OPTIMIZED VIDEO FACIAL EXPRESSION RECOGNITION USING GENERATIVE ADVERSARIAL NETWORKS WITH STOCHASTIC GRADIENT DESCENT MOMENTUM

Dr. Aswathy K Cherian[1], M. Vaidhehi[2], Arshey M[3]
Assistant Professor, SRM Institute of Science and Technology, Chennai, (Corresponding Author),
Assistant professor, SRM Institute of Science and Technology
Assistant Professor in CSE, SCMS School of Engineering and Technology

Abstract:

Video data serves as a valuable asset across diverse settings, ranging from live broadcasts on personal blogs to security surveillance in manufacturing facilities. The integration of machine learning tools for video processing has become a prevalent practice in various applications. Noteworthy strides in computer vision through machine learning have been made, particularly in item identification, object categorization, and image segmentation, rivaling or surpassing human capabilities. However, challenges persist in effectively identifying human emotions. This study focuses on emotion recognition using still images and motion pictures through various machine learning techniques. Neural networks, specifically based on Generative Adversarial Networks (GAN), were employed to categorize facial emotions into seven predefined categories. Informative features from videos, such as audio and single or multiple video frames, were extracted using OpenSMILE and Inception-ResNet-v2 models. Stochastic Gradient Descent with Momentum Approach (SGDMA) was utilized to train multiple classification models. The results, compiled into a table, revealed the predominant facial expression throughout the video. GAN-SGDMA was applied for the classification of audio feature vectors, and Inception-ResNet-v2 was employed for recognizing emotions from still photographs. Experimental outcomes suggest that the distributed model GAN-SGDMA significantly enhances the speed of detecting and classifying facial expressions in videos. The effectiveness of the GAN-SGDMA approach is demonstrated through its application to GAN-structured face expression recognition datasets, yielding impressive results.

Keywords: Emotion recognition, Generative Adversarial Networks, Facial expressions, Video data, Stochastic Gradient Descent with Momentum

1.Inroduction

In recent years, there has been a growing interest in using computer vision and image processing technologies to recognize human facial expressions [1] automatically. This would be accomplished by analyzing the state aspects of images, including facial expressions. This was done by looking at photographs of people's faces. It can be employed in various scenarios, including intelligent animation synthesis, intelligent monitoring, human-computer interaction, emotional computing, video games, psychology, and medical monitoring [2]. Interpreting a person's facial expressions contributes to developing related topics and industries. Concurrently, technologies based on facial expression detection technology are garnering a lot of positive attention due to the practicality and exciting applications they have in people's day-to-day lives. Regarding computers, one of the most important things that must be researched is the development of more intricate and accurate methods for recognizing facial expressions and how they change over time.

Facial expressions are a kind of communication that is powerful and subtle. Facial expression recognition is a necessary technology to deploy the emotional computing system as the human-computer interface successfully. Facial expressions have the potential to be utilized in a wide diversity of research fields, together with but not incomplete to virtual reality, video conferencing, customer satisfaction surveys, and many more. In human communication, facial expressions are a vital component. It frequently adds various expressive mechanisms to nonverbal communication and is necessary for comprehending other people's emotional responses. As a result of developments in biotechnology and computer science, different industries have begun to implement facial expressions. The most common application sector is privacy and security, as shown by the function allowing users to unlock their mobile devices and desktop computers by using their faces. Second, facial expression recognition is also utilized in transportation to determine whether a driver is tired or impaired before an accident occurs. In addition, the technology used to identify facial expressions has numerous applications in simulated realism, medical care, and service robotics [3-5]. It should come as no surprise that the technology used to identify facial expressions is complex, with many technical challenges. Expressions on people's faces can transmit a wide range of meanings due to the wide variety of linguistic and cultural traditions around the world. In addition, the findings of the identification of facial expressions cannot be called into doubt due to the objective influence of non-structural elements such as occlusion, illumination, and focus issues. Recent research in facial recognition has focused on finding solutions to these technical problems, even though technological gains have been reasonably modest [6]. This is although technology advancements have been relatively slow.

Autoencoders and Generative Adversarial Networks are utilized in most deep-fake applications nowadays (GANs). Autoencoders can glean latent facial characteristics from images, which they may subsequently use to create photographs in which the subjects' faces are depicted displaying a wide range of emotions. During training, GANs can learn to anticipate the input distribution because they use two competing networks, the discriminator and the generator. After that, the GAN can forecast the input distribution using this knowledge. While the discriminator is taught to discriminate between false and accurate data, the generator still seeks ways to produce fake data that can fool the discriminator. As the training goes on, the discriminator won't be able to distinguish between collected data and made to look natural, so the two sorts of data will start to look the same to the discriminator. In light of this, it is possible to ignore the discriminator and use the generator instead to create unprecedentedly lifelike data [7].

The discriminator network was trained to handle the maximization problem to differentiate between genuine HR echoes and artificially created SR solutions. The architectural principles outlined in [8] were considered when designing it. In addition, the discriminator network's output was enhanced due to the relativistic GAN. Due to the a priori knowledge that half of the data in the mini-batch is fake, the generator G in a standard GAN is trained to increase the likelihood that false data is accurate while decreasing the likelihood that actual data is real [9]. This is because generator G is introduced to raise the possibility that bogus data is correct when it is prepared to do so. This is because generator G has been set to increase the likelihood that fraudulent data is authentic. The reason for this can be seen in the previous sentence. A method known as momentum, sometimes known as SGD with rate, is a technique that speeds up the convergence process by assisting in the acceleration of gradient vectors in the appropriate directions. It is one of the most often used optimization techniques, and many cutting-edge models are educated with its help.

The research study has the following contributions,

1. First, Generative Adversarial Networks (GAN) were used to categorize individual facial pictures into one of seven facial expression classes. Videos can show emotions by taking out audio, a single video frame, or multiple video frames.
2. During this process stage, the OpenSMILE and Inception-ResNet-v models are used to extract property vectors from audio and frames autonomously.
3. Thirdly, the categorization of feelings is achieved through stochastic gradient descent in conjunction with the momentum approach (SGDMA). The GAN-SGDMA model produced was used to categorize facial expressions on all of the facial images detected, and a table was constructed to classify the terms that were seen most frequently during the movie.

This study is divided into sections: Section II summarizes the literature review. Section III discusses the GAN-SGDMA proposed models and performance evaluation. Section IV uses statistical tests and previous research to discuss the proposed model's performance evaluation capabilities. Section V grants the conclusion of the planned prototypical and future studies.

2.Literature Survey

Convolutional layers were used throughout the system as a critical component. In the network developed by Stamatios Lefkimmiatis et al. [10], non-local filtering layers can be used to discover the underlying non-local self-similarity attribute of natural images. Utilizing the network's non-local filtering layers will enable this. This state-of-the-art method is built on the suggested network architecture. Several CNN and GAN-based neural networks for picture denoising have developed substantially in recent years. These networks have experienced enormous popularity.

Based on their observations that existing deep fake production methods produce low-resolution photos with unique distortions when bent to be compatible with source faces, Li et al. [11] planned a technique for detecting deep fakes. This method was developed in response to the fact that deep fake production methods already in use produce deep fakes. The program Idlib was used to identify the faces, and the following CNN representations were qualified to differentiate between false and real videos: VGG16, ResNet152, ResNet101, and ResNet50. This identification system proved vulnerable to a variety of video compression strategies.

Wen Z et al. [12] proposed a multi-head cross-attention network that would be used for recognizing facial expressions. Distract yourself. The previous methods (89.7%) employed an actual multifaceted network founded on multi-head courtesy called DAN, which was composed of trio portions: An attention fusion network for calculating a global attention map, a feature clustering network for extracting robust emotion data, and a multi-head attention network for focusing on many facial regions at the same time. Ismail et al. [13] created YIX, a hybrid technique that detects anomalies and artefacts in the spatial information of fabricated video frames and so determines the authenticity of videos. Faces are extracted from video frames with the help of this method by using the YOLO detector. After that, a significantly modified version of the InceptionResNetV2 model was used to extract features, and an XGBoost model was employed as a classifier to differentiate between deep fake movies and real ones.

The HOG descriptor served as the foundation for the CNN algorithm created by Fadl et al. [14]. It is possible to extract the spatial gradient orientation properties that characterise the local contour, silhouette, and texture details of faces using the HOG descriptor. These qualities describe the beginning in question. These qualities can be observed in a variety of looks. It has been demonstrated that the HOG is useful in image processing and computer vision applications such as action recognition, object detection, face recognition, facial expression recognition, and the detection of false films and photographs.

A temporal deepfake video detection approach was developed by Güera et al. [15]. This technique employs light incoherence over fake video frames, producing flicker artifacts in the face region. To judge whether or not the film could be considered legitimate, the LSTM was trained on the attributes extracted from the video frames by the Inception V3 and then used. This technique was successful when applied to videos with less than two seconds.

According to observations made by Sabir et al. [16], artificial face-generation algorithms typically do not require temporal coherence to be maintained during synthesis. As a result, their first recommendation was to extract faces from video frames while also cropping and aligning them. Then, on these allied faces, they employed DenseNet or ResNet50 with bidirectional GRU to analyse the time-based artefacts required to locate synthetic faces in video frames. This was done to test their ability to do so accurately.

BT Hung et al. [17]. The HOG is a local descriptor that analyses a pixel's horizontal and vertical surroundings to describe that pixel inside a face frame. It generates histograms of local intensity gradients, which are then used to designate the entrance and form of substances nearby. The HOG representation can capture the gradient structure or edge information of a rise, as well as the texture near limits, which correspond to the basic resident shape. Manipulation of the colour difference at individual pixels of focused portions of the face frame in both the x and y axes yields two gradients: one that is parallel to the x-axis and has a derivative of that axis and another that is perpendicular to the y-axis and has a result of that axis. This is how it works.

