Zero-Shot Facial Expression Recognition with Multi-Label Label Propagation[⋆]

Zijia Lu[1,2], Jiabei Zeng[1], Shiguang Shan[1,3,4], and Xilin Chen[1,3]

[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[2] NYU Shanghai, Shanghai, China
[3] University of Chinese Academy of Sciences, Beijing 100190, China
[4] CAS Center for Excellence in Brain Science and Intelligence Technology
{luzijia, jiabei.zeng, sgshan, xlchen}@ict.ac.cn

Abstract. Facial expression recognition classifies a face image into one of several discrete emotional categories. We have a lot of exclusive or non-exclusive emotional classes to describe the varied and nuancing meaning conveyed by facial expression. However, it is almost impossible to enumerate all the emotional categories and collect adequate annotated samples for each category. To this end, we propose a zero-shot learning framework with multi-label label propagation (Z-ML^{2}P). Z-ML^{2}P is built on existing multi-class datasets annotated with several basic emotions and it can infer the existence of other new emotion labels via a learned semantic space. To evaluate the proposed method, we collect a multi-label FER dataset FaceME. Experimental results on FaceME and two other FER datasets demonstrate that Z-ML^{2}P framework improves the state-of-the-art zero-shot learning methods in recognizing both seen or unseen emotions.

Keywords: Zero-Shot Learning · Facial Expression Recognition · Multi-Label Classification.

1 Introduction

Facial expression is an important part of human communication. Automatic facial expression recognition (FER) is a long-standing problem in computer vision and human-machine interaction. The FER problem is always defined as a multi-class classification that divides the facial expressions into several discrete emotional categories. Ekman Paul[8] has proposed six basic emotions: anger, disgust, fear, happiness, sadness, and surprise. However, they are insufficient to

[⋆] This work is done by Zijia Lu during his internship in Institute of Computing Technology, Chinese Academy of Sciences. We gratefully acknowledge the supports from National Key R&D Program of China (grant 2017YFA0700800), National Natural Science Foundation of China (grant 61702481), and External Cooperation Program of CAS (grant GJHZ1843). We also thank Yong Li for his help in adapting the annotation tool to Windows OS.

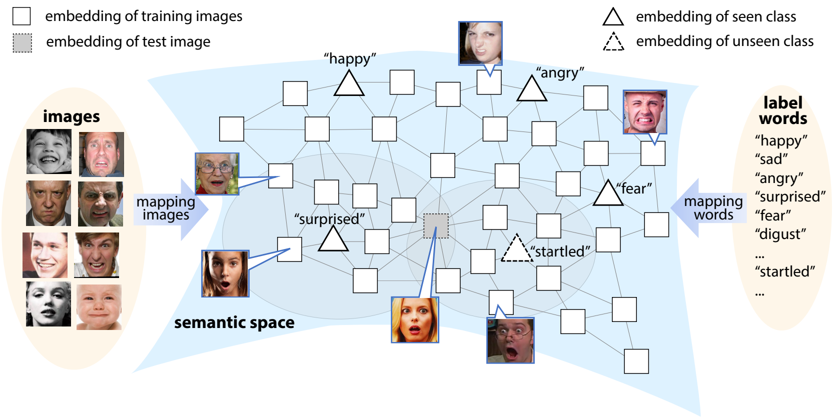


Fig. 1. The main idea of the proposed Z-ML^{2}P framework. Z-ML^{2}P embeds the facial images and all the emotion classes into a shared semantic space. Then, given a test image, we infer its labels by the proposed multi-label label propagation mechanism.

describe all the facial expressions. Some works extend the list with neutral[28], contempt[24], fatigue[20], engagement[20]. Other works annotate the facial expressions as mixtures of the basic emotions[7,22], such as "happily surprise" and "sadly fear". Unfortunately, there exist innumerable complex and subtle words to describe the nuanced emotions. It is prohibitive to gather a complete list of all emotion classes or collect adequate training samples for each emotion.

To address the issue, we propose a Zero-shot framework with Multi-Label Label Propagation on semantic space (Z-ML^{2}P). Z-ML^{2}P is built on existing multi-class datasets annotated with several basic emotions and can generalize to new emotions according to their relations with the basic emotions. Fig. 1 shows the main idea of our Z-ML^{2}P framework. It embeds the images and emotion classes into a shared semantic space. The class embeddings are mapped from the word vectors of the emotions' names, which implicitly encode the relation information between the emotions. The image embeddings are forced to be close to the embeddings of their belonging classes and similar images. In the end, a graph is built upon the embeddings to capture the manifold structure of the semantic space and propagate labels from class embeddings to the image embeddings via a proposed Multi-Label Label Propagation (ML^{2}P) mechanism. Our contributions are summarized as the following.

1. We build a Z-ML^{2}P framework to recognize the unseen emotion categories of facial images. To our knowledge, it is the first work to address facial expression recognition with emotions that are unseen during the training.
2. We construct a shared semantic space for facial images and the emotion classes, and then propose a novel multi-label label propagation (ML^{2}P) schema that can efficiently infer the seen or unseen emotions of a query image. Previous label propagation methods require a unlabelled dataset of new, unseen classes. We successfully remove the requirement.

3. We collect a *Fac*ial *e*xpression dataset with *M*ultiple *E*motions (FaceME) to evaluate the Z-ML^{2}P framework. Experimental results on FaceME and two other multi-label facial expression datasets demonstrate that our Z-ML^{2}P framework improves the state-of-the-art zero-shot learning methods in recognizing both seen or unseen emotions.

