Towards Interpretable Facial Emotion Recognition

Sarthak Malik
Indian Institute of Technology
Roorkee India
sarthak_m@mt.iitr.ac.in

Puneet Kumar
Indian Institute of Technology
Roorkee India
pkumar99@cs.iitr.ac.in

Balasubramanian Raman
Indian Institute of Technology
Roorkee India
bala@cs.iitr.ac.in

ABSTRACT

In this paper, an interpretable deep-learning-based system has been proposed for facial emotion recognition. A novel approach to interpret the proposed system's results, Divide & Conquer based Shapley additive explanations (DnCShap), has also been developed. The proposed approach computes 'Shapley values' that denote the contribution of each image feature towards a particular prediction. The Divide and Conquer algorithm has been incorporated for computing the Shapley values in linear time instead of the exponential time taken by the existing interpretability approaches. The experiments performed on four facial emotion recognition datasets, i.e., FER-13, FERG, JAFFE, and CK+, resulted in the emotion classification accuracy of 62.62%, 99.68%, 91.97%, and 99.67%, respectively. The results show that DnCShap has consistently interpreted the highly relevant facial features for the emotion classification for various datasets.

CCS CONCEPTS

• **Computer systems organization** → *Neural networks*; • **Computing methodologies** → Supervised learning by classification; **Cognitive science**; **Machine learning**; **Computer vision**; • **Information systems** → *Sentiment analysis.*

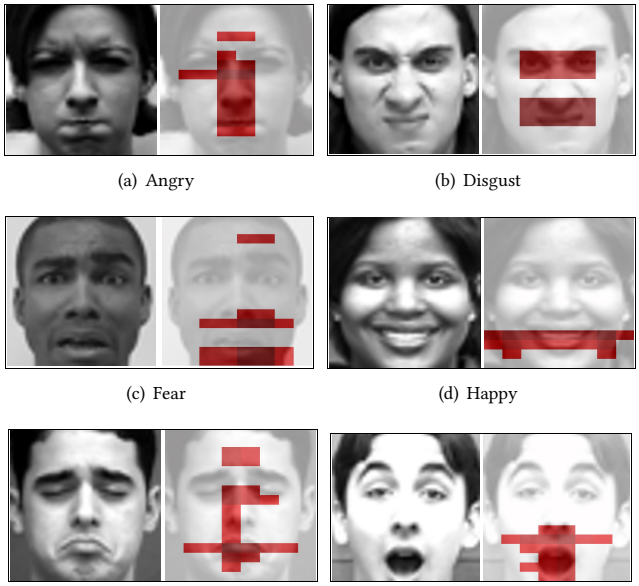
KEYWORDS

Facial Expressions, Emotion Recognition, Deep Network Interpretability, Computer Vision, Affective Computing.

ACM Reference Format:
Sarthak Malik, Puneet Kumar, and Balasubramanian Raman. 2021. Towards Interpretable Facial Emotion Recognition. In *Proceedings of 12th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'21),* Chetan Arora, Parag Chaudhuri, and Subhransu Maji (Eds.). ACM, New York, NY, USA, Article 14, 9 pages. https://doi.org/10.1145/3490035.3490271

1 INTRODUCTION

Today, machine learning (ML) has a more significant impact on day-to-day life. Deep neural networks (DNN) have demonstrated even more significant potential in solving problems related to computer vision, emotion analysis, medical imaging, language processing, and speech analysis. However, one of the biggest challenges associated with the use of DNNs is their 'Black-box' nature because of the lack of explanations of their predictions [45]. While machine learning models are used in an increasing number of fields, DNNs are at a disadvantage because of being less interpretable [26]. The use of DNNs in essential applications related to medical diagnostics, legal judgments, decision support systems, etc., may involve a considerable risk if we have no idea on 'Why?' and 'how?' we get a specific prediction [33]. This fact makes the DNNs' interpretability significantly important. When we interpret a model, we can account for Fairness, Accountability, and Transparency in the model's predictions which can help build this trust in them [18]. In this paper, we have focused on the interpretability of DNNs for their applications in Facial Emotion Recognition (FER), which is an important application in Affective Computing and Computer Vision. The problem of explainable facial emotion recognition is illustrated in Fig. 1.



(a) Angry (b) Disgust (c) Fear (d) Happy (e) Sad (f) Surprise

Figure 1: Illustration of the problem. The goal is to compute the most important visual features for recognizing the emotions portrayed by the images containing facial expressions.

The problem of interpretable ML models and DNNs has gained much interest in recent times, and it has emerged as a new research area, 'Interpretable Artificial Intelligence (XAI)' [33]. In this direction, DNNs interpretability has been explored for various research domains such as computer vision, medical imaging, natural language processing, etc. [1, 9, 55]. However, only a handful of DNN explainability approaches exist in the literature applicable to FER. The most commonly used interpretability approaches are SHAP [30], LIME [35], and Grad-CAM [36]. They assign the importance

ICVGIP'21, December 2021, Jodhpur, India

ACM ISBN 978-1-4503-7596-2.
https://doi.org/10.1145/3490035.3490271

scores to every feature according to specific rules. SHAP uses 'Shapley values,' which is a solution concept inherited from cooperative game theory [37]. However, it is computationally expensive and takes exponential computation time for the computation of exact Shapley values. In this paper, we have developed a computationally efficient FER interpretability approach, DnCShap, and compared its performance against the aforementioned approaches.

The proposed FER interpretability approach, DnCShap, is a faster way to compute approximated Shapley values that denote the contribution of each image feature towards a particular prediction. The Divide and Conquer algorithm has been incorporated with the SHAP algorithm that enabled the proposed approach to compute the Shapley values in linear time instead of the theoretical requirement of exponential time. The experiments have been performed on four FER datasets, and the proposed approach's performance has been compared with the existing FER methods and interpretability techniques. It has resulted in the emotion classification accuracy of 62.62%, 99.68%, 91.97%, and 99.67% for FER-13, FERG, JAFFE, and CK+, respectively. The facial features contributing towards the emotion recognition were computed faster by the proposed approach. However, their relevance was comparable to the features discovered by the SHAP algorithm.