P. Korshunov and colleagues [18] developed a procedure that detects deep fake lip-sync performances. The examination employed the distances between mouth landmarks and the Mel Frequency Cepstral Coefficients as visual structures. The MFCC was used to analyze the sound systems. After that, the principal constituent study decreased the dimensionality of the combined visual-audio property vector. This vector was used as a contribution in various classifiers, such as the multilayer perceptron, the LSTM, the support vector machine, and the Gaussian mixed model. Results from this method with the LSTM classifier were superior to those from the others. But the results became progressively poorer as the number of training cases decreased.

Patch-gated CNN is the method that Li et al. [19] suggest using for FER. They do this by employing landmarks to locate small areas of interest on the face, and once they have done this, they embed patch-gated learning units. This allows them to accomplish what they set out to do. The weights given to these patches for the region were calculated to consider the information from those patches.

Jain et al. [20] recommend a protracted deep neural network that has the potential to achieve an accuracy level of 95.23% on average. If the network is trained correctly, it can achieve this level of accuracy. If the network is adequately trained, it can achieve this level of accuracy. If the network is given the proper training, it can achieve this level of accuracy. If the network is trained correctly, it can reach this level of accuracy independently.

Wang et al.[21] the idea was "Region Attention Networks for Posture and Occlusion Resistant Facial Expression Recognition." Several deep learning-based super-resolution (SR) approaches have been presented, from a primary method based on CNN to a more modern system based on generative adversarial networks (GAN), both of which can generate precise touches during the super-resolving of a single image. However, the issue of weather radar echo super-resolution has only been explored by a limited number of earlier articles from the perspective of deep learning. Consequently, it would be fascinating to compare the performance of these deep learning super-resolution models to that of conventional super-resolution methods applied to weather radar level-II data products without prior knowledge.

2.1 Limitations for Existing system

- The quality of the video data significantly impacts the accuracy and reliability of facial expression analysis. Lighting, video resolution, camera angles, and occlusions can all affect the visibility and identification of face characteristics, making it difficult to extract accurate and consistent facial expressions.
- Facial expressions are highly dynamic and vary significantly between people, countries, and settings. The interpretation and classification of facial expressions can be subjective, as certain expressions may have multiple meanings or interpretations based on the person being analyzed's cultural and social background.

- Facial expressions are influenced by contextual cues such as body language, voice tone, and the surrounding environment, in addition to the movement of the face. Analyzing facial expressions in isolation from other cues can result in limited or erroneous emotional judgments.
- While facial landmarks and features can be retrieved from video frames, their accuracy and robustness vary. Occlusions, fluctuations in instance and head motions, and poor image quality might make it difficult to correctly identify and track facial features, lowering overall analysis accuracy.
- Datasets are heavily used in developing and evaluating video-based face expression analysis systems. On the other hand, the absence of standardized datasets that capture different facial expressions in real-world circumstances can interfere with development in this field and limit the generalizability of produced models.
- Access to video recordings of individuals is frequently required for video-based facial expression analysis, presenting privacy problems. Proper consent and ethical considerations must be considered to ensure the responsible use and protection of personal data.
- Analyzing video data for facial expressions can be computationally demanding, necessitating much computing power and storage space. Real-time analysis or large-scale applications may provide computing efficiency and scalability problems.

2.2 Problem identification for Existing system

- One of the most challenging issues in video-based facial expression analysis is the need for a large and diverse dataset. Creating a large-scale dataset with diverse facial expressions taken in various scenarios and lighting conditions can be time-consuming and costly. The restricted dataset makes developing and testing accurate and robust face expression recognition models difficult.
- In video-based facial expression analysis, facial occlusions such as spectacles, facial hair, masks, or hands obscuring areas of the face represent a considerable obstacle. These occlusions can impede crucial facial landmarks, making identifying and interpreting facial expressions challenging. Furthermore, natural variations in facial features, such as varied head postures, different intensities of facial expressions, and tiny movements, complicate recognizing and assessing facial expressions.

- Facial expressions are fluid and change throughout time. Analyzing facial expressions in video sequences necessitates capturing the temporal dynamics of words and comprehending the context in which they occur. Recognizing and effectively interpreting these dynamic shifts is critical for capturing the nuances and subtleties of various emotions. However, correctly modeling and extracting temporal information from video data remains a significant issue in video-based facial expression analysis.
- Individuals' facial expressions vary depending on their cultural background, psychological qualities, and individual distinctions. Because of these inter-subject differences, developing a generic facial expression recognition model that can reliably recognize and classify facial expressions for a large variety of individuals is difficult. Addressing these variations and

developing models resistant to individual variability is critical in video-based facial expression analysis.

- Affective computing, human-computer interaction, mental health evaluation, and emotion-driven technology are real-world uses of video-based facial expression analysis. However, employing facial expression analysis algorithms in real-world applications introduces other obstacles, such as fluctuating illumination conditions, uncontrolled environments, and real-time processing needs. It is critical to effectively implement video-based facial expression analysis in practical applications to develop accurate and efficient algorithms that can manage these real-world obstacles.
- Analyzing facial expressions in films raises ethical questions about privacy and the potential exploitation of gathered data. To ensure the safety of personal information, the storage and analysis of face data from individuals require careful consideration of privacy laws and regulations. In video-based facial expression analysis, developing solid frameworks for data anonymization and assuring informed permission and openness are critical topics to address.

3. Proposed System

3.1.GAN Method

The purpose of the GANs developed by [22] was to train a generating prototypical G to mislead the discriminator model D. The first step, known as the generator, takes in a noisy image as input and develops solutions for an image that is indistinguishable from a ground truth image. The second step, the discriminator, takes in both a crushed truth picture and a picture created and attempts to differentiate the developed appearance from the ground truth image. The following is something that can be said regarding the purpose of the contest between generator G and the discriminator D:

$$\min_{H}\max_{J}\mathop{\mathrm{E}}_{x\square \mathrm{P}_r}[\log(F(J(x))]+\mathop{\mathrm{E}}_{\tilde{x}\square \mathrm{P}_h}[\log(1-J(\tilde{x}))] \qquad (1)$$

Where P_r mentions to the delivery of the actual information x and P_h refers to the distribution of the model, which is specified by $\tilde{x}=H(z)$. The distribution of the input noise variable z is denoted by the letter P. (z). GANs have garnered much attention due to their remarkable capacity to produce photorealistic images of a high standard. On the other hand, training a GAN is complicated by issues such as vanishing gradients and mode collapse [23]. To have good training, it is critical to maintain a healthy equilibrium between Generator G and Discriminator D during the training time. Otherwise, the activity may fail to be successful. Even though a great deal of work has been put into finding solutions to those issues, the strategies that rely on innovative architectural frameworks such as DenseNet may only sometimes be successful. Arjovsky et al. [24] stated that GAN training was unsuccessful due to the JS divergence calculation. To solve this problem, they devised the idea of Wasserstein-GAN, which improved the Earthmover (EM) detachment for a cost purpose. This, in turn, completed the exercise procedure fail-proof and produced pictures of similar excellence as the unique GAN did. In addition, it has been recognized that the EM detachment between accurate and manufactured images from the discriminator is a highly essential metric of the shaped sample's excellence. In the course of our job, we make use of WGAN to direct training.

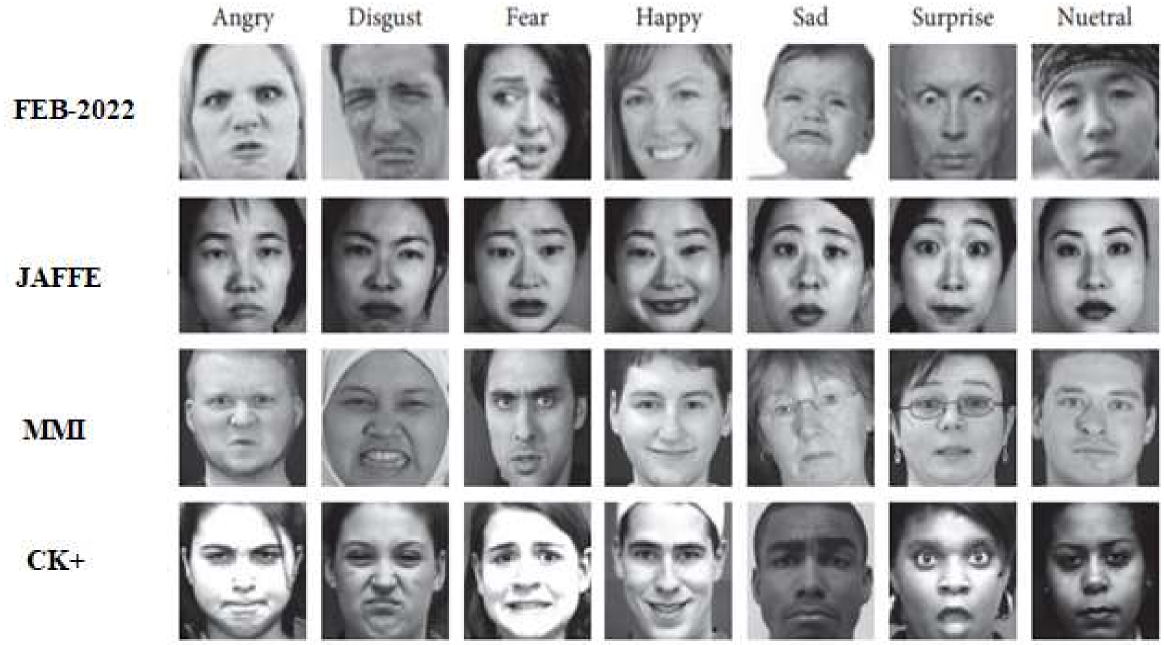

Figure 1: Sample images from four datasets

3.1.1. Generator network

The Generator Network is the central component of GAN and is responsible for producing the end output. The producer system has to obtain more specific data from the adjacent pixels to build a photo-realistic picture of high value. In this scenario, the essential component should create high-quality photographs using a fantastic architecture containing numerous deep convolutional neural networks. Fig. 2 displays the proposed figure of the GAN-SGDMA method.

