This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

DeepBE: Learning Deep Binary Encoding for Multi-Label Classification

Chenghua Li
NLPR
Institute of Automation, CAS
lichenghua2014@ia.ac.cn

Qi Kang
Beijing Institute of Technology
kangqi@bit.edu.cn

Guojing Ge
NLPR
Institute of Automation, CAS
guojing.ge@nlpr.ia.ac.cn

Qiang Song
NLPR
Institute of Automation, CAS
qiang.song@nlpr.ia.ac.cn

Hanqing Lu
NLPR
Institute of Automation, CAS
luhq@nlpr.ia.ac.cn

Jian Cheng*
NLPR
Institute of Automation, CAS
jcheng@nlpr.ia.ac.cn

Abstract

The track 2 and track 3 of ChaLearn 2016 can be considered as Multi-Label Classification problems. We present a framework of learning deep binary encoding (DeepBE) to deal with multi-label problems by transforming multi-labels to single labels. The transformation of DeepBE is in a hidden pattern, which can be well addressed by deep convolutions neural networks (CNNs). Furthermore, we adopt an ensemble strategy to enhance the learning robustness. This strategy is inspired by its effectiveness in fine-grained image recognition (FGIR) problem, while most of face related tasks such as track 2 and track 3 are also FGIR problems. By DeepBE, we got 5.45% and 10.84% mean square error for track 2 and track 3 respectively. Additionally, we proposed an algorithm adaption method to treat the multiple labels of track 2 directly and got 6.84% mean square error.

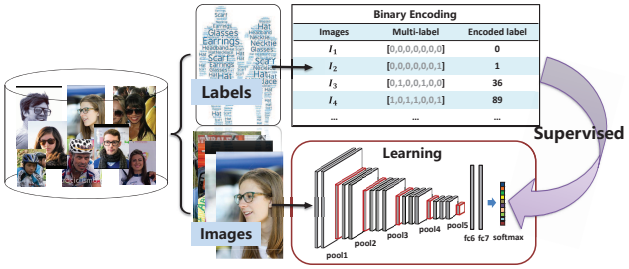

Figure 1. The framework of Learning Deep Binary Encoding for Multi-Label Classification problems. Given a training dataset with images of human upper bodies and multi-labels. The multi-label is a seven bits binary number indicates whether wearing seven corresponding accessories including earrings, glasses, hats and etc. There are two main steps: (1) Encode the multi-labels into single labels by the proposed DeepBE; (2) Learning the encoded labels with CNNs. For a test image, the learned model will output an encoded label. The final multi-label prediction can be done by a simple decoding process. See Figure 2 for track 3 testing process.

1. Introduction

Face images analysis has widely applied in our life, such as access control, identification systems, surveillance and even more attractive applications such as mood, preference analysis, pervasive computing. In pursuance of these goals, there are quantities of researches being carried out, *e.g.* face or gender recognition, age estimation, and expression analysis .

In ChaLearn 2016 [3], track 2 is for accurate accessories classification and track 3 aims at classifying gender and smile simultaneously. Both tasks have practical applications as well as research value for machines to understand human via face images. For instance, machines can help salesman determine what products should be promoted and when to promote, which will highly improve efficiency of promotions, especially in a huge exhibition with quite a lot of people. And these applications are more complicated than conventional face recognition tasks since they need more analysis on fine-grained image information and more semantic understandings.

As for track 2, there are 7 accessories including earrings, hats, glasses, necklaces, headbands, scarves, and neckties to be classified to yes or no, which indicates that whether a specific accessory is shown in an image or not. That is to say, each image has 7 labels to be predicted. Similarly, track 3 requires to predict two labels (gender and smile) for each image.

