Received: April 24, 2020. Revised: May 19, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Facial Expressions Recognition Via CNNCraft-net for Static RGB Images

Ahmed Hesham Mostafa[1]* Hala Abdel-Galil El-Sayed[2] Mohamed Belal[2]

[1]*Computer Science and Artificial Intelligence, Helwan University Cairo, Egypt*
[2]*Professor of Computer Science Department, Computer Science and Artificial Intelligence, Helwan University, Egypt*
* Corresponding author's Email: ahmed.hisham@fci.helwan.edu.eg

Abstract: Facial Expression Recognition (FER) is one of the most important research problems in computer vision and Artificial Intelligence (AI) due to its potential applications, many studies were proposed for the FER, whether based on using handcrafted (Craft) features with traditional machine learning techniques or using end to end convolution neural network (CNN). In this paper, we proposed a new model called CNNCraft-net based on combining the advantages of CNN and traditional models by concatenating features outputs from CNN, autoencoder, and handcrafted features such as scale-invariant feature transform (SIFT), speed up robust feature (SURF) and Oriented Fast Rotated Brief (ORB), computed by the bag of visual words (BOVW) to recognize eight facial expressions for static RGB images. For the comparative analysis, multiple metrics were used such as Accuracy, Loss, F-measure, precision, and recall. The high imbalanced AffectNet and FER2013 datasets were used to evaluate the proposed model where the proposed model achieves accuracy 61.9% for eight expressions and 65% for seven expressions for AffectNet and 69% for FER2013.

Keywords: Convolutional neural network, Deep learning, Emotion recognition, FER, Facial expression recognition, Bag of visual words.

1. Introduction

Automatic facial expression recognition (FER) is one of the interesting problems in computer science because of its potential applications such as human-computer interaction (HCI), virtual reality (VR), augmented reality (AR), advanced driver assistant systems (ADASs), and video games, etc. So, there is significant interest in AI research to recognize facial expressions [1, 2]. But it is still a difficult and complex problem in computer science because of the complexity of translating the shape of facial muscles into emotions.

Many techniques have been proposed to detect emotions based on automatic facial expression recognition, but these techniques are based on traditional machine learning techniques such as support vector machines (SVM), Bayesian classifiers, etc. These techniques work well when recognizing expressions in a controlled environment, but these techniques cannot recognize expressions from untrained images [1, 2].

The lack of generalizability for these techniques because many techniques depending on datasets that are collected and designed in controlled environments that have tightly controlled illumination and pose conditions. Also, datasets often have limited numbers of subjects, few sample images or videos, or limited variation between samples [1, 2].

Moreover, Traditional techniques depend on handcrafted features that are designed by programmers, so they can ignore many important features or cannot capture them.

Recently, trends of research in various fields have begun to transfer to deep learning techniques, Due to having many benefits such as it can learn and capture features automatically, robustness to natural variations in the data is automatically learned and generalizability, the same model can be used for many applications and scalability, where the performance improves with more data.

Although deep learning is powerful, there are still problems when applied to FER. First, deep neural networks need a huge number of training images to prevent overfitting. However, the existing facial expression databases are not sufficient to train the well-known deep learning models that achieved the most promising results in object recognition tasks [1].

Moreover, deep learning-based approaches require more a higher-level and massive computing device than convention approaches to operate training and testing [2].

Additionally, variations in pose, illumination, and occlusions are common in unconstrained facial expression scenarios. These factors increase the requirement of deep networks to address the large intra-class variability [1].

As well, the need for a large dataset can lead to an imbalance in the distribution of facial expression samples. Because of the nature of expressions, the number of obtained images for the major expressions is larger than the minor expressions, for example, the Happy expression represents 46.26 % of the AffectNet [5] dataset, while Neutral expression represents 25.84%, so expressions like happy and natural will dominate during the training that leads model to perform well on dominant emotions, and poorly on the under-represented expressions.

So, in this paper, we present a new model called CNNCraft-net to enhance facial expressions recognition accuracy, CNNCraft-net is based on combining the advantages of CNN and traditional models to recognize eight facial expressions "Neutral, Happiness, Sadness, Surprise, Fear, Disgust, Anger and Contempt" for static RGB images.

CNNCraft-net concatenates features output from pretrained model Densent169 [3] and features from the encoder part from the proposed autoencoder model and handcrafted features such as scale-invariant feature transform (SIFT) [4], speed up robust feature (SURF) [5] and oriented FAST Rotated BRIEF (ORB) [6] computed by the bag of visual words (BOVW).

Also, in this paper, to overcome the imbalanced dataset problem, we proposed to weight the categorical cross-entropy loss function where the weight is calculated for each class or expression in the dataset by dividing the number of all images in the dataset by the number of images in the class multiplied by the number of expressions.

Also, in this paper, to overcome the limitation of hardware capabilities where it is not sufficient to train the proposed model with large datasets. So, we presented $BOVW\_Batch\_Geneator$ method that can load images and extract handcraft features using $BOVW$ method on the fly during the training of the deep learning model.

For the comparative analysis, multiple metrics were used such as Accuracy, Loss, F-measure, confusion matrix, precision, and recall. To evaluate the proposed model, we used the high imbalanced AffectNet [7] dataset and FER2013 [8] dataset

The rest of this paper is organized as follows: Section 2 discusses the related work; Section 3 introduces the proposed CNNCraft-net model and the proposed algorithms section 4 presents the dataset while section 5 discusses the experiment and results and finally conclusion and future work in section 6.

2. Related work

Many studies used and inspired the convolution neural networks for FER problem whether with fine-tuning or modifying the architecture or ensembling with other architectures such as Han, Byungok, et al. [9] proposed a cross-dataset adaptation method for merging different datasets to get sufficient sample size to train a deep learning model and reduce biases that exist across different datasets via proposed separate feature extractor and pseudo-label extractor.