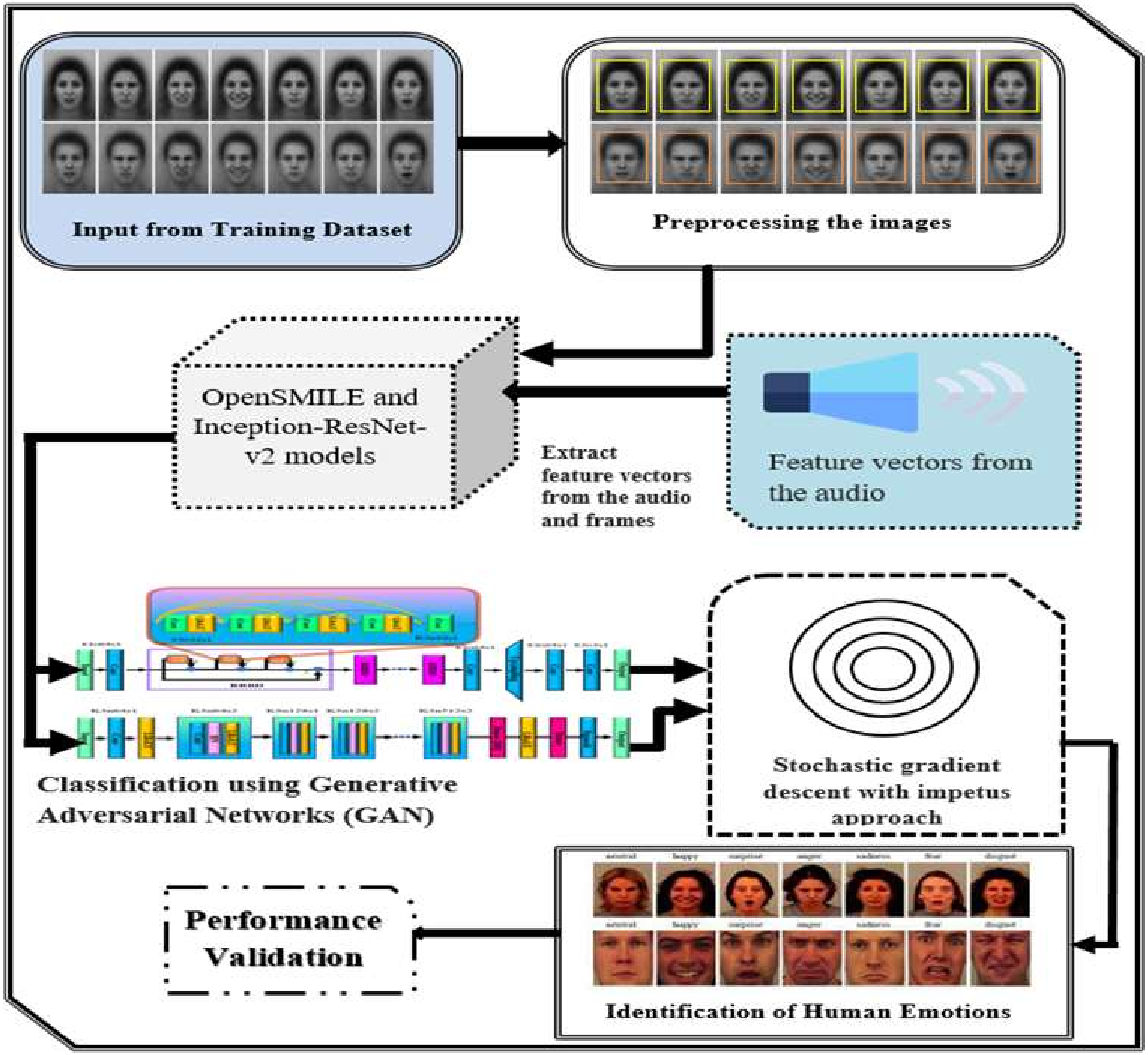
Input from Training Dataset
Preprocessing the images
OpenSMILE and Inception-ResNet-v2 models
Extract feature vectors from the audio and frames
Feature vectors from the audio
Classification using Generative Adversarial Networks (GAN)
Stochastic gradient descent with impetus approach
Performance Validation
neutral
happy
surprise
anger
sadness
fear
disgust
Identification of Human Emotions

Figure 2: Proposed diagram of GAN-SGBMA method

Fig.2 depicts the block diagram for the GAN-SGBMA technique that has been proposed. The procedure that is involved can be shown very clearly in the illustration. The training dataset is treated as input and undergoes additional preprocessing steps. Before making practical use of the data, preparation is necessary. The transformation of raw data into an organized and error-free data set is what's meant by the term "data preparation." Before the method is executed, the dataset is preprocessed so that missing values, noisy data, and other inconsistencies can be checked for and remedied. The first step in the workflow for deep learning typically involves preprocessing the data. It transforms unprocessed data into a format that the network can utilize.

After being preprocessed, the data are transmitted to Open SMILE and Inception ResNet-v2 to extract their features. The process of transforming unprocessed data into numerical features that can be used while preserving the integrity of the original data set is known as feature extraction. The term "generative adversarial network" (GAN) refers to a model in machine learning (ML) where two neural networks compete against one another to improve prediction accuracy. The data is optimized even further by using a technique known as SGDMA, which is an abbreviation for stochastic gradient descent with momentum. In the end, the data are validated to determine whether or not it is superior to other ways.

SRDenseNet [25], whose design is represented in Figure 3, places a significant amount of importance on the architecture of our Generator Network. A group normalization layer [26], a Relu beginning, and a convolution layer make up individually individual blocks. By utilizing skip connections, which mitigates the slope vanishing/exploding problem and increases property propagation in neural networks, we provender a particular layer with all the preceding layers (excluding the last layer).

Deep networks Low-level structures are extracted from the contribution picture, which is noisy, using the first convolution layer. After that, eight thick blocks are utilized to acquire knowledge of the higher-level aspects. After that, a layer known as a bottleneck, an essential component, is introduced. A 1 x 1 convolution layer is the absolute best for reducing the number of input property maps [27]. It is feasible to acquire a property fusion at a cheap calculation charge, which is one of the benefits of using such a layer. The final step is a three-by-three convolution layer used to build the production images.

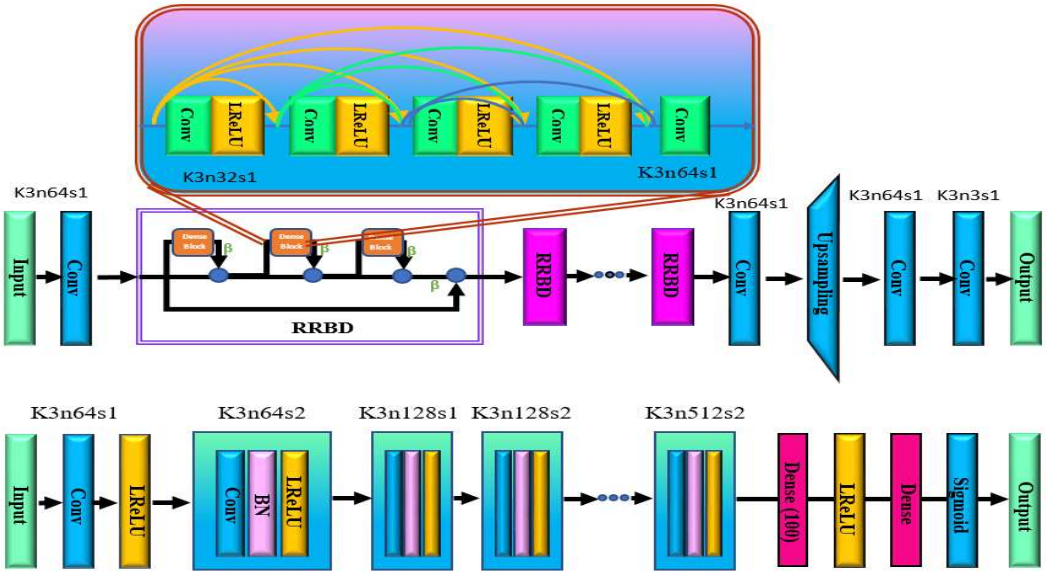

Figure 3: The generative adversarial network's architecture

3.1.2 Discriminator network

To produce a denoised image that is visually pleasing, the discriminator network's job is to determine whether or not a given input image is a genuine photograph or a computer-generated one. Therefore, the Discriminator Network ensures that the likelihood rate allocated to accurate picture information is as near to one as is practically conceivable. In contrast, the probability rate assigned to samples produced by the network should be as close to zero as is practically possible.

Our Discriminator Network is quite similar to SRGAN's in structure and operation. The LeakyReLU activation (with a value of 0.2) and Layer Normalization have been incorporated in response to the recommendation. Due to WGAN-GP, however, we substitute batch norm layers with layer norm layers in the work that we have done. It consists of eight convolutional layers, each with three times three kernels. The top dual layers are entirely linked layers that calculate the probability that a generator network generated the input image or that it was the ground truth. Another contrast is that the Sigmod activation technique is not used in the final layer since WGAN-GP is used instead.