Obviously, both tasks of track 2 and track 3 belong to

*correponding author

Multi-Label Classification problems, where each instance is associated with multiple labels, *e.g*. each image has a gender label and a smile label for track 3. An intuitive method is to train multiple independent classifiers, namely Binary Relevance (BR) [20], *e.g*. training N binary classifiers for a N-label classification problem (one label one classifier). However, BR doesn't directly encode any relations between labels, while these relations are the most important ingredients of multi-label problems.

In this challenge, we proposed a Deep Binary Encoding (DeepBE) Learning framework (Figure 1) to encode inner relations between multiple labels. DeepBE can transform a multi-label to a single label in a way similar to turning a binary number to a decimal. This is a new type of Problem Transformation Method of Multi-Label Classification problems [20]. By DeepBE, track 2 and track 3 are transformed to a 2^7 and $3 \cdot 2$ single label classification problems respectively.

As face related tasks belong to Fine-Grained Image Recognition (FGIR) problems and inspired by the great progress of FGIR [9] and related applications [10] [17] in recent years, we adopt a specific FGIR learning scheme. The learning is implemented by fine-tuning four CNN models (including VGG19 [16], ResNet-152 [5], GoogleNet [18], Inception-V3 [19]) independently supervised by the DeepBE labels and then applying an ensemble of them to boost the accuracy.

On the other hand, it is obvious that the proposed DeepBE has exponential complexity, as there will be 2^N classes with N labels, that is to say it will probably suffer curse of dimensionality problem when N is large. Fortunately, both track 2 and track 3 are multi-label problems with small N. Furthermore, there is another way to deal with multi-label problems by directly handle the multiple labels [20] by specific Algorithm Adaptation Method (AAM). Specifically for track 2, an intuitive AAM to handle all accessories simultaneously is detection. We trained a faster-RCNN [14] multi-instances detector, which can detect all the seven accessories and also face simultaneously. By carefully setting a threshold of classification probability of detection, we could tell whether an accessory is shown in an image or not. By using AAM of detection, a set of spatial constraints can then be added to improve final performance.

The proposed DeepBE is a simple but an effective way to encode closely related multiple labels of an image. Moreover, it doesn't need face detection by our learning framework. On the ChaLearn 2016 provided validation set, we got 10.8% mean square error of track 3. And on a splitted validation set of track 2, we got 5.45% mean square error by DeepBE and 6.84% by detection.

2. Related works

Multi-Label Classification is very common in our life, such as a photograph can belong to sunsets and beaches at the same time in semantic scene classification. G. Tsoumakas and I. katakis [20] give a better review of Multi-Label Classification. They grouped the methods into two main categories: Problem Transformation Methods and Algorithm Adaptation Methods.

Problem Transformation Methods aims to transform the mulit-label classification problem either into one or more single-label classification or regression problems. While Algorithm Adaptation Methods tries to extend specific learning algorithms in order to handle multi-label data directly, such as Adaboost.MH and Adaboost.MR [15], ML-KNN [24] and etc.

The most common transformation method is Binary Relevance (BR). BR dealt with multi-label problems by learning a set of binary classifiers, one for each different label, such that each binary model is trained to predict a particular label. In this way, any single-label classifier can be used to suit requirements. However, BR didn't directly model correlations between labels in the training data, which would lose many useful informations for the final classification target.

Another two transformation methods are binary pairwise classification approach (PW) [4] and label combination or label power-set method (LC) [1] [12] [21]. PW trained a binary model for each pair of labels, which can result more naturally in a set of pairwise preferences. Although PW performs well in several domains, it faces quadratic complexity in terms of the number of labels and, for this reason, is usually intractable for large problems. LC transforms a multi-label problem into a single-label (multi-class) problem by treating all label sets as atomic labels, which is able to model more label correlations in the training data than PW. Obviously, LC suffers exponential computational complexity. However, most applications in computer vision such as scene classification, track 2 and track 3 are multi-label tasks with small number of labels. The complexities of these methods are listed in Table 1, where we can see that as the size of multi-label datasets grows, PW and LC are challenged by the growth in the number of possible cor-