Georgescu, et al. [10] proposed a method that combines handcrafted features extracted by BOVW with features learned by CNN, and they used a combination of three CNN architectures VGG-13, VGG-f, and VGG-Face that pertained on face net dataset they used only one type of handcraft features, they used dense sift features extractor and descriptor for handcraft features, then for classification part, they used proposed local learning framework that works according to the following steps First, a KNN model is applied to select the nearest training samples for an input test image. Then, the (SVM) classifier is trained on the selected training samples, then the SVM classifier is used to classify the class label only for the test image it was trained for it and for imbalanced dataset problem they used downsampling method and focal loss function.

Also, Radu, et al. [11] uses BOVW to extract dense SIFT descriptors and built a Linear kernel model combined with multi-class one-versus-all SVM with local learning while Li, Jing, et al. [12] present a novel convolutional neural network that consists of local binary patterns and improved Inception-ResNet layers for automatic facial expression recognition.

Li, Yong, et al. [13, 14] propose a CNN with attention mechanism (ACNN) that can distinguish the occlusion parts of the face and focus on not occluded parts. It merges the multiple representations from facial parts where it extracts 24 interested regions from that features maps that computed via VGG-16 Net and for each representation is weighted via a

proposed Gate Unit with predefined weights for each region in the face.

Mollahosseini, et al. [7] uses AlexNet and achieved better results using weighted cross-entropy loss function where they calculate the weights by dividing the number of samples of the class by the number of samples in the most under-represented class, while Charlie, et al. [15] proposed many deep learning models architecture based on AlexNet, VGGnet, and MobileNet and used the weighted loss proposed by Mollahosseini [7] while Y. Tang, et al. [16] the winner of ICML 2013 hold by Kaggle, where Y. Tang. [16] proposed a novel CNN architecture with replacing the SoftMax layer with a Linear SVM classifier.

Hua, Wentao, et al. [17] proposed an ensemble deep learning model that consists of three subnetworks with different depths. Also, Wei, Zijun, et al. [18] proposed an emotion net network with restnet50 as the backbone network for facial expression recognition. Siqueira, et al. [19] proposed a method for ensemble convolutional neural networks with shared representation.

Also, Ngo, et al. [20] use deep transfer learning techniques by using a squeeze-and-excitation network (SENet) model SE-ResNet-50 which pre-trained for using the largest dataset for human face VGGFace2 and proposes a new loss function and named weighted-cluster loss.

Also W. Xiaohua, et al [21] propose a two-level attention network for facial expression recognition in a static image, the first level used to extract the position of features while the second level is a Bi-directional Recurrent Neural Network for utilizing the relation between all features between all layers.

Zeng, et al. [22] proposed a framework to train the model from different inconsistently labeled datasets and large-scale unlabelled images. and for each image Zeng, et al. [22] assigns more than one labels whether from human manual annotations or model predictions. Then, they propose an end-to-end CNN model with a method to discover the latent truth from the inconsistent false labels and the input images.

Also based on our previous benchmarking study Ahmed H. Mostafa, et al. [23] where we compared fourteen pretrained CNN deep learning models applied to FER problems to recognize eight expressions, the CNN deep learning model DenseNet169 achieved the best accuracy result with 52.5%.

So the main question here is the combining of the handcrafted features extracted by traditional methods, with features extracted by deep learning methods can improve the recognition accuracy for facial expression? and how can the model deal with the imbalanced expressions problem? So, In this paper, we address these research questions and investigate the effect of this idea.

3. Proposed model

As shown in Fig. 1 the proposed model "CNNCraft-net" receives input from the batch generator then sends output as a pair of two inputs to the feature extractor where we implement the BOVW Extractor as a part of Image Batches Generator as well shown in Algorithms 1, 2, and 3 then concatenates all output features into one flatten features vector following by a dense layer of eight nodes represent the eight expressions.

The feature's extractor part in CNNCraft-net extract features using three modules DenseNet169, Encoder, and BOVW Extractor. The feature's extractor part in CNNCraft-net extract features using three modules DenseNet169, Encoder, and BOVW Extractor.