2 Related Work

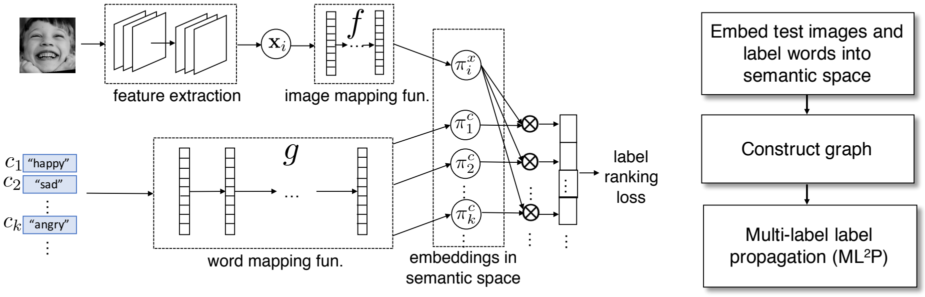
This section firstly reviews the recent work in facial expression recognition. Then, we briefly review zero-shot learning, multi-label learning, and label propagation. The three fields are technically relevant to the proposed Z-ML^{2}P.

Facial Expression Recognition(FER) Efforts have been made during the last decades in recognizing facial expressions[11,10,6,5,23,33]. Most existing methods classify the facial expressions into several discrete emotion categories, either based on hand-crafted features[41,18] or using end-to-end deep learning frameworks [38,37]. Ekman and Friesen proposed 6 basic emotions[8]. Du et al.[7] proposed 21 emotions as different combinations of the basic emotions. With these combined emotions, the FER can be solved as either a multi-class[24] or multi-label[42] classification problem. Meanwhile, some works employ Facial Action Coding System[9]. It uses a set of specific localized movements of the face, called Action Units, to encode the facial expression[43,31,40]. Some other works use three contiguous numerical dimensions to encode emotions[26,34].

Zero-Shot Learning (ZSL) ZSL methods are capable of correctly classify data of new, never-seen classes. Class embedding is the key component to connect seen classes and unseen classes. One type of representations are human-crafted attribute[30]. Yet those attributes are hard to obtain for new class as it requires expert knowledge. An alternative is using the word vectors of the class names, such as pretrained word vectors via GloVe[32] and Skip-Gram[27] methods, or learning new word vectors from textual corpus [35,15]. To compare images with class embeddings, most ZSL methods introduce a semantic space[19,14,39] where images and classes are mapped according to their semantic meanings.

Multi-Label (ML) Learning In multi-label learning, each instance can be assigned with multiple labels simultaneously. Typically, it is addressed from two approaches: *binary relevance* and *label ranking.*[16] *Binary relevance* splits a multi-label task into multiple binary classification problems. Its deficit is failure to capture class correlation and interdependence. *Label ranking* is frequently employed as an alternative, especially in large scale problems[39]. Its objective is to rank relevant labels ahead of the irrelevant ones thus requires the model to understand class relations. Thus It has been used in many multi-class ZSL models[13,19,1,36]. In this paper, we also employ label ranking method to bridge multi-class task and multi-label task.

Label Propagation (LP) Label Propagation is a graph-based semi-supervised algorithm[44]. It builds graphs to model the manifold structure in data and utilize the structure to propagate label from labelled data to the unlabelled ones. Recent works[14,21] introduce it into ZSL to propagate label from class repre-



(a) Training (b) Inference

Fig. 2. (a) The network architecture to learn the semantic space during the training stage. (b) Diagram of the inference stage in the Z-ML^{2}P framework.

sentations to test images. However, these works require an axillary unlabelled dataset of the new, unseen classes. It violates with the essential ZSL assumption that unseen classes are unknown in advance and have no available data.

3 Learning the semantic space

Figure 2 shows the training stage (Fig. 2 (a)) and the inference stage (Fig. 2 (b)) of the proposed zero-shot learning framework with multi-label label propagation on semantic space (Z-ML^{2}P). At training stage, Z-ML^{2}P learns to embed the emotion classes and images into a shared semantic space. In the semantic space, the embeddings of classes must preserve their syntactic and semantic structure. The embeddings of images must surround those of their relevant classes. During inference, Z-ML^{2}P projects the seen and unseen classes as well as the training and test images into the semantic space. It then constructs a graph upon the embeddings to perform multi-label label propagation(ML^{2}P).

In the rest part of this section, we introduce the details of learning the semantic space. In Sec. 4 and Sec. 5, we present the inference procedure for unseen classes and seen classes, respectively.

3.1 Problem Setup

Let $\mathcal{C} = \{$"*happy*", "*sad*"$, \ldots\}$ denote the name set of all the classes that are seen or unseen. Without losing generality, we assume the first s elements are the seen classes and the rest u elements are unseen classes. c_k represents the k-th class in $\mathcal{C}$. $\mathcal{S} = \{1, 2, \ldots, s\}$ and $\mathcal{U} = \{s+1, \ldots, s+u\}$ are the index sets of seen and unseen classes respectively.

The training dataset $\{\mathbf{x}_i, \mathcal{Y}_i\}_{i=1}^n$ contains n images labelled with seen classes $\mathcal{C}_\mathcal{S}$. $\mathbf{x}_i \in \mathbb{R}^h$ is the h-dimensional feature of the i-th image. $\mathcal{Y}_i \subset \mathcal{S}$ is the index set of the i-th image's relevant classes. The test dataset $\{\mathbf{x}_i^t, \mathcal{Y}_i^t\}_{i=1}^{n_t}$ contains n_t images of both seen and unseen classes. $\mathbf{x}_i^t$ is the feature of the i-th test image.

$\mathcal{Y}_i^t \subset \mathcal{S} \cup \mathcal{U}$ is the index set of the i-th test image's relevant classes. The proposed method is expected to predict whether a test image is relevant to each of the seen and unseen classes.

3.2 The Learning Architecture

In this section, we introduce the general network structure to learn a shared semantic space for images and emotion classes. As shown in Fig.2.a, there are four important components: feature extraction unit, image mapping function f, word mapping function g and the ranking loss.

The network takes images and the class names $\{c_1, c_2, \dots, c_s\}$ as inputs. The feature extraction unit is a CNN network (e.g., residual CNN[17]) and extracts feature $\mathbf{x}_i$ of a given input image. f and g learns to embed the visual feature $\mathbf{x}_i$ and the class names into the semantic space. We use $\pi_i^x = f(x_i)$ to denote the embedding of the i-th image and $\pi_k^c = g(c_k)$ to denote the embedding of the k-th class. To ensure $\{\pi_k^c\}$ capturing the class relations, they are taken as the pretrained word vectors of the class names, or transformed word vectors but with constraints to maintain their original structure.