3.1.3 Loss function

The following problems and challenges have been present throughout the history of GAN: On the one hand, the difficulties encountered during training call for careful planning of the model construction and collaboration between the many training tiers of the Generator Network and the Discriminator Network. Alternately, the harmful purpose of the Generator Network and the Discriminator Network cannot display the training process since there needs to be a relevant pointer connected with the picture excellence produced. We use the WGAN, which only makes simple trio variations compared to the traditional GAN. Using the WGAN allows us to make the training more stable.

Some things still need to be clarified between mathematical theory and the actual code implementation, even though WGAN suggests employing the Wasserstein distance as an optimization strategy for training GAN. When using the Wasserstein distance, it is necessary to fulfill the Lipschitz continuity requirement, which is a strong continuity condition. This requirement must be satisfied; thus, To ensure Lipschitz continuity, the author limits the weights to a particular range. It's crucial to be aware of these potential dangers because they can lead to unforeseen challenges like vanishing or exploding gradients. The idea is terrific, yet, the results could be better. As a result, Gulrajani et al. [28] recommend the addition of a gradient penalty word in its place. The technique allows GAN training to adapt to diverse neural network designs and only requires minor hyper-parameter changes, which favours our network.

$$\min_{H} \max_{J \in J} \underset{X \square \mathrm{P}_r}{\mathrm{E}} [J(x)] - \underset{\tilde{x} \square \mathrm{P}_h}{\mathrm{E}} [J(\tilde{x})] \tag{2}$$

Where the entire set of 1-Lipschitz functions is represented by the letter D., The objective here is to get value to be close to K W (Pr, Pθ), where K is a Lipschitz persistent, and W (Pr, Pθ) is a Wasserstein detachment. This should be a doable goal. To prevent the gradient of the Discriminator Network from being more significant than K, a slope consequence term has been included:

$$\lambda \underset{\tilde{x}\square P_{\tilde{x}}}{E}[(\| \nabla_{\tilde{x}} J(\tilde{x}) \|_2 - 1)^2] \quad (3)$$

Our overall loss function is comprised of two different types of losses: content losses and adversarial losses:

$$Loss = \lambda . Loss_{GAN} + Loss_{Content} \quad (4)$$

Where λ, for simplicity, we have set the hyperparameter to be 0.0001.

Content loss: In most cases, the material is either lost at the L1 level MAE or the L2 level MSE. We use the L2 loss, which is the alteration between the picture generated and the image considered to be the ground truth. Calculating the MSE loss per pixel looks like this:

$$Loss_{Content} = \frac{1}{WG}\sum_{x=1}^{W}\sum_{y=1}^{G}(I_{x,y}^{Orgin} - H(I^{Input})_{x,y})^2 \quad (5)$$

In this case, W and H define the picture's dimensions. The variable Origin represents the concept of the ground truth, whereas the image containing noise is denoted by the variable I Input.

Adversarial loss: In our work, the WGAN-GP performs the critic function. This is true throughout all of our projects. The following formula is used to compute the total loss to the image denoising:

$$Loss_{GAN} = -D_{WGAN,\theta}(H(I^{Input})) \quad (6)$$

Where DWGAN is the output digit of the discriminator that WGAN-GP generates to do image denoisin0067

3.2 Feature Extraction

Structures need to be extracted appropriately before training classifiers. In this investigation, OpenSMILE is used to pull out audio features, and a deep CNN model that has already been prepared and fine-tuned is used to pull out facial image features.

3.2.1 Audio Features

One of the techniques used to extract audio features is called OpenSMILE, an abbreviation for "Speech and Music Interpretation by Large-space Extraction." OpenSMILE is also the name of the program itself. The property taken from an audio example that is dual seconds long has 1582 sizes.

3.2.2 CNN Deep Features

3.2.2.1 Inception ResNet V2 network

Inception A ResNet V2 network trained using the ImageNet database is utilized for deep property removal. In addition to the start of deep convolutional design, the Inception-ResNet-v2 network operates residual connections. Inception-ResNet-v2 presently has the leading organization accurateness on the ILSVRC picture organization benchmark (top-1 accurateness: 80.4%, top-5 accuracy: 95.3%). As a consequence of this, it is capable of extracting features from photographs that contain a wide variety of subject matter. For the sake of this undertaking, the limitation rate of the ImageNet-pre-trained Inception-ResNet-v2 model will serve as the initial rate for the Inception-ResNet-v2 model. The architecture of the Discriminator System is shown in Figure 4.

However, because ImageNet has 1000 classes and this study used the seven universal sentiments as classes, the parameters in the entire associated layer of the ImageNet pre-trained prototypical are not appropriate for this prototypical. This is because this study used the seven universal emotions as classes. As an alternative to being brought back from a previously-trained prototypical, each limit in the 'AuxLogits' and 'Logits' blocks will be generated randomly instead of reinstated.

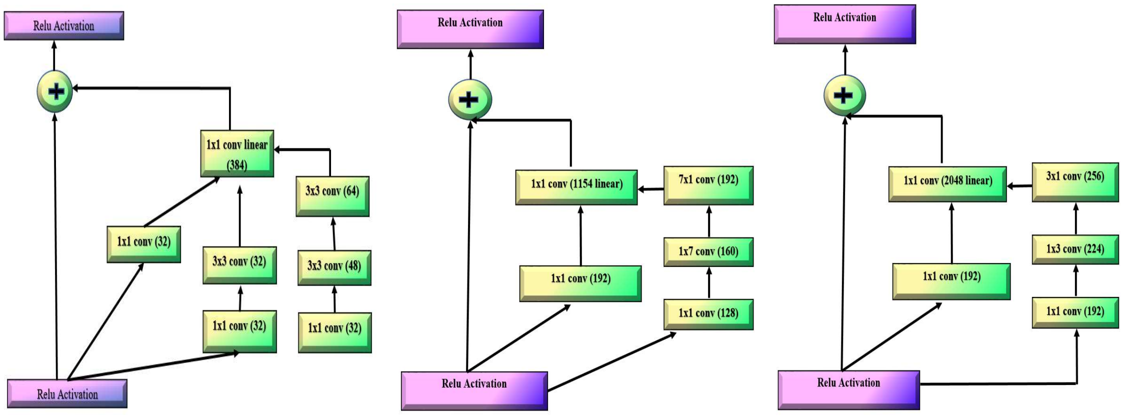

Figure 4: The structure of Inception-ResNet-v2 network blocks.

3.2.2.2 Fine-tune

To improve the Inception-ResNet-v2 model's capacity to extract facial expression characteristics, a fine-tuning procedure using the ImageNet-pre-trained Inception-ResNet-v2 model and the Facial Expression Recognition 2013 (FER2013) data set is carried out. This is accomplished by feeding the FER2013 data set into the Inception-ResNet-v2 model.

The entire pre-trained Inception-ResNet-v2 prototypical was tweaked with the FER2013 database, including all its layers. The FER2013 data collection contains a train set with 28709 images and a validated set with 7188 replications. For stages 1-30000, the learning rate will remain at 0.01 throughout. The model should then be fine-tuned for a total of 2000 steps using a learning rate of 0.0001.

Once fine-tuned, this model can distinguish between different facial expressions based on still photographs. The output layer will produce the classification outcome, while the output of the convolution layers will consist of the profound emotional characteristics of facial pictures.

3.3. Stochastic Gradient Descent with Momentum Algorithm

A method for speeding gradient vectors in the appropriate directions, leading to faster convergence, is called "momentum." This method is sometimes referred to as "SGD with momentum." It is a well-known optimization procedure utilized in training a wide variety of innovative models.

SGD *SGD with Momentum*

$$\theta_j \leftarrow \theta_j - \in \nabla_{\theta_j}\Gamma(\theta) \qquad \begin{aligned} v_{t+1} &\leftarrow pv_t + \nabla_{\theta_j}\Gamma(\theta) \\ \theta_j &\leftarrow \theta_j - \in v_{t+1} \end{aligned} \tag{7}$$

Equation 7 shows the equation for updating the weights using conventional stochastic gradient descent. The rules for weight updates based on the SGD with momentum are depicted in the equation that can be found to the right. An additional form of p times v has been added to the usual update rule to indicate momentum. Introducing this momentum element allows our gradient to acquire some state of velocity v while it is being trained. This makes intuitive sense. The velocity is calculated by running the p-weighted sum of the gradients.

One way to think of p is friction, which has a tiny retarding effect on the velocity. It is a common observation that speeds go up over time. Because of the concept of momentum, saddle points and local minimums pose significantly less risk to the gradient. Step sizes in the direction of the global minimum now depend not only on the rise of the loss function at the current position but also on the velocity that has accumulated over time. Previously, they just turned on the gradient. At a certain point, we are no longer traveling in the direction of the slope but rather in the course of the velocity. As a metaphor for a stochastic gradient descent with momentum, you can think of a ball rolling down a hill and picking up speed over time as a physical analog. Due to the high velocity with which this ball is moving, it will be able to roll over every obstacle it comes across, including holes and surfaces that do not slope downward. In this scenario, both the flat ground and the spot are saddle points, also known as local minimums of a loss function. I want to examine the differences between stochastic incline parentage with and without an impetus term. These techniques look for the point in three-dimensional space where the loss function is at its absolute minimum. It is essential to note that the momentum term reduces the variation of the gradient and the zigzag motions. In a broad sense, the momentum term stabilizes and quickens the convergence to optimal weights.

3.3.1AdaGrad

The AdaGrad optimization approach is still another option. The running sum of formed inclines is maintained at a constant value throughout the optimization. There is no momentum term; instead, there is only the equation g, which is the sum of the shaped gradients.