Table 1. Worst case computation complexity for common multi-label methods. This is from J.Read *et al*. [13]. The first column is multi-label methods, including Binary Relevance (BR), pairwis (PW), Label Combination (LC). The second column is the models or classifiers needed to train of each method. The last two are the number of classes and available examples of each classifier.

method	models	lables / model	examples / model
BR	L	2	N
PW	$\frac{L(L-1)}{2}$	2	$\leq N$
LC	1	$\min(N, 2^L - 1)$	N

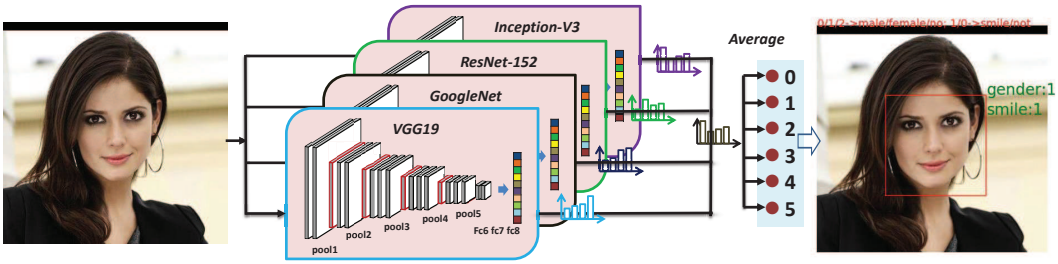

Figure 2. Deep Binary Encoding Learning of gender and smile classification. Step 1: Finetune CNNs using the provided training data and the Binary Encoding labels. Step 2: Give an image, put it into four CNNs and get the predicted softmax probabilities vector. Step 3: Apply an ensemble of four CNNs by average pool the four probability vectors. Step 4: Get the output prediction and decompose it into multi-label (see section 3.1 for detail).

relations.

Furthermore, J. Read, *et al*. [13] proposed Classifier Chains (CC) method, which can model label correlations while maintaining acceptable computational complexity.

Fine-Grained Image Recognition (FGIR) has been developed quickly recently due to the powerful feature representation ability of CNNs. For example, Lin *et al*. [11] proposed Bilinear-CNN for FGIR, which can extract specially better feature for FGIR by an outer product operation. Comparably, Spatial Transformer Network (STN) [6] added the ST block before a single CNN model to attention distinctive parts and achieves same top accuracy as Bilinear-CNN. However existing database such as CUB-200-2011 [23], Stanford Dogs [8] are all small, which is not enough for algorithms to fully learn inner variances of a specific data.

Furthermore, J. Krause [9] amazingly used noisy web data with only one single CNN model (Inception-V3 [19]) to highly boost the classification accuracy of fine-grained datasets. As some face images related tasks are highly related with FGIR, we naturally thought to apply FGIR methods into face related tasks.

3. Learning Deep Binary Encoding

The proposed framework of Learning Deep Binary Encoding is shown in Figure 1 and the specific testing process of track 3 is shown in Figure 2, which applies an ensemble of four CNN models.

3.1. Binary Encoding of Multi-labels

Let us consider x is an input image. The set $\mathcal{L} = \{1, 2, \cdots, L\}$ is the domains of possible labels. Each x is associated with a subset of these labels. This set is represented by an L-code $b = [b_1, b_2, \cdots, b_L]$ where $b_i = 0$ if and only if label i is associated with instance x, and 0 otherwise. Obviously, the L-code is just a binary number. Equivalently, we transform b into a decimal by $e = b_1 \cdot 2^{L-1} + b_2 \cdot 2^{L-2} + \cdots + b_{L-1} \cdot 2^1 + b_L \cdot 2^0$. This is a one-to-one mapping from b to e. Naturally, we take e as the binary encoding of the multi-label of x. Some examples are given in Table 2.