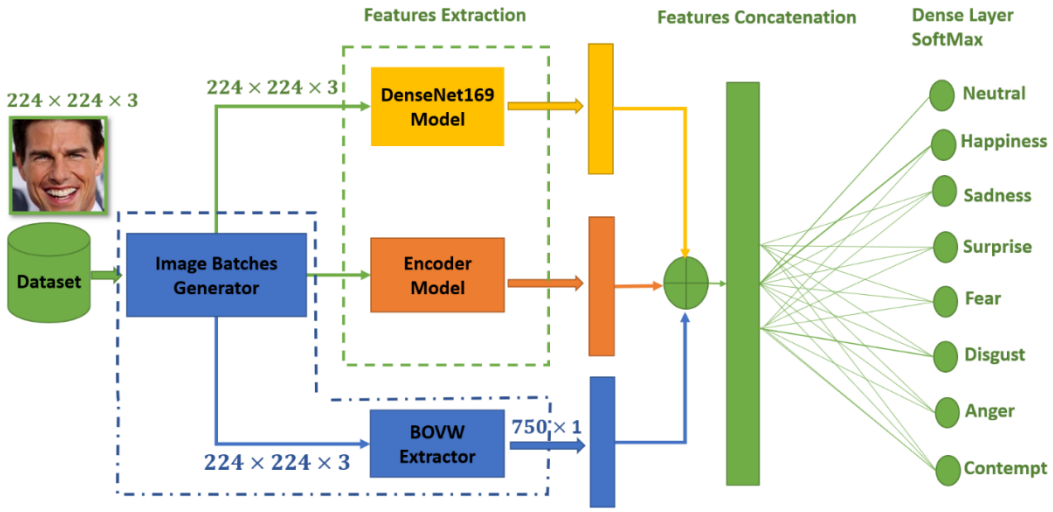


Figure. 1 Proposed CNNcraft-net model

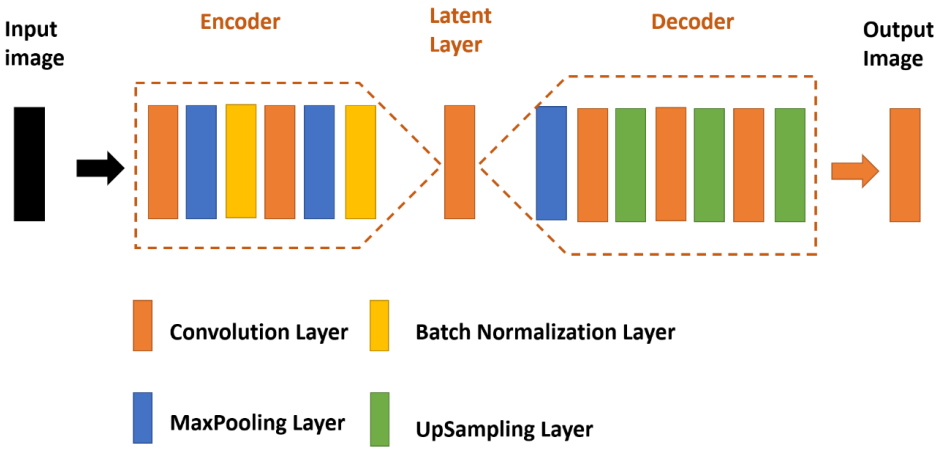


Figure. 2 Proposed autoencoder model architecture

As shown in Fig. 1 the model receives two inputs from a batch generator, the first is an RGB image in shape $224 \times 224 \times 3$ that is input for the Denesnt169 model and the encoder model while another input is the handcrafted features with shape 750×1 represent that 250 features form each of SIFT, SURF, and ORB features to be concatenated with other output from Densent169 and encoder to be input for prediction dense layer to predict or classify the eight expressions.

3.1 DenseNet169

It is one of the common pretrained CNN deep learning models, DenseNet groups that is based on connecting each layer to every other layer within a dense block. In this paper, the Densent169 model is used as a backbone model based on our previous benchmarking study Ahmed H. Mostafa, et al. [23] where the DenseNet169 achieved the best accuracy result with 52.5%.

3.2 Encoder model

Autoencoders [24] is a specific type of feedforward neural that consists of 3 components: encoder, latent, and decoder. The encoder compresses the input and produces the latent, the decoder then reconstructs the input only using this code.

As shown in Fig. 2 the autoencoder architecture is like a convolution neural network for the encoder part we used a sequence of convulsion of 3×3 following by MaxPooling 2×2 following by batch normalization where the encoded output will be the Latent Layer, while in the decoder is the same sequence but instead of MaxPooling it replaced by UpSampling while in the last convolution layer used depth with size three to output three channels image in RGB, but as shown in Fig.2 CNN model, the output of encoder followed by averaging pooling layer to select the most important features.

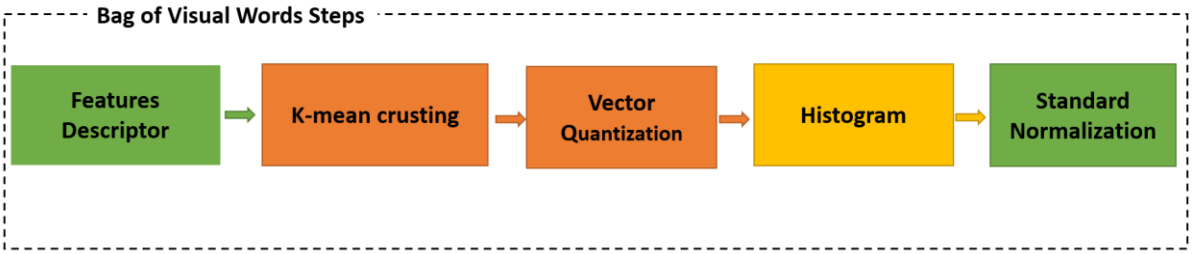
3.3 BOVW extractor

The overall idea behind the bag of visual words (BOVW) is to represent the image as a set of features [24]. Features consist of key points and descriptors.

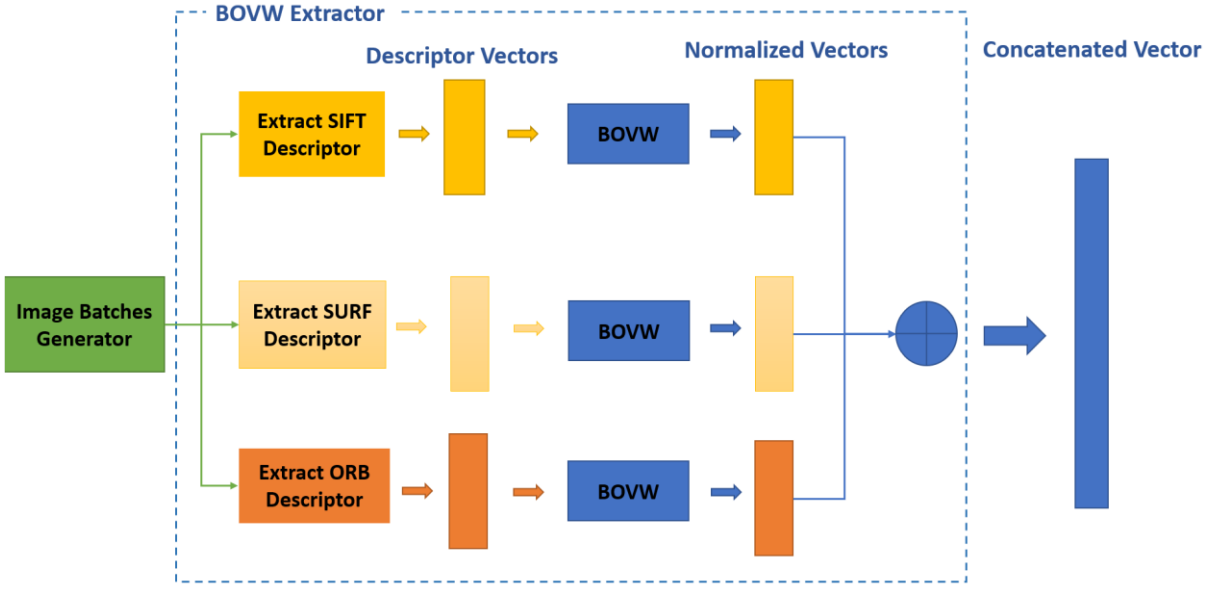
Key points are important points in an image where the feature has been detected. And descriptor is the description of the key point as an array of float numbers. Where the key points and descriptors are used to build the vocabulary and represent each image as a frequency histogram of features. From the frequency histogram, we can classify or cluster the images into different categories.

