Effective Deep Multi-source Multi-task Learning Frameworks for Smile Detection, Emotion Recognition and Gender Classification

Dinh Viet Sang and Le Tran Bao Cuong
Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam
E-mail: sangdv@soict.hust.edu.vn, ltbclqd2805@gmail.com

Keywords: multi-task learning, convolutional neural network, smile detection, emotion recognition, gender classification

Received: March 29, 2018

Automatic human facial recognition has been an active reasearch topic with various potential applications. In this paper, we propose effective multi-task deep learning frameworks which can jointly learn representations for three tasks: smile detection, emotion recognition and gender classification. In addition, our frameworks can be learned from multiple sources of data with different kinds of task-specific class labels. The extensive experiments show that our frameworks achieve superior accuracy over recent state-of-the-art methods 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.

Povzetek: Razvita je izvirna metoda globokih nevronskih mrež za tri hkratne naloge: prepoznavanje smeha, čustev in spola.

1 Introduction

In recent years, we have witnessed a rapid boom of artificial intelligence (AI) in various fields such as computer vision, speech recognition and natural language processing. A wide range of AI products have boosted labor productivity, improved the quality of human life, and saved human and social resources. Many artificial intelligence applications have reached or even surpassed human levels in some cases.

Automatic human facial recognition has become an active research area that plays a key role in analyzing emotions and human behaviors. In this work, we study different human facial recognition tasks including smile detection, emotion recognition and gender recognition. All of three tasks use facial images as input. In smile detection task, we have to detect if the people appearing 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 determine who are males and who are females in gender classification task.

In general, these tasks are often solved as separate problems. This may lead to many difficulties in learning models, especially, when the training data is not large enough. On the other hand, 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 introduce effective deep convolutional neural networks (CNNs) 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 networks with many hidden layers. At the end of each network, these tasks are separated into three branches with different task-specific losses. We combine all the losses to form a common network objective function, which allows us to train the networks end-to-end via the back propagation algorithm.

The main contributions of this paper are as follows:

1. We propose effective architectures of CNNs that can learn joint representations from different sources of data to simultaneously perform smile detection, emotion recognition and gender classification.

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 briefly review related work. In section 3, we present our proposed multi-task deep learning frameworks and describe how to train the networks from multiple data sources. Finally, in section 4, we show the experimental results on popular datasets and compare our proposed frameworks with recent state-of-the-art methods.

2 Related work

2.1 Deep convolutional neural networks

In recent years, deep learning has been proven to be effective in many fields, and particularly, in computer vision. Deep CNNs are one of the most popular models in the family of deep neural networks. LeNet [21], and AlexNet [20] are known to be the earliest CNN architectures with not many hidden layers.

Latest CNNs such as VGG [33], Inception [35], ResNet [13] and DenseNet [16] tend to be deeper and deeper. In ResNet, residual blocks can be stacked on top of each other with over 1000 layers. Meanwhile, some other CNN architectures like WideResNet [41] or ResNeXt [40] tend to be wider. All these effective CNNs have demonstrated their impressive performances in one of the biggest and the most prestigious competitions in computer vision - the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

2.2 Smile detection

Traditional methods often detect smile based on a strong binary classifier with low-level face descriptors. Shan et al. [32] propose a simple method that uses the intensity differences between pixels in the gray-scale facial images and then combines them with AdaBoost classifier [39] for smile detection. In order to represent faces, Liu et al. [23] use histograms of oriented gradients (HOG) [10], meanwhile, An et al. [4] use local binary pattern (LBP) [3], local phase quantization (LPQ) [25] and HOG. Both of them [23, 4] then apply SVM classifier [9] to detect smiles. Jain et al. [18] propose to use Multi-scale Gaussian Derivatives (MGD) and SVM classifier as well for smile detection.

Some recent methods focus on applying deep neural networks to smile detection. Chen at al. [6] use deep CNNs to extract high-level features from facial images and then use SVM or AdaBoost classifiers to detect smiles as a classification task. Zhang et al. [42] 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. [30] proposes an effective VGG-like network, called BKNet, to detect smiles. BKNet achieves better results than many other state-of-the-art methods in smile detection.