Table 2. Examples of Binary Encoding of multi-labels. x is the input images. b is the multi-label of x. e is the Binary Encoding of b.

x	b	e
x_1	[0, 0, 0, 0, 0, 0, 0]	0
x_2	[0, 0, 0, 0, 0, 0, 1]	1
x_3	[0, 1, 0, 0, 1, 0, 0]	36
x_4	[1, 0, 1, 1, 0, 0, 1]	89
x_5	[0, 1, 0, 0, 1, 0, 1]	37
x_6	[1, 1, 1, 1, 1, 1, 1]	127

We assume a set of training data S of N labelled images $S = \{(x^{(i)}, b^{(i)}) | i = 1, 2, \cdots, N\}$, where $b^{(i)} = [b_1^{(i)}, b_2^{(i)}, \cdots, b_L^{(i)}]$ indicates the multi-label of $x^{(i)}$.

Hence, in accessories classification (track 2), there are seven accessories of each instance with a 7-bits binary multi-label, which will be transformed to a $2^7 = 128$ classes single label classification problem. Specifically, the track 3 will be encoded to a classification problem of $3 \cdot 2 = 6$ classes due to there are three types of gender (**G:** male, female, uncertain) and two types of smile (**S:** no or yes). The encoding of the multi-label of gender and smile is $e = G \cdot 2^1 + S \cdot 2^0$, which is also an one-to-one map and results in $3 \cdot 2 = 6$ classes (see Table 6). For a test image, we first get a prediction of an encoded label, and then obtain its multi-label by decomposing the predicted label in a way similar to transfer a decimal number to a binary.

3.2. Deep Binary Encoding Learning

Traditional classification problems aim at distinguishing different objects, which have obviously different parts, for instance, a car has a wheel and a cup doesn't. Differently, FGIR aims at distinguishing sub-categories of a particular object such as different breeds of dogs [9]. FGIR

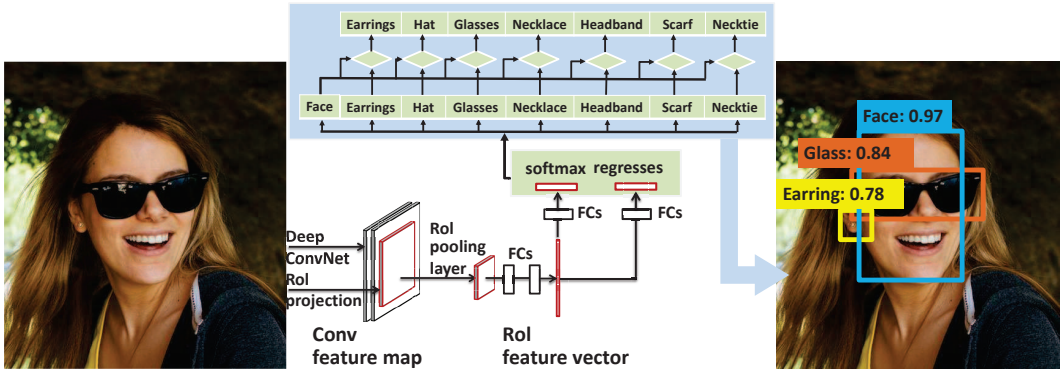

Figure 3. **Accessories Classification by Detection.** Step 1: Detect face, earrings, hat, glasses, necklace, headband, scarf, necktie in the picture by faster-RCNN model; Step 2: Give the scores and boxes of those accessories; Step 3: Choose appropriate thresholds for those accessories; Step 4: Wipe off some doubtful accessories by the threshold; Step 5: Verify the existence of accessories by the location relation between the face and the accessory.

needs more distinguishing informations, *e.g*. the hair types of dogs, the shape of beak of birds and etc. Besides, face image related tasks such as face recognition, age estimation and also the accessories, gender and smile recognition in this challenge are naturally FGIR problems. These tasks are much more difficult due to faces are very similar to each other than any other existing fine-grained datasets like birds, flowers, cars or dogs.