The contributions of this paper are summarized as follows:

- A deep-learning-based facial emotion recognition system has been proposed to classify a given facial image into discrete emotion classes, i.e., 'Anger,' 'Disgust,' 'Fear,' 'Happy,' 'Neutral,' 'Sad,' and 'Surprise.'
- A novel approach to interpreting the working of a deep neural network, 'DnCShap,' has been developed and applied to the proposed facial emotion recognition system. The developed interpretability technique identifies the highly relevant parts of the image that contribute the most to recognizing the emotion classes.
- We have also incorporated the 'Divide and Conquer' algorithm with the proposed DnCShap approach, which has enabled it to compute the approximated Shapley values for the input images in linear time instead of the exponential time taken by the existing SHAP algorithm.

2 RELATED WORK

The recognition of emotions portrayed through facial expressions is a well-known research area. Various research attempts have been made in this direction [3, 10, 15, 19, 31]. While traditional FER methods focused on extracted and processing handcrafted features, Deep Learning-based end-to-end FER approaches attempt to explain and interpret the underlying architectures as well [42]. With that in mind, this section surveys the state-of-the-art FER approaches and various interpretability methods.

2.1 Explainable Facial Expression Recognition

The pioneer works of Paul Ekman [13, 14] serves as the reference for deciding the discrete emotion classes in which researchers attempt to classify a given image. Seven basic emotion classes, i.e., 'Happy,' 'Sad,' 'Fear,' 'Disgust,' 'Surprise,' 'Anger,' and 'Neutral' have been considered in this context.

Traditional FER approaches follow a three-step process involving pre-processing, Feature extraction, and Classification [8]. The pre-processing phase is most commonly performed using the histogram equalization method because of its capability of conquering any illumination-related variations in the images [19, 47]. Feature extraction has been achieved using various techniques such as histogram of oriented gradients [21], local binary patterns [7], and Gabor wavelets [4]. However, these approaches have limited usage in the presence of intra-class variations such as partial removal of few facial features, use of glasses, etc.

Deep Learning-based FER approaches have been able to overcome the aforementioned challenge of handling intra-class variations in the input data [48]. In this context, Khorrami et al. [24] demonstrated better FER accuracy for Cohn-Kanade and the Toronto Face datasets using a deep convolutional neural network with zero-bais. In another work, Aneja et al. [3] devised a Deep Learning-based model of facial expressions for animated characters and incorporated the modeling of human faces, animated images, and human to animated images' mapping. The attention mechanism-based DNNs have also been used to compute the importance of the salient features for the input images [31].

In the context of interpreting the working of the DNNs deployed for FER, Zhou et al. [56] developed a deep regression network to identify the severity scores of the input image's pixels towards portraying depressive emotions. Various interpretability techniques have been developed for understanding the dynamics that occur during convolutional-based prediction tasks [22, 44, 55]. There has been significant progress in making the interpretable systems for the generic DNNs. However, the interpretability can be explored in more depth in the context of FER. With that inspiration, we have explored various interpretability methods available in the literature and developed a new efficient approach for FER interpretability.

2.2 Attribution-based Interpretability methods

Attribution methods attempt to assign the relevance, also known as the attribution values, to each input feature of the network for a given input. 'Shapley Values' [37] is an example of frequently used attribution values in this context. In this context, Lundberg et al. [30] used Shapley values and developed an interpretability system known as SHAP (SHAPley Additive exPlanations). The SHAP system calculates the contribution of each feature towards the prediction by calculating its Shapley values. For calculating Shapley values for a network containing n features, a total of 2^n models need to be trained, which turns out to be exponential in the number of features, making it computationally very expensive [6, 30]. Various approximation-based methods have been developed to fasten up the computation of Shapley values, such as Shapley values sampling [6] and KernelSHAP [30].

Lately, the attribution-based methods have gained much importance for analyzing the local interpretability. Local interpretability focuses on explaining a single instance instead of explaining the general model [12]. Formally local interpretability oriented attribution-methods assign a scalar value to each feature called the 'importance of the feature' or its 'contribution' toward the output. Over the last decade, various attribution methods are being developed. For instance, Lundberg et al. [29] proposed a tree-based attribution

method, TreeSHAP, that calculates the Shapley values for tree-based machine learning models. However, it is model-specific and can not be used for non-tree-based models. Furthermore, a few more attribution-based interpretability methods have been developed specifically for deep neural networks, such as Deeplift, GradientExplainer, and DeepExplainer [30, 39]). The attribution-based methods have two types based on perturbation and backpropagation, which are explored in the following sections.

2.3 Perturbation-based Interpretability

Perturbation-based methods study the effect of a slight change in the input and try to interpret it based on the insights obtained [16, 17, 51]. Local Interpretable Model-agnostic Explanations (LIME) [35] is the most commonly used perturbation-based method. For explaining a new instance, LIME generates a new dataset by perturbing the given instance that is to be explained. It calculates the output by passing from the original model. LIME then trains an interpretable model on this new perturbed data, which is weighted by the closeness of perturbed data point from our original instance. The learned model's weights approximately tell about the importance or contribution of each feature. LIME is model-agnostic, and it can be used with any machine learning model. However, it needs to perform a large number of evaluations to interpret the model, making it computationally expensive.

2.4 Backpropagation-based Interpretability

In these methods, attributions are calculated by backpropagating once or more times through the network. Several backpropagation-based interpretability approaches have been developed recently. One such method is 'Saliency Map' [40] where attributions are the absolute gradient of the label output of each input feature. Another method is Gradient-weighted Class Activation Map ('Grad-CAM') that calculates an activation map and uses it to assign importance score to each feature or pixel [36]. Grad-CAM does not backpropagate back to the image, unlike other gradient-based methods but goes until the last convolutional layer. It produces a localization map highlighting features of the image considered necessary by the model. However, Grad-CAM does not generate a unique feature map, and it generates an entirely different feature map when the image is even slightly changed [25].

Some of the backpropagation-based methods have given particular emphasis to network visualization to visually demonstrate each pixel's unique quality [44, 53]. For example, Yosinsk et al. [50], and Hu et al. [22] tried understanding deep networks through deep visualization. Visual decomposition-based approaches are also frequently used for the interpretation of various computer vision-related tasks [55]. Kindermans et al. [25] discussed the limitations of the visualization-based backpropagation methods in terms of high computational cost.