In this paper, to build the vocabulary we follow the steps in Fig. 3 (a) where We first extract descriptors from each image in the dataset and build a visual dictionary. To detect features and extracting descriptors in an image can be done by using feature extractor algorithms SIFT, SURF, and ORB, after extracting features descriptor from each image, the next step will be constructing the vocabulary of visual words by clustering the features descriptors into clusters using the k-means algorithm and the result of k-mean is considered as the dictionary of visual words, in this research we used only 250 as K value for clusters number for the three features type due to the limitation on memory size.

After that, as shown in Fig.3 (b) BOVW Extractor extracts feature descriptors using SIFT, SURF, and ORB and to compute its nearest neighbor in the dictionary of visual words that were created in the previous step, and this is normally done by calculating the distance between each features' descriptor vector and visual word vector using the Euclidean Distance, this process is called vector



(a)



(b)

Figure. 3: (a) BOVW steps and (b) proposed BOVW extractor

quantization. [25]

The next step is computing the frequency histogram. By counting the frequency of feature descriptors compared with the vocabulary and the resulting histograms are considered the bag of visual words (BOVW).

Finally, the Standardization of histograms even individual features do not less or more look like standard normally distributed data after that concatenate all standardized features for SIFT, SURF, and ORB as a vector with shape 750×1.

3.4 Proposed batch_geneartor

Due to the dataset has a huge number of images and hardware capabilities is not sufficient to train the proposed model with this large dataset, So in the implementation of our work we used from Keras framework the IgamgeDataGenerator [26] but it can load only images without handcrafted features, so we presented a modified version of batch generator by adding $BOVW\_Batch\_Geneator$ method that can load image and extract handcraft features using the proposed $BOVW$ method on the fly during the training as shown in Algorithms 1, 2, and 3.

In the beginning, Algorithm-1 uses the method $Creat\_voc(X, t)$ to create the vocabulary of visual words voc

Algorithm-1 Create the Vocabulary

Input: X the set of all images and t type of features extractor

Output: voc vocabulary of *visual words*

$\boldsymbol{Creat\_voc}(X, t)$:
  $D \leftarrow \{\}$
  $k \leftarrow 250$
  for $each\ x_i \in X_i$ ***do***
    $f_i \leftarrow Extract\_Features\_decriptor(t, x_i)$
    $D \leftarrow append\_to\_list(f_i)$
  end for
  $voc \leftarrow kmean(D, k)$
  $save\_voc(t, voc)$

where X refers to the set of all images in the dataset while t refer to the type of features extractor after extracting the features' descriptor f_i for each image x_i and appending them in the features descriptor list D_b, the $kmean$ algorithm is used to cluster them into k clusters or visual words then save

the vocabulary voc to be used later in $BOVW$ method.

While Algorithm-2 is used to extract a bag of visual words by using the proposed method $BOVW(t, XB)$ where t refer to the type of feature and XB refer to the batch of images, So it loops over each image x_i in the batch XB_b to extract the features descriptor f_i based on type t and append the extracted features description into descriptor list D_b, then load the list of voc for features of type t after that for each descriptor $D_b[j]$, in the descriptor list D_b it clusters the content of the descriptor $D_b[j]$ by measuring the distance between them and the voc using function $vector\_quintization$ and assignment the clustering result in W_b , then it calculates the histogram H_b by counting the frequency for each word w_i in the W_b, after that, it is standardizing the histogram H_b and set result in a standardized histogram f_b.

Algorithm -2 Extract the Bag of Visual Words

Input: t type of feature and XB the batch of images
Output: standardized histograms f_b

$D_b \leftarrow \{\}$
$k \leftarrow 250$
for $each\ x_i \in XB_b$ **do**
  $f_i \leftarrow Extract\_Features\_decriptor(t, x_i)$
  $D_b \leftarrow append\_to\_list(f_i)$
end for
$l = length(D_b)$
$H_b[l][k] \leftarrow \emptyset$
$voc \leftarrow load\_voc(t)$
for j to l **do**
  $W_b \leftarrow vector\_quintization(D_b[j], voc)$
  $for\ each\ w_i \in W_b$:
    $H_b[j][w_i] \leftarrow H_b[j][w_i] + 1$
end for
$f_b \leftarrow standarize(H_b)$
return f_b

While in Algorithm-3 the $Batch\_Geneator$ is used to generate batches of handcrafted features with images where it receives three-parameter X represent the source for all images, T is a list of strings that represents the type of feature SIFT, SURF, and ORB while Z represents the size of batch in this research 50 batch size is used.

The ***BOVW_Batch_Geneat*** work as following it get a batch of images x_b where b refer to the index of the batch such as $x_0, x_1, \ldots x_n$ where the number of batched determined by the size of the images dataset X divided by the batch size Z as listed in Algorithm-3 ($len(X)/Z$), then for feature extractor t_i in the list T, where the bag of visual words (handcraft features) are extracted using the function $BOVW(t, x_b)$ where for each batch of images x_b the function extracts the 250 features of type SIFT and set the result in the list f_b then merge features in f_b with the features in the list ft_b , then extracts the other 250 features of type SURF then merge it with features in the list ft_b and finally, it extracts the other 250 features of type ORB then merge them with features that exist in the list ft_b to have a list of 750 handcrafted features, then return a batch of images x_b with the corresponding bag of visual words ft_b.