2.3 Emotion recognition

Classical approaches to facial expression recognition are often based on Facial Action Coding System (FACS) [11]. FACS includes a list of Action Units (AUs) that describe various facial muscle movements causing changes in facial appearance. Cootes et al. [38] propose a model based on an approach called the Active Appearance Model [8] that creates over 500 facial landmarks. Next, the authors apply PCA algorithm to the set of landmarks and derive Action Units (AUs). Finally, a single layered neural network is used to classify facial expressions.

In Kaggle facial expression recognition competition [1], the winning team [36] proposes an effective CNN, which uses the multi-class SVM loss instead of the usual cross-entropy loss. In [31], Sang et al. propose the so-called BKNet architecture for emotion recognition and achieve better performance compared to previous methods.

2.4 Gender classification

Conventional methods for gender classification often take image intensities as input features. [26] combines the 3D structure of the head with image intensities. [15] uses image intensities combined with SVM classifier. [5] tries to use AdaBoost instead of SVM classifier. [12] introduces a neural network trained on a small set of facial images. [37] uses the Webers Local texture Descriptor [7] for gender classification. More recently, Levi et al. [22] present an effective CNN architecture that yields fairly good performance in gender classification.

2.5 Multi-task learning

Multi-task learning aims to solve multiple classification tasks at the same time by learning them jointly, while exploiting the commonalities and differences across the tasks. Recently, Kaiser et al. [19] propose a big model to learn simultaneously many tasks in nature language processing and computer vision and achieve promising results. Rothe et al. [28] propose a multi-task learning model to jointly learn age and gender classification from images. Zhang et al. [2] propose a cascaded architecture with three stages of carefully designed deep convolutional networks to jointly detect faces and predict landmark locations. Ranjan et al. [27] introduce a multi-task learning framework 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 frameworks

3.1 Overall architecture

In this work, we propose effective deep CNNs that can learn joint representations from multiple data sources to solve different tasks at the same time. The merged dataset (Fig. 1) is fed into a block called “CNN Shared Network", which can be designed by using an arbitrary CNN architecture such as VGG [33], ResNet [13] and so on. The motivation of the CNN Shared Network is to help the networks learn the shared features from multiple datasets across different tasks. It is thought that the features learned in the shared block can generalize better and make more accurate predictions than a single-task model. Moreover, thanks to joint representation learning, the tasks with less data can largely benefit from other tasks with more data.

After the shared block, each network is separated into three branches associated with three different tasks. Each branch learns task-specific features and has its own loss function corresponding to each task.

3.2 Multi-task BKNet

Our first multi-task deep learning framework called Multi-task BKNet has been previously described in [29] (Fig. 3), which is based on the BKNet architecture [30, 31]. We construct the CNN shared network by eliminating three last fully-connected layers of BKNet (Fig. 2).

CNN Shared Network. In this part, we use four convolutional (conv) blocks. The first conv block includes two conv layers with 32 neurons 3×3 with the stride 1, followed by a max pooling layer 2×2 with the stride 2. The second conv block includes two conv layers with 64 neurons 3×3 with the stride 1, followed by a max pooling layer 2×2 with the stride 2. The third conv block includes two conv layers with 128 neurons 3×3 with the stride 1, followed by a max pooling layer 2×2 with the stride 2. Finally, the last conv block includes three conv layers with 256 neurons 3×3 with the stride 1, followed by a max pooling layer 2×2 with the stride 2. Each conv layer is followed by a Batch normalization layer [17] and a ReLU (Rectified Linear Unit) activation function [24]. The Batch normalization layer reduces the internal covariant shift, and, hence, allows us to use higher learning rate when applying the SGD algorithm to accelerate the training process.

