Neural Computing and Applications (2022) 34:21481–21501
https://doi.org/10.1007/s00521-021-06003-9

S.I. : NCACVIP

Understanding cartoon emotion using integrated deep neural network on large dataset

Nikita Jain[1] · Vedika Gupta[1] · Shubham Shubham[1] · Agam Madan[1] · Ankit Chaudhary[1] · K. C. Santosh[2]

Received: 25 September 2020 / Accepted: 31 March 2021 / Published online: 21 April 2021

Abstract

Emotion is an instinctive or intuitive feeling as distinguished from reasoning or knowledge. It varies over time, since it is a natural instinctive state of mind deriving from one's circumstances, mood, or relationships with others. Since emotions vary over time, it is important to understand and analyze them appropriately. Existing works have mostly focused well on recognizing basic emotions from human faces. However, the emotion recognition from cartoon images has not been extensively covered. Therefore, in this paper, we present an integrated Deep Neural Network (DNN) approach that deals with recognizing emotions from cartoon images. Since state-of-works do not have large amount of data, we collected a dataset of size 8 K from two cartoon characters: *'Tom'* & *'Jerry'* with four different emotions, namely happy, sad, angry, and surprise. The proposed integrated DNN approach, trained on a large dataset consisting of animations for both the characters (*Tom* and *Jerry*), correctly identifies the character, segments their face masks, and recognizes the consequent emotions with an accuracy score of 0.96. The approach utilizes Mask R-CNN for character detection and state-of-the-art deep learning models, namely ResNet-50, MobileNetV2, InceptionV3, and VGG 16 for emotion classification. In our study, to classify emotions, VGG 16 outperforms others with an accuracy of 96% and F1 score of 0.85. The proposed integrated DNN outperforms the state-of-the-art approaches.

Keywords Animation · Cartoon · Character Detection · Convolutional Neural Network · Emotion · Face Segmentation · Mask R-CNN · VGG16

1 Introduction

An emotion is a physiological state of mind that is subjective and is constituted by associated thoughts, feelings, and behavioral responses that are homogeneous. Recognizing emotions is highly usable in artificially intelligent systems, to enable such systems in recognizing and predicting human emotions to enhance productivity and effectiveness of working with computers. Recently, automatic recognition of emotion has become a popular research field involving researchers from the industry, as well as academia, specializing in artificial consciousness, computer vision, brain computing, physiology, and, more recently, deep learning. Its ubiquity emerges from extensible regions of potential applications. Ekman and Friesen [1] has conducted one popular research work in the field of Emotion Recognition, who have classified the emotions into six basic expressions of happiness, sadness, disgust, anger, and fear and stated they are universal. Their research

To the best of our knowledge, these are the largest data for cartoon emotion classification and are available for research purpose.

✉ K. C. Santosh
santosh.kc@usd.edu

Nikita Jain
nikita.jain@bharatividyapeeth.edu

Vedika Gupta
vedika.gupta@bharatividyapeeth.edu

Shubham Shubham
shbhm3199@gmail.com

Agam Madan
madanagam@gmail.com

Ankit Chaudhary
ankitchaudhary3010@gmail.com

[1] Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, Dehi, India

[2] KC's PAMI Research Lab - Computer Science, University of South Dakota, 414 E Clark St, Vermillion, SD 57069, USA

has become a benchmark for the evaluation of studies conducted in the field of emotion detection.

Recognizing emotions is gaining thrust since the past few years. It can be performed by analyzing data in any medium: text [2–5], audio or speech [6–9], video, or images [10–13]. The research in the field of emotion recognition has provided valuable data giving the emotional state of patients [14], response to an advertisement, and even in times of crisis like coronavirus [15]. Emotion recognition from images generally involves facial expressions or gestures. Facial emotions are a form of non-verbal communication that conveys both the emotional state and behavioral intentions of an individual. The task of recognizing such emotions can be performed over the facial image data of human beings or animals or any real-world entity. There are several applications of this task in a variety of fields ranging from medicine [16–18] and e-learning [19–22] to entertainment [23] and marketing [24–26] and even judiciary [27].

The application of facial emotion recognition is not limited to reading human face emotions. It can also be implemented to detect the emotions of animated characters or cartoons. Cartoons are mostly made, keeping in mind the entertainment and suitability of viewers (especially children). They are often filled with various kinds of emotions that are portrayed in multiple forms by the same character.

The motivation behind current research lies in the fact that there are plenty of emotions portrayed in cartoons, even by the same character and animated cartoons provide an opportunity, where one can extract emotions from these characters (from one or more videos). This idea of identifying emotions is useful in cases where parents or guardians often want to choose a category of the cartoon (sci-fi, comic, humor, mystery, and horror) based on their child's interest or according to their suitability.

To identify human faces, an image can be segmented using the OpenCV library[1] [28]. However, the utilized algorithm misses detecting any other real-world entity [29] (cartoon, in this paper). Therefore, we propose a generic approach that uses a popular method Mask R-CNN to efficiently segment objects. Furthermore, it was improbable to find an existing dataset online owing to the time-consuming nature of data preparation from videos in this task, encompassing character identification, as well as emotion identification. Therefore, it is in this regard that a dataset is built (with two cartoon characters—*Tom* & *Jerry*, currently) that can even be utilized in different applications if publicly released on the web. Moreover, the dataset is extensible for any number of cartoon characters keeping the generic approach/implementation same.

[1] https://opencv.org/.

The current work deals with recognizing emotions from facial expressions of cartoon characters. The objective is to find out if DNNs can be deployed to extract and recognize emotions from cartoons. Even though emotion recognition has been extensively performed over human facial images; yet, recognizing emotions from cartoon images is still an under-explored area. To handle the same, a novel integrated DNN approach has been developed to identify emotions from cartoon characters wherein the faces of characters are segmented into masks using the Mask R-CNN technique. These generated masks are further used as input to the emotion recognition model to recognize the emotions of the character. For the analysis conducted in this paper, two cartoon characters—*Tom* and *Jerry*—have been taken into account. The recognized emotions fall into the following categories (Sad, Happy, Angry, and Surprise). The deployed approach gives an F-score of 0.85 when implemented on a created dataset of two characters used here viz. *Tom* and *Jerry*. The proposed approach is generic and scalable to recognize emotions.

The rest of the paper has been organized as follows: Sect. 2 puts forward the existing literature and datasets for emotion recognition from animated images, including cartoons. Section 3 presents the outline of the contributions done in this paper. Section 4 discusses materials used in this work: dataset collection, preparation and the deep neural networks in detail. Section 5 presents the methodology of the proposed work. Section 6 shows the experimental analysis, results and comparison with the state of the art. Section 7 concludes the paper with an explanation of the obtained results and their evaluation with the baseline methods with further scope of enhancement.

2 Related works

Existing research on emotion recognition from facial images is extensive. Facial expressions give accurate information allowing the viewer to differentiate between various negative and positive emotions; however, the evidence relates to posed emotions only [30].

For emotion recognition, one of the works that has been conducted by [31] where the dataset prepared from several photo sites such as *Flickr, Tumblr, and Twitter* has been classified into five emotions viz. Love, Happiness, Violence, Fear, and Sadness. The authors have tested various pre-trained Convolutional Neural Network (CNN) models like VGG-Image Net, VGG-Places205, and ResNet-50, out of which ResNet-50 has performed the best, giving an accuracy of 73% after fine-tuning. Another contribution by [32] where human emotions learned from 2D images have been transferred to animated cartoon 3D animated character for further classification into seven emotions (joy,

sad, anger, fear, disgust, surprise, and neutral). The authors have proposed a fused CNN architecture (f-CNN), giving a total recognition rate of 75.5%, having initially a CNN trained on human expression dataset followed by a transfer learning-based classification to analyze the relationship between emotion transfer from 2D human images to 3D cartoon images. The recognition rate here is a parameter that shows how well an animated character can simulate a human face emotion.

Based on the emotion transfer concept, as mentioned in the previous paragraph, authors in [33] have proposed a human animated face emotion classification where human expressions have been simulated using an animated face. The experiments have been performed on a dataset of 50,000 annotated (cartoonish) face images of several human stylized characters. Using a modified CNN architecture, the experiments obtained different expression recognition for all emotions mentioned (as compared in Section: Conclusion). However, a recent article [34, 35] has argued that recognizing facial emotion of specific cartoon characters adds another challenge to detect emotion since cartoons usually depict extreme levels of emotions that are not otherwise seen and captured from human faces. The authors have also shown through interview-based experiments that specific cartoon face emotion recognition requires a higher processing intensity and speed than real faces during the early processing stage.

The author in [29] gives another pioneer contribution specific to cartoon character emotion recognition using Haar Cascade [28] and modified CNN architecture for character detection and emotion classification, respectively, from cartoon movie videos. Classifying three emotions (Happy, Angry, and Surprise), the experiments achieved a classification accuracy of 80%. However, the authors claim that an improvement in accuracy can be achieved by transfer learning and hence proposes it as an open problem, thereby contributing a public dataset of 1600 emotion labeled cartoon character images.

Recently, several datasets [36, 37] have been contributed intended for experimentation of cartoon face detection to the state of the art. These datasets, however, only contain the character labels and do not provide any information about emotion labels for those characters. Hence, contributing a dataset having emotion labeled cartoon faces becomes another significant contribution of this work. Such a dataset can be used for prospective research if the dataset is released for the public. Applications in emotion recognition, as [38] points out, include avoidance, alerting, production, tutoring, and entertainment. The focus of this article is to address the applications of training and entertainment. This can allow a computer to recognize emotion in an animated cartoon automatically. It could generate subtitles (text or audio) explaining and teaching the emotions of the characters throughout the video to children. The latter relies on the fact that cartoons are a form of entertainment, to adults and especially children. For example, a recommendation system can be designed where an animated cartoon has an emotion rating outlining which characters possess various emotions in an episode.

Although there has been much work over human facial emotion recognition [39, 40], existing literature on emotion recognition from cartoons is limited and has scope for extensive work. The mentioned contributions do not provide any cartoon emotion labeled dataset (obtained from cartoon videos) and instead propose specific human facial expressions simulated animated data. However, contouring of features in non-human faces (such as cartoon characters) is different from human faces, which requires specific detection methods. Also, the existing character detection algorithms that enable efficient emotion classification mainly rely on default libraries and modified CNN architectures helping in feature extraction. Such methods miss detecting a real-world entity (here a specific cartoon face) thereby giving low emotion recognition accuracy (ref. Section: Results). The major contributions of this paper are presented in Sect. 3 as follows.

