# Graph Neural Networks for Image Understanding Based on Multiple Cues: Group Emotion Recognition and Event Recognition as Use Cases

Xin Guo[1], Luisa F. Polanía[2], Bin Zhu[1], Charles Boncelet[1], and Kenneth E. Barner[1]
[1] Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, USA
Email: {guoxin, zhubin, boncelet, barner}@udel.edu
[2] Target Corporation, Sunnyvale, California, USA,
Email: Luisa.PolaniaCabrera@target.com

## Abstract

*A graph neural network (GNN) for image understanding based on multiple cues is proposed in this paper. Compared to traditional feature and decision fusion approaches that neglect the fact that features can interact and exchange information, the proposed GNN is able to pass information among features extracted from different models. Two image understanding tasks, namely group-level emotion recognition (GER) and event recognition, which are highly semantic and require the interaction of several deep models to synthesize multiple cues, were selected to validate the performance of the proposed method. It is shown through experiments that the proposed method achieves state-of-the-art performance on the selected image understanding tasks. In addition, a new group-level emotion recognition database is introduced and shared in this paper.*

## 1. Introduction

Deep learning methods have shined in many computer vision tasks [17, 19, 23, 53, 55] ever since Krizhevsky *et al*. [27] achieved the top classification accuracy on the Large Scale Visual Recognition Challenge (LSVRC) [40] in 2010. This overwhelming success is due to the ability of deep learning methods to learn at different levels of abstraction from data. Different tasks need different levels of abstraction. For example, tasks such as image segmentation focus on pixel-level information, whereas tasks such as GER and event recognition require a deeper semantic understanding of image contents and the aggregation of information from facial expressions, posture, people layout and background environments [12]. Single deep models are typically sufficient to achieve excellent performance for object recognition and image segmentation tasks while the aggregation of several deep models that extract features not only from the whole image but also from salient areas is typically needed for image understanding tasks [20, 45, 50].

Understanding the meaning and content of images remains a challenging problem in computer vision. Attempts to extract high-level semantic information for image understanding include the work in [30], which proposes the Object Bank, an image representation constructed from the response of multiple object detectors. Recently, modular networks have been proposed to perform visual understanding tasks by using several reusable and composable modules that carry on different functions [34]. In a nutshell, the state-of-the-art in image understanding is based on exploiting the principle of compositionality, meaning that a set of entities and their interactions are used to understand an image.

The aggregation of information from deep models trained on different entities or cues is typically implemented through decision and feature fusion [20, 22, 45, 43]. However, such methods neglect the fact that features can interact with each other to exchange information. Recurrent neural networks (RNNs) are widely used to aggregate features [28, 43, 50], but mostly from the same model since features of different models usually have different size. Another major drawback of RNN-based approaches is that they only consider sequential information but ignore spatial relations between entities present in the image.

Motivated by addressing the image understanding problem from learning features of multiple cues jointly, we propose a GNN model, which can be seen as a generalization of RNNs from sequential to graph data [38]. Features from regions of interest corresponding to multiple cues are extracted from the images and used as the nodes of the GNN. The hidden representation of each node evolves over time by exchanging information with its neighbors. One major advantage of the proposed model is its ability to deal with different number of inputs, which is relevant because the number of entities of interest vary between images, *e.g.* the number of faces. Another advantage is that each input is allowed to have a different size, which is important because different entities may have feature representations of dif-

ferent size. The performance of the proposed approach is validated on GER and event recognition tasks.

The models closer to the proposed model are those of [33] and [31] because they also use graphs to address image understanding tasks. However, the method in [33] focuses only on the problem of object detection. The method in [31] exploits connections across semantic levels, while the proposed method exploits connections between multiple cues and between features belonging to the same cue type. The model in [31] also differs from ours in the aggregation functions that are employed. Also, it does not use RNNs to update the features.

The major contributions of this work are summarized as follows: (1) A GNN model to address the problem of image understanding based on multiple cues. (2) The topology of the graph is dynamic because the number of entities of interest varies between images. Also, the proposed GNN model is able to deal with different number of inputs, where each input is allowed to have a different size. (3) A dataset is introduced to address the GER problem in realistic scenarios. (4) Extensive experiments are conducted to illustrate the performance of the proposed GNN on GER and event recognition tasks. Code and database are available at https://github.com/gxstudy/Graph-Neural-Networks-for-Image-Understanding-Based-on-Multiple-Cues.

## 2. Related Work

### 2.1. Graph Neural Network

Graph neural networks were first proposed by Gori *et al*. [18] and detailed in Scarselli [41] as a trainable recurrent message passing network applicable to sub-graph matching, web page ranking, and some toy problems derived from graph theory. Graph neural networks extend the notion of convolution and other basic deep learning operations to non-Euclidean grids [35]. In 2015, Li *et al*. [32] proposed to modify GNNs to use gated recurrent units (GRUs) and modern optimization techniques. Their work showed successful results in synthetic tasks that help develop learning algorithms for text understanding and reasoning [52]. In [25], Kipf and Welling introduced graph convolutional networks as multi-layer CNNs where the convolutions are defined on a graph structure for the problem of semi-supervised node classification. A message passing algorithm and aggregation procedure for GNNs proposed by Glimer [17] achieved state-of-the-art results for molecular prediction. In 2018, Meng *et al*. [35] proposed a GNN model to learn relative attributes from pairs of images. Meanwhile, a GNN model was proposed by Garcia and Bruna [16] to learn valuable information from limited and scarce training samples for image classification. In [38], a 3D GNN for RGBD semantic segmentation, which leverages both the 2D appearance information and 3D geometric relations, was proposed.