Algorithm–3 Generate batches of handcrafted features with images

Input: X set of all images, T feature type, and Z batch size
Output: ***a*** batch of images x_b with a bag of visual words ft_b

BOVW_Batch_Geneator(X, T, Z):
counter← 0
while $(counter < len(X)/Z)$ ***do***
  $x_b \leftarrow get\_next\_batch(X, Z)$
  $ft_b \leftarrow \emptyset$
  for t_i ***in*** T ***do***:
    $f_b \leftarrow BOVW(t, x_b)$
    $ft_b \leftarrow merg(ft_b, f_b)$
  end for
  counter← counter + 1
end while
return $[ft_b, x_b]$

3.5 Weighted categorical cross-entropy loss

In this work we used the categorical cross-entropy (CCE) loss function, it calculates the loss by computing the following formula:

$$Loss\ (CCE) = -\sum_{i=1}^{c} y_i \times \log p_i \qquad (1)$$

Where p_i is i^{th} Scalar value in the prediction output of the model that calculated by SoftMax as shown in the equation, while $y_i$is the corresponding actual output, c is the number of classes in model output.

The dataset is very high imbalanced as shown in Table 1, one of the solutions to handle an imbalanced dataset is weighting the loss function for each of the classes by their relative proportion in the training dataset. In other words, the loss function penalizes the model by a large factor for misclassifying images

from under-represented classes, while penalizing the model by a small factor for misclassifying examples from well-represented classes.

So, in this paper, to overcome the imbalanced dataset problem the loss function Categorical cross-entropy (CCE) in Eq. (1) was modified by Weighted Categorical cross-entropy (WCCE) loss function in Eq. (4) by multiplying CCE loss function by wights w_i as in Eq. (2) where wights are calculated by balanced class wights function in Keras.

$$Loss\ (WCCE) = -\sum_{i=1}^{c} w_i \times y_i \times \log p_i \quad (2)$$

Where w_i is the class weight for class or expression i^{th} and calculated by dividing the number of all images in the dataset by the number of images in the class i^{th} multiplied by the number of expressions (the eight expressions) as shown in Eq. (3)

$$w_i = \frac{N}{n_i \times c} \quad (3)$$

Where N is the number of all training images, n_i is the number of images in class i while c is number of classes (number of expressions), So the Class Weighted Categorical cross-entropy Loss function will be Eq. (4)

$$Loss(WCCE) = -\sum_{i=1}^{c} \frac{N}{n_i \times c} \times y_i \times \log p_i \quad (4)$$

4. Experiments

4.1 Dataset

4.1.1. AffectNet [6]

Only the manually annotated images are used which represent eight expressions as shown in Fig.4 (a) where 287,651 images for training and 4000 images for validation (500 images for each class) as shown in Table-1. All the faces in images are allocated and cropped from images and resized to 224 × 224 × 3 in RGB mode.

For augmentation, we first convert the RBG image to a warm image and cold image [27] and flip each one of the images horizontally. To obtain a warm image, the values of the red channel are increased while the values of the blue channel are decreased. For obtaining a cold image the vice versa [27] as shown in Fig.5.

AffectNet dataset has several images in the standard datasets that were miscategorized. Where we used commercial [28] software to find duplicated images in different classes where we mean with duplication that the same image exists in more than one class with similarity greater than 95% where we found more than 11000 duplicated images.

Table 1. Expressions distribution

Expression	AffectNet	FER2013
	samples	Samples
Neutral	75,374	6,198
Surprise	14,590	4,002
Anger	25,382	4,953
Happy	134,915	8,989
Fear	6,878	5,121
Sad	4,303	6,077
Disgust	25,959	547
Contempt	4,250	-

(a)

(b)

Figure. 4 Samples: (a) AffectNet and (b)FER2013

(a) (b) (c)

Figure. 5 Samples of augmented images: (a) Original, (b) Warm, and (c) Cold

4.1.2. FER2013

It contains 35889 for seven types of expression as shown in Fig.4 where 28709 training images 3589 public validation and 3589 as private tests and images have size 48×48×1 with black and white color mode and all images are scaled to 224×224×3.

4.2 Hyperparameters and configuration

All experiments were done on the Google Colab platform [29] and we used Keras [26] and OpenCV frameworks. The multistage training method proposed in Ahmed H. Mostafa, et al. [23] was used to train the models by saving the models and reload them to resume training.

For all experiments we scaled all input images to 224 × 224 ×3, the number of epochs is 100, many batches sizes are used 16, 32, 50, 80, and shuffle is settled to true for training data, while for optimizer parameter the Adam optimizer is used.

The evaluation was performed by measuring the following metric from Keras: Accuracy (ACC), loss, Precision (Pre), Recall (REC), and F-measure (F1).

4.3 Results

As noticed from Table 2, that the proposed version for weighted loss (WCCE) achieved higher accuracy compared with the regular loss function (CCE) whether for public and private validation set for both seven and six expressions except when the model pretrained on random weights (None) the regular loss (CCE) achieved higher accuracy than the weighted loss function, but in general the highest accuracies achieved by regular loss (CCE) are 65.6%, 67.0% with losses values 2.3, 2.49 for public and private validation dataset respectively for seven expressions, while for six expressions achieved accuracy 71.4%, 70.3% with losses values 1.98, 2.09 for public and private validation dataset respectively.

While the highest results by WCCE are higher than the best results for CCE, where the best results of weighted version are for seven expressions 65.7%, 69.0 with losses values 2.46, 2.56 for public and private validation set respectively and for six expressions 72.0%, 73.0% with losses values 2.19, 1.76 for public and private validation set respectively. Also, it can be noted that transfer learning help in improving the accuracy of the model where the accuracy is increasing regularly from None to ImageNet to AffectNet.

Also, it can be noted from Table 2, all best result achieved whether by regular or weighted loss function achieved when the model pretrained on AffectNet dataset that proves that when transfer learning model pretrained on data like the problem, it will improve the accuracy and performance of the model although the difference between the AffectNet dataset and FER2013 in the image dimensions, number of samples and the number of expressions.

So, based on the previous results on FER2013 from Table 2 we excluded the experiments by using random weights (None) for AffectNet experiments and we used the ImageNet wights only as shown in Table 3.