The ranking loss ensures the image embeddings are close to its relevant classes and far from its irrelevant classes. It requires that, for every relevant-irrelevant label pair, a image is embedded closer to its relevant class than the irrelevant one. The ranking loss is proved to have advantages in capturing class interdependence[3]. Moreover, ranking loss is applicable to multi-class problems and enables us to train a multi-label network on the existing multi-class datasets. There exist many variants of the label ranking loss. Here we present two of the widely-used formulas: max margin loss $\mathcal{L}_{\max}$ and soft margin loss $\mathcal{L}_{\mathrm{soft}}$:

$$\mathcal{L}_{\max}(\mathbf{x}_i, \mathcal{Y}_i) = \alpha_i \sum_{k_+ \in \mathcal{Y}_i} \sum_{k_- \in \mathcal{S} - \mathcal{Y}_i} \max\left(0, \mathit{\Delta}_{k_+ k_-} + S_{ik_-} - S_{ik_+}\right), \tag{1}$$

$$\mathcal{L}_{\mathrm{soft}}(\mathbf{x}_i, \mathcal{Y}_i) = \alpha_i \sum_{k_+ \in \mathcal{Y}_i} \sum_{k_- \in \mathcal{S} - \mathcal{Y}_i} \log\left(1 + \exp\left(S_{ik_-} - S_{ik_+}\right)\right), \tag{2}$$

where k_+ and k_- are the indices of the image's relevant and irrelevant classes, respectively. S_{ik} is the similarity between the π_i^x and π_k^c, such as the result of inner product $\langle \pi_i^x, \pi_k^c \rangle$. $\mathit{\Delta}_{k_+ k_-}$ in Eq. (1) is the margin between class k_+ and k_-. It is either predefined or set to 1 for all class pairs. α_i is a weight term. By varying it from 1, weighted loss can be formed. Both the losses force the image embedding to be closer to its relevant class k_+ than its irrelevant class k_- for every (k_+, k_-) pair.

4 Inference with ML^2P for unseen classes

During the test stage, we are given some new classes that are unseen during the training. Our goal is to predict whether a test image belongs to the unseen classes or not. Typical zero-shot learning methods[13,1,36,4,39,2] classify the image to its nearest classes in the semantic space. However, the nearest neighbor method is likely to fail when the embeddings are lying on a non-linear manifold. In the multi-label label propagation (ML^{2}P) mechanism, we construct a graph to model the manifold structure of each class.

To fully capture the manifold structure, we need large amount of images to build the graph. Yet, in zero-shot learning, we have no access to the images labelled with unseen classes. In ML^{2}P, the training images are used to estimate the unseen classes' manifolds. There are two reasons supporting our method. Firstly, although un-annotated, the unseen emotions or its resembling expressions are probable to occur in the training images because emotions are no-exclusive and highly correlated. Secondly, ML^{2}P benefits from the manifolds of seen classes revealed by the training images. The manifold structure of related seen emotions embodies valuable information to the unseen classes. If an unseen emotion is not related to any training images, ML^{2}P has a better estimation of the manifold shape than the methods with nearest neighbor mechanism.

4.1 Graph Construction

To predict whether a test image $\mathbf{x}$ belongs to an unseen class c, a directed graph $\mathcal{G}(\mathbf{x}, c) = \{V, E, \mathbf{W}, \mathbf{r}\}$ is constructed. V is the set of vertices, including the embeddings of the test image, training images and a subset of classes. Each vertex has a score indicating its relevance to class c. Thus the score of class c is 1 and those of the other classes are 0. The scores of the test image and training images are unknown and to be inferred. We denote the vertices scores as $\mathbf{r} \in [0,1]^{|V|} = \begin{bmatrix} \mathbf{r}_l \\ \mathbf{r}_u \end{bmatrix}$, where $\mathbf{r}_l$ denotes the known scores and $\mathbf{r}_u$ denotes the to-be-learned scores. E is the edge set. The weight on a edge indicates the similarity of vertices on the two ends. A larger weight indicates two more similar vertices. $\mathbf{W} \in \mathbb{R}^{|V| \times |V|}$ denotes the weight matrix. The weight of edge from vertex j to i is computed as:

$$w_{ij} = \frac{\exp(\alpha_{ij}/\tau)}{\sum_{p=1}^{|V|} \exp(\alpha_{jp}/\tau)} \tag{3}$$

where α_{ij} is the similarity of the i-th and j-th vertices, measured by inner product or negative euclidean distance and τ is a length-scale hyper-parameter.

Vertices of Class Embeddings it is straightforward to include all class embeddings in the graph and set the label score of class c to 1 and the other scores to 0. However, if a synonymous or related class of c is present, setting its score to 0 will misguide label propagation. To address the issue, we employ a simple yet effective method: exclude classes from $\mathcal{G}(\mathbf{x}, c)$ that are semantically-close to class c. However, for a large class set, identifying those classes itself is cumbersome. Therefore, we first divide embeddings of all seen and unseen classes into a few groups via KNN then a selected representative class embedding for each

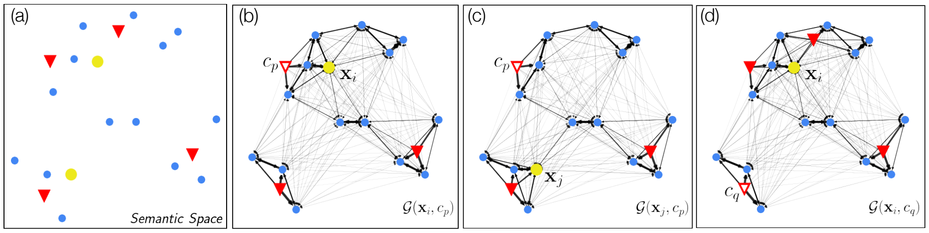

(a)
Semantic Space
(b)
c_p
$\mathbf{x}_i$
$\mathcal{G}(\mathbf{x}_i, c_p)$
(c)
c_p
$\mathbf{x}_j$
$\mathcal{G}(\mathbf{x}_j, c_p)$
(d)
$\mathbf{x}_i$
c_q
$\mathcal{G}(\mathbf{x}_i, c_q)$