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

Each branch consists of two fully connected layers with 256 neurons and a final fully connected layer with C neurons, where C is the number of classes in each task ($C = 2$ for smile detection and gender classification branch, and $C = 7$ 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 being used. These kinds of loss function are described in detail in the next section. Similar with the CNN shared network, each fully connected layer in all branches (except the last one) is followed by a Batch Normalization layer and ReLU. Dropout [34] is also utilized in all fully connected layers to reduce overfitting.

3.3 Multi-task ResNet

ResNet [13] is known as one of the most efficient CNN architectures so far. In order to enhance the information flow between layers, ResNet uses shortcut connections between layers. The original variant of ResNet is proposed by He et al. in [13] with different numbers of hidden layers: ResNet-18, ResNet-34 or ResNet-50, ResNet-101 and ResNet-152. He et al. then introduce an improved variant of ResNet (called ResNet_v2) in [14] which shows that the pre-activation order “conv - batch normalization - ReLU" is consistently better then post-activation order “batch normalization - ReLU - conv".

Inspire by the design concept of ResNet_v2, we propose a multi-task ResNet framework to jointly learn three tasks: smile detection, emotion recognition and gender classification. Since the amount of facial data is not large, we choose ResNet-50 (with bottleneck layer) as the base architecture to design our multi-task ResNet framework. In the original ResNet_v2-50 architecture, there are 4 residual blocks, each of which consists of some sub-sampling blocks and identity blocks. The architectures of identity blocks and sub-sampling blocks are shown in Fig. 4a and Fig. 4b. For both these two kinds of blocks, we use the bottleneck architecture with *base depth* m that consists of three conv layers: a 1×1 conv layer with m filters followed by a 3×3 conv layer with m filters and a 1×1 conv layers with $4m$ filters. The identity blocks and sub-sampling blocks are distinguished by the stride value in the second conv layer and the shortcut connection. In sub-sampling blocks, we use a conv layer with stride 2 instead of stride 1 as in identity blocks. The first residual block of ResNet-50 contains only 3 identity blocks and has no sub-sampling block. The next three residual blocks of ResNet-50 have a sub-sampling block at the top, followed by 3, 5 and 2 identity blocks, respectively.

Based on the aforementioned ResNet_v2-50 architecture, we propose two versions of multi-task ResNet framework. In the first version, which is abbreviated as Multi-task ResNet ver1, we use all of 4 residual blocks to build the CNN shared network to learn joint representations for three tasks. Like in multi-task BKNet, for each task in branch network, we use two fully connected layers with 256 neurons combined with a softmax classifier. Fig. 5a illustrates the architecture of Multi-task ResNet ver1.

In the second version, which is abbreviated as Multi-task ResNet ver2, we only use first three residual blocks to build the CNN shared network. For each task in the branch network, we use a separate residual block combined with global average pooling layer and a softmax classifier. Fig. 5b illustrates the architecture of Multi-task ResNet ver2.

3.4 Multi-source multi-task training

In this paper, we propose effective deep networks that can learn to perform multi tasks from different data sources. All data sources are mixed together and form a large common training set (Fig. 1). Generally, each sample in the mixing training set 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, $t = 1, 2, ..., T$.