Table 2. Shows the accuracy(ACC) in percentage for Fer 2013 for 7 and 6 expressions (expr) for both public (pub) and private (Prv) sets when using CCE and WCCE losses

		7 expr		6 expr	
Loss	Pre-train	Pub	Prv	Pub	Prv
CCE		Acc	Acc	Acc	Acc
	None	63.5	64.3	66.1	67.5
	Imagnet	64.6	66.1	69.1	69.5
	AffectNet	65.6	67.0	71.4	70.3
WCCE	None	60.9	61.0	67.3	64.7
	Imagnet	65.3	68.5	72.0	73.0
	AffectNet	65.7	69.0	71.9	72.0

It can be noted from Table 3, the accuracy is increased when the WCCE loss is used where the highest accuracies achieved by CCE loss are 53.0, 58.7 for eight and seven expressions respectively with losses values 2.9, 2.10 while in the weighted version the highest the accuracies are 58.0%, 62.0% for eight and seven expressions respectively, the weighted version increases the accuracy of the model by 5% and 3%, also it shows when the handcrafted features added to feature extracted with deep learning models increase the accuracy from 52.5% [23]. to 53% with CCE and 58% with WCCE.

Also, it can note the weighted version WCCE increases the accuracy of the model when the augmented and cleaned version of AffectNet is used where the accuracy by regular CCE loss is 58.5 is increased by 3% when the weighted version the WCCE loss is used.

As shown in Tables 3 and 4, the results show that when the model is trained using the AffectNet dataset the highest accuracy is when the model is pre-trained on ImageNet where the accuracy is 58.0%, 62.0% for eight and seven expressions respectively with losses values of 5.92, 6. Also, it can be noted that when AffectNet dataset was cleaned from incorrectly classified and perform augmentation in the classes have very few numbers of samples the classes 4,5 and 7 the accuracy of the model is increases where the highest accuracy is 61.9%, 65.0% for eight and seven expressions respectively with losses 2.51, 2.1.

As shown in Fig. 6 (a) and (b) for eight expressions the model improves the rate for correctly classified expressions when trained on the augmented version where the number of correctly classified for expressions, natural 316, 300, surprise 351, 301, fear

Table 3. Accuracy (ACC) in percentage results for AffectNet and augmented and cleaned (Aug) version

			8 expr	7 expr
Data	Loss	Pre-train	Acc	Acc
AffectNet	CCE	Imagnet	53.0	58.7
	WCCE	Imagnet	58.0	61.6
AffectNet Aug	CCE	Imagnet	58.5	62.2
	WCCE	Imagnet	**61.9**	65.4

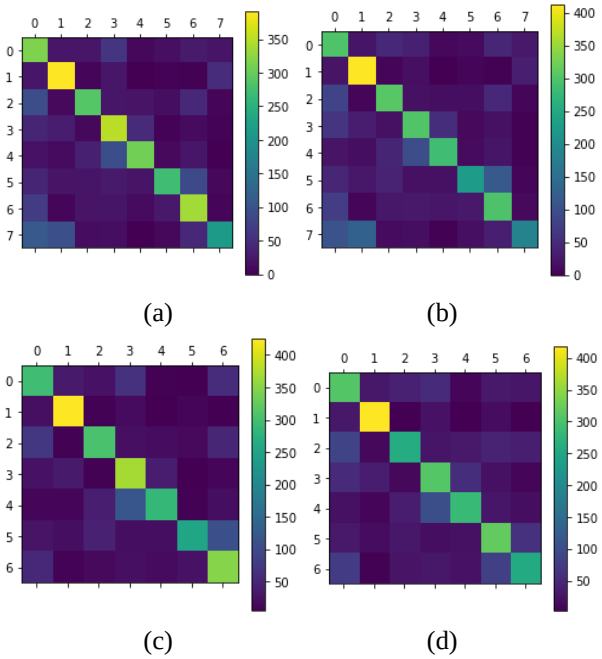


Figure. 6 Confusion matrices for the best model for AffectNet: (a) 8 expr (Aug), (b)8 expr without (Aug), (c) 7 expr (Aug), and (d) 7 expr without (Aug)

308, 286, disgust 272, 229, Anger 338, 298 and contempt 214, 185 where it is decreased for classes happy 392, 413 and sadness 288, 305.

Where for seven expressions as shown in Fig.6 (c) and (d) that the model improves the rate for correctly classified when trained on the augmented version for all classes except one class the class natural 298, 307.

Based on the previous results and the results shown in Table 4 the best accuracy achieved by the proposed CNNCraft model for eight expressions are 58% with Precision 0.58, Recall 0.58, and F-measure 0.58 when the model trained on AffectNet using the weighted loss WCCE and when trained on cleaned and augmented AffectNet using the weighted loss WCCE achieved accuracy 61.9% with 0.62 Precision, 0.62 Recall, and 0.62 F-measure.

Also, to show the performance of the proposed CNNCraft model, it was tested via a crossing different dataset as shown in Table 5, the model was trained via dataset and the validation accuracy is

Table 4. Precision (Pre), Recall (Rec), and F measure (F) for best results achieved for AffectNet using WCCE

	AffectNet			AffectNet (Aug)		
expr	Pre	Rec	F	Pre	Rec	F
7	0.65	0.65	0.65	0.61	0.61	0.61
8	0.58	0.58	0.58	0.62	0.62	0.62

Table 5. Accuracy crossing datasets validation

Training	Validation data			
	Fer2013		AffectNet	
	Pub	Prv	7 expr	8 expr
AffectNet	53.9	52.1	62.0	58.0
AffectNet (Aug)	55.1	54.7	65.0	61.9
Fer2013	66.0	69.0	40.6	-

calculated on the validation data for other dataset and Table 5 summarizes the results of validation over different datasets, where when the model was trained on AffectNet dataset and it was tested via the validation sets for FER2013 whether, for public and private validation dataset, the results were 53.9%, 52.1%, respectively. While when Fer2013 was used to train the model and validation set for AffectNet was used to evaluate the proposed model, the accuracy for 7 expressions was 46.7%.

