Multi-task learning for smile detection, emotion recognition and gender classification

Conference Paper · December 2017

DOI: 10.1145/3155133.3155207

Dinh Viet Sang
Hanoi University of Science and Technology

Multi-task learning for smile detection, emotion recognition and gender classification

Dinh Viet Sang
Hanoi University of Science and Technology
Hanoi, Vietnam
sangdv@soict.hust.edu.vn

Le Tran Bao Cuong
Hanoi University of Science and Technology
Hanoi, Vietnam
ltbclqd2805@gmail.com

Vu Van Thieu
Hanoi University of Science and Technology
Hanoi, Vietnam
thieuvv@soict.hust.edu.vn

ABSTRACT

Facial expression analysis plays a key role in analyzing emotions and human behaviors. Smile detection, emotion recognition and gender classification are special tasks in facial expression analysis with various potential applications. In this paper, we propose an effective architecture of Convolutional Neural Network (CNN) which can jointly learn representations for three tasks: smile detection, emotion recognition and gender classification. In addition, this model can be trained from multiple sources of data with different kinds of task-specific class labels. The extensive experiments show that our model achieves superior accuracy over recent state-of-the-art techniques in all of three tasks on popular benchmarks. We also show that the joint learning helps the tasks with less data considerably benefit from other tasks with richer data.

CCS CONCEPTS

• **Computing methodologies** → **Object recognition**; **Multi-task learning**; **Neural networks**; *Batch learning*;

KEYWORDS

Multi-task Learning, Multi-source Learning, Convolutional Neural Network, Image Classification.

ACM Reference Format:
Dinh Viet Sang, Le Tran Bao Cuong, and Vu Van Thieu. 2017. Multi-task learning for smile detection, emotion recognition and gender classification. In *SoICT '17: Eighth International Symposium on Information and Communication Technology, December 7–8, 2017, Nha Trang City, Viet Nam.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3155133.3155207

1 INTRODUCTION

In recent years, artificial intelligence (AI) has been exploding thanks to breakthroughs in the field of machine learning and data science. A wide range of AI products have helped improve labor productivity, improve the quality of human life, and save human and social resources. Many artificial intelligence applications in computer vision, voice recognition, or natural language processing have reached or even surpassed human levels in some cases.

In this work, we study different human facial analysis tasks including smile detection, emotion recognition and gender recognition. All of three tasks use facial images as input. In smile detection task, we must detect if the people in a given image are smiling or not. We then classify their emotions into seven classes: angry, disgust, fear, happy, sad, surprise and neutral in emotion recognition task. Finally, we indicate who are males and who are females among them in gender classification task.

These tasks have been studied for a long time using both conventional and modern approaches. Nevertheless, in previous works, these tasks are often solved as separate problems. This may lead to many difficulties in training models, especially, when the training data is not so much. In general, the data of different facial analysis tasks often shares many common characteristics of human faces. Therefore, joint learning from multiple sources of face data can boost the performance of each individual task.

In this paper, we present an effective architecture of CNN to simultaneously learn common features for smile detection, emotion recognition and gender classification. Each task takes input data from its corresponding source, but all the tasks share a big part of the network with many hidden layers. At the end of the network, these tasks are separated into three branches with different task-specific losses. We combine all the losses to form a common network loss, which allows us to train the network end-to-end via the backprop algorithm.

The main contributions of this paper are as follows:

(1) We propose an effective multi-task CNN architecture that performs smile detection, emotion recognition and gender classification simultaneously.
(2) We conduct extensive experiments and achieve new state-of-the-art accuracies in different tasks on popular benchmarks.

The rest of the paper is organized as follows. In section 2, we review related work. In section 3, we present our proposed multi-task network and describe how to train the model from multiple data sources. Finally, in section 4, we show the experimental results on popular datasets and compare the proposed model with recent state-of-the-art methods.

SoICT '17, December 7–8, 2017, Nha Trang City, Viet Nam
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5328-1/17/12. . . $15.00
https://doi.org/10.1145/3155133.3155207

2 RELATED WORK

2.1 Deep convolutional neural networks

In recent years, deep learning has proved its superior power in many fields, especially, in computer vision. Along with Recurrent Neural Networks (RNNs), deep CNNs are thought to be the brightest stars in the deep network family. The first well-known CNN models can be mentioned are LeNet [20], and AlexNet [19] - the winner of ImageNet ILSVRC challenge in 2012.