Fig. 3. Illustration of ML^{2}P Graphs. (a) is an example of embeddings in the semantic space. Triangles are class embeddings. Small blue dots are training image embeddings and large yellow dots are test images embeddings. (b), (c), (d) demonstrate three graphs created for different test images and classes. The edge widths represent the magnitude of edge weights.

group, which can either be a real class embedding or an average of all embeddings in the group. This set of representative class, $\mathcal{C}^r$, are expected to cover the semantic meaning of most of the classes and it is much easier to find the semantically-close classes on this smaller set. Finally, class embeddings in $\mathcal{G}(\mathbf{x}, c)$ is $\{\pi_c\} \cup \{\pi_k | k \in \mathcal{C}^r;\ k,\ c \text{ is not semantically-close}\}$ and the label of class c is 1 and the others' are 0.

4.2 Multi-label Label Propagation

The multi-label label propagation (ML^{2}P) mechanism propagates labels from labelled vertices (e.g., vertices of classes) to unlabelled vertices (vertices of unlabelled images). To achieve it, on graph $\mathcal{G}(\mathbf{x}, c)$, ML^{2}P estimates the unlabelled vertices' scores $\mathbf{r}_u$ by minimizing

$$\min_{r_i \in \mathbf{r}_u} \frac{1}{2} \sum_{ij} w_{ij} (r_i - r_j)^2 \tag{4}$$

where r_i is i-th element of the $\mathbf{r} = [\mathbf{r}_l; \mathbf{r}_u]$, denoting the score of i-th vertex belonging to class c. Eq. (4) has a close-form solution

$$\mathbf{r}_u = (I - \mathbf{W}_{uu})^{-1} \mathbf{W}_{ul} \mathbf{r}_l \tag{5}$$

where $\mathbf{r}_l$ is the scores of labelled vertices. $\mathbf{W}_{uu}$ are the weights of edges between two unlabeled vertices and $\mathbf{W}_{ul}$ are those of edges from labelled ones to unlabelled ones. The weight matrix can be rewritten as $\mathbf{W} = \begin{bmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{bmatrix}$.

Acceleration To predict the relevance of a test image $\mathbf{x}$ to a class c, we have to construct a graph $\mathcal{G}(\mathbf{x}, c)$ and solve Eq.(5). The inversion of $I - \mathbf{W}_{uu}$ is hard to compute, with complexity $\mathcal{O}(n^3)$. n is the number of training images. To get the scores of all test images and all classes, the overall complexity is $\mathcal{O}[n_t \cdot (s+u) \cdot n^3]$. To reduce the computation cost, we propose the following acceleration method.

We observe that, for a fixed class c, the only difference among the graphs of different test images $\{\mathcal{G}(\cdot, c)\}$ is the vertex of the test image. The vertices of embeddings of classes and training images are the same. As the number of those shared vertices is much larger than 1(the number of test image vertex), it is safe to assume the scores of training images have little change when the test image changes. Therefore, we decompose the matrix $I - \mathbf{W}_{uu}$ to $\begin{bmatrix} I - \mathbf{W}_{tr,tr} & -\mathbf{w}_{tr,i} \\ -\mathbf{w}_{i,tr}^\top & 1 - w_{i,i} \end{bmatrix}$ and set $\mathbf{w}_{tr,i} = 0$, meaning that the label score of test image does not influence those of training images. $\mathbf{W}_{tr,tr}$ denotes the edge weights between training images and $\mathbf{w}_{tr,i}$ denotes the edge weights between training images and the test image i. When the test image changes, $I - \mathbf{W}_{tr,tr}$ remains the same and only the last row of $I - \mathbf{W}_{uu}$ is updated. With this property, the score of the test image can be efficiently obtained by Eq.(6) using Bordering Method[12].

$$r_i = (1 - w_{ii})^{-1} \left[\mathbf{w}_{i,l}^\top, \mathbf{w}_{i,tr}^\top\right] \begin{bmatrix} \mathbf{r}_l \\ \mathbf{r}_{tr} \end{bmatrix} \tag{6}$$

where $\mathbf{r}_l$ is the scores of labelled vertices.$\mathbf{W}_{ul} = \begin{bmatrix} \mathbf{W}_{tr,l} \\ \mathbf{w}_{i,l}^\top \end{bmatrix}$ is the edge weights between unlabelled and labelled vertices. $\mathbf{r}_{tr} = (\mathbf{I} - \mathbf{W}_{tr,tr})^{-1} \mathbf{W}_{tr,l}\mathbf{r}_l$ is the label scores of training images. It is the same for all graphs $\{\mathcal{G}(\cdot, c)\}$. Therefore, for a given class c, we only need to compute matrix inversion once to obtain $\mathbf{r}_{tr}$ with complexity $\mathcal{O}(n^3)$ and share it in all graphs. Given $\mathbf{r}_{tr}$, computing r_i only requires vector product with complexity $\mathcal{O}(n + s + u)$. The overall complexity is reduced to $\mathcal{O}[(s + u) \cdot n^3 + (s + u) \cdot n_t \cdot (n + s + u)]$.

Beta Normalization ML^{2}P propagates the label scores with respect to multiple classes. The score distributions *w.r.t.* different classes are often skewed and of different scales. Therefore, scores *w.r.t.* different classes are not comparable. We propose Beta Normalization(BN) to align the ranges of the distributions and remove the skewness. For each class, we estimate the score distribution as a Generalized Beta Distribution. Then the test images' scores are converted from absolute values to the percentile ranks by the cumulative distribution function.