Thanks to recently powerful deep models (*e.g*. CNNs) and very large datasets (*e.g*. ImageNet), traditional classification tasks have reached a very high accuracy. Based on the pre-trained deep models, even much more difficult FGIR tasks have got huge progresses. Thus algorithms in FGIR must have a good directive significance to the problems in face analysis.

Motivated by Binary Encoding and FGIR, in this paper we firstly transform both multi-label problems into single label problems by the proposed DeepBE. Then four CNN models are utilized to learn the single label classification. Finally, we obtain multi-label predictions of input instances by applying an inverse process of DeepBE. DeepBE learning can encode the hidden inner relations of the multiple labels. It can also

It is obvious that there will be 2^N classes with N labels problems, and there are also some combinations of labels that rarely occur. For example, the uncertain gender has only 93 instances in track 3 (Table 7). This data unbalance may affect the whole classification accuracy of a CNN model. This is unavoidable and adding more training data or some augmentation procedure is necessary to reduce the influence.

On the other hand, Algorithm Adaptation Method (AAM) is another way to deal with multi-label problems by directly handle all labels [20]. Specifically for track 2, an intuitive method to handle all accessories simultaneously is detection. Figure 3 shows the main ideas of track 2 by detection. We trained a faster-RCNN [14] multi-instances detector, which can detect all the seven accessories simultaneously. By carefully setting a rule, we could tell whether an accessory is shown or not in the image. Furthermore, a set of spatial constraints could be added to improve final performance.

4. Experiments

We did our experiments on a machine with the following settings. System of Ubuntu-15.10, with gcc version 5.2.1, CUDA-7.5, MATLAB-2015b, and GeForce GTX TITAN Black/PCIe/SSE2. Our training experiments were applied on a CentOS-6.5 with two TITAN X GPU (12G, 12G) machine. We used the public tools including caffe [7] and mxnet [2].

As for the evaluation of the challenge, a mean square error between the prediction and the ground-truth will be computed. This error will range between 0 (gender and smile have been correctly classified) and 1 (gender and smile prediction are all wrong). Not predicted faces are evaluated with 1. We will report either the top-1 accuracy, the error or both of our experiments.

4.1. Accessories Classification

In the challenge, our submission of track 2 adopted a detection framework by using faster-RCNN, given in section 4.1.3 amply. Thus we simply give a version of DeepBE with only VGG19, as shown in section 4.1.2.

It is important to point out that the sparsity of the accessory labels may affect the performance of the detection framework. Adding more training data will be an effective way to reduce this sparsity. For example, face bounding box labels is dense, as each image has at least one face. Then the face detection achieves 98.30% average precision (Table 3)

with a specific threshold, which is much better than other accessories .

However, DeepBE can deal with this type of data efficiently, where a single VGG19 model achieved about 1.4% mean square error less than the detection framework, see Table 5).

4.1.1 Data Preprocessing

The provided training data of the accessories classification data has totally 8,477 images. There are seven accessories: earrings, hats, glasses, necklaces, headbands, scarves, and neckties, where we randomly select six images from the training set and show it in Figure 4 (a).

To get a little more training data, we split the data of track 2 into two parts: 6,476 for training and 2,001 for validation, which is almost equivalent to the provided splitting of the data in the challenge. Most of the training instances are upper body images with multi-label of seven accessories, and we didn't add any other preprocessing procedure before inputting them to CNNs.