Some latest CNNs such as VGG [31], Inception [33], ResNet [13] and DenseNet [15] tend to be deeper and deeper. Meanwhile, some other CNN architectures like WideResNet [39] or ResNeXt [38] tend to be wider. All of these CNNs have demonstrated their impressive performances in one of the most prestigious competitions in computer vision - the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

2.2 Smile detection

Conventional approaches often extract low-level face descriptors and detect smile based on a strong binary classifier. Shan et al. [30] propose a simple method that uses the intensity differences between pixels in the gray-scale face images and then combines them with AdaBoost classifier [37] for smile detection. In order to represent face images, Liu et al. [22] use histograms of oriented gradients (HOG) [10], meanwhile, An et al. [3] use local binary pattern (LBP) [2], local phase quantization (LPQ) [24] and HOG. Both of them [3, 22] then apply SVM classifier [9] to detect smiles. Jain et al. [17] exploit Multi-scale Gaussian Derivatives (MGD) combined with SVM classifier.

Some recent works focus on applying deep learning approach to smile detection. Chen at al. [6] use deep CNN to extract high-level features from face images and then use SVM or AdaBoost classifiers to detect smiles as a classification task. Zhang et al. [40] introduce two efficient CNN models called CNN-Basic and CNN 2-Loss. The CNN-2Loss is a improved variant of the CNN-Basic, that tries to learn features by using two supervisory signals. The first one is recognition signal that is responsible for the classification task. The second one is expression verification signal, which is effective to reduce the variation of features which are extracted from the images of the same expression class.

In [28], we propose an effective VGG-like network, called BKNet, to detect smiles. BKNet outperforms recent state-of-the-art techniques.

2.3 Emotion recognition

Classical approaches for facial expression recognition are often based on Facial Action Coding System (FACS) [11], which involves identifying various facial muscles causing changes in facial appearance. It includes a list of Action Units (AUs). Cootes et al. [36] propose a model based on an approach called the Active Appearance Model [8]. Given input image, preprocessing steps are performed to create over 500 facial landmarks. From these landmarks, the authors perform PCA algorithm and derive Action Units (AUs). Finally, they classify facial expressions using a single layered neural network.

In facial expression recognition competition in Kaggle [1], the winning team [34] propose an effective CNN model that sets the state-of-the-art on the Kaggle FERC-2013 dataset. In [34], the authors use multi-class SVM loss instead of usual cross-entropy loss, and exploit augmentation techniques to create more training data. In [29], we also successfully apply our BKNet architecture to emotion recognition problem and achieve better accuracy then previous methods.

2.4 Gender classification

Conventional methods for gender classification often use image intensities as input features. In [25], the authors combine the 3D structure of the head with image intensities. He et al. [14] only use image intensities combined with SVM classifier. [4] tries to use AdaBoost instead of SVM classifier. [12] introduces a neural network trained on a small set of face images. [35] uses the Webers Local texture Descriptor [7] for gender classification. More recently, Levi et al. [21] present an effective CNN architecture that yields good results in recognizing gender.

2.5 Multi-task learning

The concept of Multi-task learning was first mentioned in [5]. Multi-task learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. Recently, Kaiser et al. [18] propose a big model to learn simultaneously many tasks in nature language processing and computer vision and achieve very promising results. Rothe et al. [27] proposed a multi-task learning model to jointly learn age and gender classification from images. Ranjan et al. [26] introduce a multi-task learning framework, so-called hyperface, for face detection, landmark localization, pose estimation, and gender recognition. Nevertheless, the hyperface is only trained from a unique source of data with full annotations for all tasks.

3 OUR PROPOSED METHOD

3.1 Overall architecture

Our proposed network takes input from multiple data sources. After merging these datasets, we input them into a block called *CNN Shared Block* which learns joint representations for all tasks from all the sources of data. This block can be any arbitrary known CNN architecture such as VGG [31], Resnet [13], DenseNet [15]. After the shared block, the network is separated into three branches associated with three different tasks. Each branch then learns task-specific features and has its own loss corresponding to each task. Fig. 1 describes the pipeline of our network.

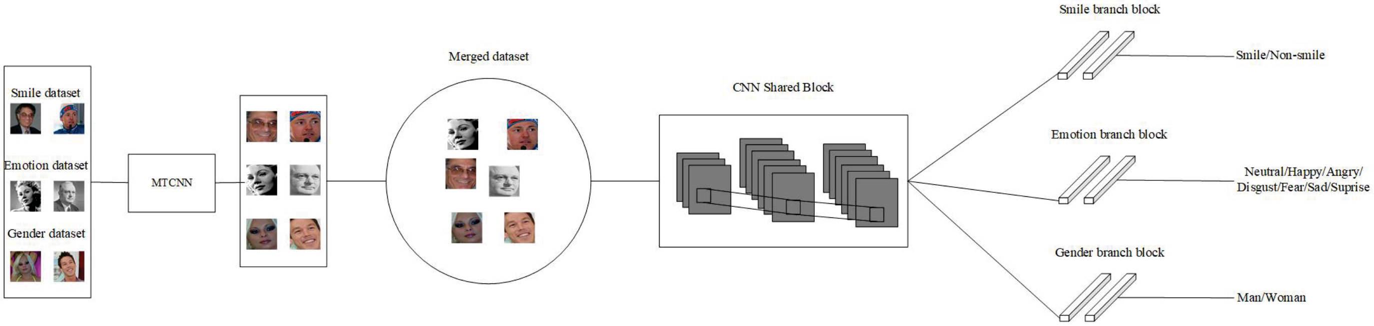


Figure 1: Our proposed network pipeline.