### 2.2. Group-level emotion recognition

Group-level emotion recognition has gained popularity in recent years due to the large amount of data available on social networks, which contain images of groups of people participating in social events. In addition, GER has applications in image retrieval [8], shot selection [9], surveillance [5], event summarization [9], social relationship recognition [21], and event detection [47], which motivates the design of automatic systems capable of understanding human emotions at the group level. Group emotion recognition is challenging due to face occlusions, illumination variations, head pose variations, varied indoor and outdoor settings, and faces at different distance from the camera which may lead to low-resolution face images.

Contextual information is crucial for the GER problem. In Figure 1, it would be difficult to infer the group emotion by only extracting information from faces, since many of the humans in the image are posing for the photo. However, it is only when contextual information is extracted, in the form of salient objects, such as demonstration posters, that the real emotion of the group is exposed.

The EmotiW Group-level Emotion Recognition Subchallenge [11] was created with the aim of advancing group-level emotion recognition. In this annual subchallenge, the collective emotional valence state is classified as positive, neutral, or negative using the Group Affect Database 2.0 [11, 10, 13]. In 2017, the winner of the subchallenge proposed fused deep models based on CNNs and trained on facial regions and entire images [45]. A deep hybrid network [20] using image scene, faces and skeletons attained the second place. In 2018, the top performance of the sub-challenge was attained with a deep hybrid network [22] based on faces, scenes, skeletons, and visual attentions. Cascade attention networks [48] based on face, body and image cues attained the second place and a four-stream deep network [24] consisting of the face-location aware global stream, the multi-scale face stream, a global blurred stream and a global stream attained the third place.

### 2.3. Event Recognition

With abundance of applications such as video surveillance and content-based video retrieval [46], solutions to the problem of event recognition have evolved from using hand-engineered features to deep models for both videos [14, 15, 26] and static images [29, 4, 53, 1]. Event recognition using static images is more challenging than using video because of the lack of motion information [49]. The interest in event recognition from static images has increased due to the explosive growth of web images, driven primarily by online photo sharing services such as Flickr and Instagram. Event recognition is challenging because behaviors of interest can have a complex temporal structure. For example, a wedding event is characterized by behav-

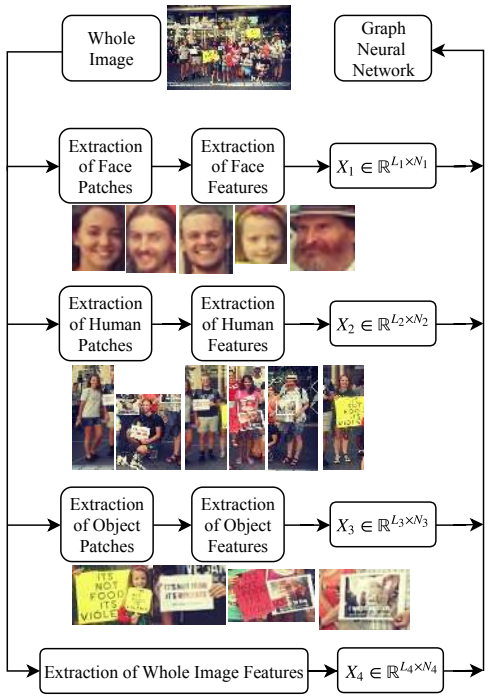


Figure 1. An illustration of how to build a complete graph from an image. Face, human, and object patches are first cropped using different object detection models, then features are extracted using CNN-based models. Each feature vector is a node in the graph.

iors that occur at different time, such as walking the bride, dancing, and flower throwing. Even though there are only 8 classes, each class encompasses many behaviors, which are visually very different from each other.

In [29], an aggregate model that jointly infers the classes of event, scene and objects from low-level features of images was proposed. Wang *et al*. [49] proposed a CNN which extracts useful information for event understanding from objects and scenes, and attained the first place in the task of cultural event recognition at the ChaLearn Looking at People (LAP) Challenge [4] in 2015. A framework that discovers event concept attributes from the web and use them to extract semantic features from images and classify them into social event categories was proposed in [2].

## 3. Proposed Graph Neural Network

Motivated by previous works [53, 29, 49] that show image understanding benefits from the extraction of information from multiple cues, a GNN-based model is designed to jointly learn feature representations from multiple cues.

Given an image $I$, assume that there are $T$ different cue types of interest for a certain image understanding task. For example, Figure 1 illustrates $T = 4$ cue types, namely, facial cues, human body cues, object cues and whole image cues. For each cue type $i$, $N_i$ features are extracted using deep models. For example, for the facial cues, $N_i$ may correspond to the number of detected faces in the image. The feature extraction operation for the $i$th cue is defined as

$$X_i = \psi_i(I), \tag{1}$$

where, $X_i = [x_{i,1}, \ldots, x_{i,N_i}] \in \mathbb{R}^{L_i \times N_i}$ and $\psi_i$ denotes the set of $L_i$-dimensional features and the feature extractor operator corresponding to the $i$th cue type, respectively. For example, for facial cues, a candidate for $\psi_i$ may be an operator that detects face patches in the image and aligns them, runs the face patches through a fine-tuned VGG-FACE model [37] and extracts the outputs from the fully-connected layer *fc7* to generate features.

To build the complete graph, each feature $x_{i,j}$ represents a node and every pair of distinct nodes is connected by an undirected edge. Note that $N_i$ may change across different images, for example, the number of faces changes across images, and therefore, every image has their own graph morphology. Since the feature length $L_i$ depends on the cue type, a function $f_i(\cdot)$ that converts the features to fixed-size vectors is needed. Although there are many options for the implementation of $f_i(\cdot)$, in this paper, the function is implemented with a single layer neural network as follows

$$h^0_{i,j} = f_i(x_{i,j}) = \text{ReLU}(W_i x_{i,j} + b_i), \tag{2}$$

where $h^0_{i,j} \in \mathbb{R}^{L_h}$ is the fixed-length feature vector associated to $x_{i,j}$, $W_i \in \mathbb{R}^{L_h \times L_i}$ and $b_i \in \mathbb{R}^{L_h}$ are the cue-type-specific weight matrix and bias, respectively. The vectors $h^0_{i,j}$ will hereafter be referred to as the hidden states of the nodes. Note that $W_i$ and $b_i$ are shared across nodes corresponding to the same cue-type and ReLU can be replaced with other functions.