Based on the literature survey, SHAP, LIME, and Grad-CAM are the most widely used interpretability approaches for ML models. Considering the high computation cost involved with LIME and the inability of Grad-CAM in withstanding small changes in the input image, SHAP comes out to be the most suitable method. We have further improved the SHAP algorithm by incorporating the Divide and Conquer mechanism with it. It resulted in the development of a computationally efficient interpretability approach, DnCShap. The proposed approach discovers comparable feature maps; however, their computation is significantly faster than SHAP.

3 PROPOSED APPROACH

3.1 Problem Formulation

Given a Deep Neural Network $\mathbb{F}$ that maps an n-dimensional feature space $\mathbb{X}$ to m labels where $\mathbb{X} = \{x[1],\ x[2],\ x[3]....x[n]\}$ and the labels are represented as $y_1,\ y_2,\ ...\ y_m$. The goal of the proposed approach is to find the importance of each feature $x[i]$ in determining y_j where $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m\}$.

3.2 Architecture

The architecture of the proposed method has been depicted in Fig. 2. It has been built on top of VGG16 [41] by adding four residual layers to it. In total, the proposed architecture is composed of five blocks, B_k where $k \in \{1, 2, \ldots, 5\}$. The block B_1 consists of two convolutional layers with 64 channels, and B_2 is made up of two convolutional layers with 128 channels. Blocks B_3 and B_4 contain three convolutional layers, and they are made up of 256 and 512 channels, respectively. The configuration of B_5 is same as block B_4. The filters of size 3×3 are incorporated in each of these five blocks, and a max-pool layer of stride $(2, 2)$ is added after them. The output on convolution layers is flattened, and then three fully connected layers of size $(1, 2048)$, $(1, 2048)$, and $(1, 7)$ and included.

Four residual layers, R_l where $l \in \{1, 2, 3, 4\}$ have been added after B_1, B_2, B_3 and B_4, respectively to help to prevent vanishing and exploding gradients that is faced by very DNNs [20]. The layer R_1 consists of four max-pool layers and one convolution layer of 512 channels with 3×3 filters. Similarly, R_2 consists of three max-pool layers and one convolution layer of 512 channels with 3×3 filters. Likewise, R_3 and R_4 contain one convolutional layer each and two and one max-pool layers, respectively. The output from B_5 and the Residual Module is concatenated, flattened, and fed to the dense layer.

3.3 DnCShap

We have proposed a novel Interpretability framework, Divide & Conquer based Shapley additive explanations (DnCShap), which incorporates the Divide and Conquer mechanism with the SHAP algorithm for faster computation of approximated Shapley values. The Shapley values denote the importance of each pixel in performing the classification. Though they can be proven theoretically, their exact computation is NP-hard [37]. The time complexity of exact Shapley values' computation is theoretically exponential. The SHAP algorithm calculates the approximate Shapley values; however, it requires a quadratic time complexity. The proposed approach, DnCShap, computes the approximate Shapley values in linear time. The process to compute the same has been described below.

3.3.1 Calculating Shapley values. The Fig 3 shows an example network to demonstrate the computation of Shapley values. The network contains two features f_1 and f_2 it contains the model *Node1* with null features, model *Node2* and *Node3* with one feature each, i.e., f_1 and f_2, respectively and model *Node4* containing both the features f_1 and f_2. The difference between the prediction of two

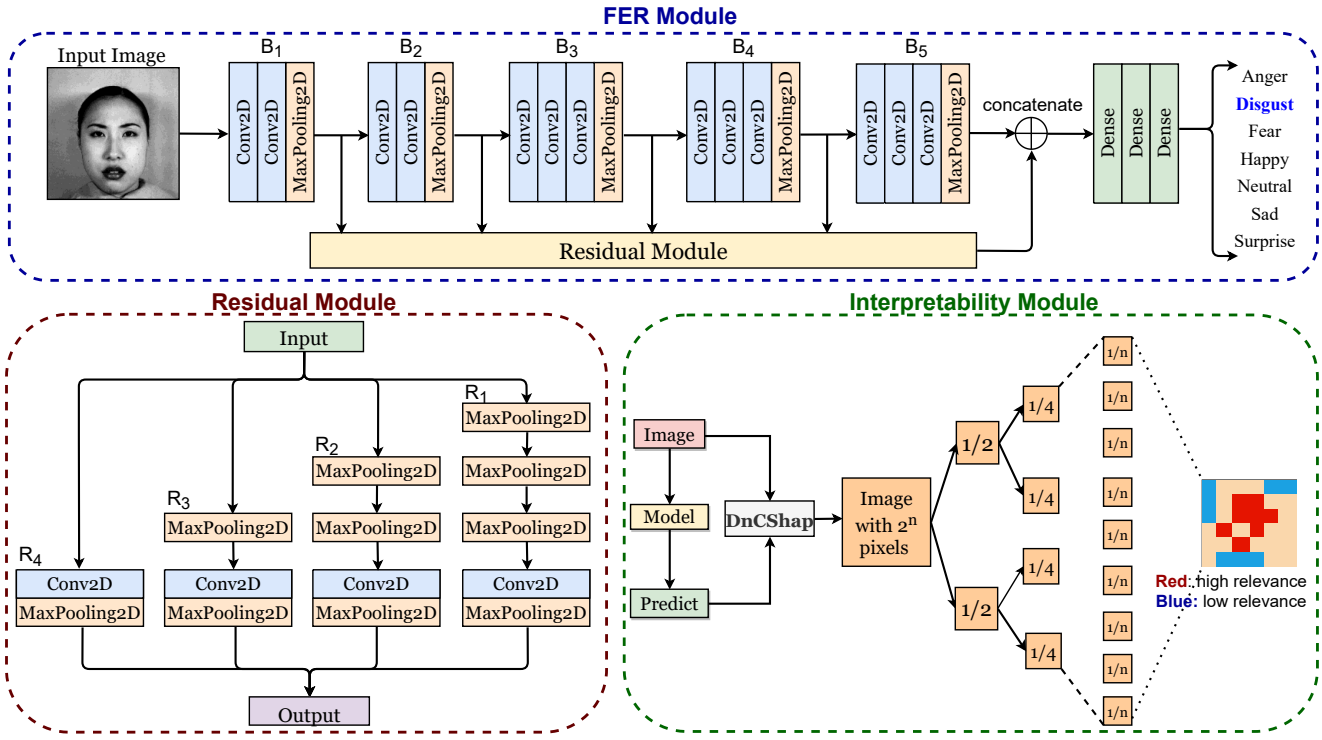


Figure 2: Schematic architecture of proposed method

nodes connected by an edge is said to be the 'marginal contribution' of that feature.