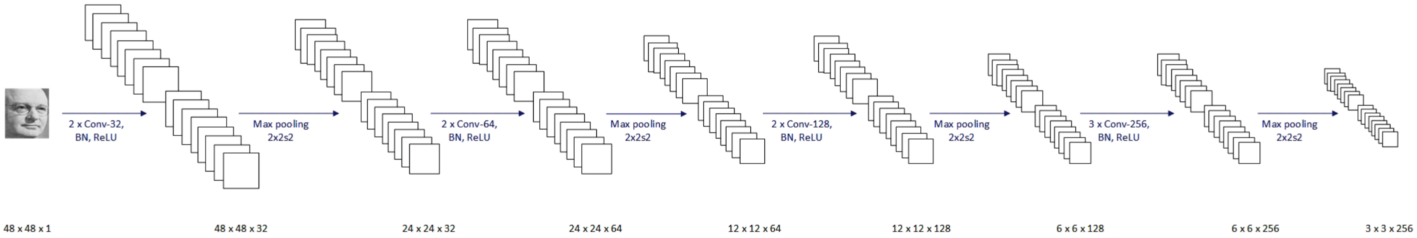


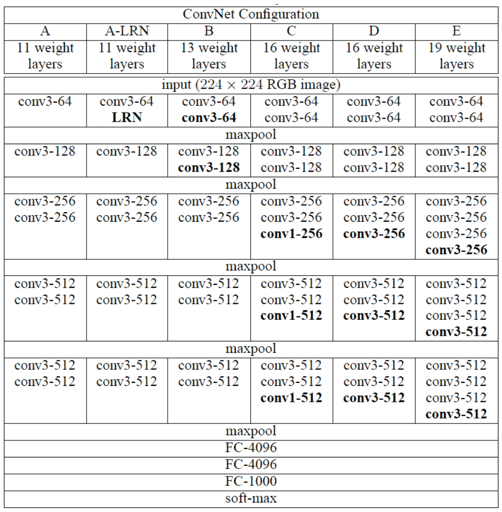
Figure 2: CNN Shared Block for multi-tasks in our proposed network.

3.2 CNN Shared Block

The motivation of the CNN Shared Block is to help the network learn the shared features from multiple datasets. It is generally believed that the features learned in this block can generalize better and make more accurate predictions than a model for a single task. Moreover, thanks to joint learning, the tasks with less data can largely benefit from other tasks with more data.

We can design this block based on some famous architectures such as VGG, ResNet, Google Inception or DenseNet. In this paper, we use our previous BKNet architecture [28, 29], which is a VGG-like network (Fig. 4), to design the CNN shared block by eliminating three last fully-connected layers of BKNet.

Our proposed architecture for *CNN Shared Block* is illustrated in Fig. 2. In this block we use four convolutional blocks. The first convolutional (conv) block includes two conv layers with 32 neurons 3 × 3 with the stride of 1, followed by a max pooling layer 2 × 2 with the stride of 2. The second conv block includes two conv layers with 64 neurons 3 × 3 with the stride of 1, followed by a max pooling layer 2 × 2 and the stride of 2. The third conv block includes two conv layers with 128 neurons 3 × 3 with the stride of 1, followed by a max pooling layer 2 × 2 and the stride of 2. Finally, the last conv block includes three conv layers with 256 neurons 3 × 3 with the stride of 1, followed by a max pooling layer with kernel size of 2 × 2 and the stride of 2. Each conv layer is followed by a Batch normalization layer [16] and a ReLU (Rectified Linear Unit) activation function [23]. The Batch normalization layer is applied to reduce the internal covariant shift. It allows us to use higher learning rate and, therefore, accelerates the training process.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3: VGG architecture.

3.3 Branch Block

After *CNN Shared Block*, we split our network into three branches corresponding to separate tasks, *i.e.*, smile detection, emotion recognition and gender classification. While *CNN Shared Block* can learn joint representations across three tasks from multiple datasets, each branch tries to learn individual features corresponding to each specific task.