3 Contribution outline

Integrating DNN and validating it on a fairly (with respect to state-of-the-art works) large amount of data to understand and analyze emotions are the primary aim of the study. This allows us to have multiple objectives:

(a) The proposed integrated DNN includes Mask R-CNN for cartoon character detection and well-known deep learning architectures/models, namely VGG16, InceptionV3, ResNet-50, and MobileNetV2 for emotion classification. Compared to the state-of-the-art works, the use of Mask R-CNN makes a difference in terms of performance (ref. Section Results). Further, employing multiple deep learning architectures/models provides a fair comparison among them.

(b) As no state-of-the-art works provide large amount of data for validation, we created a dataset[2] of 8,113 images and annotated them for emotion classification. This brings us a solution to quantify/test DNN appropriately and is available for research purpose (upon request).

[2] https://github.com/TheSSJ2612/Cartoon_Emotion_Dataset/releases.

The proposed approach, as depicted in Fig. 1, works by collecting and preparing dataset by downloading videos from a popular YouTube channel (ref. Sect. 4.1) followed by character detection and consequent emotion classification through an integrated DNN.

4 Materials

In this paper, a custom dataset consisting of the images extracted from *Tom* and *Jerry* episodes was created. The images needed pre-labeled emotion classes for supervised classification. The labeling process is defined in Sect. 4.1. Currently, there is no active dataset in this regard available. To correctly recognize cartoon emotion from a given input video or image frame, the first and the foremost step includes character detection followed by accurate face segmentation procedures.

4.1 Dataset collection and preparation

The episodes for extracting the images have been identified and then downloaded from the Jonni Valentayn channel[3] on YouTube using a downloading tool named Videoder[4] in MP4 format. The frames were obtained at a ratio of 1:15 in the JPG format using OpenCV from the downloaded videos. From the frames extracted, a training dataset has been generated for the Mask R-CNN model, which is later used to classify and segment *Tom* & *Jerry's* faces from the input frames. Regarding the dataset prepared for training the Mask R-CNN model, the frames extracted were classified into two classes, *Tom* and *Jerry*. Each frame has an associated class, which specifies that the cartoon character's face is present in the frame. A frame can have *Tom's* face, *Jerry's* face, or both.

For preprocessing, each data frame is augmented with a JSON file that stores the frame name, cartoon character name, and the X–Y coordinates of the corresponding cartoon face. The X–Y coordinates of the face were marked using a labeling tool, i.e., VGG Image Annotator (VIA).[5] The marked regions were *Tom's* face and *Jerry's* face. The Mask R-CNN model learns this Region of Interests through the X–Y coordinates of cartoon character's faces marked through VIA tools. Frames with unknown faces were left unmarked. Figure 2 shows the screenshot of the output given by the VGG annotator tool via where Tom's face is marked.

The dataset consists of 10 k images, i.e., 28 animations of Tom and Jerry. The dataset generated from the Mask R-CNN model is annotated into four emotions. These emotions are Happy, Angry, Sad, and Surprise. For both cartoon characters, around 1000 images depicting each out of the four emotions were manually segregated. In total, 8113 images (or masks) of size 256 × 256 were used for the purpose of training after discarding the poor-quality images.

4.2 Mask Annotation for obtaining emotion labels

Three independent annotators were employed to annotate the masked faces manually. These annotators are skilled and well versed in the field of animation and multimedia.

The annotators were asked to interpret each masked face with one of the identified labels in the dataset. The quality of annotation was measured by two standard agreement parameters: Inter-Indexer Consistency (IIC) [41] and Cohen's Kappa [42], with values of 92.06 and 84.9%, respectively. The emotions of both the cartoon character masked faces, namely *Tom* and *Jerry,* were annotated and then labeled in the following order of Sad, Happy, Angry, and Surprise, respectively, in which the emotions of *Jerry* have recorded first and then emotions of *Tom* next in a similar manner. The masked face images with their respective labels so obtained after annotation (as described above) were used to train the emotion recognition model. Figure 3 shows the distribution of emotion labels used for training purposes. A fair distribution of emotions was used to have balanced supervised classification training. The further process to obtain the masked faces from the character images (as prepared) is explained in Sect. 4.3.

4.3 Basics of CNN

This section provides an explanation on the working of CNN along with the other fundamental concepts used.

(i) *Convolutional Operation* The main reason for performing convolution is feature extraction. Applying convolutional operation on an image of size $H \times W \times D$, where H, W, D represent height, width, depth of image, respectively, with convolution filter of size $f \times f \times D$, taking stride size equal to s with padding similar to p gives an output of size $(\frac{H-f+2p}{s}+1) \times (\frac{W-f-2p}{s}+1) \times n_f$ where n_f is the number of filters. The operation of convolution between image matrix P of size (I, J) and kernel matrix Q of size (M, N) gives the output O given by:

[3] https://www.youtube.com/user/TomAndJerryWarner.

[4] https://www.videoder.com/.

[5] http://www.robots.ox.ac.uk/∼vgg/software/via/.

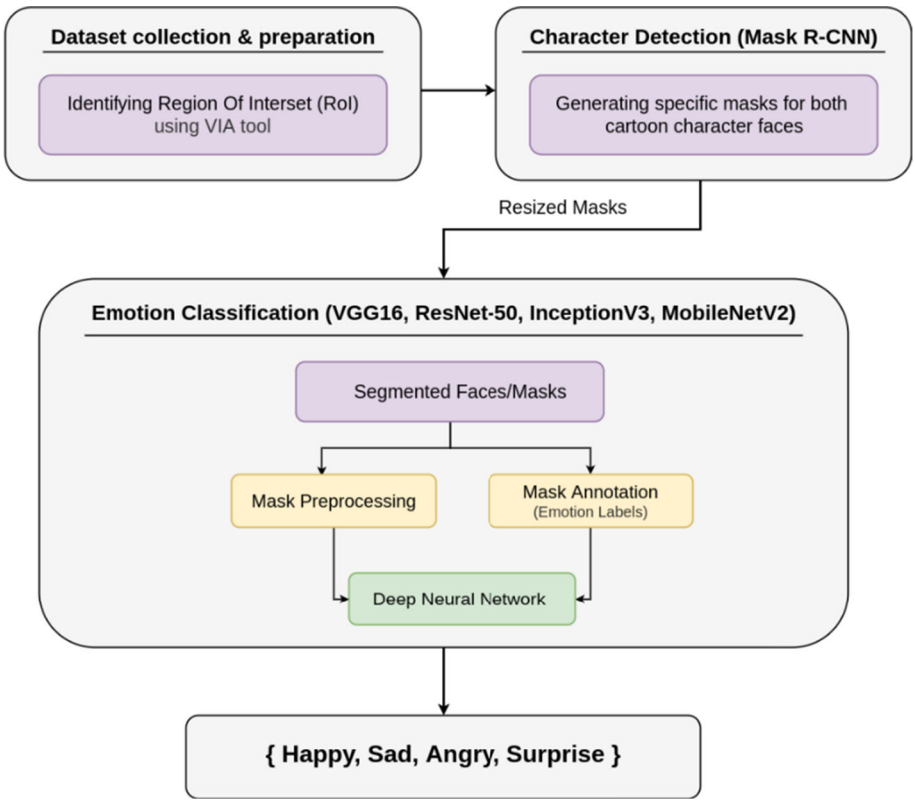


Fig. 1 Workflow of the proposed approach

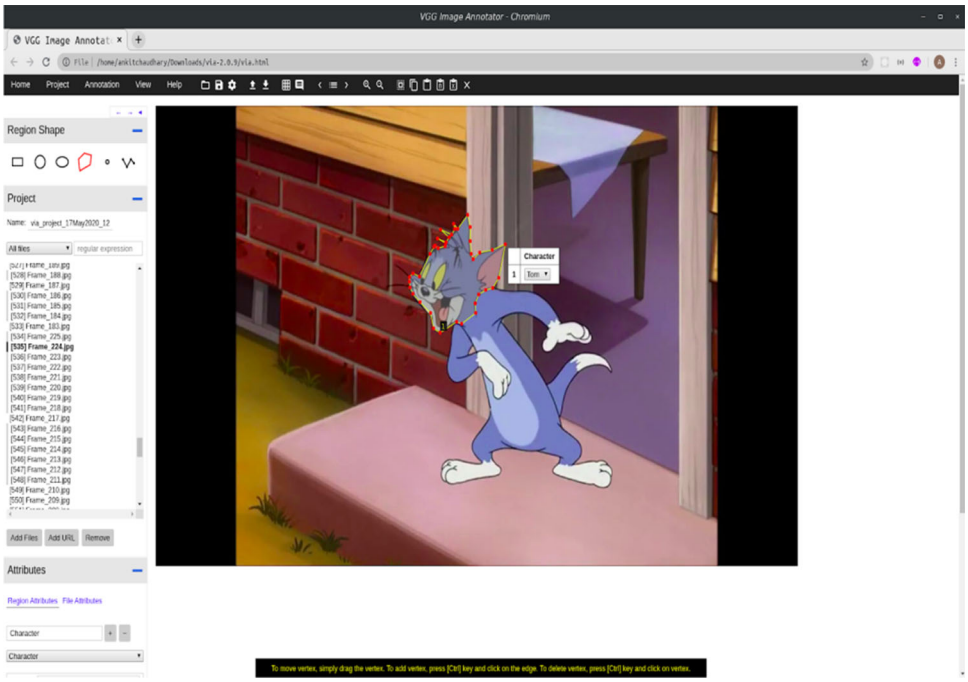


Fig. 2 Screenshot of the VGG Image Annotator

$$O(a,b) = (P * Q)[a,b] = \sum_{i=0}^{I}\sum_{j=0}^{J} P(i,j) * Q(a-i,b-j) \quad (1)$$

where $0 \le a \le I + M + 1$ and $0 \le b \le J + N + 1$.

(ii) *Pooling Operation* Along with convolutions layers, CNN's use pooling layers like max-pooling and average pooling. Thus, for an image of size $H \times W$ with a filter size of k and stride size s, the size of the output is $\left(\frac{H-k}{s} + 1\right) \times \left(\frac{W-k}{s} + 1\right)$.

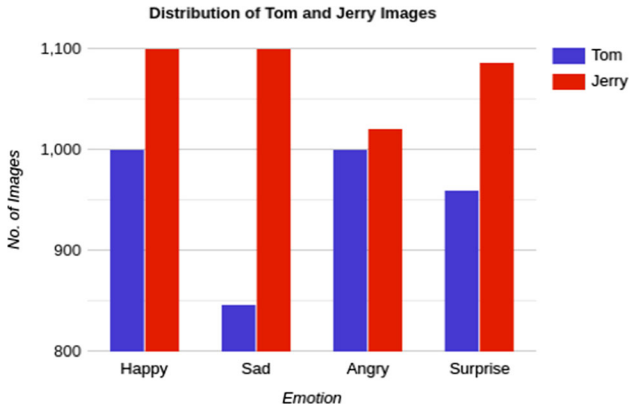


Fig. 3 Distribution of emotion labels after annotation, for the dataset

(iii) *Classification* For the classification, the fully connected layers serve as classifiers on top of the extracted features given by hidden layers. This layer then generates the final probabilities for determining the class for the input image, after applying the weights over the output generated by feature detectors.