SGD with Momentum *AdaGrad*

$$\begin{aligned} v_{t+1} &\leftarrow pv_t + \nabla_{\theta_j}\Gamma(\theta) \\ \theta_j &\leftarrow \theta_j - \in v_{t+1} \end{aligned} \qquad \begin{aligned} g_0 &= 0 \\ g_{t+1} &\leftarrow g_t + \nabla_{\theta_j}\Gamma(\theta) \\ \theta_j &\leftarrow \theta_j - \in \frac{\nabla_{\theta_j}\Gamma(\theta)}{\sqrt{g_{t+1}} + 1e^{-5}} \end{aligned} \tag{8}$$

The term's root is subtracted from the gradient whenever one of the weight parameters is modified. Consider a loss function existing in a two-dimensional space, where the rise of the loss function in one direction is exceedingly low. In contrast, the gradient in the other direction is considerable. When the squared sum of the angles along the axis where the slopes are tiny is added together with the sum of all the gradients, the gradients get smaller. During the update step, the value obtained is very high if the current angle is divided by a tiny sum of squared gradients g. The opposite is true for the axis that has high gradient values. Consequently, we make the algorithm update in any given direction while maintaining the same proportions.

This indicates that increasing the gradient along an axis with tiny inclines speeds up the informing process. Nonetheless, informs along the axis with the significant angle are slowed down slightly. Nevertheless, this optimization approach has a drawback. Consider what would occur if the training time for the sum of gradients squared was lengthy. Over time, the scope of this phrase would expand. The weight update step becomes extremely small when the current angle is divided by such a large number. It's as if we're employing shallow learning, which gets even lower as training progresses. In the worst-case scenario, we'd be forced to use AdaGrad, and the movement would continue indefinitely.

3.3.2. RMSProp

The issue that AdaGrad is prone to be addressed by a variant of AdaGrad called RMSProp. RMSProp maintains the running sum of squared gradients, but during the algorithm's training phase, instead of allowing it to increase continuously, we will enable it to decrease gradually.

$$\begin{array}{ll} AdaGrad & RMS\ \Pr op \\ g_0 = 0 & g_0 = 0, \alpha \square 0.9 \\ g_{t+1} \leftarrow g_t + \nabla_{\theta_j} \Gamma(\theta) & g_{t+1} \leftarrow \alpha . g_t + (1-\alpha) \nabla_{\theta_j} \Gamma(\theta) \\ \theta_j \leftarrow \theta_j - \in \dfrac{\nabla_{\theta_j} \Gamma(\theta)}{\sqrt{g_{t+1}} + 1e^{-5}} & \theta_j \leftarrow \theta_j - \in \dfrac{\nabla_{\theta_j} \Gamma(\theta)}{\sqrt{g_{t+1}} + 1e^{-5}} \end{array} \qquad (9)$$

In the RMSProp algorithm, the current gradient is weighted by a parameter, mixed with the sum of squared angles, and then multiplied by a decay rate. The action that has to be taken to update RMSProp is identical to the step that needs to be taken to correct AdaGrad. In both instances, the current gradient is divided by the sum of squared angles, which causes motion along one dimension to speed up. In contrast, activity along the other dimension causes movement to slow down.

Let's compare RMSProp to SGD and then SGD with momentum to determine the appropriate weights for your portfolio. Although SGD with rate may locate the global minimum in a shorter time, this algorithm follows a significantly longer path, which may be dangerous. A more comprehensive approach enables the creation of a more significant number of saddle points and local minima. On the other hand, the RMSProp algorithm bypasses all possible side excursions en route to arriving at the local minimum of the loss function.

3.3.3. Adam Optimizer

Until this point, we have relied on the moment term to determine the velocity of the gradient so that we may adjust the weight parameter according to the direction of this velocity. We scaled the current rise using the sum of the squared angles in the case of AdaGrad and RMSProp. This allowed weight updates with the same ratio in each dimension. These two approaches are very innovative in their execution.

$$m_0 = 0, v_0 = 0$$

$$m_{t+1} \leftarrow \beta_1 m_t + (1-\beta_1)\nabla_{\theta_j}\Gamma(\theta) \quad \textit{Momemtum} \tag{10}$$

$$v_{t+1} \leftarrow \beta_1 v_t + (1-\beta_1)\nabla_{\theta_j}\Gamma(\theta)^2 \quad RMS\ \mathrm{Pr}\,op \tag{11}$$

$$\theta_j \leftarrow \theta_j - \in \frac{\nabla_{\theta_j}\Gamma(\theta)}{\sqrt{g_{t+1}} + 1e^{-5}} \qquad RMS\ \mathrm{Pr}\,op + Momemtum \tag{12}$$

The first equation is similar to the SGD with momentum. The terms, in this case, would be velocity and friction. In Adam's case, the initial rate is merely a hyperparameter. The factor (1- ***β1***) the present incline increases the difference between SGD with and SGD without momentum. The second part of the equation is RMSProp, where we keep the consecutive sum of shaped angles. There is also a factor (1-***β2***) that is multiplied by the gradient squared. This is the second momentum used in the equation and functions as a hyperparameter. The RMSProp and SGD growth rates and speed are both components of the ultimate update equation. Up to this point, Adam has accomplished a successful blending of the benefits of the two previous optimization strategies. The problem is that the initialization of the first and second momentum terms needs to be corrected. Following the initial update of the dual momentum, this term is extremely close to zero. A minimal double momentum factor v is divided by the previous equation to update the weight parameters, which results in a relatively high first update step. This extraordinarily significant initial update was produced because the first and second momentum were reset to zero. Adam adds the following correction clause to deal with the problem of first update steps that take too long:

$$m_{t+1} \leftarrow \beta_1 m_t + (1-\beta_1)\nabla_{\theta_j}\Gamma(\theta) \quad \textit{Momemtum} \tag{13}$$

$$v_{t+1} \leftarrow \beta_1 v_t + (1-\beta_1)\nabla_{\theta_j}\Gamma(\theta)^2 \quad RMS\ \mathrm{Pr}\,op \tag{14}$$

$$\overset{\square}{m}_{t+1} \leftarrow \frac{m_{t+1}}{1-\beta_1^{t}} \qquad \textit{Bias Correction}$$

$$\hat{v}_{t+1} \leftarrow \frac{v_{t+1}}{1-\beta_2^{t}} \tag{15}$$

$$\theta_j \leftarrow \theta_j - \in \frac{\nabla_{\theta_j}\Gamma(\theta)}{\sqrt{g_{t+1}} + 1e^{-5}} \qquad RMS\ \mathrm{Pr}\,op + Momemtum \tag{16}$$

By taking into account the most recent time step in calculating the initial update of the first and second momentums, we can get an estimate of these momentums that is free from bias. Because of these correction factors, the first and second momentum values are initially more significant than they would have been in the scenario where the bias correction was not applied. As a direct consequence, the first update step of the neural network parameters is not all that significant. Consequently, our training is not corrupted at an early stage. The Adam Optimizer in its complete form can now be constructed thanks to the extra bias adjustments. Now that we have all the algorithms let's evaluate them against one another based on how well they locate the global minimum of the loss function.

4. Result and Discussion

4.1 Dataset and data preparation

The following database was utilized to demonstrate how generalizable the deep learning prototypical is.

4.1.1. Training data

Pictures from DIV2K, a contest for the most excellent possible resolve achieved by a single image, were used to compile the training information set. Included in this package are a total of 800 photos. Due to a scarcity of training and evaluation datasets for single-picture denoising, we chose a 6464 crop from the original images at random for training. The range [-1,1] is applied to all pixels. Three different levels of Gaussian noise are added during the training process to generate noisy photos. The noisy images come in as input, and the original pictures come out as the ground truth.

4.1.2. Test Data

Perform a study of the prototypical presentation utilizing both the Set 5 and Set 14 datasets. After applying some Gaussian noise to them, which was done to evaluate picture super-resolution, these pictures may now be only incorporated into our model. The picture-denoising training can then be finished with the generation of denoised imageries.

4.2. Experimental Results

In the research on facial emotion recognition, the primary metrics investigated were precision, recall, execution time, peak signal-to-noise ratio (PSNR), and accuracy. Root mean square mistake was also investigated (RMSE). To verify that our technique is not required, we put it through its paces in competition against more conventional methods of recognizing facial expressions and against a neural network series of processes that recognize facial expressions. A comparison was made between the CNN, R-CNN, LSTM, and RNN approaches. During the period of training and tuning, the individual system was trained individually without the gratitude component so that the overall performance of the particular method could be evaluated.