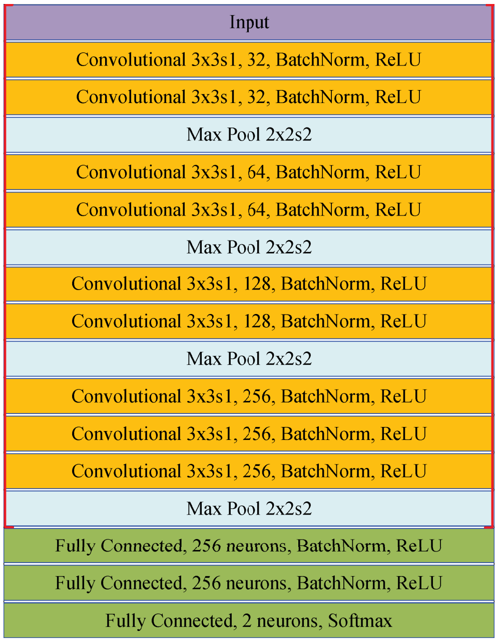

Input
Convolutional 3x3s1, 32, BatchNorm, ReLU
Convolutional 3x3s1, 32, BatchNorm, ReLU
Max Pool 2x2s2
Convolutional 3x3s1, 64, BatchNorm, ReLU
Convolutional 3x3s1, 64, BatchNorm, ReLU
Max Pool 2x2s2
Convolutional 3x3s1, 128, BatchNorm, ReLU
Convolutional 3x3s1, 128, BatchNorm, ReLU
Max Pool 2x2s2
Convolutional 3x3s1, 256, BatchNorm, ReLU
Convolutional 3x3s1, 256, BatchNorm, ReLU
Convolutional 3x3s1, 256, BatchNorm, ReLU
Max Pool 2x2s2
Fully Connected, 256 neurons, BatchNorm, ReLU
Fully Connected, 256 neurons, BatchNorm, ReLU
Fully Connected, 2 neurons, Softmax

Figure 4: The CNN shared block is just the top part (marked by red lines) of our previous BKNet architecture [28], excluding the last three fully-connected layers.

Each branch consists of two fully connected layers with 256 neurons and a final fully connected layer with n neurons, where C is the number of classes in each task ($C = 2$ for smile detection and gender classification branch, $C = 7$ for emotion recognition branch). Note that, after the last fully connected layer, we can either use an additional softmax layer as a classifier or not, depending on what kind of loss function is used. These kinds of loss function are described in detail in the next section.

Similar with *CNN Shared Block*, each fully connected layer in all branches (except the last one) is followed by a Batch Normalization layer and ReLU. Dropout is also used [32] for all fully connected layers to reduce overfitting.

3.4 Multi-source Multi-task training

In this paper, we propose a deep network that can learn to perform multi tasks from different data sources. All data sources are mixed together and form a large common training set. It should be emphasized that in the mixed training set, generally, each sample is only related to some of the tasks.

Suppose that:

- T is the number of tasks ($T = 3$ in this paper);
- L_t is the individual loss corresponding to the t^{th} task $t = 1, 2, ..., T$.
- N is the number of samples from all training datasets;
- C_t is the number of classes corresponding to the t^{th} task ($C_1 = C_3 = 2$ for smile detection and gender classification task, $C_2 = 7$ for emotion recognition task);
- $\mathbf{s}_i^t$ is the vector of class scores corresponding to i-th sample in t^{th} task;
- l_i^t is the correct class label of i-th sample in t^{th} task;
- $\mathbf{y}_i^t$ is the one-hot encoding of the correct class label of i-th sample in t^{th} task ($y_i^t(l_i^t) = 1$);
- $\widehat{\mathbf{y}}_i^t$ is the probability distribution over the classes of i-th sample in t^{th} task, which can be obtained by applying the softmax function to $\mathbf{s}_i^t$.
- $\alpha_i^t \in \{0, 1\}$ is the sample type indicator ($\alpha_i^t = 1$ if the i^{th} sample is related to the t^{th} task, $\alpha_i^t = 0$ in the other case).

Note that, if the i^{th} sample is not related to t^{th} task, then the true label does not exist, and we can ignore l_i^t and $\mathbf{y}_i^t$. To ensure the mathematical correctness in this case, we can set them to arbitrary values, for instance, $l_i^t = 0$ and $\mathbf{y}_i^t$ is a zero vector.

In this paper, we try two kinds of loss: soft-max cross entropy or multi-class SVM loss.

Cross-entropy loss requires to use a softmax layer after the last fully-connected layer of each branch. Cross-entropy loss L_t corresponding to t^{th} task is defined as follows:

$$L_t = -\frac{1}{N}\sum_{i=1}^{N}\left(\alpha_i^t \sum_{j=1}^{C} \mathbf{y}_i^t(j) log(\widehat{\mathbf{y}}_i^t(j))\right), \qquad (1)$$

where $\mathbf{y}_i^t(j) \in \{0, 1\}$ indicates whether j is the correct label of i-th sample; $\widehat{\mathbf{y}}_i^t(j) \in [0, 1]$ expresses the probability that j is the correct label of i-th sample.