4.4 Comparing to the state of art

Compared to State of Art methods as shown in Table 6, Our method achieves little better results than many of the existing methods for the AffectNet dataset where CNNCraft-net overpass the best results by a range of 1% to 2%, also regards the size of our proposed model is not exceeding 146 megabytes that it is considered smaller size than other State of Art models that based on ResNet or VGG such as Georgescu, et al [10], Radu Tudor, et al. [11], Li, Yong, et al [13, 14], Charlie, et al. [15], Hua, Wentao, et al. [17] and Ngo, et al [20].

Also, the proposed model shows that the use of handcrafted features with features extracted by the deep learning model (DL+Craft) can improve the recognition accuracy for facial expressions instead of using only Deep learning (DL) methods.

Also, the proposed weighted loss function achieved high accuracy compared to other methods to handles imbalanced dataset problems such as focal loss function in Georgescu, et al [10] and downsampling method in Mollahosseini, et al. [7] and Georgescu, et al [10] and, the oversampling method in Mollahosseini, et al. [7] and the weighted loss function in Mollahosseini, et al. [7].

Also, the proposed $BOVW\_Batch\_Geneator$ method that can load images and extract handcraft features using $BOVW$ method on the fly during the

training of the deep learning model instead of extracting it manually and then merging it with features extracted from deep learning model then loading both to train the model.

Where Georgescu, et al [10] achieved top accuracy of 75.42% on FER2013 59.58% on the AffectNet (augmented and down-sampling) dataset for eight expressions and 63.31% for 7 expressions. While, Han, Byungok, et al. [9] achieved accuracy for 58.89% for seven expressions in AffectNet.

Li, Yong, et al. [13] achieved a 55.33% accuracy result for 7 expressions for AffectNet while in [14] 58.78%. While Mollahosseini, et al. [7] achieved better results using weighted loss 58% while with down-sampling approach achieved accuracy (50%). While the framework proposed by Zeng [22] achieved an accuracy of 58% with AffectNet. While Hua, Wentao, et al. [17] propose an ensemble deep learning model, where they achieved 62.11% accuracy for 7 expression recognition for AffectNet and 71.91% for Fer2013.

While the ensemble model proposed by Siqueira, et al. [19] achieved 59.3% accuracy for eight expressions recognition of AffectNet. While Charlie, et al. [15] achieve an accuracy of 58% with the VGGnet model to recognize eight expressions for AffectNet and the two-level attention model proposed by W. Xiaohua, [21] achieved an accuracy of 48%.

While emotion net network proposed by Wei, Zijun, et al. [18] achieved accuracy 53.43%, 45.57% for eight and seven expressions respectively with AffectNet. While Ngo, et al. [20] used the transfer learning techniques and proposed a loss function called weighted-cluster where the proposed loss achieved accuracy 60.7% while the weighted - SoftMax achieved 59.72% for eight expressions recognition in the AffectNet dataset.

While for Fer2013, our method achieved accuracy with 69% and overpass many of existing state of art methods such as Radu et al. [13] 67.48%, Li, Jing, et al. [12] 67.71% Mollahosseini, [7] 66.4% but in general, our proposed model did not achieve a good result compared to many of existing state of art methods for Fer2013 such as Hua, Wentao, et al. [17] 71.91%, Y. Tang. [16] the winner of ICML 2013 with public test achieve an accuracy of 69.4% and for private test achieved an accuracy of 71.2% and Georgescu, et al [10] achieved top accuracy of 75.42% on FER 2013.

While for Fer2013 our method did not achieve a good result compared to many of the existing state of art methods and this due to several reasons, first the model is trained on the images which have

Table 6. State of art Accuracy (ACC) for Fer 2013 and whether based on Deep learning (DL) or combined with Handcraft features (DL+Craft)

Model	#EN	Method	Acc	Data
Hua,Wentao, et al. [17]	7	DL	71.91 %	Fer2013
Li, Jing, et al. [12]	7		67.71 %	
Mollahosseini, et al. [7]	7		66.4 %	
Y. Tang. [16]	7		71.2 %	
Radu et al. [11]	7	DL+Craft	67.48 %	
Georgescu, et al. [10]	7		75.42 %	
Proposed model	7		**69.0** %	
Han, Byungok, et al. [9]	8	DL	58.89 %	AffectNet
Li, Yong, et al. [13]	7		55.33 %	
Mollahosseini, et al [7]	8		58.0 %	
Zeng, et al [22]	8		57.31 %	
Hua, Wentao, et al. [17]	7		62.11 %	
Wei, Zijun, et al. [18]	7		53.43 %	
Wei, Zijun, et al. [18]	8		45.57 %	
Ngo, et al [20]	8		60.70 %	
Siqueira. et al [19]	8		59.3 %	
Charlie, et al. [15]	8		58 %	
W. Xiaohua, et al [21]	8		48 %	
Li, Yong, et al [14]	**7**		58.78 %	
Georgescu, et al [10]	7	DL+Craft	63.31 %	
Georgescu, et al [10]	8		59.58 %	
Proposed Model	**7**		**62** %	
Proposed Model	**8**		**58** %	
Proposed model (Aug)	**7**		**65.0** %	
Proposed model (Aug)	**8**		**61.9** %	

dimensions 224×224×3, while the images in FER2013 have 48×48×1dimensions, the second reason that the BOVW extracted create trained and create the vocabulary using AffectNet dataset and trained on images with dimensions 224×224×3 and there is a huge gap between handcrafted features extracted from black and white images in FER2013 with size 48×48×1 and handcraft features extracted from RGB images with the size 224×224×3, all these reasons effect on the performance and accuracy of the proposed model when was trained on the FER2013 dataset.

5. Conclusion and future work

In this paper, we proposed a new model called CNNCraft-net based on combining the advantages of CNN and traditional models by concatenating features output from CNN, autoencoder, and handcrafted features SIFT, SURF, and ORB computed by the bag of visual words (BOVW) to

recognize eight facial expressions in static RGB images. We proposed a batch generator that can load images and extract handcraft features using $BOVW$ on the fly during the training, also we proposed a modified version of categorical cross-entropy loss by adding class wights which are calculated by balanced class function in Keras. We used the high imbalanced AffectNet and FER2013, datasets to evaluate the proposed model where the proposed model achieves accuracy 61.9% for eight expressions and 65% for seven expressions for AffectNet, and 69% for FER2013.