4.3. Performance Metrics

Precision is a legal assessment metric, especially when the planned machine learning prototype must be validated using the predicted and actual results. It determines the proportion of expected positives that correspond to definite positives. Consequently, it is dependent on TP and FP standards. When it is essential to establish the total number of positives that may be anticipated with a level of fair accuracy,

$$\Pr ecision = \frac{TP}{TP+FP} \tag{17}$$

Another helpful evaluation indicator is recalled, which indicates the percentage of correctly classified positives. The TP and FP values are what are used to measure recall.

$$\text{Re}\,call = \frac{TP}{TP+FN} \tag{18}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{19}$$

1.Precision

Table 1: Precision Analysis of GAN-SGDMA method with existing systems

No Of Data From Dataset	CNN	R-CNN	LSTM	DBN	RNN	GAN-SGDMA
50	67.11	70.32	74.62	78.43	83.13	90.11
100	67.89	70.84	74.11	79.12	84.45	91.45
150	68.12	71.45	75.87	79.53	85.12	92.61
200	68.45	72.36	75.99	80.13	86.33	92.81
250	69.72	73.12	76.12	81.45	87.12	93.15
300	69.91	73.97	76.45	82.62	88.43	94.32

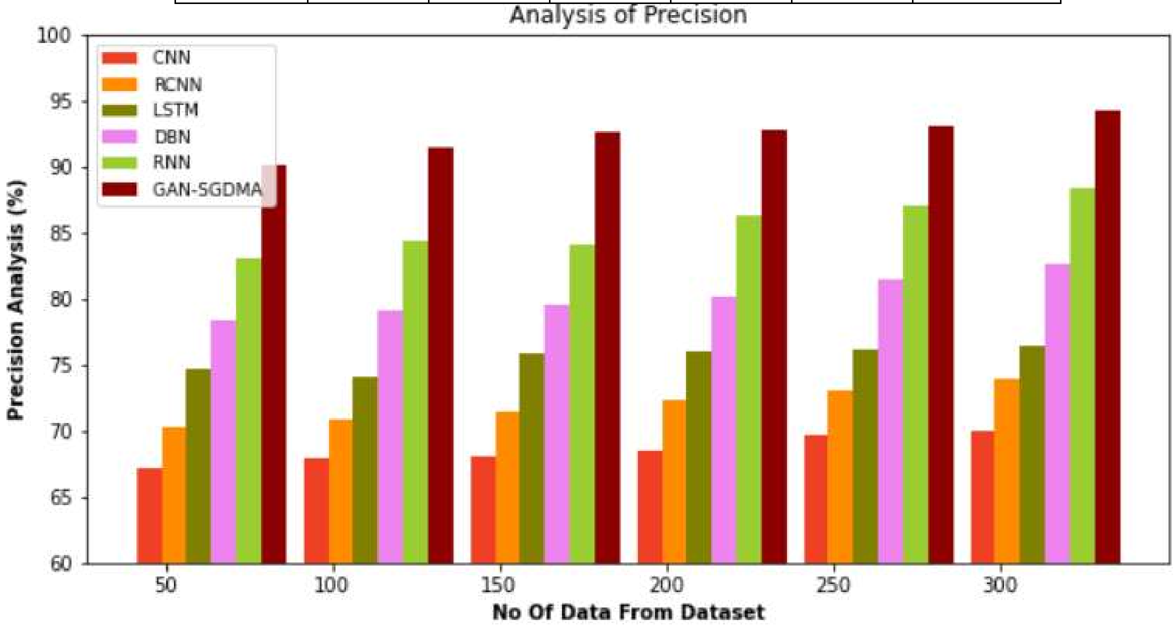

Figure 5: Precision Analysis of GAN-SGDMA method with existing systems

Fig.5 and Tab.1 compare the precision of the GAN-SGDMA approach to that of other existing methods. The results show that the machine-learning approach surpasses different systems in terms of precision. When considering 50 data points, the GAN-SGDMA technique achieves a precision value of 90.11%, outperforming the CNN, RCNN, LSTM, DBN, and RNN models, which reach accuracy values of 67.11%, 70.32%, 74.62%, 78.43%, and 83.13%, respectively. Notably, the GAN-SGDMA prototype consistently outperforms other prototypes across various data set sizes. Similarly, with 300 data points, GAN-SGDMA achieves a precision value of 94.32%, beating the CNN, RCNN, LSTM, DBN, and RNN models, which produce accuracy values of 69.91%, 73.97%, 76.45%, 82.62%, and 88.43%, respectively.

2.Recall

Table 2: Recall Analysis of GAN-SGDMA method with existing systems

No Of Data From Dataset	**CNN**	**R-CNN**	**LSTM**	**DBN**	**RNN**	**GAN-SGDMA**
50	70.12	73.65	80.11	82.62	85.15	91.42
100	70.36	73.89	80.25	82.15	86.72	92.11
150	70.11	74.12	80.43	83.46	87.32	93.45
200	71.45	74.65	80.61	83.87	88.13	93.61
250	72.12	74.86	81.72	84.12	89.71	94.82
300	72.89	74.92	81.91	84.66	90.16	95.66

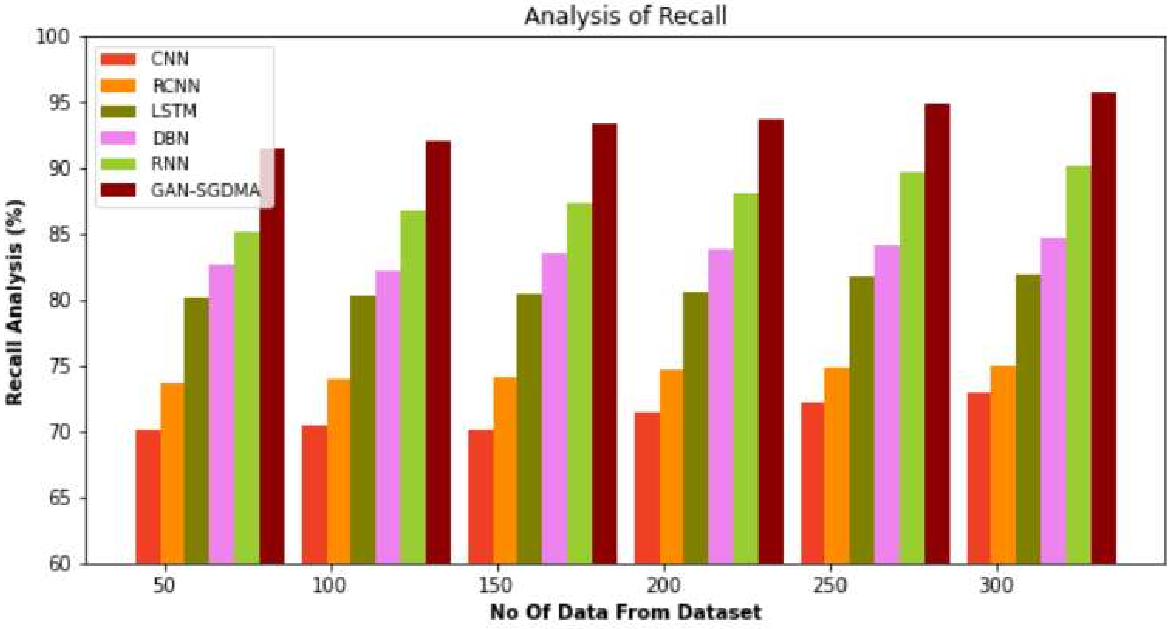

Figure 6: Recall Analysis of GAN-SGDMA method with existing systems

Figure 6 and Table 2 compare the GAN-SGDMA technique to other existing methods in terms of recall performance. The results show that the machine learning approach performs better with higher recall rates. GAN-SGDMA achieves a recall value of 91.42% when examining 50 data points, outperforming other models such as CNN (70.12%), RCNN (73.65%), LSTM (80.11%), DBN (82.62%), and RNN (85.15%). This demonstrates the usefulness of GAN-SGDMA in improving recall outcomes. Furthermore, GAN-SGDMA regularly performs excellently across a wide range of data volumes. GAN-SGDMA, for example, achieves a recall value of 95.66% when reviewing a dataset of 300 data points. CNN, RCNN, LSTM, DBN, and RNN models have recall rates of 72.89%, 74.92%, 81.91%, 84.66%, and 90.16%, respectively. These results demonstrate GAN-SGDMA's durability and exceptional performance across various data volumes.

3. Peak signal-to-noise ratio (PSNR)

Table 3: PSNR Analysis of GAN-SGDMA method with existing systems

No of Data From Dataset	CNN	RCNN	LSTM	DBN	RNN	GAN-SGDMA
100	67.834	70.244	73.832	78.435	79.948	81.632
200	68.133	70.627	73.432	76.995	78.960	81.732
300	67.437	71.222	74.656	78.831	79.900	83.465
400	68.956	72.759	75.162	78.463	80.432	84.500
500	69.327	72.351	76.822	78.822	81.300	83.651
600	69.748	71.847	77.165	79.932	81.435	85.732

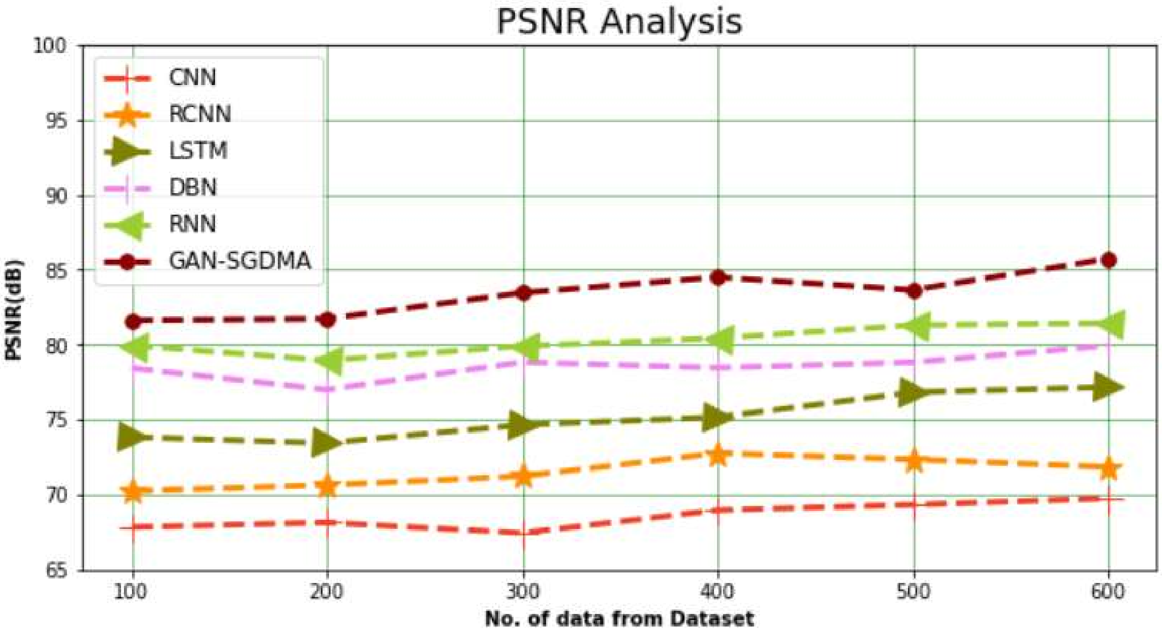

Figure 7: PSNR Analysis of GAN-SGDMA method with existing systems

When the PSNR performance of the GAN-SGDMA technique is compared to that of other proven methods, Figure 7 and Table 3 show that the machine learning approach regularly outperforms the others. GAN-SGDMA, for example, achieves a PSNR value of 81.632dB when using 100 data points, outperforming other models such as CNN (67.834dB), RCNN (70.24dB), LSTM (73.832dB), DBN (78.435dB), and RNN (79.948dB). Notably, the performance of the GAN-SGDMA prototype varies significantly across dataset sizes. With 600 data points, GAN-SGDMA achieves a PSNR of 85.732dB, outperforming CNN (69.748dB), RCNN (71.847dB), LSTM (77.165dB), DBN (79.93dB), and RNN (81.435dB) models.