The multi-class SVM loss function is used when the last fully connected layer in each task-specific branch do not use any activation function. Multi-class SVM loss function corresponding to t^{th} task can be defined as follows:

$$L_t = \frac{1}{N}\sum_{i=1}^{N}\left(\alpha_i^t \sum_{j \neq l_i} max(0, \mathbf{s}_i^t(j) - \mathbf{s}_i^t(l_i^t) + 1)^2\right), \qquad (2)$$

where $\mathbf{s}_i^t(j)$ indicates the score of class j in the i-th sample; $\mathbf{s}_i^t(l_i^t)$ defines the score of true label l_i in the i-th sample.

The total loss of the network is computed as the weighted sum of the three individual losses. In addition, we also add L2 weight decay term associated with all network weights $\mathbf{W}$ to the total loss to reduce overfitting. The overall loss can be defined as follows:

$$L_{total} = \sum_{1}^{T} \mu_t L_t + \lambda \|\mathbf{W}\|_2^2, \qquad (3)$$

where μ_t is the importance level of the t^{th} task in the overall loss; λ is the weight decay coefficient.

We train the network end-to-end via the standard back propagation algorithm.

3.5 Data pre-processing

All the images from the datasets that we use later are portraits. Nevertheless, our network only works with faces. That is reason why we do data pre-processing to crop faces from the original images in the datasets. Here we use Multi-task Cascaded Convolutional Networks (MTCNN) [41] to detect faces in each image. Fig. 5 shows some examples of using MTCNN for cropping faces.

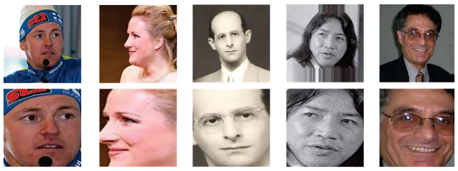


Figure 5: MTCNN for face detection. The top row is original images. The bottom row are cropped faces using MTCNN.

After that, the cropped images are converted to grayscale and resized to 48 × 48 ones.

3.6 Data augmentation

Due to small amount of samples in the dataset, we use data augmentation techniques to generate more new data for the training phase. These techniques help us to reduce overfitting and, hence, to train a more robust network.

We used 3 main methods for data augmentation as follows:

- Randomly crop: We add margins to each image in the datasets and then crop a random area of that image with the same size as the original image;
- Randomly flip an image from left to right.
- Randomly rotate an image by a random angle from -15° to 15°

In practice, we find that applying augmentation techniques greatly improves the performance of the model.

4 EXPERIMENTS AND EVALUATION

4.1 Datasets

4.1.1 GENKI-4K dataset.

GENKI-4K is a well-known dataset used in smile detection task. This dataset includes 4000 labelled images of human face from different ages, and races. Among these pictures, 2162 images were labeled as smile and 1838 images were labeled as non-smile. The images in this dataset are taken from the internet with different real-world contexts (unlike other face data sets, often taken in the same scene), which makes the detection more challenging. However, some images in the dataset are unclear (not sure whether smile or not). In some previous works, some unclear images are eliminated during the training and testing phases. It is obviously that keeping wrong samples in the dataset intuitively makes the model more likely to be confused during the training phase. In the testing phase, the wrong samples might considerably reduce the overall accuracy, when the model makes true predictions but the data says no. Despite that fact, in this work we still retain all the images in the original dataset in both phases. Fig. 6 shows some examples from GENKI-4K dataset.

Figure 6: Some samples in the GENKI-4K dataset. The top two rows are examples of smile faces and the bottom two rows are examples of non-smile faces.

4.1.2 FERC-2013 dataset.

FERC-2013 dataset is provided on the Kaggle facial expression competition. The dataset consists of 35,887 gray images of 48x48 resolution. Kaggle has divided into 28,709 training images, 3589 public test images and 3589 private test images. Each image contains a human face that is not posed (in the wild). Each image is labeled by one of seven emotions: angry, disgust, fear, happy, sad, surprise and neutral. Some images of the FERC-2013 dataset are showed in Fig. 7 shows some examples from FERC-2013 dataset.

Figure 7: Some samples in the FERC-2013 dataset.

4.1.3 IMDB and Wiki dataset.

In this work, we use IMDB and Wiki datasets as data sources for gender classification task.

The IMDB dataset is a large face dataset that includes data from celebrities. The authors take the list of the most popular 100,000 actors as listed on the IMDB website and (automatically) crawl from their profiles date of birth, name, gender and all images related to that person. The IMDB dataset contains about 470.000 images. In this paper, we only use 170.000 images from IMBD.