The crucial idea of GNNs is that the vectors $h^0_{i,j}$ are iteratively updated by trainable nonlinear functions that depend on the hidden states of the neighbor nodes. This is accomplished by a GRU model in this paper. At every time step $k$, the hidden states are updated with a new $h^k_{i,j}$. Since the fixed-size features are the initial state input to the GRU, $L_h$ is also the number of hidden units in the GRU. As shown in Figure 2, a GRU unit takes the previous hidden state of the node $h^{k-1}_{i,j}$ and a message $m^k_{i,j}$ as input at each iteration, and outputs a new hidden state $h^k_{i,j}$. The message $m^k_{i,j}$, generated at time step $k$, is the aggregation of messages from the neighbors of the node, and is defined by the aggregation function $\phi(\cdot)$ as

$$m^k_{i,j} = \phi(\{h^{k-1}_{q,p} \mid \forall (q,p), (q,p) \neq (i,j)\}), \tag{3}$$

$$= \sum_{\substack{q,p \\ (q,p)\neq(i,j)}} W^e_q h^{k-1}_{q,p}, \tag{4}$$

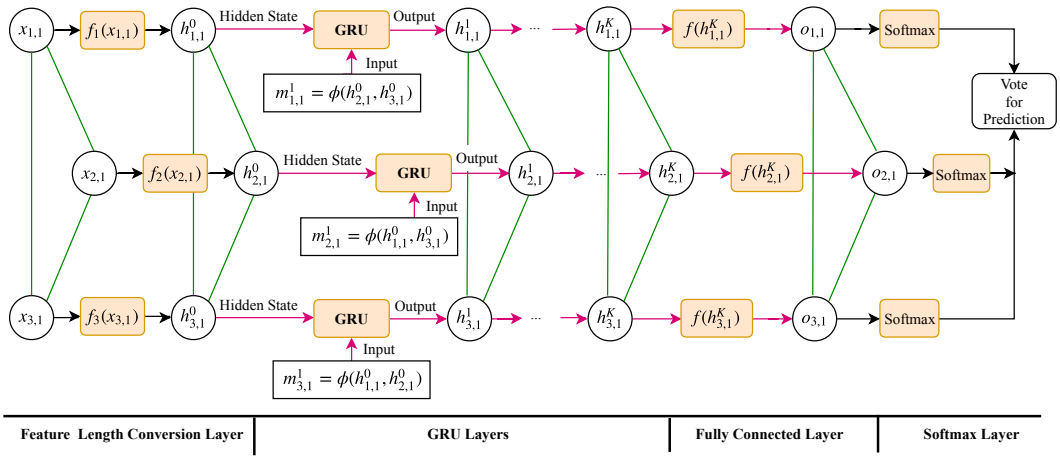


Figure 2. Illustration of a graph with 3 nodes. The feature vectors associated to the nodes are $x_{1,1}$, $x_{2,1}$ and $x_{3,1}$. The single layer neural networks used to convert the features from different cues to vectors of the same size are $f_1$, $f_2$ and $f_3$. The resulting fixed-length vectors are $h^0_{1,1}$, $h^0_{2,1}$ and $h^0_{3,1}$. At each time step, GRUs take both the previous hidden states $h^{k-1}_{1,1}$, $h^{k-1}_{2,1}$, $h^{k-1}_{3,1}$ and the messages $m^k_{1,1}$, $m^k_{2,1}$, and $m^k_{3,1}$ as inputs, and output the updated hidden states $h^k_{1,1}$, $h^k_{2,1}$ and $h^k_{3,1}$. After $K$ time steps, the hidden states are fed to a fully connected layer to output class scores $o_{1,1}$, $o_{2,1}$ and $o_{3,1}$. The Softmax layer normalizes the class scores between 0 to 1, and majority voting over the nodes determines the final prediction.

where $W^e_q \in \mathbb{R}^{L_h \times L_h}$ is the weight matrix associated to the neighbors whose cue type is $q$. Note that the neighbors are all the other nodes since the graph is complete. The cue-dependent matrices $W^e_q$ are learned during training.

The computations within the GRU, which allow the network to adaptively reset or update its memory content, are formally expressed as follows:

$$\begin{aligned} z^k_{i,j} &= \sigma(W_z m^k_{i,j} + U_z h^{k-1}_{i,j}), \\ r^k_{i,j} &= \sigma(W_r m^k_{i,j} + U_r h^{k-1}_{i,j}), \\ \tilde{h}^k_{i,j} &= \tanh(W_h m^k_{i,j} + U_h(r^k_{i,j} \odot h^{k-1}_{i,j})), \\ h^k_{i,j} &= (1 - z^k_{i,j}) \odot h^{k-1}_{i,j} + z^k_{i,j} \odot \tilde{h}^k_{i,j}, \end{aligned} \tag{5}$$

where $r^k_{i,j}$ and $z^k_{i,j}$ are the reset and update gates, $\tilde{h}^k_{i,j}$ is the candidate memory content, $\sigma(\cdot)$ is the logistic sigmoid function, and $\odot$ denotes the element-wise multiplication operation, and matrices $W_z$, $W_r$, $W_h$, $U_z$, $U_r$, and $U_h$ are model parameters. The update gate $z^k_{i,j}$ controls how much of the previous memory content is to be forgotten and how much of the candidate memory content is to be added. The model parameters of the GRU are shared across all nodes, thus providing an explicit control on the number of parameters. After training the GRU for $K$ time steps, all the nodes have learned from their neighbors during $K$ iterations. Note that the functions that define the update of the hidden states specify a propagation model of information inside the graph.