4. Accuracy

Table 4: Accuracy Analysis of GAN-SGDMA method with existing systems

No Of Data From Dataset	CNN	R-CNN	LSTM	DBN	RNN	GAN-SGDMA
50	70.32	75.11	77.14	79.92	85.13	92.33
100	71.54	75.39	77.53	81.14	86.54	93.66
150	73.65	75.63	78.12	82.35	88.73	94.11
200	74.15	76.15	79.43	82.64	89.11	95.32
250	74.83	76.84	79.11	83.72	90.43	96.54
300	74.91	76.91	79.60	84.20	91.11	97.23

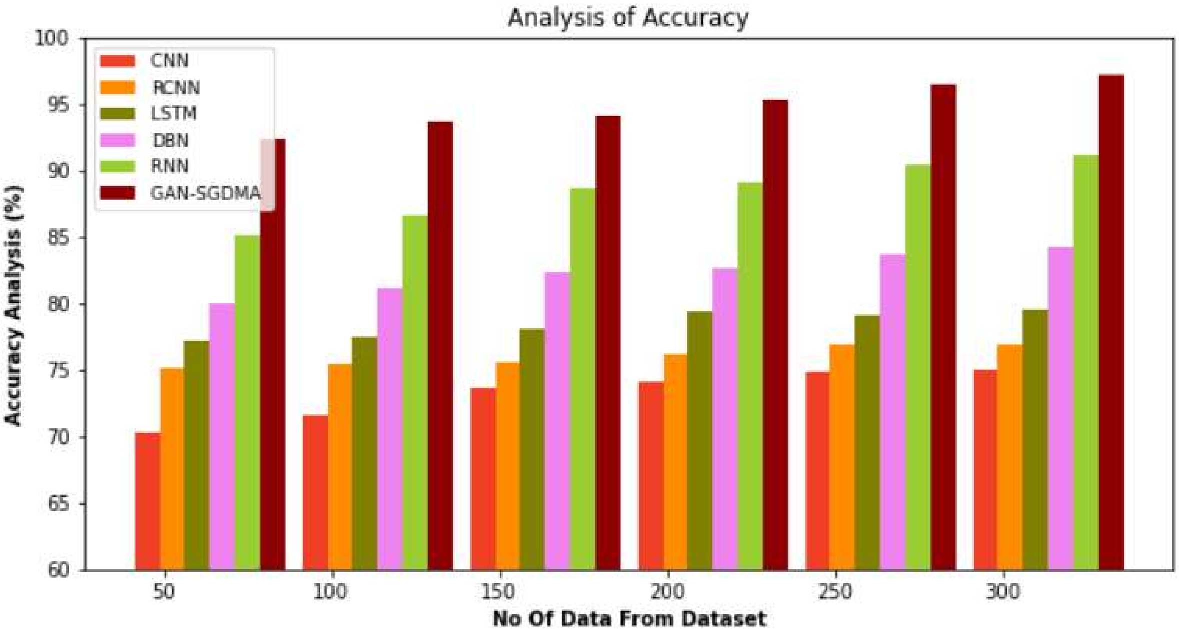

Figure 8: Accuracy Analysis of GAN-SGDMA method with existing systems

Fig.8 and Tab.4 compare the GAN-SGDMA approach to other existing methods in terms of proportional correctness. The results show that the machine learning strategy outperformed the different approaches regarding accuracy. When evaluating 50 data points, GAN-SGDMA achieves an accuracy of 92.33%, whereas CNN, RCNN, LSTM, DBN, and RNN models reach accuracies of 70.32%, 75.11%, 77.14%, 79.92%, and 85.13%, respectively. Notably, the GAN-SGDMA model performs admirably across various data set sizes. Similarly, with 300 data points, GAN-SGDMA achieves an accuracy of 97.23%, while the CNN, RCNN, LSTM, DBN, and RNN models reach accuracies of 74.91%, 76.91%, 79.60%, 84.20%, and 91.11%, respectively.

5. Execution Time

Table 5: Execution Time Analysis of GAN-SGDMA method with existing systems

No Of Data From Dataset	CNN	R-CNN	LSTM	DBN	RNN	GAN-SGDMA
50	9.321	8.362	6.732	5.680	4.839	2.162
100	9.456	8.565	6.932	5.732	4.932	2.132
150	9.532	8.676	7.132	5.839	4.969	3.435
200	9.548	8.832	7.666	5.132	5.132	2.111
250	9.652	8.965	7.932	6.399	5.322	4.732
300	9.731	8.979	8.435	6.562	5.612	4.831

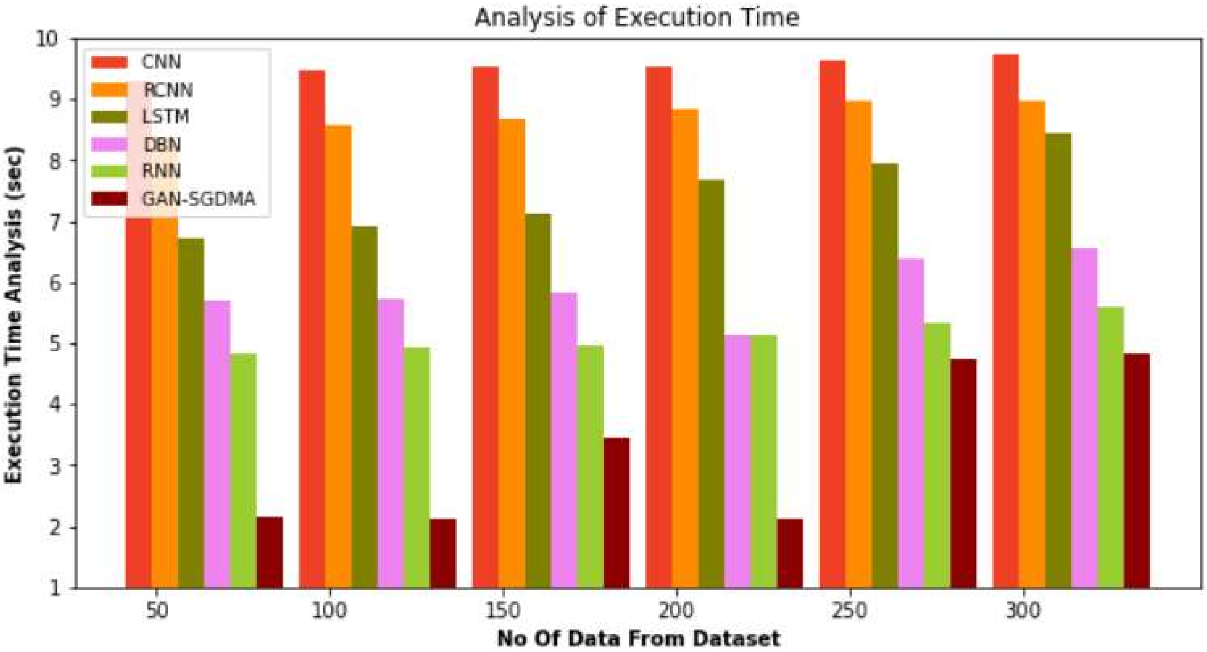

Figure 9: Execution Time Analysis for GAN-SGDMA method with existing systems

Fig. 9 and Tab. 5 describe the execution time analysis of the GAN-SGDMA technique with existing methods. The data shows that the GAN-SGDMA method has outperformed the other techniques. For example, with 50 data points, the GAN-SGDMA method has taken only 2.162 sec to execute, while the different existing techniques like CNN, RCNN, LSTM, DBN, and RNN have an execution time of 9.321 sec, 8.362 sec, 6.732 sec, 5.680 sec, and 4.839 sec, respectively. Similarly, for 300 data points, the GAN-SGDMA method has an execution time of 4.831 sec, while the other existing techniques like CNN, RCNN, LSTM, DBN, and RNN have 9.731 sec, 8.979 sec, 8.435 sec, 6.562 sec, and 5.612 sec of execution time, respectively.