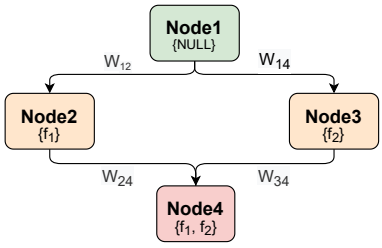


Figure 3: Sample model for Shapley values computation

For the predicted label c, the marginal contribution of feature f_1 to the model containing only f_1 is calculated as per Eq. 1.

$$MC_{f_1,\{f_1\}} = score_{\{f_1\}} - score_{\{\phi\}}$$
$$\begin{cases} where: \\ score_{\{f_1\}} = prediction\ for\ label\ c \\ using\ the\ model\ with\ feature\ f_1 \end{cases} \quad (1)$$

To calculate overall contribution, the weighted average of all 'marginal contributions' is calculated as per Eq. 2.

$$SHAP_{\{f_1\}} = w_{12} \times MC_{f_1,\{f_1\}} + w_{34} \times MC_{f_1,\{f_1,f_2\}}$$
$$\begin{cases} where: \\ w_{12}\ and\ w_{34}\ are\ the\ coefficient\ weights \end{cases} \quad (2)$$

The i^{th} model's coefficient weight, w_i is calculated as per Eq. 3.

$$w_i = \frac{|S|!(|F| - |S| - 1)!}{|F|!}$$
$$\begin{cases} where: \\ |S| = Number\ of\ features\ in\ the\ model \\ |F| = Total\ number\ of\ weights \end{cases} \quad (3)$$

The values of w_{12} and w_{34} described in Eq. 2 are computed using Eq. 3 and we get the Shapley values as follows.

$$w_{12} = (0!(2-0-1)!)/(2!) = 1/2$$
$$w_{34} = (1!(2-1-1)!)/(2!) = 1/2$$
$$SHAP_{\{f_1\}} = (\frac{1}{2}) \times MC_{f_1,\{f_1\}} + (\frac{1}{2}) \times MC_{f_1,\{f_1,f_2\}} \quad (4)$$

3.3.2 Approximating Shapley values. Theoretically, to calculate the Shapley values, the model needs to be re-trained for a different number of features. In case of the network containing two features f_1 and f_2, the models need to be retrained for $\{\phi\}$, $\{f_1\}$, $\{f_2\}$, $\{f_1, f_2\}$. However, in the proposed approach, we are going to perturb the input instead. The 'Divide and Conquer' algorithm [43] has been incorporated with the proposed approach for computing the Shapley values in linear time instead of the exponential time taken by the existing interpretability approaches. Algorithm 1, 'RecSHAP,' is a recursive procedure to divide the image into parts and find Shapley values for each part. Algorithm 2 provides the pseudo-code of DnCShap that computes the approximate Shapley values.

Algorithm 1: RecSHAP (recursive procedure)

Define *model*: DNN model,
Define *data*: Image pixels,
Define *wd*, *ht*: Image's width & height,
Define *pred_b*, *pred_f*: prediction with both/no parts perturbed,
Define *x_s*, *x_e* initial & final dimensions for width
Define *y_s*, *y_e* initial & final dimensions for height
Define *times*: number of parts the image is to be divided,
Define *arg_max*: predicted label,
Define *score*: SHAP value of the part,
Define *value*: activation map.

procedure RecSHAP (*model*, *data*, *wd*, *ht*, *times*, *i*, *x_s*, *x_e*, *y_s*, *y_e*, *pred_b*, *pred_f*, *score*, *arg_max*, *value*)

if $times == 0$ **then**
  $value[x\_s : x\_e,\ y\_s : y\_e] = score$;
else if $i == 1$ **then**
  $z = y$;
else
  $z = x$;
end
▷ Perturb by dividing image into two parts, along x or y axis
$data\_1, data\_2 = get\_parts(data,\ wd,\ ht,\ z)$
$pred\_1 = model.predict(data\_1)[0][arg\_max]$
$pred\_2 = model.predict(data\_2)[0][arg\_max]$
$score\_1 = (((pred\_1 - pred\_b) + (pred\_f - pred\_2))/2)/2$
$score\_2 = (((pred\_2 - pred\_b) + (pred\_f - pred\_2))/2)/2$
$times = times - 1$
$i = abs(i - 1)$
RecSHAP (*model*, *data*, *wd*, *ht*, *times*, *i*, *x_s*, *x_e*, *y_s*, *y_e*, $pred\_b + pred\_2,\ pred\_1 + pred\_f$, *score_2*, *arg_max*, *value*)
RecSHAP (*model*, *data*, *wd*, *ht*, *times*, *i*, *x_s*, *x_e*, *y_s*, *y_e*, $pred\_b + pred\_1,\ pred\_2 + pred\_f$, *score_2*, *arg_max*, *value*)

Algorithm 2: DnCShap

Define *model*: DNN model,
Define *data*: Image pixels,
Define *wd*, *ht*: Image's width & height,
Output *SHAP_value*.

procedure DnCShap (*model*, *data*, *wd*, *ht*, *times*)
$data\_b = np.zeros([wd, ht, 3])$
$data\_f = data$
$data\_f = data\_f.reSHAPe(1, wd, ht, 3)$
$data\_b = data\_b.reSHAPe(1, wd, ht, 3)$
▷ Perturb by dividing image into two parts, along x or y axis
$data\_1, data\_2 = get\_parts(data, width, height, x)$
$pred = model.predict(data\_f)$
$arg\_max = np.argmax(pred)$
$pred\_f = pred[0][arg\_max]$
$pred\_b = model.predict(data\_b)[0][arg\_max]$
$pred\_1 = model.predict(data\_1)[0][arg\_max]$
$pred\_2 = model.predict(data\_2)[0][arg\_max]$
$score\_1 = ((pred\_1 - pred\_b) + (pred\_f - pred\_2))/2$
$score\_2 = ((pred\_2 - pred\_b) + (pred\_f - pred\_1))/2$
$SHAP\_value = np.zeros([width, height])$
$times = times - 1$
RecSHAP $(model, data, wd, ht, times, 1, 0, x\_m, 0, ht, pred\_b+$ pred_2, $pred\_1 + pred\_f, score\_1, arg\_max, SHAP\_value)$
RecSHAP $(model, data, wd, ht, times, 1, x\_m, wd, 0, ht, pred\_b+$ pred_1, $pred\_2 + pred\_f, score\_2, arg\_max, SHAP\_value)$
return *SHAP_value*