The final stage of the GNN consists in pushing the last hidden states through a fully-connected (FC) layer followed by a Softmax layer to generate the class probabilities. The total number of classes is denoted as $C$. The FC layer is represented with the function $f(\cdot)$, which is defined as

$$o_{i,j} = f(h^K_{i,j}) = W h^K_{i,j} + b, \tag{6}$$

where $W \in R^{C \times L_h}$ and $b \in R^C$ are the weights and bias term of the FC layer and are the same for all the nodes in the network. The class probabilities are generated by the Softmax layer as follows,

$$p^c_{i,j} = \frac{e^{W_{(c)} h^K_{i,j} + b_{(c)}}}{\sum_{l=1}^{C} e^{W_{(l)} h^K_{i,j} + b_{(l)}}}, \tag{7}$$

where $p^c_{i,j}$ is the probability for class $c$, $W_{(c)}$ is the $c$th row of $W$ and $b_{(c)}$ is the $c$th component of $b$. The predicted class of a node is the class with the largest probability, and the final prediction of the GNN is computed by using majority voting over the class predictions of the nodes. Figure 2 illustrates the structure of the proposed GNN.

The GNN is trained using backpropagation through time and the cross entropy loss function for multiple cues, which is defined, for each training sample, as

$$L = -\frac{1}{\sum_i N_i} \sum_{i,j} \sum_c y_c log(p^c_{i,j}), \tag{8}$$

where $y_c$ is the ground-truth for class $c$.

## 4. Experiments

In this section, the GroupEmoW database is introduced. Details of the implementation of the proposed GNN and comparisons with baseline and state-of-the-art methods are also provided.

| Dataset | Partition | Neg | Neu | Pos |
|---|---|---|---|---|
| Group Affect Database 2.0 [13] | Train | 2759 | 3080 | 3977 |
| | Val | 1231 | 1368 | 1747 |
| | Test | 1266 | 916 | 829 |
| MultiEmoVA [36] | – | 68 | 73 | 109 |
| GroupEmoW | Train | 3019 | 3463 | 4645 |
| | Val | 861 | 990 | 1327 |
| | Test | 431 | 494 | 664 |

Table 1. Dataset distribution of the proposed GroupEmoW dataset and currently available datasets for GER, where column names Neg, Neu and Pos correspond to negative, neutral and positive, respectively.

## 4.1. Datasets

### 4.1.1 GroupEmoW: A New GER Dataset

Datasets are crucial for building deep learning models. Even though there are many images of groups of people on social media and a strong interest in GER, labeled data is still scarce. In this paper, a new group-level emotion dataset in the wild, referred to as GroupEmoW, is introduced. The images are collected from Google, Baidu, Bing, and Flickr by searching for keywords related to social events, such as funeral, birthday, protest, conference, meeting, wedding, etc. Collected images form an in-the-wild dataset, with different image resolutions. The labeling task was performed by trained human annotators, including professors and students. Each image is labelled by 5 annotators, and the ground-truth is determined by consensus. Images are removed from the dataset if a consensus is not reached.

The collective emotion of the images are labeled between negative, neutral, and positive valence states. The total number of $15,894$ images in the GroupEmoW database is divided into train, validation and test sets with $11,127$, $3,178$ and $1,589$ images, respectively. The distribution of samples and comparison with currently available datasets for the GER problem are shown in Table 1. Sample images of the GroupEmoW database are shown in Figure 3.

### 4.1.2 Group Affect Database 2.0

The Group Affect Database $2.0$ [12] contains $9,816$, $4,346$ and $3,011$ images in the train, validation and test sets, respectively. These images are associated to social events, such as convocations, marriages, parties, meetings, funerals, protests, etc. This is the dataset employed by the GER sub-challenge of the Emotion Recognition in the Wild (EmotiW) Grand Challenge [13]. The labels of train and validation sets are provided while the labels of the test set are unknown. The size of the Group Affect Database $2.0$ was increased from $6,467$ in 2017 to $17,173$ in 2018.

Figure 3. GroupEmoW samples. First row: negative valence state. Middle row: neural valence state. Last row: positive valence state.

### 4.1.3 Social Event Image Dataset (SocEID)

The Social Event Image Dataset (SocEID) [2] is a large-scale dataset that consists of $37,000$ images belonging to $8$ event classes (birthdays, graduations, weddings, marathons/races, protests, parades, soccer matches and concerts). It was collected by querying Instagram and Flickr with tags related to the event of interest. This dataset also contains some relevant images from the NUS-WIDE dataset [7] and the Social Event Classification subtask from MediaEval 2013 [39]. SocEID contains $27,718$ and $9,254$ samples in the train and test sets, respectively.

## 4.2. Implementation and Results

Three baseline methods are proposed as follows:

(1) A fine-tuned CNN model based on whole images, referred to as CNN-Image. The selected pre-trained CNN is SE-ResNet-50 [23], which is a $50$-layer version of the SENet-154 model [23], which was trained on the ImageNet-1K database and achieved the highest accuracy in the ILSVRC 2017 image classification challenge*. All the learning parameters are adopted from the original model, with the exception of the size of the last FC layer, which is set the same as the number of classes of the problem of interest (3 for GER and 8 for event recognition), and the learning rate, which is initialized to $0.0005$.