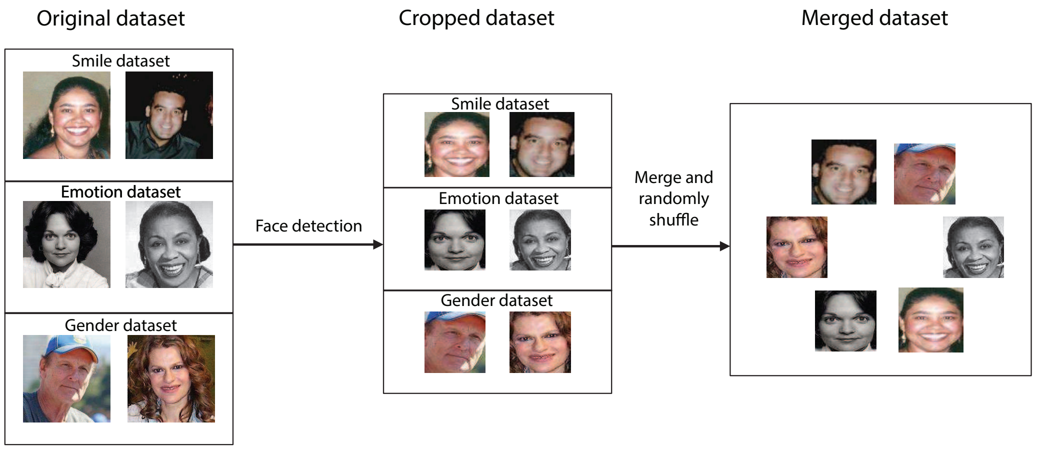
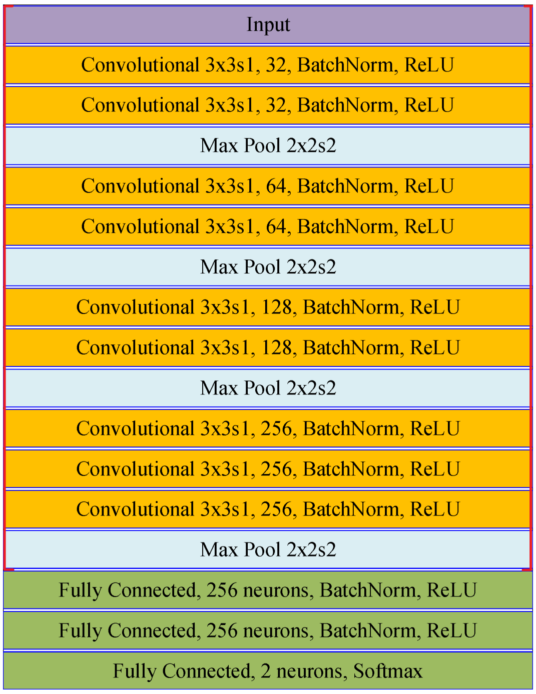


Figure 1: Merged dataset



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 2: The CNN shared network in Multi-task BKNet is just the top part (marked by red lines) of the BKNet architecture [30], excluding the last three fully-connected layers.

- 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 ($y_i^t(l_i^t) = 1$);