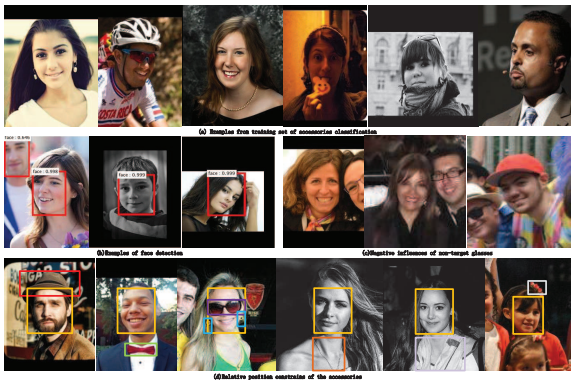

Figure 4. Examples of accessories classification images in training dataset. (a) The images in the above row is selected from the training set, which have earrings, a pair of glasses, a necklace, a headband, a scarf and a necktie from left to right. (b) Examples of face detection. (c) Negative influences of non-target glasses. (d) Relative position constrains of the accessories.

4.1.2 Learning Deep Binary Encoding for Accessories Classification

We adopted VGG19 [16], which was pre-trained on the ImageNet dataset and fine-tuned it using the splitted training images and encoded labels by DeepBE.

By DeepBE, we transform track 2 task (accessories classification) to a single label classification problem. As there are seven accessories, final encoding space have 2^7 classes. That is to say, we transform track 2 as a 2^7 classification problem. This method utilizes not only the pixel information of an image, but also the relations between different labels. For example, a rider always wears a hat and a pair of glasses; a girl always wears a necklace and earrings.

By only VGG19, we got 70.68% top-1 accuracy on the splitted validation set. After decoding all the predictions, we got multi-label predictions with 5.45% mean square error compared with ground truth by using the provided evaluation script in the challenge.

4.1.3 Detection by faster-RCNN for Accessories Classification

Besides, it's necessary to provide Bounding Boxes labels for training faster-RCNN detectors. Thus, we manually annotated all the Bounding Boxes in accordance with the provided labels of track 2 in the challenge. We also asked annotators to follow two rules. Firstly, the annotation of Bounding Box of an object should be compact. Secondly, although the track only has one target person in an image, we should mark all accessories in the pictures.

Intuitively, a faster-RCNN detector can be trained to simultaneously detect all the accessories simultaneously. By setting a proper thresholds of each accessory detection probability, we can tell the existence of the accessories in the images. Furthermore, Detection by faster-RCNN schema detects all accessories and the target face meanwhile. Thus we can fully use prior location relations between accessories and face to improve final policy decision.

In our experiments, we used VGG-16 [16] as a main model of the faster-RCNN [14]. For each image in the training dataset, the target face is defined by the provided Bounding Box, which is not the Ground Truth of face locations. The Average Precision of face detection can be evaluated by the ratio of instances of Intersection over Union (IoU) larger than a specific threshold. We used the provided face Bounding Boxes in the training, and Table 3 gives the performance of face detection by faster-RCNN. From Table 3, We can see that the proposed solution of Detection by faster-RCNN can detect face with high accuracy if we set threshold to 0.5, which also reflects that faster-RCNN could achieve better performance if the label is dense. And we also found that the face with the highest score is always the target face.

Table 3. Performance of face detection by faster-RCNN (%).

Threshold	0.7	0.6	0.5	Mean
Average Precision	86.71	95.45	**98.30**	77.47

Similar to face, we can define a set of thresholds to judge the existence of an accessory in a test image. Table 4 lists the adopted threshold of each label.

We can get an relatively accurate result through the detection. But there are also some inaccurate judgements. For example, there are three persons in the picture, the one who

Table 4. Detection Probability Thresholds of each accessory.

Accessories	Threshold
earrings	0.75
hats	0.9
glasses	0.85
headbands	0.9
necklaces	0.9
scarves	0.96
neckties	0.95

we are interested in doesn't wear glasses but the other person wears (see examples in Figure 4 (c)). Then the result will be influenced. So we make some regulations about the accessories.