The summation of Shapley values of all the features is equal to the difference of predicted values and average values. For calculating *score*1 and *score*2, i.e., Shapley values of both halves, the value to *pred_b* and *pred_f* are chosen in such a way the property in Eq. 5 holds for each and every part.

$$SHAP_{\{f_1\}} = score_{\{f_1,f_2\}} - score_{\{\phi\}} \tag{5}$$

To calculate the Shapley values for an input image X with dimensions (w, h), we divide the image in two halves x_1, x_2, each with dimensions $(w/2, h)$. These two parts are considered as two features, and Shapley values are computed using Eq 2. Now x_1 and x_2 are further divided, and their Shapley values are computed. This process continues until the required number of partitions (specified as a hyperparameter) of the image are created.

The time complexity of the DnCShap algorithm can be expressed as $T(n) = 2T(n/2) + O(1)$ which comes out to be O(n) in the worst case where n is the number of features or pixels. This affirms the linear time complexity for computing the approximate Shapley values using the proposed approach.

4 EXPERIMENTS AND RESULTS

4.1 Datasets and Training Strategy

The FER has been performed on the following datasets.

- **FER13** [5]: The Facial Expression Recognition 2013 dataset consisting of 35,887 facial images labeled with the emotion class. FER2013 is a diverse dataset containing variations such as facial occlusions, objects such as glasses, gestures, and partial faces.
- **FERG** [3]: The FERG dataset contains 55,767 animated and annotated face images of six stylized characters labeled with various emotion classes.
- **JAFFE** [23]: This dataset includes 213 images posed by Japanese female models with various facial expressions.
- **CK+** [28]: The Cohn Kanade extended dataset contains 927 posed and spontaneous images of 123 subjects. Three frames and an emotion label of each image are provided.

The implementation has been carried out using NVIDIA Tesla T4 GPU with 2560 CUDA cores, 2320 Tensor cores, and 16 GB Virtual RAM. The experiments have been performed using 5-fold cross-validation, Adam optimizer, Categorical Cross entropy loss, and batch-size of 32 for FER13 & 16 for the rest of the datasets. The model has been trained up to 100 epochs, using a learning rate from 8×10^{-5} to 8×10^{-4} and ReduceLROnPlateau learning rate scheduler.

4.2 Ablation Study

As depicted in Table 1, an extensive ablation study has been performed to decide the architecture of the proposed approach. First, pre-trained VGG16 and ResNet-50 networks were deployed, and then a baseline architecture was implemented on top of VGG-16. The baseline architecture contained five blocks containing convolutional and max-pool layers and residual layers integrated with the last two blocks. It performed better than pre-trained VG and ResNet models; however, the dimensions $(1, 4096)$ for the penultimate and

anti penultimate fully connected layers proved to be too large for the model to continue learning smoothly.

Table 1: Summary of the ablation study

Model	FER-13	FERG	JAFFE	CK+
VGG-16	26.98%	99.20%	99.20%	90.82%
ResNet-50	30.50%	88.80%	71.00%	75.52%
Baseline	60.12%	99.68%	90.16%	99.02%
Proposed	62.62%	99.68%	91.97%	99.67%

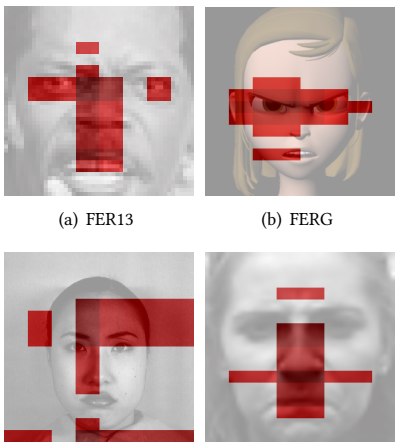
Finally, the residual layers were incorporated after each block, and the penultimate fully connected layer's dimensions were reduced to $(1, 2048)$. The architecture thus emerged has been chosen for the proposed FER model as it performed better than pre-trained and baseline models.

4.3 Results

The quantitative results have been discussed in terms of accuracy and confusion matrices for the proposed FER model. The feature maps generated by DnCShap have been presented, and the qualitative results have been discussed. The results have been compared with the existing FER models and interpretability approaches.

4.3.1 Accuracy and Confusion Matrices. The proposed FER system has resulted in the emotion classification accuracy of 62.62%, 99.68%, 91.97%, and 99.67% for FER-13, FERG, JAFFE, and CK+, respectively. The confusion matrices have been shown in Fig. 4.

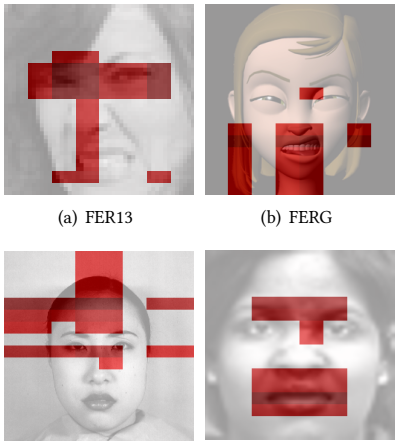
4.3.2 Interpretable Feature Maps. Fig. 5 to 11 show the feature maps for 'Anger,' 'Disgust,' 'Fear,' 'Happy,' 'Sad,' 'Surprise,' and 'Neutral' emotion classes. The image features with the maximum contribution towards the prediction of the emotion class are marked red in the feature maps. As observed from Fig. 5 to 11, each emotion class is associated with particular feature patterns. For example, the emotion class 'Happy' is associated with smile patterns, and 'Surprise' is associated with eyes and upper lips. The features leading to the prediction of the 'Sad' emotion class are distributed throughout the face vertically, while the features contributing towards 'Anger' are also vertically distributed with more emphasis on the eyes.