(2) GRU and long short-term memory (LSTM) models trained on single cue types. For example, for facial cues, these models treat facial features within one image as one input sequence. The output of the RNN, either GRU or LSTM, is connected to an FC layer followed by a Softmax layer to generate predictions. The learning rate and length of the hidden state vectors of these models are set to $0.0001$ and 128, respectively. The GRUs trained for faces and objects are referred to as GRU-Face and GRU-Object, respectively. The LSTM models trained for faces and objects are referred to as LSTM-Face and LSTM-Object, respectively.

*The SE-ResNet-50 and SENet-154 pre-trained models are downloaded from https://github.com/hujie-frank/SENet.

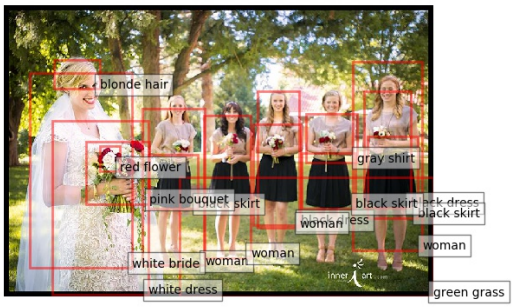


Figure 4. Salient object areas within one image.

Even though GRU and LSTM models can handle variable-length input sequences, each input must have the same feature size, which lead us to train GRU and LSTM models only on single cues.

(3) Additional baselines referred to as CNN-VGG-F and CNN-Skeleton are described in Sections 4.2.1 and 4.2.2, respectively.

In addition, the performance of the proposed GNN is also compared with state-of-the-art methods for GER and event recognition. All experiments are performed 10 times, using 20 epochs each time. For the GroupEmoW database, the model at the epoch with the highest average accuracy on the validation set is selected. Results on the test set using the selected model are reported. For SocEID, given that the dataset is divided into two partitions only, and for the Group Affect Database 2.0, given that the test labels are unknown, the model at the epoch with the highest average accuracy on the training set is selected. Results on the validation set are reported for the Group Affect Database 2.0., while results on the test set are reported for SocEID. For all the GNN models, the learning rate, the number of time steps $K$, and the length of the hidden state vectors $L_h$ are set to 0.0001, 4, and 128, respectively. The learning rate was selected using grid search in $\{0.00001, 0.0001, 0.001, 0.01\}$. The performance metric used for evaluation is the classification accuracy.

#### 4.2.1 Experiments on the GroupEmoW Database

Three cue types, namely, facial, object and whole image cues are explored for the GER task using the GroupEmoW database. Face patches are extracted and aligned using MTCNN [54]. A VGG-FACE model [37] initially trained on 2.7M images for face recognition is fine-tuned in the same way as described in [22] but using the training set of the GroupEmoW database. Once fine-tuned, the features of the FC layer *fc7* are extracted from each face patch and used as input to the GRU, LSTM, and GNN models. A baseline method, referred to as CNN VGG-F and described in [22], is implemented by running the face patches of an image through the fine-tuned VGG-FACE model and averaging the generated class probabilities across faces to finally select the class with the largest average class probability.

Let $i = 1$ be the index assigned to the facial cue type. The number of face feature vectors $N_1$ extracted from an image is restricted to be less or equal than $N_1^{\max}$. During training, to ensure that all of the faces from most of the training images are selected, $N_1^{\max}$ is set to 16 since $84.68\%$ of the training images in the GroupEmoW database contain less than 16 faces. If more than $N_1^{\max} = 16$ faces are detected in an image, then $N_1^{\max} = 16$ faces are randomly selected to extract features from them. During testing, $N_1^{\max}$ is set to 48 since $98.46\%$ of the testing images contain less than 48 faces. If more than 48 faces are detected in a testing image, the first 48 faces to be detected are selected. Therefore, faces are selected in a deterministic fashion during testing. The reason for using a smaller $N_1^{\max}$ for training than for testing is to prevent images with a large number of faces to excessively influence the learning of the network.

Facial features within one image are treated as one input sequence by the GRU and LSTM models. Therefore, the maximum sequence length is 16 for training and 48 for testing. The number of time steps for the GRU and LSTM models are equal to the input sequence length, which is the number of extracted face patches $N_1$.

For the object cues, the attention mechanism proposed in [3] is used to extract the salient objects. The SENet-154 model [23] trained on the ImageNet-1K database is employed to extract a 2048-dimensional feature representation for each salient object by using the output of layer *pool5/7x7_s1*. As shown in Figure 4, the attention mechanism is able to detect salient objects, such as humans, bouquet and grass. The areas detected by the attention mechanism are sorted by the confidence of the predictions. Let $i = 2$ be the index of the object cues. The number of feature vectors for the salient objects is restricted to be less or equal than $N_2^{\max}$; therefore, if more than $N_2^{\max}$ salient objects are detected by the attention mechanism, then only the salient objects with the top $N_2^{\max}$ scores are selected for feature extraction. The value of $N_2^{\max}$ is set to 16 for the experiments in this section.

For the whole image cues, in order to show that the proposed GNN is able to handle features of different length, an Inception-V2 [44] model pre-trained on the ImageNet-1K database is fine-tuned as described in [22] but using the training set of the GroupEmoW database. Once fine-tuned, the features of the *global_pool* layer with dimension 1024 are extracted and used as input to the GNN models.