(iv) *Dropout Regularization* This technique refers to dropping out neurons (both hidden and visible) in a neural network randomly.

(v) *Activation Function* ReLU is the most regularly used nonlinear function since it provides better performance than its alternatives. Some of the commonly used activation functions are mathematically expressed as follows:

$$\text{Simple ReLU} : f(x) = \max(0, x) \tag{2}$$

$$\text{Leaky ReLU} : f(x) = \begin{cases} x & x \geq 0 \\ 0.01x & \text{otherwise} \end{cases} \tag{3}$$

$$\text{Parameteric ReLU} : f(\alpha, x) = \begin{cases} \alpha x \text{ for } x < 0 \\ x \text{ for } x \geq 0 \end{cases} \tag{4}$$

4.3.1 Mask R-CNN

Mask R-CNN is an extension of Faster R-CNN. Faster R-CNN gives two outputs for every object—a class label and the bounding box coordinates. In Mask R-CNN, a third branch for the output of the object mask is added, which enables it to perform image instance segmentation. The third added branch also shows the prediction of the object mask in parallel with existing branches performing classification and localization. For appropriate instance segmentation, pixel-level segmentation is performed, which requires precise alignment as compared to just the boundary boxes. Hence, Mask R-CNN uses RoI (Region of Interest) pooling layer known as RoIAlign Layer, so much more precise regions can be mapped for segmentation. The backbone of Mask R-CNN is a standard convolutional neural network (like ResNet-50 or Resnet-101), and it helps in the extraction of features. These early layers detect features like edges and corners. After passing through the backbone network, the image is transformed to a $32 \times 32 \times 2048$ feature map from the given image. Figure 4 is the visual model for the architecture of Mask R-CNN. The Region Proposed Network (RPN) is a lightweight neural network that checks the image in a sliding door fashion to find the regions that contain the objects. RPN scans over these regions (also known as anchors) using the backbone feature map instead of directly scanning over the image, enabling it to run faster and more efficiently. Consequently, it avoids duplicate calculations by reusing extracted features. The use of RoIAlign Layer fixes the location misalignment caused due to quantization in the case of RoIPool used in Faster R-CNN. Figure 5 shows the use of spatial transformer and bilinear sampling kernel. Bilinear interpolation is used to compute the exact floating-point location values of input features at four regular sampled locations in each RoI bin and then aggregates the result. Figure 6 shows the use of spatial transformer and bilinear sampling kernel in RoIAlign operation. The following output, i.e., Target feature value at location i in channel c, is obtained from the use of a sampling kernel from the sampler:

$$Q_i^c = \sum_b^H \sum_a^W P_{ba}^c k\left(x_i^s - a; z_x\right) k\left(y_i^s - b; z_y\right) \forall i \in [1 \ldots H' W'] \forall c \in [1 \ldots C] \tag{5}$$

where x_i^s and y_i^s are sampling coordinates at location i.

After applying the bilinear sampling kernel to the above output, the equation transforms to:

$$Q_i^c = \sum_b^H \sum_a^W P_{ba}^c \max\left(0, 1 - \left|x_i^s - a\right|\right) \max\left(0.1 - \left|y_i^s - b\right|\right) \tag{6}$$

where the bilinear sampling kernel copies the value at the nearest pixel to (x_i^s, y_i^s) to the output location(x_i^o, y_i^o). The objective multitasking function, which includes classification lossL_{cls}, bounding box location loss L_{bbox}, and the segmentation loss for mask L_{mask} is defined. The loss equation can represent this:

$$L = L_{cls} + L_{bbox} + L_{mask} \tag{7}$$

or it can also be written as:

$$L(c, g, b^g, z) = L_{cls}(c, g) + \tau[g \geq 1] L_{loc}(b^g, z) \tag{8}$$

where c is the predicted class, g is GT (ground truth) class, b^g is predicted bounding box for class g, z is the GT bounding box, the classification loss is be defined as:

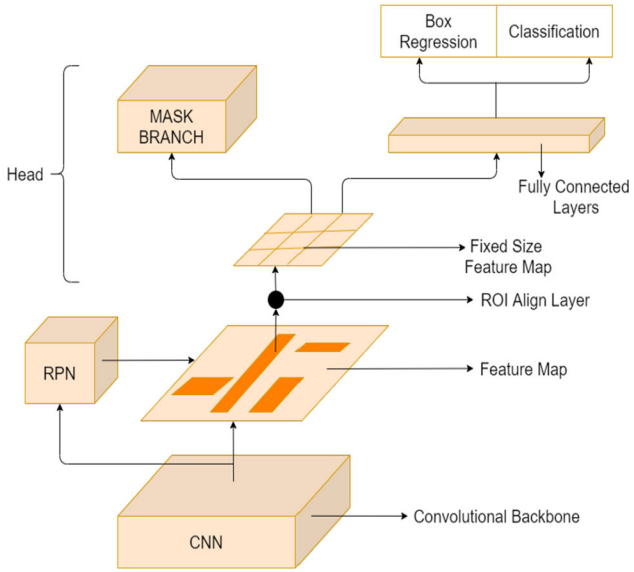


Fig. 4 Architecture of Mask R-CNN

$$L_{cls}(c, g) = -logc_g \tag{9}$$

$$L_{loc}(b^g, z) = \sum_{i \in (x,y,w,h)} smooth_{L1}(b_i^g - z_i) \tag{10}$$

$$smooth_{L1}(a) = \begin{cases} 0.5(a^2) & if|a| < 1 \\ |a| - 0.5 & \text{otherwise} \end{cases} \tag{11}$$

L_{mask} is the mean binary cross-entropy $k \bullet (a \times a)$ the sigmoid output helps in pixel-wise binary classification and allows one mask for each class and hence eliminates competition.

This definition of L_{mask} allows Mask R-CNN to generate masks for every class without competition between the classes; only the classification branch is used to predict the class label of the output mask. This process disconnects mask and class prediction. Since the considered case includes a per-pixel sigmoid and a binary loss, the masks across classes do not compete and hence provide good instance segmentation results.

4.3.2 Popular DNNs

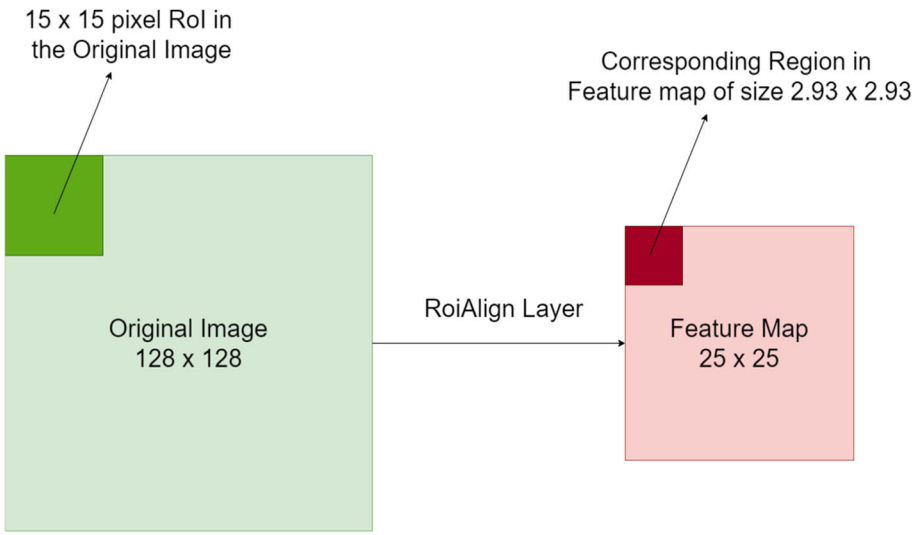
(a) VGG16- VGG16 is a deep convolutional neural network which is 16 layers deep as its name suggests. It is trained on ImageNet database and takes an input of size 224 × 224. An image is passed through a group of convolutional layers with small receptive field, i.e., it uses 3 × 3 kernel with stride size 1. It uses three fully connected layers after the convolutional layers to perform classification. Figure 7 shows the complete architecture of VGG16 with all the layers.

(b) InceptionV3- InceptionV3 is made with computational power efficiency in mind and thus less number of parameters are generated. It uses techniques like factorized convolutions, smaller convolutions layers, parallel computations, reduction in dimensions, and regularization to make the network more efficient. This network is 48 layers deep.

(c) ResNet-50- The full form of ResNet is Residual Networks. Instead of relying on depth of network to learn more features, the residual networks try to lean features from the residual of previous layer which helps in improving accuracy and also helps to solve the problem of vanishing gradient. ResNet has many variants out of which ResNet-50 has been used which is 50 layers deep.

(d) MobileNetV2- MobileNetV2 further improves on MobileNetV1 by using lightweight depth-wise separable convolutions along with linear bottlenecks and