(a) FER13 (b) FERG

(c) JAFFE (d) CK+

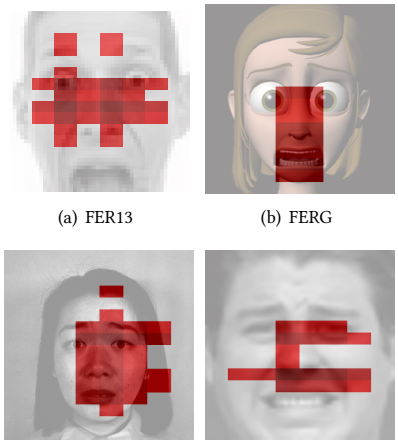
Figure 5: Feature maps for 'Angry' emotion class



(a) FER13 (b) FERG

(c) JAFFE (d) CK+

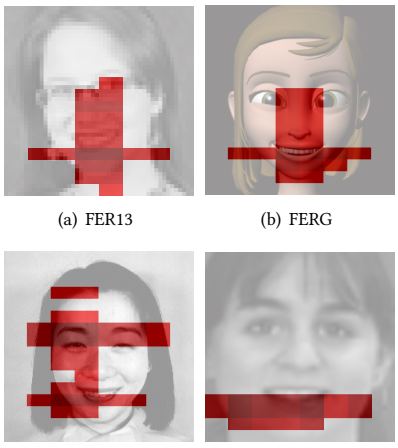
Figure 6: Feature maps for 'Disgust' emotion class



(a) FER13 (b) FERG

(c) JAFFE (d) CK+

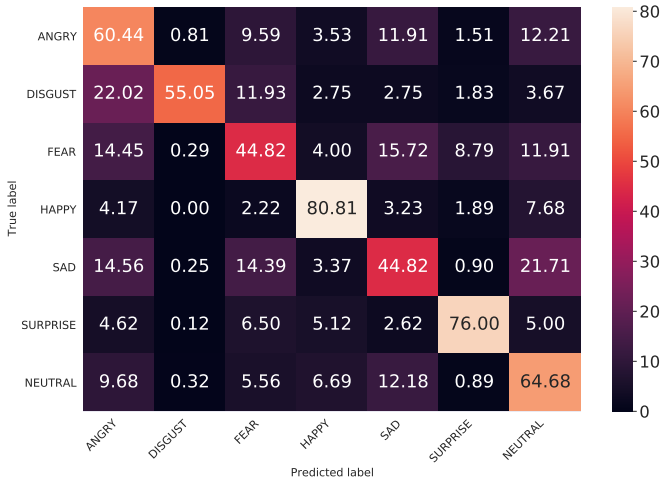
Figure 7: Feature maps for 'Fear' emotion class



(a) FER13 (b) FERG

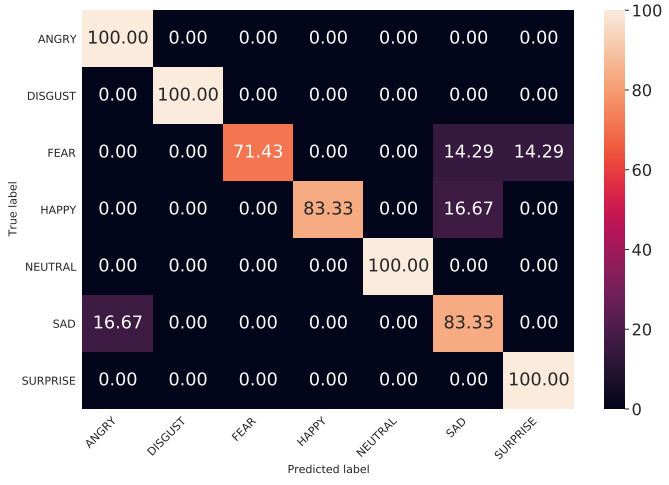
(c) JAFFE (d) CK+

Figure 8: Feature maps for 'Happy' emotion class



(a) FER13

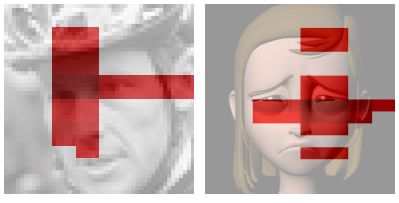
(b) FERG



(c) JAFFE

(d) CK+

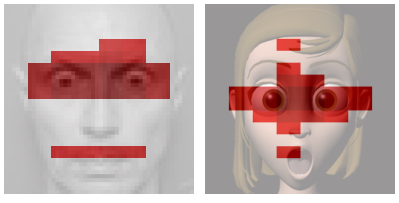
Figure 4: Confusion Matrices



(a) FER13 (b) FERG

(c) JAFFE (d) CK+

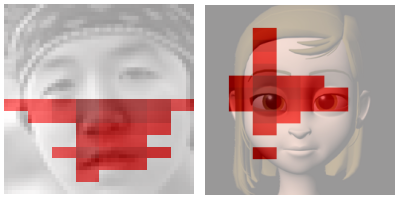
Figure 9: Feature maps for 'Sad' emotion class



(a) FER13 (b) FERG

(c) JAFFE (d) CK+

Figure 10: Feature maps for 'Surprise' emotion class



(a) FER13 (b) FERG

(c) JAFFE

Figure 11: Feature maps for 'Neutral' emotion class

It is to be noted that the CK+ dataset does not contain the images labeled as 'Neutral.' Hence, it has not been presented in Fig. 11.

4.4 Results Comparison

The performance of the proposed FER system has been compared with the existing FER approaches in Table 2 while Table 3 compares DnCShap against the existing interpretability approaches.

Table 2: Comparing with existing FER approaches

Model	FER13	FERG	JAFFE	CK+
Deep-Emotion [31]	70.02%	99.30%	92.80%	98.00%
Mollahosseini et al. [32]	66.40%	-	-	93.20%
DOG+CNN [38]	58.96%	-	-	-
Domain Adaptation [49]	63.50%	-	-	-
LBP+ORB features [34]	-	-	88.50%	93.20%
DeepExpr [3]	-	89.02%	-	-
Ensemble [54]	-	97.00%	-	-
Adversarial Net [15]	-	98.20%	-	-
Local Dominant Patterns [46]	-	-	87.60%	-
Salient Patches [19]	-	-	91.80%	-
Pseudo Annotations [52]	-	-	-	92.45%
Occlusion-aware FER [27]	-	-	-	97.03%
Proposed	62.62%	99.68%	91.97%	99.67%

As observed from Table 2, the proposed FER system has performed better or is comparable to the existing FER approaches.