The performance of the GNN model is evaluated by progressively adding cues of different type. First, the performance of the GNN using facial cues only, referred to as GNN-Face, and object cues only, referred to as GNN-

| Method | Avg_V | Max | Min | Med | Avg |
|---|---|---|---|---|---|
| CNN-Image | 80.14 | 82.38 | 79.11 | 81.25 | 81.22 |
| CNN-VGG-F | 83.17 | 82.52 | 81.95 | 82.27 | 82.26 |
| GRU-Face | 85.66 | 85.65 | 84.83 | 85.15 | 85.28 |
| LSTM-Face | 85.58 | 85.27 | 84.45 | 84.70 | 84.86 |
| **GNN-Face** | 85.54 | 85.02 | 84.14 | 84.64 | 84.68 |
| GRU-Object | 85.38 | 85.58 | 83.95 | 84.58 | 84.83 |
| LSTM-Object | 85.25 | 85.52 | 83.95 | 84.77 | 84.92 |
| **GNN-Object** | 85.93 | 86.21 | 85.08 | 85.71 | 85.66 |
| **GNN F+O** | 89.71 | 89.80 | 88.35 | 89.03 | 89.06 |
| **GNN F+O+I** | **89.79** | **89.93** | **88.60** | **89.11** | **89.14** |

Table 2. Experimental results on the GroupEmoW dataset. Avg_V refers to the average accuracy on the validation set, while Max, Min, Med, Avg are maximal, minimal, median and average accuracy on the test set. F, O, and I refer to face, object and whole image cues, respectively.

Object, is evaluated. Next, the performance of the GNN that uses both object and facial cues, referred to as GNN F+O, is evaluated. The last model to be evaluated is the GNN that uses face, object and whole image cues, referred to as GNN F+O+I. Results shown in Table 2 demonstrate that the proposed GNN F+O+I model outperforms the baseline methods. Each cue type adds information that is needed to improve the overall accuracy. Note that both GRU-Face and LSTM-Face slightly outperform GNN-Face, while GNN-Object outperforms both GRU-Object and LSTM-Object. This may be due to the fact that similarity between face patches is much higher than similarity between salient objects, and therefore, the task of predicting group-level emotion from faces may benefit from a simpler model. Instead, relations between salient objects are more semantic and may need more elaborate models.

#### 4.2.2 Experiments on the Group Affect Database 2.0

In addition to the three cues used for the GroupEmoW Database, skeleton cues are also used for the Group Affect Database 2.0. Skeleton images have been used in [20, 22] for group level emotion recognition and offer crucial information related to people layout and postures. Skeleton images only contain the landmarks of the faces and limbs and their connections (Figure 5). OpenPose [6, 42, 51] is used to extract skeleton images in the same way as described in [20, 22]. The SE-ResNet-50 is fine-tuned on skeleton images in the same way as described in [22]. Once fine-tuned, the features of the *pool5/7x7_s1* layer are extracted from each skeleton image and used as one of the inputs of the GNN model. The CNN model trained on skeleton images, described in [22], and referred to as CNN-Skeleton, is used as a baseline method in Table 3.

As in Section 4.2.1, the number of features for the facial and object cues is also restricted to be less or equal than $N_1^{\max}$ and $N_2^{\max}$, respectively. During training, to ensure that all of the faces from most of the training images are selected, $N_1^{\max}$ is set to 16 since $86.72\%$ of the training images in the Group Affect Database 2.0 contain less than 16 faces. During testing, $N_1^{\max}$ is set to 48 since $98.58\%$ of the testing images contain less than $48$ faces. For the object cues, $N_2^{\max}$ is set to 16 for both training and testing.

Figure 5. A sample image for the negative valence state from the Group Affect Database 2.0. and its corresponding skeleton image.

As in Section 4.2.1, the performance of the GNN is evaluated by progressively adding cues of different type. Other than the comparisons with the baseline models in Table 3, GNN is also compared to state-of-the-art methods. Since the methods described in [13, 24, 48, 22] report their best predictions across different experiments on the validation set, their results are placed in the column that reports the maximum accuracy in Table 3. We are unable to evaluate the performance of the proposed GNN on the test set since the test labels are unavailable. In terms of average and median accuracy, experimental results show that GNN-based models outperform GRU and LSTM models trained on single cues. The proposed model that exploits face, object, whole image and skeleton cues, referred to as GNN F+O+I+S, outperforms all the state-of-the-art methods in Table 3, except the model in [48], which attains high accuracy on the validation dataset but lower accuracy than the model in [22] on the test set.

#### 4.2.3 Experiments on the SocEID Database

The same cues used in Section 4.2.1 are employed for the event recognition task, with the exception of the facial cues, which are replaced by human body cues since faces are not as important as human bodies when it comes to recognizing activities and scene categories. Human body bounding boxes are detected and cropped in the following way: face and body keypoints are first detected using OpenPose [6, 42, 51], the width and height of the bounding boxes for the detected keypoints are calculated, and then increased by $20\%$. Any bounding box region that lies outside the image is cropped to fit within the image.

Since the average number of humans in the SocEid dataset is only 2, human body cues-based CNNs are not trained in this paper. Instead, the human body features are

| Method | Max | Min | Median | Avg |
|---|---|---|---|---|
| CNN-Image [22] | 68.16 | – | – | – |
| CNN-Skeleton [22] | 64.42 | – | – | – |
| CNN-VGG-F [22] | 68.28 | – | – | – |
| GRU-Face | 75.34 | 74.68 | 74.99 | 75.05 |
| LSTM-Face | 75.45 | 74.22 | 75.06 | 75.03 |
| **GNN-Face** | 75.48 | 73.96 | 75.24 | 75.00 |
| GRU-Object | 68.45 | 65.09 | 66.78 | 66.89 |
| LSTM-Object | 67.39 | 66.06 | 66.68 | 66.77 |
| **GNN-Object** | 69.16 | 67.76 | 68.34 | 68.32 |
| Inception-Img [13] | 65.00 | – | – | – |
| Multi-Models [24] | 78.39 | – | – | – |
| Multi-Models [48] | 86.90 | – | – | – |
| Multi-Models [22] | 78.98 | – | – | – |
| **GNN F+O** | 78.34 | 76.32 | 77.58 | 77.83 |
| **GNN F+O+I** | 78.87 | 76.65 | 77.97 | 77.96 |
| **GNN F+O+I+S** | **79.08** | **77.09** | **78.00** | **78.16** |