Mathematically speaking, let r_{ik} denote the score of the i-th test image *w.r.t.* class k. $\mathbf{R}^u \in \mathbb{R}^{n_t \times u} = [r_{ik}]$ is the score matrix for all the test images and unseen classes. $\mathbf{r}_{*k}$ is $\mathbf{R}^u$'s k-th column, representing all test images' scores of belonging to class c_k. We assume r_{ik} follows Generalized Beta Distribution[25] with parameter $\boldsymbol{\theta}_k$ as $r_{ik} \sim GB(\boldsymbol{\theta}_k)$. The normalized score $\mathbf{r}'_{*k}$ is computed as

$$\mathbf{r}'_{*k} = F\left(\mathbf{r}_{*k}; \boldsymbol{\theta}_k\right) \tag{7}$$

where F is the cumulative distribution function. It converts $\mathbf{r}_{*k}$ as absolute values to $\mathbf{r}'_{*k}$ as percentile ranks. After the normalization, the distributions of $\mathbf{r}'_{*k}$ for all the classes are of same range with little skewness.

To estimate score distribution, adequate amount of data are required. In our settings of zero-shot learning, only one test image is available at inference time.

Thus we are not able to directly estimate the score distributions of test data. Instead, we take the score distributions of training images as an approximation and learn an approximated $\hat{\boldsymbol{\theta}}_k$ on training images then use it in Eq. (7).

5 Inference with ML^2P for Seen Classes

Similar to unseen classes, given a test image $\mathbf{x}$ and a seen class c, a graph $\mathcal{G}(\mathbf{x}, c) = \{V, E, \mathbf{W}, \mathbf{r}\}$ is built. V includes the embeddings of all training images, the test image and all seen classes. In contrast to the case of unseen classes, vertices of classes and training images are all labelled and the test image is the only unlabelled vertex. As labelled vertices are rich, class exclusion mentioned in section 4.1 is no longer needed. Then the new close-form solution is

$$r_u = (1 - w_{uu})^{-1} \mathbf{w}_{ul} \mathbf{r}_l \tag{8}$$

which does not involve matrix inversion so the acceleration step is omitted as well. Collecting r_u for all test images and seen classes, we have $\mathbf{R}^s$ as the score matrix for seen classes.

In terms of beta normalization, the distribution parameter $\hat{\boldsymbol{\theta}}$ cannot be directly approximated from training images as they are labelled vertices with discrete scores, 0 or 1. To address this issue, we use K-fold method: we divide training images into K folds and first pretend the images in the first fold are unlabeled while the others are still labelled, then perform LP to get the label scores for the first fold. Repeating this process to other folds, we get the label scores for all training images as if they are unlabeled. In the end, these scores are utilized to learn $\hat{\boldsymbol{\theta}}$.

After beta normalization, the normalized score of seen classes, $\mathbf{R}^{s\prime}$, is obtained. The final output for all classes $\mathbf{R}' = \left[\mathbf{R}^{s\prime}, \mathbf{R}^{u\prime}\right] \in \mathbb{R}^{n_t \times (s+u)}$.

6 Experiments

6.1 Datasets

We chose AffectNet Dataset as our training dataset and evaluated our model on RAF, Emotic and FaceME datasets. The latter is a multi-label dataset collected by ourselves. **AffectNet**[28] is a multi-class dataset with 287,618 training images and 8 basic emotion classes: {*neutrality, happiness, sadness, anger, surprise, fear, disgust, contempt*}. These 8 emotions are our seen classes. Please note we did not have large amount of multi-labelled images for training. Each AffectNet image only has one of the 8 emotion labels.

RAF[22] is a multi-class dataset containing of a 7-class basic emotion part and a 12-class compound emotion part. We used the compound label part in our experiments, where the labels are formed by combining 2 of the 6 *Seen* classes: {*happy, sad, anger, surprise, fear, disgust*}. We regarded the compound emotion classification problems as a multi-label problem. There are 3956 images with compound labels.

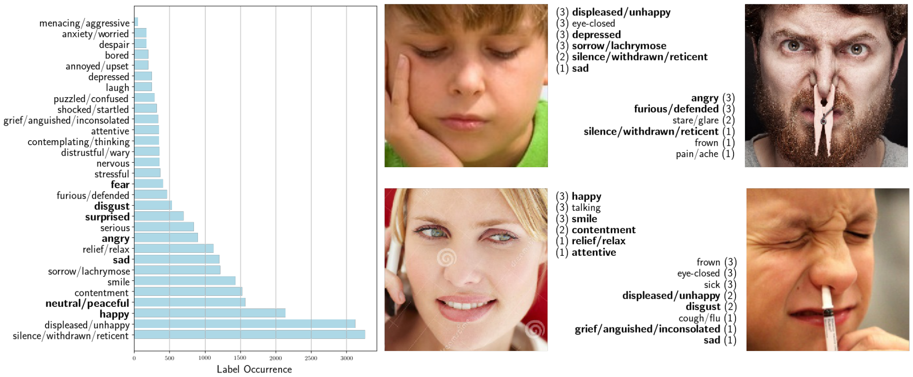


Fig. 4. FaceME dataset sample images and label occurrence of the selected 30 labels. Texts aside the images are their labels. The numbers in the parenthesis denote how many annotators choose that label.

Emotic[20] is a multi-label dataset with 26 emotion labels and images of people in real environments. Emotions have to be inferred from the context and in many of the images, faces are covered or invisible. To evaluate our model, we used a face detector to crop out the faces and got 7134 cropped images in the end. 20 of the 26 labels are unseen classes and 6 of them are seen classes: {*peace*(correspondent to *neutral*), *anger, fear, happiness, sadness, surprise*}. It represents a hard scenario with noisy label.

FaceME is a multi-label facial expression dataset collected by ourselves. It has 10062 images and 85 labels. The labels not only include emotions, but also labels about action, health and inward thoughts. Each image is labelled by 3 annotators. In the experiments, we discarded the labels annotated by only one person. From the 85 labels, we selected out 30 labels for evaluation that are emotion-related and have adequate amount of images and 7 of them are seen classes (not including *"contempt"*). Excluding the images not having the selected 30 labels, there are 9687 images left. In Fig. 4, the label occurrence distribution and sample images are shown.