The Wiki dataset also includes data from celebrities, which are crawled data from Wikipedia. The Wiki dataset contains about 62.000 images and in this work we will use about 34.000 images from this dataset.

Figure 8: Some samples in the IMDB and Wiki datasets.

Fig. 8 shows some samples from IMDB and Wiki datasets.

4.2 Implementation detail

We will discuss about our experiment setup to jointly learn three tasks: smile detection, emotion recognition and gender classification. We use GENKI-4K dataset for smile detection, FERC-2013 for emotion recognition. We separately use one of the two IMDB and Wiki datasets for gender classification task.

Our experiments is conducted using Python programing-language on computers with the following specifications: Intel Xeon E5-2650 v2 Eight-Core Processor 2.6GHz 8.0GT/s 20MB, Ubuntu Operating System 14.04 64 bit, 32GB RAM.

Preparing data set: Firstly, we merge three datasets (GENKI-4K, FERC-2013, gender dataset IMDB/Wiki) to make a large dataset. We then create a marker vector to define sample type indicators α_i^t. We always keep the number of training data for each task equally to help the learning process stability. For example, if we train our model with two dataset: dataset A with 3000 samples, dataset B with 30000 samples, we will duplicate dataset A 10 times to make a big dataset with total 60000 samples.

In our work, we divide each dataset into training set and testing set. With GENKI-4K dataset, we use 3000 samples for training and 1000 samples for testing. With FERC-2013 dataset we use data distribution similar with competition in Kaggle. With Wiki dataset, we use 30000 samples for training and about 4200 samples for testing. With IMDB dataset, we use 150000 samples for training and about 20000 samples for testing.

Training phase: Our model is trained end-to-end by using SGD algorithm with momentum 0.9. We set the batch size equal to 128. We initialize all weights using a Gaussian distribution with zero mean and standard deviation 0.01. The L2 weight decay is $\lambda = 0.01$. All the tasks have the same importance level $\mu_1 = \mu_2 = \mu_3 = 1$. The dropout rate for all fully connected layers is set to 0.5. Moreover, we apply an exponential decay function to decay the learning rate through time. The learning rate at step k is calculated as follows:

$$curLr = initLr * decayRate^{m/decayStep}, \qquad (4)$$

where $curLr$ is the learning rate at step m; $initLr$ is the initialization learning rate at the beginning of training phase; $decayStep$ is the number of steps when the learning rate decayed.

In our experiment, we set $initLr = 0.01$, $decayRate = 0.8$ and $decayStep = 10000$.

Testing phase: In the testing phase, our model is evaluated by k-fold cross-validation algorithm. This method splits our original data into k parts of the same size. The model evaluation is performed through loops, each loop selects $k-1$ parts of data as training data and the rest is used for testing model. For the convenience of doing comparison between different methods, we use 4-fold cross-validation algorithm as previous works. We will report the average accuracy and the standard deviation after 4 iterations. Moreover, we test our model with two different loss functions mentioned above.

Furthermore, we ensemble different checkpoints obtained during the training phases to infer for our model. In the paper, we keep 10 last checkpoints corresponding to 10 last training epochs for inference.

4.3 Experimental results

In this work, we set up two experiment cases. The first one, we train our model with GENKI-4K, FERC-2013 and Wiki dataset. The second one, we train our model with GENKI-4K, FERC-2013 and IMDB dataset. Table 1 shows our experiment setup.

We report our result and compare with previous methods in Table 2. As we can see, using cross-entropy loss function gives better result than using SVM loss function in all cases.

In smile detection task, the best accuracy we archive is **96.23 ± 0.58 %** when we train our model with GENKI-4K, FERC-2013 and IMDB dataset. Moreover, in all experiment cases, we get a better result than all previous work.

In emotion recognition task, the best accuracy we archive is **71.03 ± 0.11 %** for public test and **72.18 ± 0.23 %** for private test. This result also considerably outperforms all of previous methods.

In gender classification task, to the best of our knowledge, there are no works reported result on the Wiki and IMDB datasets for gender classification. The best accuracy we get on Wiki is **96.33 ± 0.16 %** when we train our model on Wiki. The best accuracy we get on IMDB is **92.20 ± 0.11 %** when we train our model on IMDB. We also report the IMDB 's test accuracy when we train our model on Wiki, and the Wiki's test accuracy when we train our model on IMDB.