Fig. 5 RoIAlign Layer



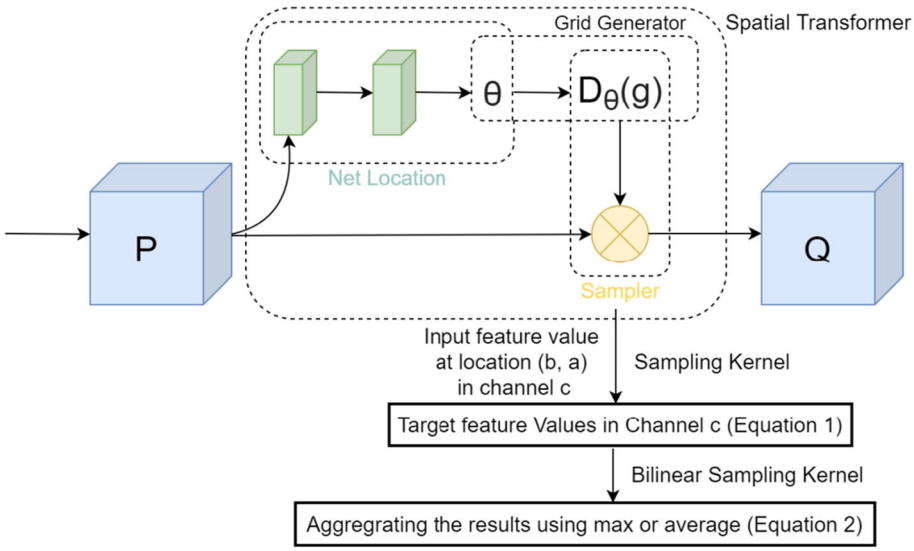


Fig. 6 Use of spatial transformer and bilinear sampling kernel

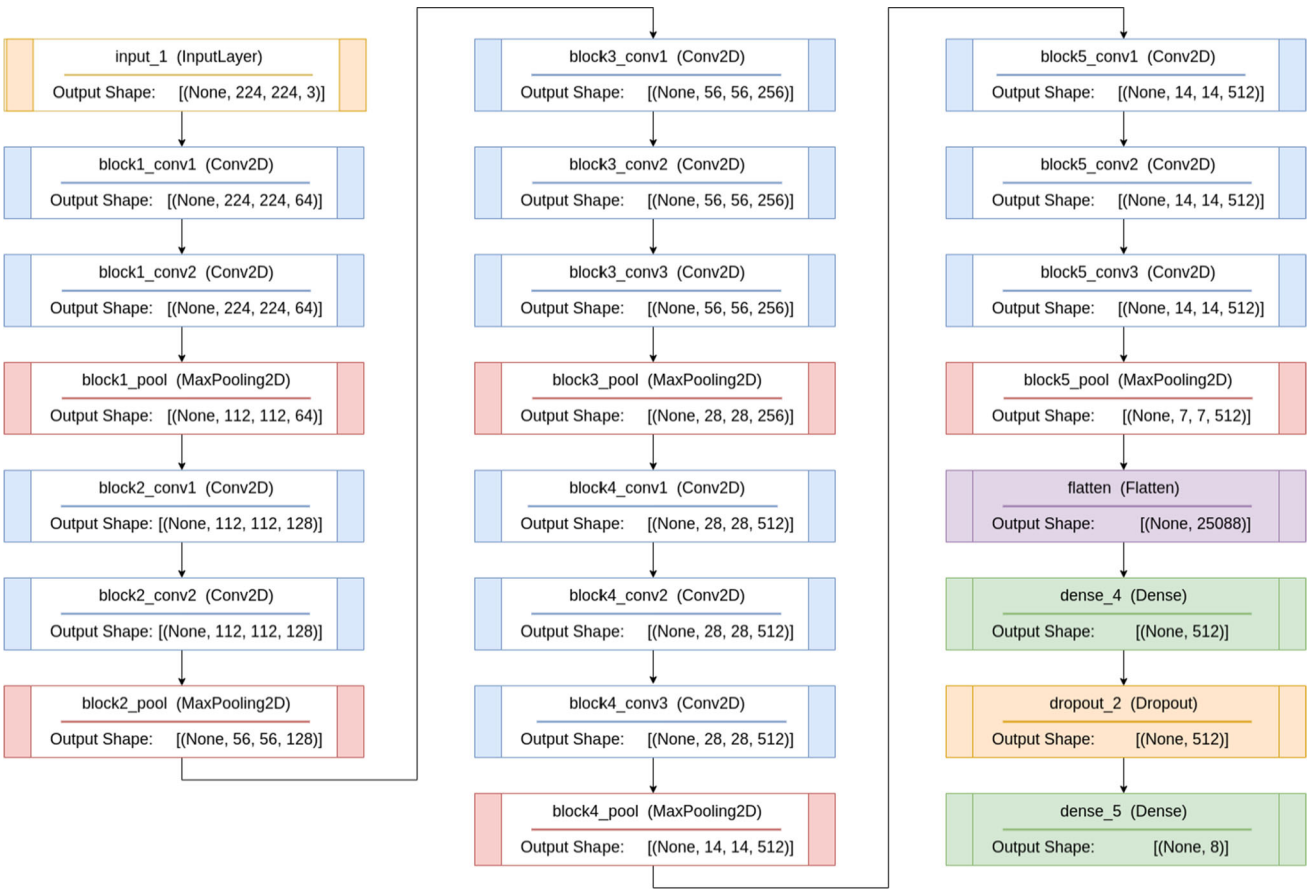


Fig. 7 Layers involved in VGG16

short connection between these bottlenecks. This allows this network to be implemented even on mobile devices. It is 53 layers deep and has also been trained on ImageNet database.

Table 1 describes the number of layer, total parameters, and trainable parameters for different CNN architectures used in this paper, i.e., VGG16, InceptionV3, ResNet50, and MobileNetV2.

Table 1 Summary for CNN architectures used in this paper

Model name	# of Layers	Total parameters	Trainable parameters
VGG16	16	19,175,207	13,863,719
InceptionV3	48	7,317,608	5,688,975
ResNet50	50	20,942,054	16,848,092
MobileNetV2	53	5,440,729	2,804,612

5 Methodology

5.1 Character face detection using mask R-CNN

Facial expressions are the significant contributor to interpersonal communication [43]. In this paper, Mask R-CNN is used for character face detection, which separates the foreground–background pixels from each other using a bounding box that segments the face [44]. The model takes images or videos as input to extract masks of *Tom*'s face or *Jerry*'s face out of it. Algorithm 1 specifies the step-wise methodology adopted for the character face detection Mask R-CNN, also explained in subsequent sections:-

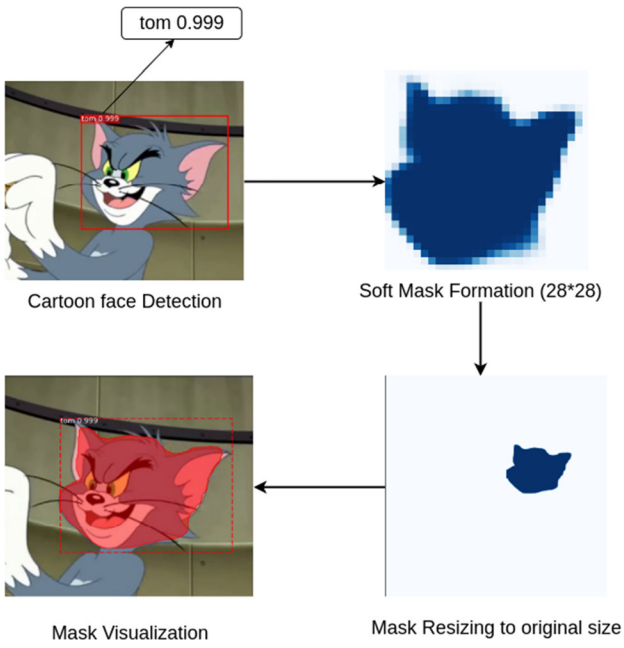


Fig. 8 Mask formation steps

(i) Frame extraction

The colored frames (from videos) are extracted using OpenCV and are then converted from BGR (Blue, Green, Red) to RGB (Red, Green, Blue) color order. Frame extraction process used here is explained in Sect. 4. Mask R-CNN takes input images of same size. Further, all the images are resized to a fixed dimension of (1280 × 1280) and the aspect ratio is maintained using padding.

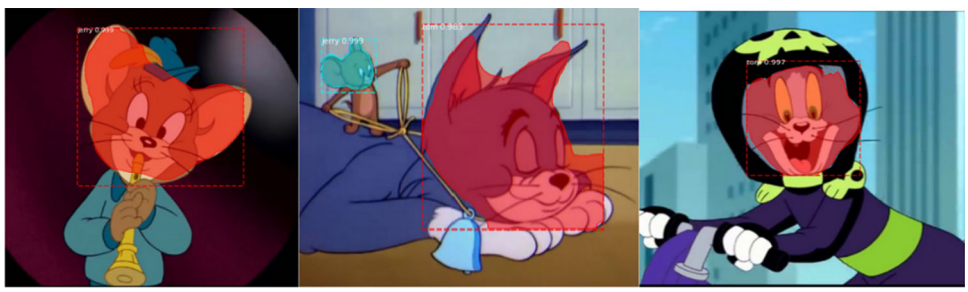
(ii) Mask generation and image storage

Then, the resized image (H' × W') is given to Mask R-CNN as input to detect faces of *Tom* and *Jerry*. It then returns a dictionary for each image with four key-value pairs (parameters) concerning the face detected:

(a) boundary box coordinates ($y1$, × 1, $y2$, × 2), which are generated around the cartoon's face,
(b) class id for both cartoon characters (1 for Tom and 2 for Jerry),
(c) binary masks.
(d) mask confidence score.

After obtaining the bounding boxes and refining them, the instance segmentation model generates the masks for each detected object. The masks are soft masks (with float pixel values) and of size 28 × 28 during training. Finally, the predicted masks are rescaled to the bounding box dimensions using padding and scaling factors to generate binary masks (0 and 1). Here, 1 denotes the region of the face detected and 0 denotes the rest of the image features. The face detected is marked with a mask confidence score that signifies the confidence in recognizing the mentioned character. As shown in Fig. 8, the score of detecting Tom's face is 0.999. These masks can be overlaid on the original image to visualize the final output. The processed masks generated are shown in Fig. 9. As can be observed from the

Fig. 9 Masks generated by Mask R-CNN



figure, the employed Mask R-CNN model detects straight faces, or tilted faces or even the faces surrounded by any object, such as helmet (ref. Fig. 9).

Using boundary box coordinates or pixel value ($y1, \times 1, y2, \times 2$) of the detected face, images($(y2 - y1) \times (x2 - x1)$) containing only faces of *Tom* and *Jerry* were cropped from the original image. Since the image size of both the original image and binary mask was the same, each RGB pixel value of the image is multiplied by the corresponding binary value present in the mask. The binary value 0, when multiplied with any pixel value, results in 0 (representing black color). This process removes extra features from the image. While binary value 1 in the mask, when multiplied with any pixel value, results in the same value. Hence, only the pixel values of the face were included in this mask (hereafter, will be called as segmented masks). After this, the cropped image and segmented masks are resized into dimension 256×256. The cropped images contained extra features (background regions), whereas segmented masks contained the required features for emotion classification.