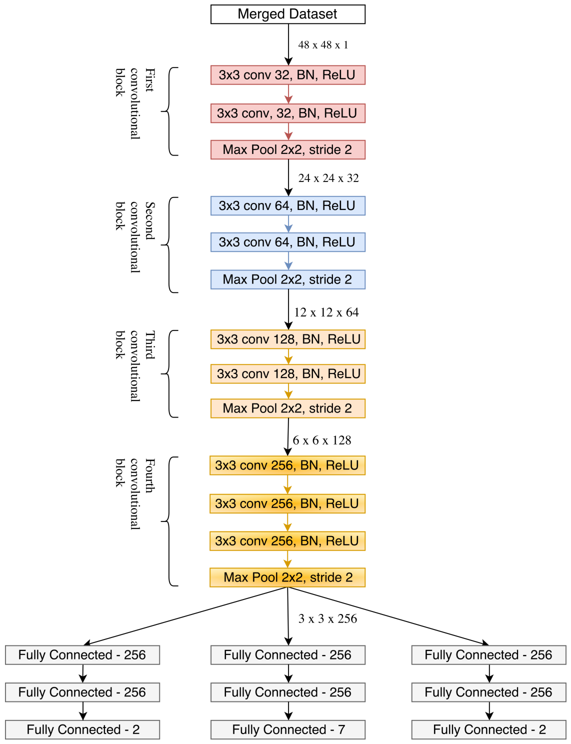
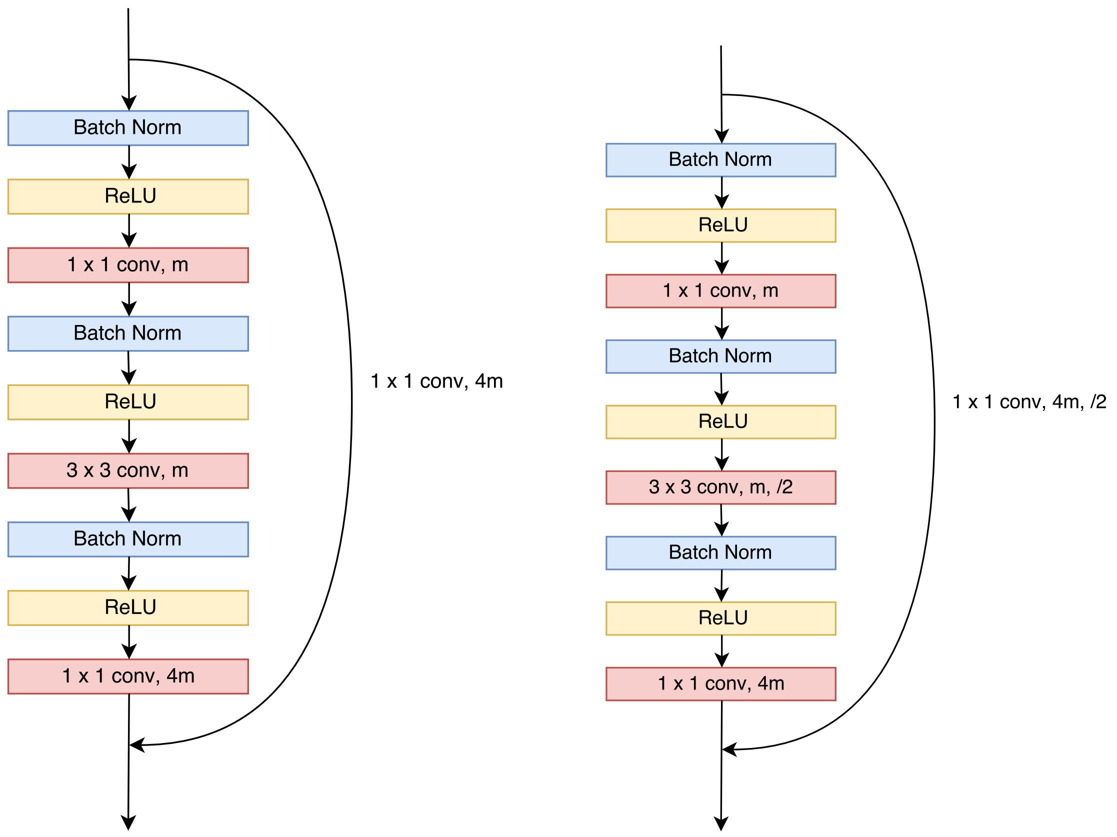


Figure 3: Our proposed Multi-task BKNet

- $\widehat{\mathbf{y}}_i^t$ 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, and $\alpha_i^t = 0$ otherwise).

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



(a) Identity block

(b) Subsampling block

Figure 4: The architectures of identity blocks and sub-sampling blocks in our Multi-task ResNet framework.

zero vector.

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

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

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

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 accompanies with no activation function. The multi-class SVM loss function corresponding to the t^{th} task can be defined as follows:

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

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^t 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 network loss to reduce overfitting. The overall loss can be defined as follows:

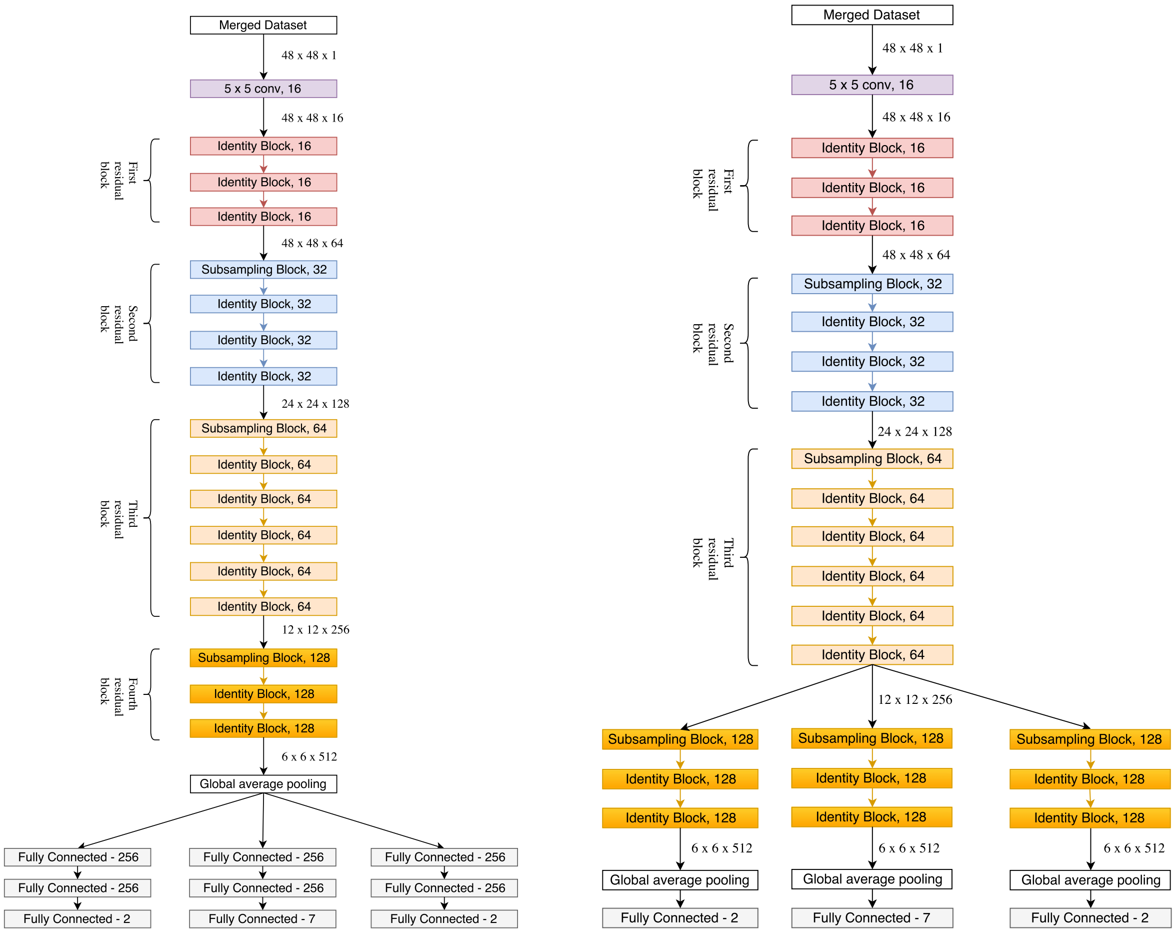
$$L_{total} = \sum_{1}^{T} \mu_t L_t + \lambda \|\mathbf{W}\|_2^2, \qquad (3)$$

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 networks works with facial regions only. Thus, we have to perform data pre-processing to crop faces from the original images in the datasets. Here we use Multi-task Cascaded Convolutional Neural Networks (MTCNN) [2] to detect faces in each image. Fig. 6 shows some examples of using MTCNN for cropping faces.



(a) First version with fully connected layers in the branch network

(b) Second version with residual blocks in the branch network

Figure 5: Our proposed Multi-task ResNet framework. The notation *"Identity block, m"* means the identity block with base depth m.

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

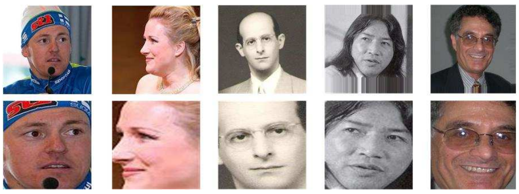


Figure 6: MTCNN for face detection. The top row is original images. The bottom row are cropped faces using MTCNN.

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 learn more robust networks.