Also, the proposed model shows that the use of handcrafted features with features extracted by the deep learning model can improve the recognition accuracy for facial expressions.

Compared to State of Art methods our method achieves little better results than many of the existing methods for the AffectNet dataset where CNNCraft-net overpasses the best results by a range of 1% to 2%.

In future work, we aim to evaluate the proposed methods on additional data sets, improve the model to accept different image sizes, finally apply and custom the proposed method to work on facial expression recognition on video.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Ahmed Hesham Mostafa wrote the paper, designed, and perform experiments, proposed models, and algorithms, and analyzed the results. Hala Abdel-Galil El-Sayed supervised the study and verified the study's findings. Mohamed Belal supervised the study and verified the study's findings.

Reference

[1] S. Li and W. Deng, "Deep facial expression recognition: A survey", *IEEE Transactions on Affective Computing*, pp. 1-20, 2020.

[2] B. Chul, "A Brief Review of Facial Emotion Recognition Based on Visual Information", *Sensors*, Vol. 18, No. 2, pp. 401, 2018.

[3] G. Huang, Z. Liu, L. Maaten, and K. Weinberger, "Densely connected convolutional networks", In: *Proc. of* International *Conf. on computer vision and pattern recognition*, pp. 4700-4708, 2017.

[4] D. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, "Speeded-up robust features (SURF)", *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346-359, 2008.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF", In: *Proc. of International Conf. on computer vision*, pp. 2564-2571, 2011.

[7] B. Hasani, and M. Mahoor, "Facial expression recognition using enhanced deep 3D convolutional neural networks", In: *Proc of the International Conf. on computer vision and pattern recognition workshops*, pp. 30-40, 2017.

[8] I. Goodfellow, D. Erhan, P. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D. Lee, and Y. Zhou, "Challenges in representation learning: A report on three machine learning contests", In: *Proc. of International Conf. on neural information processing*, pp. 117-124, 2013.

[9] B. Han, W. Yun, J. Yoo, and W. Kim, "Toward Unbiased Facial Expression Recognition in the Wild via Cross-Dataset Adaptation", *IEEE Access*, Vol. 8, pp. 159172-159181, 2020.

[10] M. Georgescu, R. Ionescu, and M. Popescu, "Local learning with deep and handcrafted features for facial expression recognition", *IEEE Access*, Vol. 7, pp. 64827-64836, 2019.

[11] R. Ionescu, M. Popescu, and C. Grozeam, "Local learning to improve bag of visual words model for facial expression recognition", In *Workshop on Challenges in Representation Learning, ICML*, 2013.

[12] J. Li, Y. Mi, J. Yu, and Z. Ju, "A novel convolutional neural network for facial expression recognition", In: *Proc. of International Conf. on Cognitive Systems and Signal Processing*, pp. 310-320. 2018.

[13] Y. Li, J. Zeng, S. Shan, and X. Chen, "Patch-gated cnn for occlusion-aware facial expression recognition", In: *Proc. of International Conf. on Pattern Recognition*, pp. 2209-2214, 2018.

[14] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using CNN with attention mechanism", *IEEE Transactions on Image Processing*, Vol. 28, No. 5, pp. 2439-2450, 2018.

[15] C. Hewitt and H. Gunes, "Cnn-based facial affect analysis on mobile devices", *ArXiv preprint arXiv*, pp. 1807-08775, 2018.

[16] Y. Tang, "Deep learning using linear support vector machines", *ArXiv preprint*, pp. 1306-0239, 2013.

[17] W. Hua, F. Dai, L. Huang, J. Xiong, and G. Gui, "HERO: Human emotions recognition for realizing intelligent Internet of Things", *IEEE Access*,Vol. *7*, pp. 24321-24332, 2019.

[18] Z. Wei, J. Zhang, Z. Lin, J. Lee, N. Balasubramanian, M. Hoai, and D. Samaras, "Learning visual emotion representations from web data", In: *Proc of International Conf. on Computer Vision and Pattern Recognition*, pp. 13106-13115, 2020.

[19] H. Siqueira, S. Magg, and S. Wermter, "Efficient facial feature learning with wide ensemble-based convolutional neural networks", In: *Proc. of International Conf. on artificial intelligence*, Vol. 34, No. 4, pp. 5800-5809, 2020.

[20] Q. Ngo and S. Yoon, "Facial Expression Recognition Based on Weighted-Cluster Loss and Deep Transfer Learning Using a Highly Imbalanced Dataset", *Sensors*, Vol. 20, No. 9, pp. 1-20, 2020.

[21] W. Xiaohua, P. Muzi, P. Lijuan, H. Min, J. Chunhua, and R. Fuji, "Two-level attention with two-stage multi-task learning for facial emotion recognition", *Journal of Visual Communication and Image Representation*, Vol. 62, pp. 217-225, 2019.

[22] J. Zeng, S. Shan, and X. Chen, "Facial expression recognition with inconsistently annotated datasets", In: *Proc. of International Conf on computer vision*, 2018.

[23] A. Hesham, H. Galil, and M. Belal, "Benchmarking of Convolutional Neural Networks for Facial Expressions Recognition", *Journal of Theoretical and Applied Information Technology*, Vol. 98, No. 18, pp. 3104-3115, 2020.

[24] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures", In: *Proc. International Conf. on unsupervised and transfer learning*, pp. 37-49. 2012.

[25] B. Davida, "Bag of visual words in a nutshell", [online] Available: https://towardsdatascience. com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb, 2021.

[26] Keras Framework Applications, [online] Available: https://keras.io/applications/, 2021

[27] M. Borcan, "Python OpenCV: Building Instagram-Like Image Filters", [online] Available: https://programmerbackpack.com/python-opencv-building-instagram-like-image-filters/, 2021.

[28] Visual Similarity Duplicate Image Finder, [online] Available: https://www.mindgems.com/, 2021.

[29] Google Colab, [online] Available: https://colab.research.google.com, 2021