ALGORITHM 1: Image segmentation using Mask R-CNN

```
This algorithm takes the extracted images
if (videos)
    images = ExtractFrames(videos)
end if.
for each image
    NewImage = Preprocessing(image)
    function Resize(NewImage, MinSize, MaxSize,  Padding)
        return ResizedImage, Window, ScalingFactor, Padding
    end function
    function Mask RCNN (ResizedImage)
        return BoundaryBox, ClassID, Score, BinaryMask
    end function
    Dictionary = GetClassID_Score(classID, Score)
    function getSpecificMask(BinaryMask, Dictionary, ResizedImage)
      if detected = Tom
        Mask = GenerateMask(BinaryMask, ResizedImage)
      else if detected = Jerry
        Mask = GenerateMask(BinaryMask, ResizedImage)
      endif
    function Resize(Image,MinSize, MaxSize, Padding)
        resized_imge = cv.resize(img, (256, 256))
    return Mask
end For
function ExtractFrames(videos){ //Frames get extracted at a frame ratio of 1:15
end function
    function Preprocessing(image){
    OpenCV.ImageRead(image)
     OpenCV.convert(image)
return ProcessedImage
function Resize(Image, MinSize, MaxSize, Padding){
  if Padding is true
    Image resized to MaxSize x MaxSize
  else
     Image resized to MinSize x MaxSize
  endif
returns Image
function GetClassID_Score(ClassID, Mask_Confidence_Score){
   Separate ClassID and Score of Tom and Jerry
return ClassID , respective Mask_Confidence_Score
end  function
function GenerateMask(Mask_Array,  Image){
  for j in range(3):
    Temp[:, :, j] = Temp[:, :, j] * Mask_Array
  end for
 return Image
```

5.2 Emotion classification using transfer learning and fine-tuning

Transfer learning uses the weights and knowledge gained from solving a specific problem and applying that knowledge to solve other similar tasks. It helps in leveraging the weights and biases of different state-of-the-art algorithms and hence uses it as an advantage without it being necessary to have vast amounts of data or extensive computation capabilities. The final step includes fine-tuning the model by unfreezing the specific parts of the model and re-training it on the new data with a small learning rate. The generic pipeline followed for emotion classification is as follows:

(a) Preprocessing of segmented masks

The segmented masks (of size 256 × 256) are received as an output from character detection stage using Mask R-CNN (ref. Sect. 5.1). These masks were resized to 224 × 224 for the classification of emotions using four baseline deep neural networks. These images are then converted into tensors. The value of these tensors is in the range 0 to 255, normalized to the range of 0 to 1. Afterward, data batches are created, each having 32 pictures for input into the emotion classification model. Figure 10 shows an example of a data batch created.

(b) Training and Classification

Four deep neural networks are trained using transfer learning from the data batches created. Based on the results (ref. Sect. 6), the best-trained model for every deep neural network is used for the classification of emotions. A snapshot of the results obtained by this proposed end-to-end approach is shown in Fig. 11. For instance, 0.96 is an emotion confidence score depicting an angry '*Tom*'.

6 Experiments

After applying deep neural network on fairly large data collection, the data go through a preprocessing phase. This phase which uses the Mask R-CNN model produces

Fig. 10 Visualizing the data batch created

Fig. 11 Visualization of emotion recognition using the proposed approach

accurate character masks as per the Algorithm 1 (ref. Sect. 5.1). Further, these masks are given as input to four deep neural network models, as described in Sect. 4.3. Then, the emotion of a particular character is recognized with an emotion confidence score (which is recognized emotion probability of the respective character) scaled in the range of 0 to 1.

6.1 Results

Classifying emotions on a large dataset is a multi-class classification problem. In this paper, each sample in the prepared dataset has been categorized into one of the eight (4×2) different classes. Standard metrics—precision, recall, F1-score, and accuracy score—have been computed for each class for the purpose of evaluation. The precision score for 'happy_jerry' label is the number of correctly recognized '*Jerry*' images with happy emotion out of total '*Jerry*' images with actual happy emotion. From Table 5 given in Appendix – I, the number of accurately detected 'happy_jerry' label is 198 out of 221 total recognized 'happy_jerry' label, resulting into precision score of 0.90 (198/221) for the proposed approach, whereas the precision score for other three models, i.e., InceptionV3, MobileNetV2, and ResNet-50 is 0.71, 0.63, and 0.74, respectively, for 'happy_jerry' label. Next, the recall for 'happy_jerry' label is the number of correctly recognized '*Jerry*' with happy emotion out of the number of actual '*Jerry*' images with happy emotion. As inferred from Table 5 given in Appendix—I, the number of correctly recognized

Table 2 Classification report for all the models with Mask R-CNN

Character	Emotion	VGG16				InceptionV3			
		Precision	Recall	F1-score	Accuracy score	Precision	Recall	F1-score	Accuracy score
Tom	Angry	0.82	0.86	0.84	0.96	0.72	0.75	0.73	0.94
	Happy	0.82	0.94	0.87	0.97	0.76	0.75	0.76	0.94
	Sad	0.87	0.66	0.75	0.95	0.72	0.68	0.7	0.93
	Surprise	0.85	0.85	0.85	0.96	0.76	0.78	0.77	0.94
Jerry	Angry	0.81	0.91	0.86	0.96	0.81	0.8	0.81	0.95
	Happy	0.90	0.87	0.88	0.97	0.71	0.69	0.7	0.91
	Sad	0.88	0.79	0.84	0.96	0.79	0.76	0.78	0.93
	Surprise	0.85	0.87	0.86	0.96	0.72	0.78	0.75	0.93
Micro-average		0.85	0.84	0.84	0.96	0.75	0.75	0.75	0.93
Weighted average		0.85	0.85	0.85	0.96	0.75	0.75	0.75	0.93
Character	Emotion	MobileNetV2				ResNet-50			
		Precision	Recall	F1-score	Accuracy score	Precision	Recall	F1-score	Accuracy score
Tom	Angry	0.71	0.66	0.68	0.93	0.62	0.65	0.64	0.91
	Happy	0.6	0.73	0.66	0.91	0.54	0.77	0.63	0.89
	Sad	0.71	0.52	0.6	0.93	0.46	0.61	0.52	0.88
	Surprise	0.65	0.72	0.68	0.92	0.84	0.37	0.52	0.92
Jerry	Angry	0.66	0.71	0.69	0.92	0.83	0.57	0.67	0.93
	Happy	0.63	0.54	0.58	0.89	0.74	0.64	0.69	0.92
	Sad	0.75	0.69	0.72	0.92	0.64	0.77	0.7	0.90
	Surprise	0.6	0.7	0.65	0.90	0.69	0.7	0.7	0.92
Micro average		0.66	0.66	0.66	0.91	0.64	0.64	0.64	0.90
Weighted average		0.66	0.66	0.66	0.92	0.67	0.64	0.64	0.91

'happy_jerry' emotion label is 198 out of 228 actual 'happy_jerry' label. This results into the recall score of 0.87 (198/228) for the proposed approach as shown in Table 2, whereas the recall score for other three models is comparatively less for 'happy_jerry' label.

For multiclass classification problem, F1-score is a preferable metric because there may be a large number of actual negatives. It gives the balance between precision and recall. For instance, the F1-score for 'happy_jerry' is 0.88, for the proposed approach. The other approaches namely InceptionV3, MobileNetV2 and ResNet-50 result in F1-score of 0.75, 0.66, and 0.64, respectively, for the same label. Among these models, VGG16 has outperformed the rest in terms of precision, recall, and F1-score as shown in Table 2. Also, the table depicts a combined classification report of the four models on which experimentation has been conducted. Accuracy score for each emotion class (here 8) for the two characters taken in this work is also shown in Table 2. The combined accuracy for a particular emotion (say 'sad') is calculated by averaging that emotion over both the characters. For example, the 'sad' emotion accuracy accounts to be 95% which is an average of two emotion classes (sad for *Tom* and sad for *Jerry*). Therefore, the combined accuracy for each emotion comes out to be Happy (97%), Sad (95%), Angry (96%), and Surprised (96%). Overall scores for each metric are calculated by averaging the scores obtained from each class depicted as weighted average and micro-average. The latter is calculated to handle the imbalance in the class distribution if any. VGG16 shows a weighted average of 0.85 across all metrics as compared to other three models showing the same as 0.75, 0.66, and 0.64, respectively, whereas the micro-average comes out to be 0.85 outperforming rest of the models.

Table 3 Classification report for all the models without Mask R-CNN

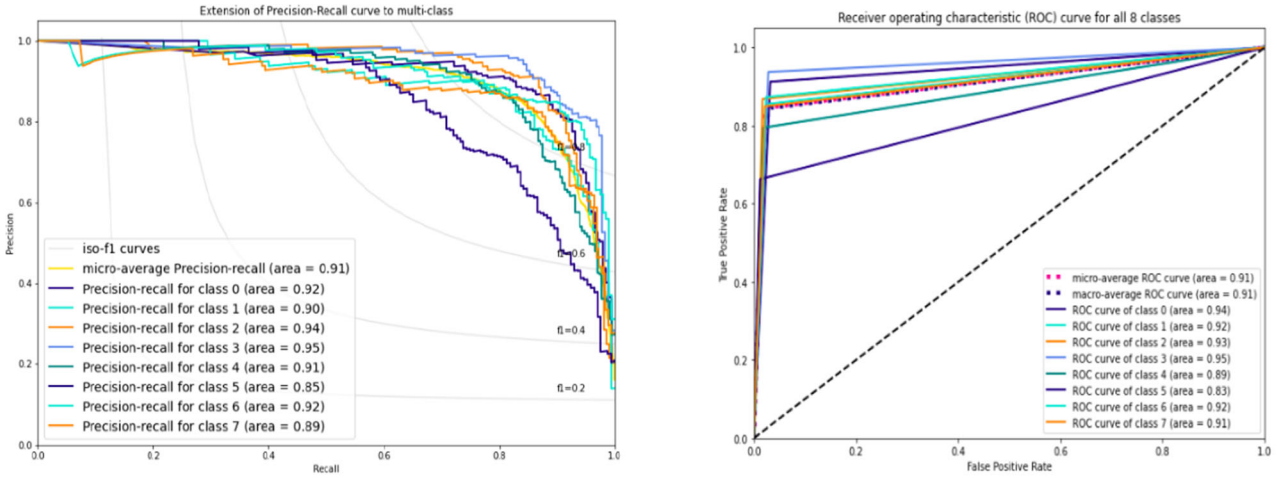
Character	Emotion	VGG16				InceptionV3			
		Precision	Recall	F1-score	Accuracy score	Precision	Recall	F1-score	Accuracy score
Tom	Angry	0.45	0.48	0.46	0.64	0.37	0.35	0.36	0.60
	Happy	0.36	0.37	0.36	0.65	0.35	0.36	0.35	0.59
	Sad	0.38	0.38	0.38	0.65	0.30	0.32	0.30	0.59
	Surprise	0.42	0.46	0.44	0.63	0.35	0.33	0.34	0.61
Jerry	Angry	0.43	0.48	0.45	0.64	0.36	0.35	0.35	0.61
	Happy	0.47	0.47	0.47	0.65	0.36	0.34	0.35	0.60
	Sad	0.45	0.38	0.41	0.63	0.37	0.37	0.37	0.58
	Surprise	0.39	0.45	0.42	0.63	0.32	0.31	0.31	0.61
Micro Average		0.42	0.42	0.42	0.64	0.35	0.34	0.34	0.59
Weighted average		0.42	0.43	0.42	0.64	0.35	0.34	0.34	0.60
Character	Emotion	MobileNetV2				ResNet-50			
		Precision	Recall	F1-score	Accuracy score	Precision	Recall	F1-score	Accuracy score
Tom	Angry	0.28	0.25	0.26	0.58	0.23	0.26	0.24	0.55
	Happy	0.25	0.24	0.24	0.58	0.24	0.25	0.24	0.55
	Sad	0.21	0.20	0.20	0.60	0.23	0.22	0.22	0.56
	Surprise	0.24	0.23	0.23	0.59	0.25	0.25	0.25	0.57
Jerry	Angry	0.26	0.25	0.25	0.59	0.26	0.22	0.24	0.56
	Happy	0.26	0.26	0.26	0.60	0.22	0.21	0.22	0.55
	Sad	0.25	0.27	0.26	0.57	0.24	0.25	0.24	0.58
	Surprise	0.22	0.22	0.22	0.58	0.33	0.27	0.30	0.58
Micro average		0.24	0.24	0.24	0.58	0.25	0.24	0.25	0.56
Weighted average		0.25	0.24	0.25	0.59	0.25	0.24	0.25	0.56

The results of these models without the use of Mask R-CNN are shown in Table 3. Without the use of Mask R-CNN, these models perform poorly as they are unable to learn the features of the character's faces which are required for emotion recognition. This happens because these models learn unnecessary features from the background of the image which are not required as the pre-processing stage, i.e., Mask R-CNN is not used to segment the faces.