Generally, hats, headbands are always on the person's head. So their centers should be on the top of the face's bounding box and the x coordinate value is between the left and the right boundary of the face. People always wear glasses on the face or on the head. So glasses' x coordinate value should be in the range of face's bounding box. In the same way, necktie, necklace should be under the face's bounding box. And the regulation is shown in Figure 4 (d). The final mean square errors of two experiments are exhibited as Table 5. We saw about 1.4% larger mean square error of the detection framework, which is due to the label sparsity and small data size.

Table 5. Mean Square Error (MSE) of Accessories Classification(%). Binary Encoding: 2^7 classes. Detection by faster-RCNN: treat seven labels directly.

	DeepBE	Detection by faster-RCNN
MSE	**5.45**	6.84

Finally, we can see the following insights. Firstly, we deal with accessaries classification problem by detection. For detection, we use faster-RCNN to detect some closely related parts of a face, which is different from detecting multiple objects. Secondly, all the accessories were detected simultaneously, which is a new Algorithm Adaption Method for the specific Multi-label Accessories Classification problem. Thirdly, we fully take advantages of prior space relations of all the accessories with face to improve performance. Finally, due to face detection is part of the task 2, a face detector can be efficiently extracted from it. This detector is based on faster-RCNN, which is more faster than the provided face detector PO-CR [22] in the challenge. Thus, it can also efficiently deal with a beforehand face detection if needed even for the tasks in track 3 (see section 4.2 for detail).

Table 6. Binary Encoding of Gender and Smile (%). Gender: male (0), female (1) and uncertain (2). Smile: no (0) or yes (1). GS: Tranformed classes of G and S by $GS = 2G + S$.

G	0	0	1	1	2	2
S	0	1	0	1	0	1
GS	**0**	**1**	**2**	**3**	**4**	**5**

Table 7. Number of training Data of Gender and Smile (%). Gender label has three types: male, female, and uncertain. Smile label has two types: no and yes.

Gender			Smile	
male	female	uncertain	no	yes
3,318	2,670	93	3,937	2,234

Figure 5. Examples of gender and smile images in training dataset. Two images of each class (see Table 6 for detail). 0: male+no (smile). 1: male+yes(smile). 2: female+no; 3: female+yes; 4: uncertain+no; 5: uncertain+yes.

4.2. Track 3: Gender and smile Classification

4.2.1 Data Processing

The provided data of gender and smile is totally 9,258 images with training 6,171 and validation 3,086. The gender label has three types: male, female and uncertain. The smile label has two types: no and yes. The number of each type is given in Table 7. There are only a small part of images with uncertain gender, which includes baby faces, clown face or some ambiguous faces. By binary encoding, we have six classes shown in Table 6. Accordingly, we select two examples of each class from the training set which are shown in Figure 5.

Specially, the provided data are almost clean images with a target upper body lying in the center. We directly use the original data to train our deep model and didn't add another preprocessing operations except the inner preprocessing operations required by each CNN model. Thus our final model will properly fit any images with the same properties with the training data. That is to say, it is necessary to add a face detection procedure and crop a big enough sub-image if given a test image with a full body. Furthermore, it is obvious that including upper body not only face is important

for gender classification in this task. As to smile, only the face is enough.

4.2.2 Results and Analysis

By using Binary Encoding, we transformed the Multi-Label Classification problems (the smile and gender classification and the accessories classification) to a common classification, which can be addressed efficiently using CNNs with a softmax classifier.

As we all know, ensemble is an effective way to improve CNN classification accuracy. Thus we carefully selected four CNN models which had very good performance in FGIR firstly. We trained 6-classes Gender-Smile (GS) classification model using VGG19 [16], ResNet-152 [5], GoogleNet [18], Inception-V3 [19]. Then we combine all the four CNNs to boost the accuracy. The results are given in Table 8.