Table 3. Comparison with baseline and state-of-the-art methods using the validation set of the Group Affect Database 2.0 dataset. Note that the multi-model method in [48] attains high metrics in the validation set but the performance on the test set is lower than that of the method in [22]. F, O, I, and S refer to face, object, whole image, and skeleton cues respectively.

| Method | Max | Min | Median | Avg |
|---|---|---|---|---|
| CNN-Image | 89.18 | 87.86 | 88.66 | 88.62 |
| GRU-Object | 90.12 | 90.27 | 90.67 | 90.69 |
| LSTM-Object | 90.90 | 90.36 | 90.71 | 90.67 |
| **GNN-Object** | 91.47 | 90.79 | 91.27 | 91.17 |
| AlexNet-fc7 [2] | – | – | – | 86.42 |
| Event concept [2] | – | – | – | 85.39 |
| **GNN O+H** | 91.96 | 90.73 | 91.38 | 91.33 |
| **GNN O+H+I** | **92.09** | **91.27** | **91.52** | **91.61** |

Table 4. Experimental results on the SocEID dataset. O, H, and I refer to object, human and whole image cues, respectively.

extracted using the output of the *pool5/7x7_s1* layer from the pre-trained SENet-154 model. The number of human bodies to be extracted from a single image is restricted to be less or equal than 16 for both training and testing.

Features for the whole image and salient object cues are extracted in the same way as described in Section 4.2.1. The number of salient objects is restricted to be less or equal than 16. As in Section 4.2.1, the performance of the GNN is evaluated by progressively adding cues of different type. In Table 4, GNN O+H refers to the model that exploits object and human body cues, while GNN O+H+I refers to the model that uses object, human body, and whole image cues. Table 4 shows that the proposed models outperform baseline and state-of-the-art methods.

## 5. Discussion and Future work

The success of CNNs is partially owed to their ability to exploit local information, by enforcing a local connectivity pattern between neurons, and to aggregate and synthesize those local attributes in the upper layers of the network to learn high-level representations. However, there is a need to transition from models that are able to extract and aggregate local attributes for tasks such as object recognition and segmentation to models that are able to extract and aggregate local attributes for reaching a complete understanding of images. Progress in that direction has been attained with attention mechanisms that help models focus on the salient areas of the image. Traditional feature fusion approaches used to aggregate features from those salient areas ignore the relations between features and their ability to learn from each other. Similarly, RNN-based approaches ignore the spatial relations between salient areas, which are better described as a set than as a sequence. **The application of GNNs to image understanding tasks effectively learns feature representations for the salient regions by exchanging information between the graph nodes. The design of the GNN allows substantial weight sharing, which helps to avoid overfitting**.

There is no guarantee that all the extracted regions from the image provide relevant information for the task of interest, some of the regions may be uncorrelated or may introduce noise. Therefore, building a complete graph may not be optimal. **Future work will address the problem of efficiently connecting the graph nodes. In the future, we may also consider jointly learning the parameters of the GNN and the CNNs used for feature extraction in an end-to-end fashion**. The proposed method can be applied to other image understanding tasks that involve aggregating information from multiple cues, such as image captioning, visual grounding, and visual question answering.

## 6. Conclusion

A GNN-based framework for image understanding from multiple cues and a new database for the GER problem were presented in this paper. Image understanding not only refers to identifying objects in an image but also to learning the underlying interactions and relations between those objects. Exploiting those relations during the feature learning and prediction stages is achieved with GNNs by propagating node messages through the graph and aggregating the results. A variety of experimental results show that the proposed model achieves state-of-the-art performance on GER and event recognition tasks.

## 7. Acknowledgements

The work is supported by the National Science Foundation under Grant No. 1319598.

## References


[1] K. Ahmad, M. L. Mekhalfi, N. Conci, G. Boato, F. Melgani, and F. G. B. D. Natale. A pool of deep models for event recognition. In *ICIP*, pages 2886–2890, Sept 2017.

[2] U. Ahsan, C. Sun, J. Hays, and I. Essa. Complex event recognition from images with few training examples. *arXiv preprint arXiv:1701.04769*, 2017.

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[4] X. Baro, J. Gonzalez, J. Fabian, M. A. Bautista, M. Oliu, H. J. Escalante, I. Guyon, and S. Escalera. ChaLearn Looking at People 2015 challenges: Action spotting and cultural event recognition. In *CVPRW*, pages 1–9, June 2015.

[5] J. Bullington. Affective computing and emotion recognition systems: the future of biometric surveillance? In *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development*, pages 95–99, 2005.

[6] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017.

[7] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. NUS-WIDE: A real-world web image database from National University of Singapore. In *ICIVR*, pages 48:1–48:9, New York, NY, USA, 2009.

[8] A. Dhall, A. Asthana, and R. Goecke. Facial expression based automatic album creation. In *ICONIP*, pages 485–492, 2010.

[9] A. Dhall, R. Goecke, and T. Gedeon. Automatic group happiness intensity analysis. *IEEE-TAC*, 6(1):13–26, 2015.

[10] A. Dhall, R. Goecke, S. Ghosh, J. Joshi, J. Hoey, and T. Gedeon. From individual to group-level emotion recognition: EmotiW 5.0. In *ICMI*, pages 524–528, New York, NY, USA, 2017.

[11] A. Dhall, R. Goecke, J. Joshi, and T. Gedeon. Emotion recognition in the wild challenge 2016. In *ICMI*, pages 587–588, New York, NY, USA, 2016.