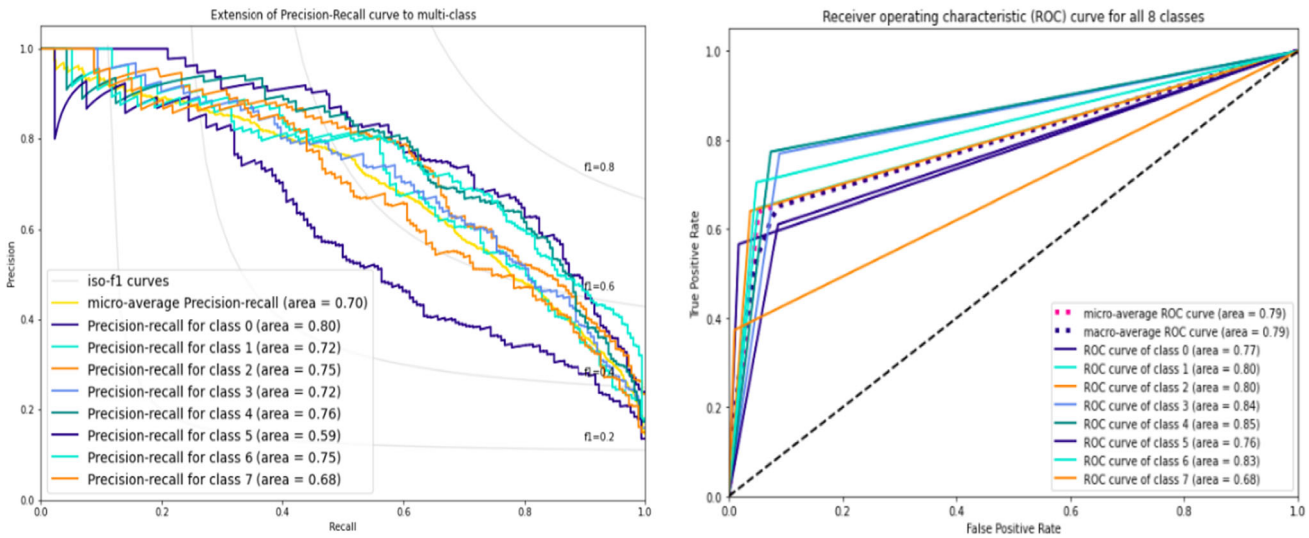
6.2 Extensions

VGG16 outperformed the other three models (ref. Section 6.1) when trained on the created dataset of 8 K labeled images of *Tom* and *Jerry*. This section draws out certain additional results for the best-performing model (VGG16). Figure 12a–d shows the plot for the precision-recall curve, which depicts the trade-off between precision and recall for all four DNNs. In an ideal scenario with high recall and precision values, a larger area under the curve can be obtained. Since the area under the curve is large, indicating high precision (accurate results) and high recall (majority positive results). This signifies that both the false-positive rate and false-negative rate are low.

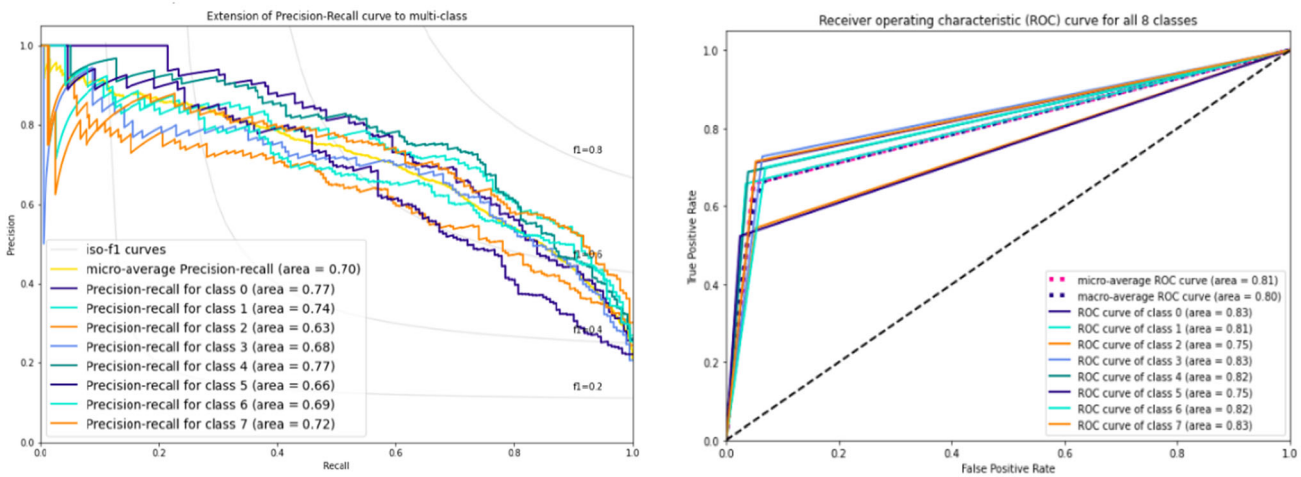
Using VGG16, the micro-average precision-recall score evaluates to 0.91 for all classes. This score accounts to be the highest among all evaluated DNNs. The correlation between the false positives and the true positives can be computed using the AUC (Area Under Curve) in a ROC curve also shown in Fig. 12a–d for all DNNs. As shown, the micro-average ROC score for VGG16 evaluates to 0.91 which exceeds scores obtained of the other three DNNs. The figure also shows a macro-average ROC-AUC score for VGG16 that returns the average without considering the proportion for each label in the dataset.