Table 8. Top-1 accuracy of each single model and their ensemble on the provided validation set with the encoding six classes classification (%).

models	top-1 accuracy
VGG19	78,26
ResNet-152	76.07
GoogleNet	74.98
Inception-V3	77.96
DeepBE	**80.29**

From Table 8, we finally got top-1 classification accuracy 80.29% on the validation set, whose corresponding gender and smile classification accuracy are 90.44% and 88.43% respectively. And by the evaluation given in the challenge, we got a final 10.8% error.

Furthermore, we also trained a 2-classes Smile classification model on the cropped faces using the bounding boxes provided in the training data. By combining VGG19, ResNet-152, GoogleNet, we have the top-1 accuracy 89.14% on the validation set (better than the above 88.43%), which indicates that face detection is helpful to smile classification. However, face detection is needed, which is time consuming and the performance is closely depends on the detection accuracy. So we didn't use it in the final evaluation. (We leave Detect Face Smile Classification as an optional choice in the programme).

The Power of Binary Encoding To verify the power of binary encoding, we compared it with the Binary Relevance. Binary Relevance is the most intuitive method of multi-label problems, which tries to train N separately classifiers for N-label problem. The results are shown in Table 9, where DeepBE is better than BR in both classification accuracy.

Table 9. The Power of Binary Encoding of VGG19 (%). BR (Binary Relevance, see section 1 for details): training 2 classifiers, Gender: 0, 1, 2, Smile: 0, 1. BE (Binary Encoding, see section 3.1 for detail.): 0, 1, 2, 3, 4, 5.

top-1	Gender	Smile
BR	87.65	85.67
DeepBE	**88.39**	**87.20**

However, the Binary Encoding is done in a hidden way by CNN. We didn't know what relations between gender and smile can boost both accuracy. This is indeed annoying, but also attracting to us to a much deeper study.

5. Conclusion

We proposed Learning Deep Binary Encoding (DeepBE) to deal with Multi-Label Classification problems. By DeepBE, we can transform a multi-label to a single-label in a hidden pattern, and apply an ensemble of some powerful CNNs to learn it. Both track 2 and track 3 in ChaLearn 2016 are Multi-Label Classification problems, which can be efficiently solved by the proposed DeepBE Learning. Our experiments also showed its efficiency. Additionally, we also proposed to solve the accessories classification by detection. This was done by training a detector using faster-RCNN to simultaneously detect all the accessories and face. We can also envisage a future where many multi-label problems in the computer vision field can be solved by DeepBE learning and ideas like classification by detection.

6. Acknowledgement

This work was supported in part by National Natural Science Foundation of China (Grant No. 61332016), and 863 program (Grant No. 2014AA015105).

References

[1] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[2] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

[3] S. Escalera, M. Torres, B. Martnez, X. Bar, H. J. Escalante, I. Guyon, G. Tzimiropoulos, C. Corneanu, M. Oliu, M. Ali Bagheri, and M. Valstar. Chalearn looking at people and faces of theworld: Face analysis workshop and challenge 2016, chalearn looking at people and faces of the world. In *CVPR workshops*, 2016.

[4] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[6] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2008–2016, 2015.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[8] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[9] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. *arXiv preprint arXiv:1511.06789*, 2015.

[10] C. Li, Q. Song, Y. Wang, H. Song, Q. Kang, J. Cheng, and H. Lu. Learning to recognition from bing clickture data. *MSR Image Recognition Challenge (IRC) @ IEEE ICME*, 2016.

[11] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[12] J. Read, B. Pfahringer, and G. Holmes. Multi-label classification using ensembles of pruned sets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 995–1000. IEEE, 2008.

[13] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

[14] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[15] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2):135–168, 2000.

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[17] Q. Song, S. Yu, C. Leng, J. Wu, Q. Hu, and J. Cheng. Learning deep features for msr-bing information retrieval challenge. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, pages 169–172. ACM, 2015.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[20] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*, 2006.

[21] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Machine learning: ECML 2007*, pages 406–417. Springer, 2007.

[22] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3659–3667. IEEE, 2015.

[23] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[24] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.