6. RMSE

Table 6: RMSE Analysis of GAN-SGDMA method with existing systems

No of Data From Dataset	CNN	RCNN	LSTM	DBN	RNN	GAN-SGDMA
50	48.632	44.981	40.132	35.500	32.635	30.132
100	49.731	45.651	41.351	36.132	32.835	30.435
150	49.611	46.132	41.632	37.432	33.132	31.632
200	49.832	46.732	42.356	38.962	33.465	31.762
250	50.652	46.832	43.731	39.432	34.652	31.785
300	51.733	47.322	43.832	39.621	34.321	31.981

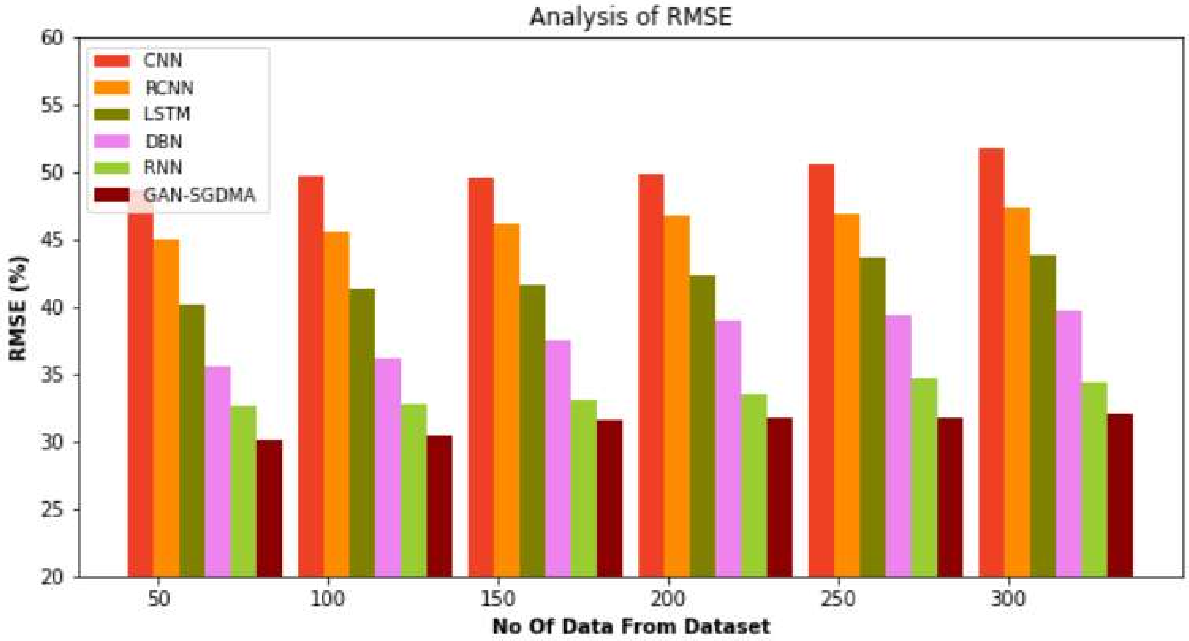

Figure 10: RMSE Analysis of GAN-SGDMA method with existing systems

An RMSE comparison of the GAN-SGDMA strategy with various known approaches is shown in Fig. 10 and Tab. 6. The graph illustrates how the machine learning strategy produced a better performance with a reduced RMSE value. For instance, the GAN-SGDMA model's RMSE value for data set 50 is 30.132%, while the RMSE values for the CNN, RCNN, LSTM, DBN, and RNN models are slightly higher at 48.632%, 44.981%, 40.132%, 35.500%, and 32.635%, respectively. However, the GAN-SGDMA model has shown maximum performance for different data sizes with low RMSE values. Similarly, under 300 data points, the RMSE value of GAN-SGDMA is 31.981%, while it is 51.733%, 47.322%, 43.832%, 39.621%, and 34.321% for CNN, RCNN, LSTM, DBN, and RNN models, respectively.

5. Conclusion

This paper developed a method that uses Generative Adversarial Networks (GAN)-based neural networks to classify each facial picture obtained from a frame into one of seven categories of facial expressions. These categories are as follows. Videos have their informative content, such as audio, single video frames, and multiple video frames, removed to be used to portray a range of emotions.

To extract feature vectors from audio and structures in a distinct manner, the OpenSMILE and Inception-ResNet-v2 models are utilized. Thirdly, using stochastic gradient descent with momentum technique, multiple models are trained to classify emotions (SGDMA). A table is constructed using the findings from each photograph to categorize which facial expression has been recognized the most frequently during the movie. To classify audio feature vectors, GAN-SGDMA is utilized. Inception-ResNet-v2 is used to acknowledge feelings conveyed by still photographs. Video emotion identification will be one of the next steps in investigating this topic. This complex problem involves many subtasks, such as detecting faces, tracking faces, recognizing faces, transforming faces into three dimensions, and other tasks. The investigation of the main task will be aided by any progress made in those subsidiary activities. None of the above subtasks can be considered state-of-the-art because of the constraints imposed by this project regarding both time and resources. A more advanced face detection technology combined with face recognition can significantly improve the results compared to the other subtasks mentioned below. This would only require distinguishing between people in a single video and appropriately titling their feelings. Future research may extend the unsupervised DPL to a semi-supervised version. This is because including real-world samples labeled with ground truth in the target dataset may enable a more accurate estimation of target data. It will also be intriguing to apply our technique to the difficulties presented by other domains of adaptation.

References

1. Boughrara H, Chtourou M, Amar CB et al (2016) Facial expression recognition based on a mlp neural network using constructive training algorithm[J]. Multimed Tools Appl 75(2):709–731

2. 4. Guo Y, Zhao G, Pietikainen M (2016) Dynamic facial expression recognition with atlas construction and sparse representation. IEEE Trans Image Process 25(5):1977–1992.

3. G. R. Alexandre, J. M. Soares, and G. A. Pereira 'e, “Sys- ´ tematic review of 3D facial expression recognition methods,” Pattern Recognition, vol. 100, Article ID 107108, 2020.

4. [3] S. A. Khan, A. Hussain, and M. Usman, “Facial expression recognition on real world face images using intelligent techniques: a survey[J],” Optik, vol. 127, no. 15, pp. 6195– 6203, 2016

5. S. Li and W. Deng, “Deep facial expression recognition: a survey,” IEEE transactions on affective computing, vol. 99, p. 1, 2020.

6. Y. Miao, “Improved deep neural network for cross-media visual communication,” Computational Intelligence and Neuroscience, vol. 2022, p. 1556352, Article ID 1556352, 2022

7. Atienza R (2020) Advanced deep learning with tensorFlow 2 and keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more. Packt Publishing Ltd

8. Jolicoeur-Martineau, A. The relativistic discriminator: A key element missing from standard GAN. arXiv 2018, arXiv:1807.00734.

9. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv 2015, arXiv:1511.06434.

10. Lefkimmiatis S (2018) Universal denoising networks: a novel CNN architecture for image denoising. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3204–3213

11. Li Y, Zeng J, Shan S, Chen X (2018) Occlusion aware facial expression recognition using cnn with attention mechanism. IEEE Trans Image Process 28(5):2439–2450

12. Wen Z, Lin W, Wang T, Xu G (2021) Distract your attention: multi-head cross attention network for facial expression recognition, arXiv:2109.07270

13. Ismail A et al (2021) A new deep learning-based methodology for video deepfake detection using XGBoost. Sensors 21(16):5413 Ismail A, Elpeltagy M, Zaki M, ElDahshan KA (2021) Deepfake video detection: YOLO-face convolution recurrent approach. PeerJ Comput Sci 7:e730. https://doi.org/10.7717/peerj-cs.730

14. Fadl S, Han Q, Qiong L (2020) Exposing video inter-frame forgery via histogram of oriented gradients and motion energy image. Multidimens Syst Signal Process 31(4):1365–1384

15. Güera D, Delp EJ (2018) Deepfake video detection using recurrent neural networks. In: 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS), IEEE, pp 1–6.

16. Sabir E, Cheng J, Jaiswal A, AbdAlmageed W, Masi I, Natarajan P (2019) Recurrent convolutional strategies for face manipulation detection in videos. Interfaces (GUI) 3(1):80–87

17. Hung BT (2021). Face recognition using hybrid HOG-CNN approach. In: Research in intelligent and computing in engineering, Springer, Singapore, pp 715–723

18. Korshunov P, Marcel S (2018) Speaker inconsistency detection in tampered video. In: 2018 26th European signal processing conference (EUSIPCO), IEEE, pp 2375–2379.

19. Y. Li, J. Zeng, S. Shan, and X. Chen, ''Patch-gated CNN for occlusionaware facial expression recognition,'' in Proc. 24th Int. Conf. Pattern Recognit. (ICPR), Aug. 2018, pp. 2209–2214.

20. D. K. Jain, P. Shamsolmoali, and P. Sehdev, ''Extended deep neural network for facial emotion recognition,'' Pattern Recognit. Lett., vol. 120, pp. 69–74, Apr. 2019.

21. K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, ''Region attention networks for pose and occlusion robust facial expression recognition,'' IEEE Trans. Image Process., vol. 29, pp. 4057–4069, Jan. 2020.

22. Goodfellow I, Pouget-Abadie J, Mirza M et al (2017) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680.

23. Salimans T, Goodfellow I, Zaremba W et al (2016) Improved techniques for training GANs. In: Advances in neural information processing systems, pp 2234–2242

24. Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." *International conference on machine learning*. PMLR, 2017.

25. Tong T, Li G, Liu X et al (2017) Image super-resolution using dense skip connections. In: IEEE international conference on computer vision (ICCV). IEEE, pp 4809–4817

26. Ioffe S, Szegedy C (2016) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp 448–456

27. Salimans T, Goodfellow I, Zaremba W et al (2016) Improved techniques for training GANs. In: Advances in neural information processing systems, pp 2234–2242.

28. Gulrajani I, Ahmed F, Arjovsky M et al (2017) Improved training of Wasserstein GANs. In: Advances in neural information processing systems, pp 5769–5779.

29.