(a) VGG-16

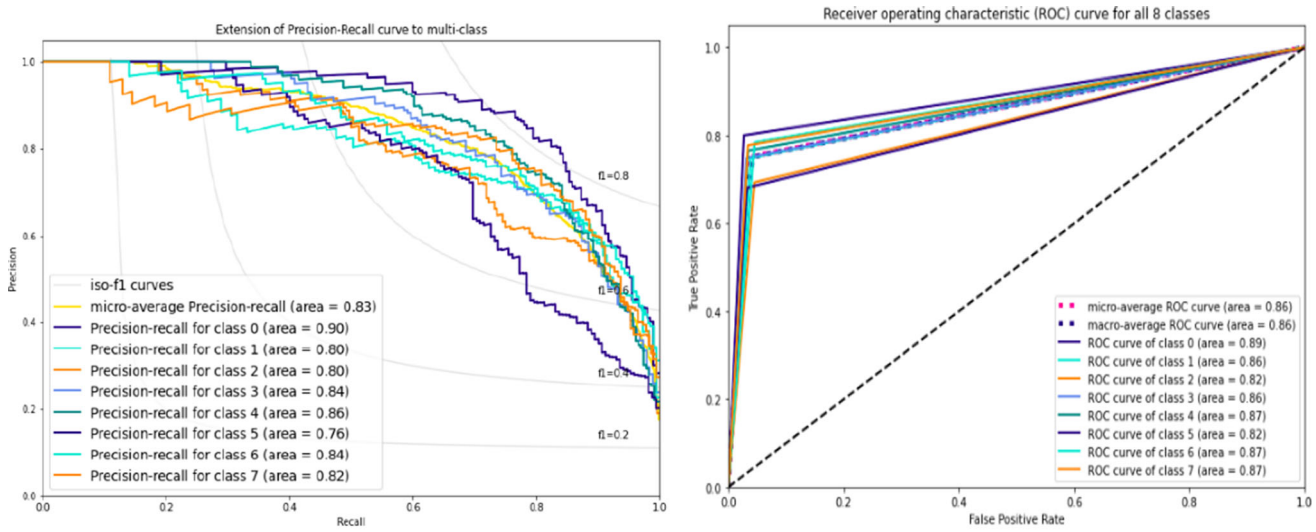


(b) ResNet-50



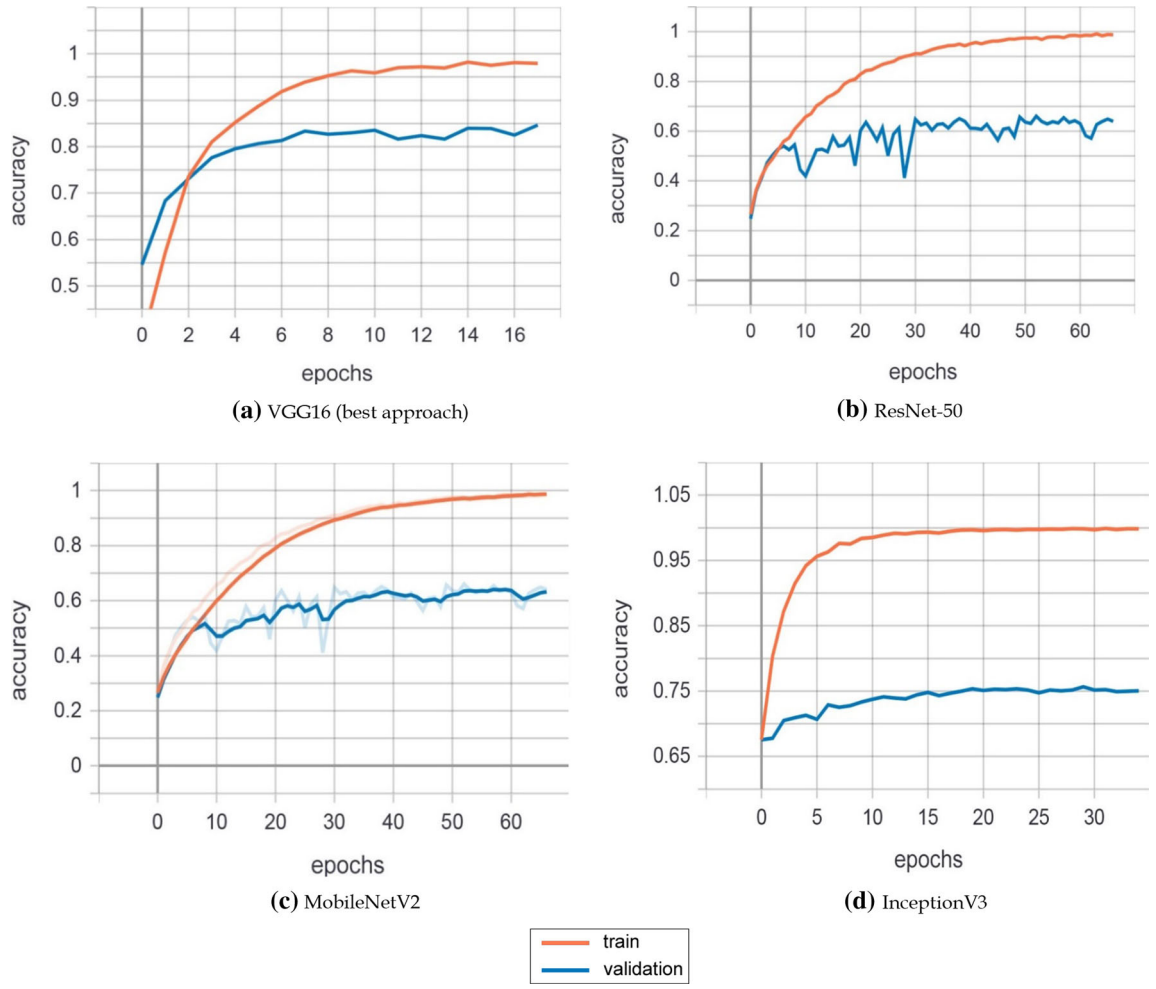
(c) MobileNetV2

Fig. 12 (**a**–**d**) Plots for precision-recall curves and ROC curves for all four DNNs



(d) InceptionV3

Fig. 12 continued



(a) VGG16 (best approach)

(b) ResNet-50

(c) MobileNetV2

(d) InceptionV3

Fig. 13 (**a–d**) Accuracy vs. the number of epochs plots for all the models

Table 4 Comparison between the proposed approach and the existing methods

Comparison Parameters	Proposed Approach	Hill [29]	Li et al. [45]	Ma et al. [46]	Aneja et al. [33]	Aneja et al. [47]
Number of emotions recognized	4 (H, S, A, & Su)	3 (H, S & A)	4 (A, H, S, & N)	6 (H, S, A, F, Su, & D)	7 (S, A, Su, D, F, N, & J)	7 (S, A, Su, D, F, N, & J)
Learning rate	0.0003	0.001	–	–	0.01	0.0001
Accuracy Score for Individual Emotion	0.97 (H), 0.95 (S), 0.96 (A), 0.96 (Su)	–	0.77 (A), 0.65 (H), 0.70 (S)	0.78 (H), 0.49 (S), 0.62 (A), 0.19 (Su)	0.89 (S), 0.85 (A), 0.95 (Su)	0.79 (S), 0.90 (A), 0.94 (Su)
Overall Model Accuracy Score	0.96	0.80	–	–	0.89	–
F1-score	0.85	–	–	–	–	–

*H → Happy, S → Sad, A → Angry, Su → Surprised, F → Fear, D → Disgusted, J → Joy, N→ Neutral

A model is said to be a good fit if it is able to generalize and learn the features from the training data without over-fitting or under-fitting. This means that the model can generalize the features and perform even on unseen data. In Fig. 13b and c, the models namely ResNet-50 and MobileNetV2 are unable to generalize the features. Hence, even if they perform well on training data with accuracy nearing 1, they are unable to give similar results on unseen data as can been seen from the fluctuation in the graph for validation accuracy curve. The proposed model depicted in Fig. 13a is able to outperform the above-stated models and is able to generalize the results giving an average accuracy score of 0.96. As shown in Fig. 13d, the InceptionV3 model is unable to learn enough features from training data as compared to our proposed model and thus gives lower scores for all the evaluation metrics. The figure also shows the train and validation accuracies recorded and plotted using tensor-board from the logs saved while training.

6.3 Comparison

Inspired by the fact that DNN models require fairly large amount of data, in our study, we have created a dataset (source:https://github.com/TheSSJ2612/Cartoon_Emotion_Dataset/releases). A fair comparison is not possible since we have created a new dataset for our proposed tool, which we call, integrated DNN. However, even though datasets and number of emotions are varied in the literature, we find it interesting to report previous works that were focused on emotion classification. In what follows, let us revisit previous works (ref. Table 4) and check how fair the comparison can be made.

In the presented comparison, one of the existing works has proposed an emotion recognition model on human animated faces [33]. The adopted methodology includes training two different CNNs to recognize human expressions and stylized characters expressions independently. A shared embedding feature space is then created by mapping human faces to character faces using transfer learning, resulting in an accuracy score of 0.89. Similarly, other contributions [45, 46] have evaluated the proposed approaches on animated characters (not specific to a cartoon character) from various sources like books, video games, etc. A different kind of approach generates 3D

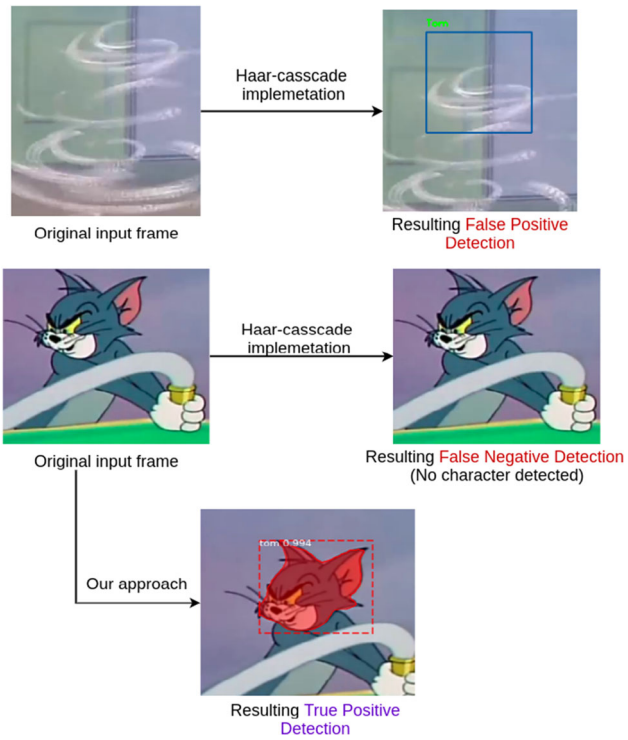


Fig. 14 Comparing our approach in detecting a cartoon character to already existing work

animated faces using human facial expressions by transferring the emotion features to the 3D character face using semi-supervised learning model 'ExprGen' [47]. The above-referred state-of-the-art methods—Li et al. [45], Ma et al. [46], and Aneja et al. [47]—do not provide their overall end-to-end model accuracy score and instead, present an accuracy score for each emotion label classified as shown in Table 4. Hill [29] is the only similar contribution that exists in the state-of-the-art methods where the author has proposed an end-to-end emotion recognition model on cartoon videos. This approach gives an overall accuracy score of 0.80. Figure 14 depicts the outperformance of the integrated DNN model (contributed in this paper) using Mask R-CNN over the already existing methodology [29]. As mentioned earlier, even though datasets are varied from one work to another, we find that our study (with integrated DNN tool) performs better on the dataset of size 8113.

7 Conclusion

Recognizing emotions from facial expressions of faces other than human beings is an interesting and challenging problem. Although the existing literature has endeavored to detect and recognize objects, however, recognizing emotions has not been extensively covered. Therefore, in this paper, we have presented an integrated Deep Neural Network (DNN) approach that has successfully recognized emotions from cartoon images. We have collected a dataset of size 8 K from two cartoon characters: *'Tom'* & *'Jerry'* with four different emotions, namely happy, sad, angry, and surprise. The proposed integrated DNN approach has been trained on the large dataset and has correctly identified the character, segmented their face masks, and recognized the consequent emotions with an accuracy score of 0.96. The approach has utilized Mask R-CNN for character detection and state-of-the-art deep learning models, namely ResNet-50, MobileNetV2, InceptionV3, and VGG 16, for emotion classification. The experimental analysis has depicted the outperformance of VGG 16 over others with an accuracy of 96% and F1 score of 0.85. The proposed integrated DNN has also outperformed the state-of-the-art approaches.

The work would be beneficial to the animators, illustrators, and cartoonists. It can also be used to build a recommender system that allows users to associatively select emotion and cartoon pair. Studying emotions encased in cartoons also extracts other allied information, which if combined with artificial intelligence can open a plethora of opportunities, for instance, recognizing emotions from body gestures.

Appendix

See Table 5 .