Table 3: Comparing existing Interpretabilty approaches

	SHAP	Lime	Grad-CAM	DnCShap
Theoretical cost	$O(n^2)$	$O(n^2)$	$O(1)$	$O(n)$
Execution Time (s)	39	34	1	26

As shown in Table 3, DnCShap came out to be faster than SHAP both theoretically and experimentally. Though Grad-CAM seems to be the fastest among the interpretability techniques under consideration, it's not the most reliable one. Even slight changes in the input images lead to an entirely different feature map generation [25]. Moreover, SHAP is the only known interpretability method that satisfies the required exempts of completeness, symmetry, linearity, continuity, and implementation invariance [2]. That's why SHAP has been considered as the basis of performance comparison. The exact computation of Shapley values is an NP-hard problem; SHAP computes approximate Shapely values in $O(n^2)$ while the proposed approach, DnCShap, is able to compute the same in $O(n)$ time, and its experimental performance is also better than SHAP.

Table 4: Feature overlap with SHAP in terms of Dice loss

	SHAP	Lime	Grad-CAM	DnCShap
SHAP	1	0.174	0.488	0.541

Furthermore, the overlap of the feature maps generated by LIME, Grad-CAM, and the proposed approach, DnCShap, has been analyzed with the feature maps of SHAP in terms of Dice Loss values [11]. As depicted in Table 4, DnCShap has shown the maximum overlap. The overlap for LIME is very low. Though Grad-CAM also demonstrates decent overlap, it is not reliable because of its inability to produce unique feature maps accomodating slight variations in the input image. SHAP generates the feature maps pixel-wise, and DnCShap generates them region-wise, considering the surrounding pixels. Generating pixel-wise maps for DnCShap resulted in a further increment in the feature overlap with DHAP at the cost of increased computation time.

4.5 Discussion

The proposed interpretability approach has demonstrated consistent results for various datasets. The distribution pattern of the facial features contributing to the prediction of a particular emotion class is similar for FER13, FERG, JAFFE, and CK+. For instance, the emotion class 'Sad' is associated with the vertically distributed features throughout the face. The features contributing towards 'Anger' and 'Surprise' emphasize the eyes similarly for the images from all of the datasets mentioned above. Likewise, the feature maps for the 'Happy' emotion class can capture the smile for the image samples from various datasets.

5 CONCLUSION

This paper proposes an interpretable deep-learning-based FER system capable of interpreting its results. The contribution of each image feature towards the predicted emotion class has been analyzed. The highly relevant facial features towards emotion recognition have been identified consistently for the image samples from various datasets. In the future, we plan to extend the proposed technique to multimodal emotion recognition, including visual, speech, and textual modalities. We also aim to make the proposed DNN interpretability approach more explanatory and apply it to other affective computing and computer vision applications.

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking Inside the Black-box: A Survey on Explainable A.I. (XAI). *IEEE Access* 6 (2018), 52138–52160.

[2] Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation. In *The International Conference on Machine Learning (ICML)*. 272–281.

[3] Deepali Aneja et al. 2016. Modeling Stylized Character Expressions via Deep Learning. In *The Asian Conference on Computer Vision (ACCV)*. Springer, 136–153.

[4] Marian Bartlett et al. 2005. Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior. In *The IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2. 568–573.

[5] Pierre-Luc Carrier, Aaron Courville, Ian J Goodfellow, Medhi Mirza, and Yoshua Bengio. 2013. FER-2013 Face Database. *Universit de Montral* (2013).

[6] Javier Castro et al. 2009. Polynomial Calculation of The Shapley Value based on Sampling. *Elsevier Computers & Operations Research* 36, 5 (2009), 1726–1730.

[7] Junkai Chen, Zenghai Chen, Zheru Chi, Hong Fu, et al. 2014. Facial Expression Recognition based on Facial Components Detection and HOG Features. In *The International Workshop on Electrical and Computer Engineering Subfields*. 884–888.

[8] Marcelo Cossetin et al. 2016. Facial Expression Recognition using a Pairwise Feature Selection and Classification Approach. In *The IEEE International Joint Conference on Neural Networks (IJCNN)*. 5149–5155.

[9] Marina Danilevsky et al. 2020. A Survey of the State of Explainable AI for Natural Language Processing. In *The 10th International Joint Conference on Natural Language Processing (IJCNLP)*. 447–459.

[10] Jia Deng, Gaoyang Pang, Zhiyu Zhang, Zhibo Pang, Huayong Yang, and Geng Yang. 2019. cGAN based Facial Expression Recognition for Human Robot Interaction. *IEEE Access* 7 (2019), 9848–9859.

[11] Ruoxi Deng et al. 2018. Learning to Predict Crisp Boundaries. In *The European Conference on Computer Vision (ECCV)*. 562–578.

[12] Finale Doshi-Velez and Been Kim. 2017. Towards a Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608* (2017).

[13] Paul Ekman. 1977. Facial Action Coding System. (1977).

[14] Paul Ekman and Wallace V Friesen. 1971. Constants Across Cultures in the Face and Emotion. *American Psychological Association Journal of Personality and Social Psychology* 17, 2 (1971), 124.

[15] Clment Feutry, Pablo Piantanida, Yoshua Bengio, and Pierre Duhamel. 2018. Learning Anonymized Representations with Adversarial Neural Networks. *arXiv preprint arXiv:1802.09386* (2018).

[16] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*. 2950–2958.

[17] Ruth C Fong and Andrea Vedaldi. 2017. Interpretable Explanations of Black-boxes by Meaningful Perturbation. In *The The IEEE/CVF International Conference on Computer Vision (ICCV)*. 3429–3437.

[18] Leilani Gilpin et al. 2018. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *The 5th IEEE International Conference on Data Science and Advanced Analytics*. 80–89.