[12] A. Dhall, J. Joshi, K. Sikka, R. Goecke, and N. Sebe. The more the merrier: Analysing the affect of a group of people in images. In *FG*, volume 1, pages 1–8, 2015.

[13] A. Dhall, A. Kaur, R. Goecke, and T. Gedeon. EmotiW 2018: Audio-video, student engagement and group-level affect prediction. In *ICMI*, 2018.

[14] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, Oct 2005.

[15] L. Duan, D. Xu, I. W. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *CVPR*, pages 1959–1966, June 2010.

[16] V. Garcia and J. Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[17] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[18] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *IJCNN*, volume 2, pages 729–734, July 2005.

[19] X. Guo, L. Polanía, and K. Barner. Smile detection in the wild based on transfer learning. In *13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 679–686, May 2018.

[20] X. Guo, L. F. Polanía, and K. E. Barner. Group-level emotion recognition using deep models on image scene, faces, and skeletons. In *ICMI*, pages 603–608, New York, NY, USA, 2017.

[21] X. Guo, L. F. Polana, J. Garcia-Frias, and K. E. Barner. Social relationship recognition based on a hybrid deep neural network. In *14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pages 1–5, May 2019.

[22] X. Guo, B. Zhu, L. F. Polanía, C. Boncelet, and K. E. Barner. Group-level emotion recognition using hybrid deep models based on faces, scenes, skeletons and visual attentions. In *ICMI*, pages 635–639, New York, NY, USA, 2018.

[23] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 7, 2017.

[24] A. S. Khan, Z. Li, J. Cai, Z. Meng, J. O'Reilly, and Y. Tong. Group-level emotion recognition using deep models with a four-stream hybrid network. In *ICMI*, pages 623–629, New York, NY, USA, 2018.

[25] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[26] D. Koller, K. Tang, and L. Fei-Fei. Learning latent temporal structure for complex event detection. In *CVPR*, volume 00, pages 1250–1257, 06 2012.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, USA, 2012.

[28] J. Li, S. Roy, J. Feng, and T. Sim. Happiness level prediction with sequential inputs via multiple regressions. In *ICMI*, pages 487–493, New York, NY, USA, 2016.

[29] L. Li and L. Fei-Fei. What, where and who? classifying event by scene and object recognition. In *ICCV*, 2007.

[30] L. Li, H. Su, Y. Lim, and L. Fei-Fei. Object bank: An object-level image representation for high-level visual recognition. *IJCV*, 107(1):20–39, 2014.

[31] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *International Conference on Computer Vision*, 2017.

[32] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[33] Y. Liu, R. Wang, S. Shan, and X. Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6985–6994, 2018.

[34] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *CVPR*, pages 4942–4950, 2018.

[35] Z. Meng, N. Adluru, H. J. Kim, G. Fung, and V. Singh. Efficient relative attribute learning using graph neural networks. In *ECCV*, pages 552–567, September 2018.

[36] W. Mou, O. Celiktutan, and H. Gunes. Group-level arousal and valence recognition in static images: Face, body and context. In *FG*, volume 05, pages 1–6, May 2015.

[37] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[38] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3D graph neural networks for RGBD semantic segmentation. In *CVPR*, pages 5199–5208, 2017.

[39] T. Reuter, S. Papadopoulos, G. Petkos, V. Mezaris, V. Kompatsiaris, P. Cimiano, C. M. D. Vries, and S. Geva. Social event detection at MediaEval 2013 : Challenges, datasets and evaluation. In *MediaEval 2013 Multimedia Benchmark Workshop*, pages 1–2, Barcelona, Spain, 2013.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[41] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE-TNN*, 20(1):61–80, Jan 2009.

[42] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, volume 1, page 2, 2017.

[43] B. Sun, Q. Wei, L. Li, Q. Xu, J. He, and L. Yu. LSTM for dynamic emotion and group emotion recognition in the wild. In *ICMI*, pages 451–457, New York, NY, USA, 2016.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[45] L. Tan, K. Zhang, K. Wang, X. Zeng, X. Peng, and Y. Qiao. Group emotion recognition with individual facial emotion CNNs and global image based CNNs. In *ICMI*, pages 549–552, New York, NY, USA, 2017.

[46] C. Tzelepis, Z. Ma, V. Mezaris, B. Ionescu, I. Kompatsiaris, G. Boato, N. Sebe, and S. Yan. Event-based media processing and analysis: A survey of the literature. *Image and Vision Computing*, 53:3–19, 2016.

[47] T. Vandal, D. McDuff, and R. El Kaliouby. Event detection: Ultra large-scale clustering of facial expressions. In *FG*, volume 1, pages 1–8, 2015.

[48] K. Wang, X. Zeng, J. Yang, D. Meng, K. Zhang, X. Peng, and Y. Qiao. Cascade attention networks for group emotion recognition with face, body and image cues. In *ICMI*, pages 640–645, New York, NY, USA, 2018.

[49] L. Wang, Z. Wang, W. Du, and Y. Qiao. Object-scene convolutional neural networks for event recognition in images. In *CVPRW*, pages 30–35, 2015.

[50] Q. Wei, Y. Zhao, Q. Xu, L. Li, J. He, L. Yu, and B. Sun. A new deep-learning framework for group emotion recognition. In *ICMI*, pages 587–592, New York, NY, USA, 2017.

[51] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, pages 4724–4732, 2016.

[52] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[53] Y. Xiong, K. Zhu, D. Lin, and X. Tang. Recognize complex events from static images by fusing deep channels. In *CVPR*, pages 1600–1609, June 2015.

[54] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.

[55] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3D object detection. *arXiv preprint arXiv:1711.06396*, 2017.