5 CONCLUSION

In this paper, we propose an effective multi-souce multi-task CNN architecture to jointly learn three facial analysis tasks including smile detection, emotion recognition and gender classification. The extensive experiments in well-known GENKI-4K, FERC-2013, Wiki, IMDB datasets show that our method achieve superior accuracy over recent state-of-the-art methods in all tasks. We also show that the smile detection

Table 2: Results on four datasets with different methods

Method	GENKI-4K	FERC-2013		Wiki	IMDB
		Public test	Private test		
CNN Basic [40]	93.6 ± 0.47%	-	-	-	-
CNN 2-Loss [40]	94.6 ± 0.29%	-	-	-	-
BKNet + Softmax [28]	95.08 ± 0.29%	-	-	-	-
CNN+SVM Loss (*team RBM*) [34]	-	69.4%	71.2%	-	-
BKNet + SVM loss [29]	-	71.0%	71.9%	-	-
Our method (Config A1)	95.25 ± 0.44%	68.60 ± 0.27%	69.28 ± 0.41%	95.25 ± 0.15%	88.18 ± 0.26%
Our method (Config A2)	95.13 ± 0.20%	69.12 ± 0.18%	69.40 ± 0.22%	95.75 ± 0.18%	88.68 ± 0.15%
Our method (Config A3)	95.52 ± 0.37%	70.63 ± 0.11%	71.78 ± 0.08%	95.95 ± 0.15%	88.83 ± 0.18%
Our method (Config A4)	95.70 ± 0.25%	**71.03** ± **0.11 %**	**72.18** ± **0.23 %**	**96.33** ± **0.16 %**	89.34 ± 0.15%
Our method (Config B1)	95.25 ± 0.43%	68.10 ± 0.14%	69.10 ± 0.57%	93.33 ± 0.19%	89.60 ± 0.22%
Our method (Config B2)	95.56 ± 0.66%	68.47 ± 0.33%	69.40 ± 0.21%	93.67 ± 0.26%	90.50 ± 0.24%
Our method (Config B3)	95.60 ± 0.41%	70.43 ± 0.19%	71.90 ± 0.36%	93.70 ± 0.37%	91.33 ± 0.42%
Our method (Config B4)	**96.23** ± **0.58 %**	70.15 ± 0.19%	71.62 ± 0.39%	94.00 ± 0.24%	**92.20** ± **0.11 %**

Table 1: Experiment setup

Name	Datasets	Loss function	Use ensemble?
Config A1	GENKI-4K, FERC-2013, Wiki	SVM loss	No
Config A2	GENKI-4K, FERC-2013, Wiki	Cross-entropy loss	No
Config A3	GENKI-4K, FERC-2013, Wiki	SVM loss	Yes
Config A4	GENKI-4K, FERC-2013, Wiki	Cross-entropy loss	Yes
Config B1	GENKI-4K, FERC-2013, IMDB	SVM loss	No
Config B2	GENKI-4K, FERC-2013, IMDB	Cross-entropy loss	No
Config B3	GENKI-4K, FERC-2013, IMDB	SVM loss	Yes
Config B4	GENKI-4K, FERC-2013, IMDB	Cross-entropy loss	Yes

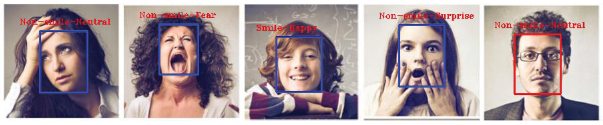


Figure 10: Some results of our multi-task learning framework. The blue box corresponds to females and the red box corresponds to males.

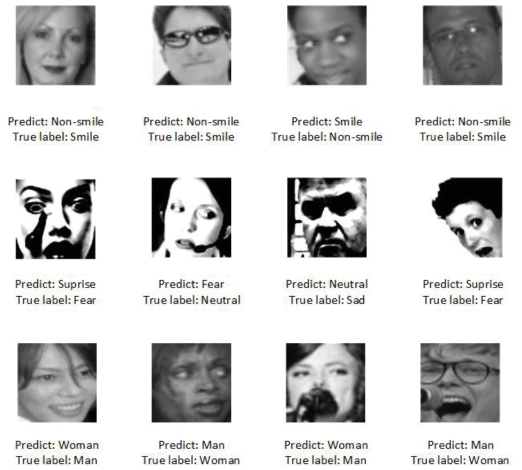


Figure 9: Some samples that our model makes wrong predictions.

task with few data largely benefit from the two other tasks with richer data.

In the future, we would like to exploit some recent effective designs such as shortcut connections in ResNet, Google inception blocks or DenseNet to build deeper and more powerful networks.

6 ACKNOWLEDGMENTS

This research is funded by Ministry of Education and Training, Vietnam, under grant number B2015-01-90.

REFERENCES

[1] 2013. Challenges in Respresentation Learning: Facial Expression Recognition Challenge. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge. (2013).

[2] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. 2004. Face recognition with local binary patterns. *Computer vision-eccv 2004* (2004), 469–481.

