model. When the trained model is used to detect smoking images in the system, the full connection layers in the model are converted into convolution layers, which improves the detection ability of the model for local small targets while maintaining considerable detection efficiency.

**HARDWARE REQUIREMENTS:**

* Processor : Intel i3 and above
* RAM : 4GB and Higher
* Hard Disk : 500GB: Minimum

**SOFTWARE REQUIREMENTS:**

* Programming Language / Platform : Python
* IDE : pycharm/jupyter

**Modules**

**Image Acquisition**

The first step of the Smoking Detection system is image acquisition. High-quality Human Smoking images need to collection from public places.

The entire sample set is divided into three parts: training samples and validation samples in the training phase and testing samples in the testing phase. Moreover, the sample set is divided into positive and negative samples—a positive sample is an image showing smoking behaviors, whereas a negative sample is a background image.

**Annotated Dataset Collection**

A Knowledge-based dataset is created by proper labeling of the collected images with unique classes.

**Image Processing**

The obtained images that will be engaged in a preprocessing step are further enhanced specifically for image features during processing. The segmentation process divides the images into several segments and utilized in the extraction of Smoking features from dataset.

**Feature-Extraction**

This section involves the convolutionary layers that obtain image features from the resize images and is also joined after each convolution with the ReLU. Max and average pooling of the feature extraction decreases the size. Ultimately, both the convolutional and the pooling layers act as purifiers to generate those image characteristics.

**Classification**

The final step is to classify images, to train deep learning models along with the labeled images to be trained on how to recognize and classify images according to learned visual patterns. The authors used an open-source implementation via the TensorFlow module, using Python and OpenCV including the VGG-16 CNN model.