6.2 Evaluation Protocol

We evaluated the models from three different aspects: discriminative separation of relevant and irrelevant labels, intra-class ranking and inter-class ranking.

Separation of relevant and irrelevant labels: it is evaluated by **F1**. Given the ground truth labels of a class and predictions for test images, we computed the Precision rate and Recall rate and the class's F1 score was the harmonic average of the two rates. The final F1 was the average of each class's F1. Since label ranking models only predict rank scores, we estimated two additional thresholds for each model to separate relevant and irrelevant classes, one for seen classes and one for unseen classes. We choose F1 instead of accuracy because the amounts of relevant and irrelevant labels are imbalanced.

Intra-class ranking: it is evaluated by **Mean Average Precision** (mAP). Given a ranked list of images for a certain class, mAP measures the area under precision-recall curve.

Inter-class ranking: we propose a new metric **Ranking F1** (rF1) to normalize the ranking accuracy according to the label imbalance. As the sample quantities of some frequent labels are much larger than the infrequent ones, if the accuracy score is not normalized, a model always ranking the frequent labels on the top of the list can reach a very high score.

Given a image with l relevant labels, we define a class c as positive if it is a relevant label and the prediction of c is positive if it is ranked as one of top l relevant labels. With these two definitions, the true positive rate and false positive rate of the rank result can be computed, so as a F1 score, which we name as Ranking F1. The final rF1 was the average of each class's rF1.

For each metrics, we reported results under three experiment setups: prediction results for the seen classes only (**S**), for the unseen classes only (**U**) and for all classes (**A**).

6.3 Implementation of Z-ML^{2}P framework

We implemented three configurations of the framework's four components introduced in Section 3.2. They correspond to three state-of-art ZSL models. In all of configurations, the feature extraction unit and f is the same. g and ranking loss follow the set-up in the original papers. We briefly introduce them below.

(1)**Fast0Tag(F0T)**[39]: It uses Soft Margin Loss and g is a lookup function which maps the class names to their word vectors.

(2)**ALE**[1]: It uses Weighted Max Margin Loss with the same g as F0T.

(3)**SYNC**[4]: It uses Structured Max Margin Loss in which margins are predefined according to class relations and g first maps the class names to the word vectors then embed them into semantic space (Model Space in [4]) via phantom classes.

The feature extraction part is an 80-layer residual CNN. It is pretrained on AffectNet by minimizing cross-entropy loss. The image features $\{\mathbf{x}_i\}$ are its last Conv layer's output. f is implemented as a single FC layer. It differs from the set-up in [39,1,4] only in the number of layers. GloVe vectors used in g are 300 dimension vectors pretrained on Wikipedia corpus. During training, the CNN and GloVe vectors are freezed. The other parts of the framework are optimized w.r.t the ranking losses via SGD on AffectNet. In AffectNet, each image is assigned to only one class. In ranking loss, this class is treated as a relevant label and the other classes as irrelevant ones.

For the representative class set of label propagation model, we found using the 8 basic seen classes is good enough. To estimate the hyperparameters, since the three datasets are relatively small and many classes have low occurrences, we did not divide validation and test set. Instead, for seen classes, we split the

Table 1. Results on RAF dataset.

	SYNC	SYNC+LP	SYNC+ML^{2}P	F0T	F0T+LP	F0T+ML^{2}P	ALE	ALE+LP	ALE+ML^{2}P
mAP	71.8	73.1	**73.1**	74.3	75.5	**75.5**	73.4	75.8	**75.8**
rF1	62.1	63.6	**63.6**	**64.4**	63.1	63.9	**64.3**	63.3	64.0
F1	61.9	64.3	**64.3**	64.2	63.5	**65.0**	63.7	64.8	**65.8**

data via KFold and tuned the hyperparameters on one fold then applied them onto the others and for unseen classes, we split classes via KFold.

For each configuration, we evaluated the prediction results by network only, by network and label propagation without beta normalization and by network and ML^{2}P.

6.4 Experimental Results

The results on RAF, Emotic and FaceME datasets are summarized in Table (1, 2, 3), respectively. In the tables, X denotes the result of network of certain configuration. *X+LP* denotes the result of network and our label propagation method without beta normalization. *X+ML^{2}P* denotes the result of network and ML^{2}P. It can be observed that our ML^{2}P model gives impressive results on intra-class, inter-class ranking and relevant-irrelevant label separations and on both seen and unseen classes.

It is worth noting that our Z-ML^{2}P framework is an efficient method to address multi-label zero-shot learning and applicable to various real-world tasks such as image annotation and image retrieval. With our acceleration method, the inference time is short. On an Intel i7 CPU, computing $\mathbf{r}_{tr}$ in Eq.(6) takes 0.3s for a seen class and 0.5s for a unseen class. Then it takes only 100μs to infer the labels of a test image.

Evaluations on Seen Classes For seen classes, the ML^{2}P models shows improvements on almost all evaluation metrics against their network baselines, especially on FaceME dataset. It shows, although the networks are directly optimized on seen classes, there still exists information that is missed by the network yet captured in the data manifold structure. In terms of rF1 score on RAF dataset, ML^{2}P is slightly worse than the network baseline. It is caused by incomparability of the scores of different classes mentioned in Sec. 4.2. Compared to LP method, ML^{2}P mitigates the issue although a small gap remains. The improvement in mAP and F1 still proves the effectiveness of our algorithm.