[19] SL Happy and Aurobinda Routray. 2014. Automatic Facial Expression Recognition using Features of Salient Facial Patches. *IEEE Transactions on Affective Computing (TAC)* 6, 1 (2014), 1–12.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *The IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[21] Paul VC Hough. 1962. Method and Means for Recognizing Complex Patterns. US Patent 3,069,654.

[22] Junjie Hu, Yan Zhang, and Takayuki Okatani. 2019. Visualization of Convolutional Neural Networks for Monocular Depth Estimation. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*. 3869–3878.

[23] Miyuki Kamachi, Michael Lyons, and Jiro Gyoba. 1998. The Japanese Female Facial Expression (JAFFE) Database. *URL www.kasrl.org/jaffe.html* 21 (1998), 32.

[24] Pooya Khorrami et al. 2015. Do Deep Neural Networks Learn Facial Action Units when Doing Expression Recognition?. In *The IEEE/CVF International Conference on Computer Vision-workshop (ICCVw)*. 19–27.

[25] Pieter-Jan Kindermans et al. 2019. The (Un)reliability of Saliency Methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. 267–280.

[26] Puneet Kumar, Vishesh Kaushik, and Balasubramanian Raman. 2021. Towards the Explainability of Multimodal Speech Emotion Recognition. In *INTERSPEECH*. 1748–1752. https://doi.org/10.21437/Interspeech.2021-1718

[27] Yong Li, Jiabei Zeng, Shiguang Shan, and Xilin Chen. 2018. Occlusion Aware Facial Expression Recognition using CNN with Attention Mechanism. *IEEE Transactions on Image Processing (TIP)* 28, 5 (2018), 2439–2450.

[28] Patrick Lucey et al. 2010. The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion Specified Expression. In *The IEEE/CVF International Conference on Computer Vision and Pattern Recognition-workshops (CVPRw)*. 94–101.

[29] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv preprint arXiv:1802.03888* (2018).

[30] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *The 31st International Conference on Neural Information Processing Systems (NeurIPS)*. 4768–4777.

[31] Shervin Minaee, Mehdi Minaei, and Amirali Abdolrashidi. 2021. Deep-Emotion: Facial Expression Recognition using Attentional Convolutional Network. *MDPI Sensors* 21, 9 (2021), 3046.

[32] Ali Mollahosseini, David Chan, and Mohammad H Mahoor. 2016. Going Deeper in Facial Expression Recognition using Deep Neural Networks. In *The Winter Conference on Applications of Computer Vision (WACV)*. 1–10.

[33] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. Definitions, Methods, and Applications in Interpretable Machine Learning. *Proceedings of the National Academy of Sciences* 116, 44 (2019), 22071–22080.

[34] Ben Niu et al. 2021. Facial Expression Recognition with LBP and ORB features. *Computational Intelligence and Neuroscience* 2021 (2021).

[35] Marco Ribeiro et al. 2016. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1135–1144.

[36] Ramprasaath Selvaraju et al. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*. 618–626.

[37] LS Shapley. 1953. A Value for n-person Games, Contributions to the Theory of Games II, AW Tucker, HW Kuhn.

[38] Minchul Shin, Munsang Kim, and Dong-Soo Kwon. 2016. Baseline CNN structure analysis for Facial Expression Recognition. In *The 25th IEEE International Symposium on Robot and Human Interactive communication*. 724–729.

[39] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not Just a Black box: Learning Important Features Through Propagating Activation Differences. *arXiv preprint arXiv:1605.01713* (2016).

[40] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034* (2013).

[41] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).

[42] Chandan Singh, W James Murdoch, and Bin Yu. 2018. Hierarchical Interpretations for Neural Network Predictions. In *The 6th International Conference on Learning Representations (ICLR)*.

[43] Douglas R Smith. 1985. The Design of Divide and Conquer Algorithms. *Elsevier Science of Computer Programming* 5 (1985), 37–58.

[44] Suraj Srinivas and François Fleuret. 2019. Full Gradient Representation for Neural Network Visualization. In *The 33rd International Conference on Neural Information Processing Systems (NeurIPS)*. 4124–4133.

[45] Erico Tjoa and Cuntai Guan. 2020. A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* (2020).

[46] Ying Tong and Rui Chen. 2019. Local Dominant Directional Symmetrical Coding Patterns for Facial Expression Recognition. *Computational Intelligence and Neuroscience* (2019).

[47] Ayşegül Uçar, Yakup Demir, and Cüneyt Güzeliş. 2016. A New Facial Expression Recognition based on Curvelet Transform and Online Sequential Extreme Learning Machine Initialized with Spherical Clustering. *Springer Neural Computing and Applications (NCAA)* 27, 1 (2016), 131–142.

[48] Monu Verma et al. 2018. EXPERTNet: Exigent Features Preservative Network for Facial Expression Recognition. In *The 11th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*. 1–8.

[49] Xiaoqing Wang, Xiangjun Wang, and Yubo Ni. 2018. Unsupervised Domain Adaptation for Facial Expression Recognition using Generative Adversarial Networks. *Computational Intelligence and Neuroscience* 2018 (2018).

[50] Jason Yosinski, Jeff Clune, Thomas Fuchs, and Hod Lipson. 2015. Understanding Neural Networks through Deep Visualization. In *The 32nd International Conference on Machine Learning-workshop (ICMLw)*.

[51] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *The European Conference on Computer Vision (ECCV)*. 818–833.

[52] Jiabei Zeng, Shiguang Shan, and Xilin Chen. 2018. Facial Expression Recognition with Inconsistently Annotated Datasets. In *The European conference on computer vision (ECCV)*. 222–237.

[53] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpretable Convolutional Neural Networks. In *The The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8827–8836.

[54] Hang Zhao et al. 2018. Transfer Learning with Ensemble of Multiple Feature Representations. In *The 16th International Conference on Software Engineering Research, Management and Applications*. 54–61.

[55] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. 2018. Interpretable Basis Decomposition for Visual Explanation. In *The European Conference on Computer Vision (ECCV)*. 119–134.

[56] Xiuzhuang Zhou, Kai Jin, Yuanyuan Shang, and Guodong Guo. 2018. Visually Interpretable Representation Learning for Depression Recognition from Facial Images. *IEEE Transactions on Affective Computing (TAC)* 11, 3 (2018), 542–552.