[3] Le An, Songfan Yang, and Bir Bhanu. 2015. Efficient smile detection by extreme learning machine. *Neurocomputing* 149 (2015), 354–363.

[4] Shumeet Baluja, Henry A Rowley, et al. 2007. Boosting sex identification performance. *International Journal of computer vision* 71, 1 (2007), 111–119.

[5] Rich Caruana. 1998. Multitask learning. In *Learning to learn*. Springer, 95–133.

[6] Junkai Chen, Qihao Ou, Zheru Chi, and Hong Fu. 2017. Smile detection in the wild with deep convolutional neural networks. *Machine vision and applications* 28, 1-2 (2017), 173–183.

[7] Jie Chen, Shiguang Shan, Chu He, Guoying Zhao, Matti Pietikainen, Xilin Chen, and Wen Gao. 2010. WLD: A robust local image descriptor. *IEEE transactions on pattern analysis and machine intelligence* 32, 9 (2010), 1705–1720.

[8] Timothy F Cootes, Cristopher J Taylor, et al. 2004. Statistical models of appearance for computer vision. (2004).

[9] Corinna Cortes and Vladimir Vapnik. 1995. Support vector machine. *Machine learning* 20, 3 (1995), 273–297.

[10] Oscar Déniz, Gloria Bueno, Jesús Salido, and Fernando De la Torre. 2011. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters* 32, 12 (2011), 1598–1603.

[11] Paul Ekman and Erika L Rosenberg. 1997. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA.

[12] Beatrice A Golomb, David T Lawrence, and Terrence J Sejnowski. 1990. SEXNET: A Neural Network Identifies Sex From Human Faces.. In *NIPS*, Vol. 1. 2.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[14] Xiaofei He and Partha Niyogi. 2004. Locality preserving projections. In *Advances in neural information processing systems*. 153–160.

[15] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2016. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993* (2016).

[16] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*. 448–456.

[17] Varun Jain and James L Crowley. 2013. Smile detection using multi-scale gaussian derivatives. In *12th WSEAS International Conference on Signal Processing, Robotics and Automation*.

[18] Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One Model To Learn Them All. *arXiv preprint arXiv:1706.05137* (2017).

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[21] Gil Levi and Tal Hassner. 2015. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 34–42.

[22] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen. 2012. Enhancing expression recognition in the wild with unlabeled reference data. In *Asian Conference on Computer Vision*. Springer, 577–588.

[23] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[24] Ville Ojansivu and Janne Heikkilä. 2008. Blur insensitive texture classification using local phase quantization. In *International conference on image and signal processing*. Springer, 236–243.

[25] Alice J O'toole, Thomas Vetter, Nikolaus F Troje, and Heinrich H Bülthoff. 1997. Sex classification is better with three-dimensional head structure than with image intensity information. *Perception* 26, 1 (1997), 75–84.

[26] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. 2016. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249* (2016).

[27] Rasmus Rothe, Radu Timofte, and Luc Van Gool. 2015. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 10–15.

[28] Dinh Viet Sang, Le Tran Bao Cuong, and Do Phan Thuan. 2017. Facial Smile Detection Using Convolutional Neural Networks. In *The 9th International Conference on Knowledge and Systems Engineering (KSE 2017)*. 138–143.

[29] Dinh Viet Sang, Nguyen Van Dat, and Do Phan Thuan. 2017. Facial Expression Recognition Using Deep Convolutional Neural Networks. In *The 9th International Conference on Knowledge and Systems Engineering (KSE 2017)*. 144–149.

[30] Caifeng Shan. 2012. Smile detection by boosting pixel differences. *IEEE transactions on image processing* 21, 1 (2012), 431–436.

[31] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[32] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15, 1 (2014), 1929–1958.

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[34] Yichuan Tang. 2013. Deep learning using support vector machines. *CoRR, abs/1306.0239* 2 (2013).

[35] Ihsan Ullah, Muhammad Hussain, Ghulam Muhammad, Hatim Aboalsamh, George Bebis, and Anwar M Mirza. 2012. Gender recognition from face images with local wld descriptor. In *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*. IEEE, 417–420.

[36] Hans Van Kuilenburg, MA Wiering, and Marten Den Uyl. 2005. A model based method for automatic facial expression recognition. In *Proceedings of the 16th European Conference on Machine Learning (ECML'05)*. Springer, 194–205.

[37] Paul Viola and Michael Jones. 2002. Fast and robust classification using asymmetric adaboost and a detector cascade. In *Advances in neural information processing systems*. 1311–1318.

[38] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2016. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431* (2016).

[39] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).

[40] Kaihao Zhang, Yongzhen Huang, Hong Wu, and Liang Wang. 2015. Facial smile detection based on deep learning features. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*. IEEE, 534–538.

[41] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503.