Evaluations on Unseen Classes The performance on unseen classes is the most important aspect of ZSL methods. ML^{2}P substantially outperforms the network baseline on both FaceME and Emotic datasets and on all metrics. The improvement in mAP confirms that, without the need of auxiliary data of unseen classes, the manifold structure of training images indeed facilitates label

Table 2. Results on FaceME dataset

		SYNC	SYNC+LP	SYNC+ML^{2}P	F0T	F0T+LP	F0T+ML^{2}P	ALE	ALE+LP	ALE+ML^{2}P
mAP	S	46.9	57.3	**57.3**	50.1	57.2	**57.2**	49.3	59.0	**59.0**
	U	12.1	17.5	**17.5**	13.0	19.5	**19.5**	12.5	19.3	**19.3**
	A	20.2	26.8	**26.8**	21.6	28.3	**28.3**	21.1	28.6	**28.6**
rF1	S	48.8	58.1	**65.3**	54.8	62.4	**67.1**	55.4	64.5	**67.6**
	U	9.7	3.6	**20.4**	11.4	3.3	**19.5**	11.2	2.8	**20.2**
	A	14.1	13.1	**27.8**	17.4	12.8	**27.4**	17.3	12.9	**28.2**
F1	S	39.1	47.7	**53.1**	46.5	49.4	**53.6**	45.9	51.5	**53.6**
	U	13.1	10.6	**19.3**	15.0	12.4	**21.2**	13.6	12.2	**19.9**
	A	19.2	19.3	**27.3**	22.3	21.0	**28.8**	21.1	21.4	**27.8**

Table 3. Results on Emotic Dataset

		SYNC	SYNC+LP	SYNC+ML^{2}P	F0T	F0T+LP	F0T+ML^{2}P	ALE	ALE+LP	ALE+ML^{2}P
mAP	S	20.5	21.5	**21.5**	22.0	22.4	**22.4**	20.4	21.6	**21.6**
	U	10.7	11.5	**11.5**	11.3	12.5	**12.5**	11.1	12.4	**12.4**
	A	12.9	13.8	**13.8**	13.8	14.8	**14.8**	13.2	13.6	**14.6**
rF1	S	26.5	33.1	**37.9**	32.4	37.4	**39.0**	33.2	37.5	**38.0**
	U	14.0	4.0	**15.5**	16.3	4.1	**18.1**	16.3	4.0	**17.5**
	A	13.7	8.3	**16.9**	15.7	9.1	**18.6**	15.4	8.6	**18.1**
F1	S	17.3	22.7	**22.9**	18.4	23.2	**23.2**	18.4	22.4	**22.6**
	U	13.0	12.9	**14.4**	11.3	11.7	**15.1**	13.2	9.5	**15.5**
	A	14.0	15.1	**16.3**	13.0	14.4	**16.9**	14.4	12.5	**17.1**

propagation. mAP of LP and ML^{2}P are the same because Beta Normalization does not change the results of intra-class ranking. rF1 score shows the increase in inter-class ranking accuracy. ML^{2}P successfully addresses the score incomparability issue of unseen classes.

Effectiveness of Beta Normalization Comparing the results of LP and ML^{2}P, we can observe the effectiveness of beta normalization. Although LP has higher mAP than the network baseline, its rF1 degrades seriously. It shows the score incomparability issue again. For the network baseline, the incomparability exists between seen classes and unseen classes. On Emotic dataset, the rF1 of networks' predictions for all classes is lower than either the score of seen classes only or unseen classes only. Beta normalization resolves the issue by correctly aligning the score distributions, as shown in the result of ML^{2}P. Moreover, it helps F1 score as well. Since all unseen classes or seen classes share the same threshold to separate relevant and irrelevant labels, if the scores are not aligned, there exists no good universal threshold.

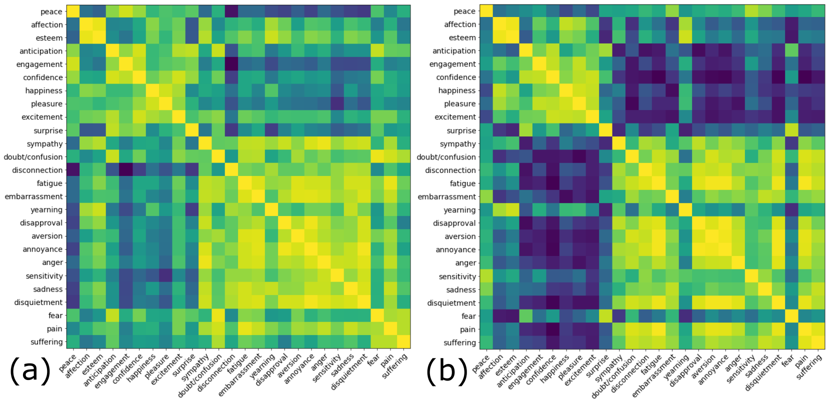


Fig. 5. Label Score Correlation Coefficient Matrix of ALE configuration on Emotic Dataset.(a) and (b) plot the matrix of network baseline and ML^{2}P. Darker in color means lower correlation.

6.5 Detailed Analysis

In this section, we first measured how well a model captures the relations among classes. We plotted the correlation coefficient of predicted scores between different classes in Fig. 5.a and 5.b of ALE configuration on Emotic dataset. On the correlation matrix of network baseline (Fig. 5.a), two bright blocks can be weakly identified, one for positive emotions and the other for the negative ones. In contrast, this structure is much more obvious on ML^{2}P's matrix (Fig. 5.b). It shows our method have strong capacity for learning class relations and is suitable for multi-label learning problems.

Due to space limitation, matrix of other configurations and the per-class mAP scores are presented in supplement. In addition, we showed the framework is robust to the feature extraction unit by replacing the residual CNN with VGG-Face network[29]. The results are also included in the supplement.