Table 5 Confusion matrix for all the models

		VGG16									InceptionV3								
Character		Jerry Predicted labels				Tom Predicted labels				**Total Actual Labels**	Jerry Predicted labels				Tom Predicted labels				**Total Actual Labels**
	Emotion	**Angry**	**Happy**	**Sad**	**Surprise**	**Angry**	**Happy**	**Sad**	**Surprise**		**Angry**	**Happy**	**Sad**	**Surprise**	**Angry**	**Happy**	**Sad**	**Surprise**	
Jerry Actual labels	**Angry**	187	6	8	3	1	0	0	0	205	164	14	19	8	0	0	0	0	205
	Happy	12	198	3	15	0	0	0	0	228	14	158	16	39	0	1	0	0	228
	Sad	27	5	186	15	0	0	1	0	234	15	22	179	16	1	1	0	0	234
	Surprise	4	12	11	186	0	0	0	0	213	8	26	12	167	0	0	0	0	213
Tom Actual labels	**Angry**	1	0	2	0	160	17	4	3	187	0	0	0	1	140	18	14	14	187
	Happy	0	0	0	0	10	179	2	0	191	1	0	0	0	23	143	12	12	191
	Sad	1	0	0	0	14	16	114	27	172	0	0	0	0	15	18	117	22	172
	Surprise	0	0	1	0	11	7	10	164	193	0	1	0	0	16	6	20	150	193
Total Predicted labels		232	221	211	219	196	219	131	194		202	221	226	231	195	187	163	198	

		MobilenetV2									Resnet_50								
Character		Jerry Predicted labels				Tom Predicted labels				**Total Actual Labels**	Jerry Predicted labels				Tom Predicted labels				**Total Actual Labels**
	Emotion	**Angry**	**Happy**	**Sad**	**Surprise**	**Angry**	**Happy**	**Sad**	**Surprise**		**Angry**	**Happy**	**Sad**	**Surprise**	**Angry**	**Happy**	**Sad**	**Surprise**	
Jerry Actual labels	**Angry**	146	22	22	14	1	0	0	0	205	116	12	52	11	9	2	3	0	205
	Happy	30	124	14	60	0	0	0	0	228	6	146	32	32	2	5	5	0	228
	Sad	31	18	161	24	0	0	0	0	234	8	13	181	21	4	2	5	0	234
	Surprise	14	32	18	149	0	0	0	0	213	6	26	18	150	4	5	3	1	213
Tom Actual labels	**Angry**	0	1	0	0	123	42	8	13	187	1	0	0	1	121	31	29	4	187
	Happy	0	0	0	0	18	139	10	24	191	1	0	0	0	22	147	18	3	191
	Sad	0	0	0	0	14	30	90	38	172	1	0	0	1	18	41	105	6	172
	Surprise	0	0	0	0	17	20	18	138	193	1	1	0	2	14	41	62	72	193
Total Predicted labels		221	197	215	247	173	231	126	213		140	198	283	218	194	274	230	86	

Authors' contributions All authors have equally contributed toward the formation of this paper.

Funding Not Applicable.

Declarations

Conflicts of interest The authors declare that they have no competing interests.

Availability of data and material https://github.com/TheSSJ2612/Cartoon_Emotion_Dataset/releases.

References

1. Ekman P, Friesen WV (1976) Measuring facial movement. Environ psychol nonverbal behav 1(1):56–75
2. Shivhare S. N., Khethawat S. (2012). Emotion detection from text. arXiv preprint. arXiv:1205.4944
3. Gupta V, Singh VK, Mukhija P, Ghose U (2019) Aspect-based sentiment analysis of mobile reviews. J Intell Fuzzy Syst 36(5):4721–4730
4. Piryani R, Gupta V, Singh VK (2017) Movie Prism: A novel system for aspect level sentiment profiling of movies. J Intell Fuzzy Syst 32(5):3297–3311
5. Rao Y, Xie H, Li J, Jin F, Wang FL, Li Q (2016) Social emotion classification of short text via topic-level maximum entropy model. Inf Manag 53(8):978–986
6. Venkataramanan K., Rajamohan H. R. (2019). Emotion Recognition from Speech. arXiv preprint, arXiv:1912.10458
7. Gupta V, Juyal S, Singh GP, Killa C, Gupta N (2020) Emotion recognition of audio/speech data using deep learning approaches. J Inf Optim Sci 41(6):1309–1317
8. Casale S., Russo A., Scebba G., Serrano S. (2008). Speech emotion classification using machine learning algorithms. In: 2008 IEEE international conference on semantic computing, pp. 158–165
9. Jiang D. N., Cai L. H. (2004). Speech emotion classification with the combination of statistic features and temporal features. In: IEEE International Conference on Multimedia and Expo, Vol. 3, pp 1967-1970
10. Kim, M. H., Joo, Y. H., Park, J. B. (2005). Emotion detection algorithm using frontal face image. International Conference on Control and Robotics Systems, 2373–2378.
11. Bargal S. A., Barsoum E., Ferrer C. C., Zhang C. (2016). Emotion recognition in the wild from videos using images. In: Proceedings of the 18th ACM International Conference on Multimodal Interaction. pp 433–436.
12. Elngar, A. A., Jain, N., Sharma, D., Negi, H., Trehan, A., & Srivastava, A. (2020). A deep learning based analysis of the big five personality traits from handwriting samples using image processing. Journal of Information Technology Management, 12(Special Issue: Deep Learning for Visual Information Analytics and Management.), 3–35.
13. Guo Y., Gao H. (2006). Emotion recognition system in images based on fuzzy neural network and HMM. In: 5th IEEE International Conference on Cognitive Informatics, Vol. 1, pp 73–78.
14. Lisetti C, Nasoz F, LeRouge C, Ozyer O, Alvarez K (2003) Developing multimodal intelligent affective interfaces for tele-home health care. Int J Hum Comput Stud 59(1–2):245–255
15. Gupta V, Jain N, Katariya P, Kumar A, Mohan S, Ahmadian A, Ferrara M (2021) An emotion care model using multimodal textual analysis on COVID-19. Chaos, Solitons & Fractals, p 110708
16. Derntl B, Seidel EM, Kryspin-Exner I, Hasmann A, Dobmeier M (2009) Facial emotion recognition in patients with bipolar I and bipolar II disorder. Br J Clin Psychol 48(4):363–375
17. Jain R, Jain N, Aggarwal A, Hemanth DJ (2019) Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images. Cogn Syst Res 57:147–159
18. Jain, N., Chauhan, A., Tripathi, P., Moosa, S. B., Aggarwal, P., & Oznacar, B. (2020). Cell image analysis for malaria detection using deep convolutional network. Intelligent Decision Technologies, (Preprint), 1–11.
19. Bahreini K, Nadolski R, Westera W (2016) Data fusion for real-time multimodal emotion recognition through webcams and microphones in e-learning. Int J Hum-Comput Interact 32(5):415–430
20. Ray A., & Chakrabarti A. (2012). Design and implementation of affective e-learning strategy based on facial emotion recognition. In: Proceedings of the International Conference on Information Systems Design and Intelligent Applications, pp 613–622.
21. Chu HC, Tsai WWJ, Liao MJ, Chen YM (2018) Facial emotion recognition with transition detection for students with high-functioning autism in adaptive e-learning. Soft Comput 22(9):2973–2999
22. Shen L, Wang M, Shen R (2009) Affective e-learning: Using "emotional" data to improve learning in pervasive learning environment. J Educ Technol Soc 12(2):176–189
23. Piryani R, Gupta V, Singh VK, Ghose U (2017) A linguistic rule-based approach for aspect-level sentiment analysis of movie reviews. In: Bhatia SK, Mishra KK, Tiwari S, Singh VK (eds) Advances in computer and computational sciences. Springer, Singapore, pp 201–209
24. Ren F, Quan C (2012) Linguistic-based emotion analysis and recognition for measuring consumer satisfaction: an application of affective computing. Inf Technol Manage 13(4):321–332
25. Piryani R, Gupta V, Singh VK (2018) Generating aspect-based extractive opinion summary: drawing inferences from social media texts. Computación y Sistemas 22(1):83–91
26. Garbas J. U., Ruf T., Unfried M., Dieckmann A. (2013). Towards robust real-time valence recognition from facial expressions for market research applications. In: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, (pp 570–575).
27. Robinson L, Spencer MD, Thomson LD, Sprengelmeyer R, Owens DG, Stanfield AC, Johnstone EC (2012) Facial emotion recognition in Scottish prisoners. Int J Law Psychiatry 35(1):57–61
28. Peleshko, Dmytro, Kateryna Soroka. (2013). Research of usage of Haar-like features and AdaBoost algorithm in Viola-Jones method of object detection. International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics.
29. Hill JW (2017) Deep Learning for Emotion Recognition in Cartoons(Unpublished master's dissertation). The University of Lincoln, Lincoln School of Computer Science, UK
30. Ekman P, Oster H (1979) Facial expressions of emotion. Annu Rev Psychol 30(1):527–554
31. Gajarla V, Gupta A (2015) Emotion detection and sentiment analysis of images. Georgia Institute of Technology, Atlanta
32. Minaee, S., & Abdolrashidi, A. (2019). Deep-emotion: Facial expression recognition using attentional convolutional network. arXiv preprint, arXiv:1902.01019
33. Aneja D., Colburn A., Faigin G., Shapiro L., Mones B. (2016). Modeling stylized character expressions via deep learning. In: Asian conference on computer vision, pp 136–153

34. Zhao J, Meng Q, An L, Wang Y (2019) An event-related potential comparison of facial expression processing between cartoon and real faces. PLoS ONE 14(1):e0198868
35. Kendall LN, Raffaelli Q, Kingstone A, Todd RM (2016) Iconic faces are not real faces: enhanced emotion detection and altered neural processing as faces become more iconic. Cognitive Res: Princ Implic 1(1):19
36. Li, S., Zheng, Y., Lu, X., & Peng, B. (2019). iCartoonFace: A Benchmark of Cartoon Person Recognition. arXiv preprint, arXiv:1907.13394
37. Zhou Y., Jin Y., Luo A., Chan S., Xiao X., Yang X. (2018). ToonNet: a cartoon image dataset and a DNN-based semantic classification system. In: Proceedings of the ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, pp. 1–8.
38. Cowie R, Douglas-Cowie E, Tsapatsoulis N, Votsis G, Kollias S, Fellenz W, Taylor JG (2001) Emotion recognition in human-computer interaction. IEEE Signal Process Mag 18(1):32–80
39. Xu C, Cui Y, Zhang Y, Gao P, Xu J (2020) Person-independent facial expression recognition method based on improved Wasserstein generative adversarial networks in combination with identity aware. Multimedia Syst 26(1):53–61
40. Siddiqi MH, Ali R, Khan AM, Kim ES, Kim GJ, Lee S (2015) Facial expression recognition using active contour-based face detection, facial movement-based feature extraction, and non-linear feature selection. Multimedia Syst 21(6):541–555
41. Rolling L (1981) Indexing consistency, quality and efficiency. Inf Process Manage 17(2):69–76
42. Byrt T (1996) How good is that agreement? Epidemiol 7(5):561
43. Pantic M, Rothkrantz LJM (2000) Automatic analysis of facial expressions: the state of the art. IEEE Trans Pattern Anal Mach Intell 22(12):1424–1445
44. Lin K, Zhao H, Lv J, Li C, Liu X, Chen R, Zhao R (2020) Face Detection and Segmentation Based on Improved Mask R-CNN. Discrete Dyn Nat Soc 2020:1–11
45. Li Y., Yu F., Xu Y. Q., Chang E., Shum H. Y. (2001). Speech-driven cartoon animation with emotions. In: Proceedings of the ninth ACM international conference on Multimedia, pp 365–371.
46. Ma X., Forlizzi J., Dow S. (2012). Guidelines for depicting emotions in storyboard scenarios. In: International design and emotion conference.
47. Aneja D., Chaudhuri B., Colburn A., Faigin G., Shapiro L., Mones B. (2018). Learning to generate 3D stylized character expressions from humans. In: IEEE Winter Conference on Applications of Computer Vision, pp 160–169.