7 Conclusion

We proposed a novel Zero-shot learning framework with Multi-Label Label Propagation (Z-ML^{2}P) and collect a new *Fac*ial *e*xpression dataset with *M*ultiple *E*motions (FaceME). Our framework for the first time addresses multi-label zero-shot FER problem using existing multi-class emotion dataset only and successfully adopts label propagation to multi-label task and shows impressive improvement on both seen classes and unseen classes.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-Embedding for Image Classification. IEEE T-PAMI **38**(7), 1425–1438 (2016)
2. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In Proc. of CVPR (2015)
3. Bucak, S.S., Mallapragada, P.K., Jin, R., Jain, A.K.: Efficient multi-label ranking for multi-class learning: Application to object recognition. In Proc. of ICCV (2009)
4. Changpinyo, S., Chao, W.L., Gong, B., Sha, F.: Synthesized Classifiers for Zero-Shot Learning. In Proc. of CVPR (2016)
5. Cohn, J.F., De la Torre, F.: The Oxford Handbook of Affective Computing, chap. Automated Face Analysis for Affective Computing (2014)
6. De la Torre, F., Cohn, J.F.: Guide to Visual Analysis of Humans: Looking at People, chap. Facial Expression Analysis. Springer (2011)
7. Du, S., Tao, Y., Martinez, A.M.: Compound facial expressions of emotion. Proceedings of the National Academy of Sciences **111**(15), E1454–E1462 (2014)
8. Ekman, P., Friesen, W.V.: Constants across cultures in the face and emotion. Journal of personality and social psychology **17**(2), 124–9 (1971)
9. Ekman, P., Rosenberg, E.L.: What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press (1997)
10. Eleftheriadis, S., Rudovic, O., Pantic, M.: Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition. IEEE Transactions on Image Processing **24**(1), 189–204 (2015)
11. Fabian Benitez-Quiroz, C., Srinivasan, R., Martinez, A.M.: Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In Proc. of CVPR (2016)
12. Faddeev, D.K., Faddeeva, V.N.: Computational methods of linear algebra. Journal of Soviet Mathematics **15**(5), 531–650 (1981)
13. Frome, A., Corrado, G., Shlens, J.: Devise: A deep visual-semantic embedding model. In Proc. of NIPS (2013)
14. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Transductive Multi-View Zero-Shot Learning. IEEE T-PAMI **37**(11), 2332–2345 (2015)
15. Gaure, A., Gupta, A., Verma, V.K., Rai, P.: A Probabilistic Framework for Zero-Shot Multi-Label Learning. In Proc. of UAI (2017)
16. Gibaja, E., Ventura, S.: A tutorial on multilabel learning. ACM Computing Surveys **47**(3), 52:1–52:38 (2015)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In Proc. of CVPR (2015)
18. Kacem, A., Daoudi, M., Amor, B.B., Alvarez-Paiva, J.C.: A novel space-time representation on the positive semidefinite cone for facial expression recognition. In Proc. of ICCV (2017)
19. Kodirov, E., Xiang, T., Gong, S.: Semantic Autoencoder for Zero-Shot Learning. In Proc. of CVPR (2017)
20. Kosti, R., Alvarez, J.M., Recasens, A., Lapedriza, A.: Emotion recognition in context. In Proc. of CVPR (2017)
21. Li, A., Lu, Z., Wang, L., Xiang, T., Li, X., Wen, J.R.: Zero-Shot Fine-Grained Classification by Deep Feature Learning with Semantics. CoRR **abs/1707.00785**, 1–10 (2017)

22. Li, S., Deng, W., Du, J.: Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In Proc. of CVPR (2017)
23. Liu, M., Shan, S., Wang, R., Chen, X.: Learning expressionlets via universal manifold model for dynamic facial expression recognition. IEEE Transactions on Image Processing **25**(12), 5920–5932 (2016)
24. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Proc. of CVPR (2010)
25. McDonald, J.B., Xu, Y.J.: A generalization of the beta distribution with applications. Journal of Econometrics **66**(1), 133 – 152 (1995)
26. Mehrabian, A.: Framework for a comprehensive description and measurement of emotional states. Genet. Soc. Gen. Psychol. Monogr. **121**(3), 339–361 (1995)
27. Mikolov, T., Corrado, G., Chen, K., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In Proc. of ICLR (2013)
28. Mollahosseini, A., Hasani, B., Mahoor, M.H.: Affectnet: A database for facial expression, valence, and arousal computing in the wild. IEEE Transactions on Affective Computing (2017)
29. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In Proc. of BMVC (2015)
30. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In Proc. of CVPR (2012)
31. Peng, G., Wang, S.: Weakly supervised facial action unit recognition through adversarial training. In Proc. of CVPR (2018)
32. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation. In Proc. of EMNLP (2014)
33. Sang, D.V., Dat, N.V., Thuan, D.P.: Facial expression recognition using deep convolutional neural networks. In Proc. of KSE (2017)
34. Soleymani, M., Asghari-Esfeden, S., Fu, Y., Pantic, M.: Analysis of eeg signals and facial expressions for continuous emotion detection. IEEE Transactions on Affective Computing **7**(1), 17–28 (2016)
35. Wang, P., Liu, L., Shen, C.: Multi-attention network for one shot learning. In Proc. of CVPR (2017)
36. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent Embeddings for Zero-shot Classification. In Proc. of CVPR (2016)
37. Yang, H., Ciftci, U., Yin, L.: Facial expression recognition by de-expression residue learning. In Proc. of CVPR (2018)
38. Zhang, F., Zhang, T., Mao, Q., Xu, C.: Joint pose and expression modeling for facial expression recognition. In Proc. of CVPR (2018)
39. Zhang, Y., Gong, B., Shah, M.: Fast Zero-Shot Image Tagging. In Proc. of CVPR (2016)
40. Zhang, Y., Dong, W., Hu, B.G., Ji, Q.: Classifier learning with prior probabilities for facial action unit recognition. In Proc. of CVPR (2018)
41. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE T-PAMI **29**(6), 915–928 (2007)
42. Zhao, K., Chu, W., la Torre, F.D., Cohn, J.F., Zhang, H.: Joint patch and multi-label learning for facial action unit and holistic expression recognition. IEEE Transactions on Image Processing **25**(8), 3931–3946 (2016)
43. Zhao, K., Chu, W.S., Martinez, A.M.: Learning facial action units from web images with scalable weakly supervised clustering. In Proc. of CVPR (2018)
44. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In Proc. of ICML (2003)