We used three following popular ways for data augmentation:

- 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 $15°$. The space around the rotated image is then filled with black color.

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 datasets, 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. 7 shows some examples from GENKI-4K dataset.

Figure 7: 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. 8.

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. Fig. 9 shows some samples from IMDB and Wiki datasets.

Figure 8: Some samples in the FERC-2013 dataset.

Figure 9: Some samples in the IMDB and Wiki datasets.

4.2 Implementation detail

In the experiments, 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 are 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, GPU NVIDIA TITAN X 12GB.

Preparing data: 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 datasets: 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 split as provided by 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: With Multi-task BKNet architecture, 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 $decayStep = 10000$. We train our Multi-task BKNet model in 250 epochs.

Similar to Multi-task BKNet, we train our Multi-task ResNet end-to-end by using SGD algorithm with momentum 0.9. We set the batch size equal to 128. We initialize all weights using variance scaling initializer (He initializer). The L2 weight decay is 10^{-4}. All the tasks have the same important level $\mu_1 = \mu_2 = \mu_3 = 1$. We train the Multi-task ResNet ver1 in 100 epochs and train the Multi-task ResNet ver2 in 80 epochs. The initial learning rate is 0.05 and then decreased by 10 times whenever the training loss stops improving.

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 combine different checkpoints obtained during the training phases to infer test samples. In the paper, we keep 10 last checkpoints corresponding to 10 last training epochs for inference.

4.3 Experimental results

4.3.1 Multi-task BKNet

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

We report our results 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 achieve is $96.23 \pm 0.58\%$ when we train our model with GENKI-4K, FERC-2013 and IMDB dataset. In all experiment cases, we achieve better results than previous state-of-the-art methods. Especially, the Multi-task BKNet clearly outperforms the single-task BKNet [30]. This fact proves that the smile detection task largely benefits from other tasks thanks to sharing the commonalities between data.

In emotion recognition task, the best accuracy we achieve is $71.03 \pm 0.11\%$ for public test and $72.18 \pm 0.23\%$ for private test. This result considerably outperforms all of previous methods.

In gender classification task, to the best of our knowledge, there are no previous results on the Wiki and IMDB datasets for gender classification. In this paper, we apply the single-task BKNet model [30] and achieve the accuracy $95.82 \pm 0.44\%$ and $91.17 \pm 0.27\%$ on the Wiki and IMDB datasets, respectively. The best accuracy we get on Wiki is $96.33 \pm 0.16\%$ when we train our Multi-task BKNet model on Wiki. The best accuracy we get on IMDB is $92.20 \pm 0.11\%$ when we train our model on IMDB. We also report the test accuracy on IMDB when we train the model on Wiki, and the test accuracy on Wiki when we train the model on IMDB.

In all tasks, the Multi-task BKNet yields comparative results and even better than the single-task BKNet in many cases. Furthermore, it should be emphasized that the Multi-task network can effectively solve all the three tasks by using only a common network instead of three separate ones, which would requires approximately three times more memory storage and computational complexity.

4.3.2 Multi-task ResNet

Based on the experimental results of Multi-task BKNet, we will choose the best config B4 in Table 1 to evaluate our

Table 1: Experiment setup

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

Table 2: Accuracy comparison on four datasets

Method	GENKI-4K	FERC-2013		Wiki	IMDB
		Public test	Private test		
Chen et al [6]	91.8 ± 0.95	-	-	-	-
CNN Basic [42]	93.6 ± 0.47	-	-	-	-
CNN 2-Loss [42]	94.6 ± 0.29	-	-	-	-
Single-task BKNet + Softmax [30]	95.08 ± 0.29	-	-	95.82 ± 0.44*	91.16 ± 0.27*
CNN (team Maxim Milakov - rank 3 Kaggle)	-	68.2	68.8	-	-
CNN (team Unsupervised - rank 2 Kaggle)	-	69.1	69.3	-	-
CNN+SVM Loss (*team RBM*) [36]	-	69.4	71.2	-	-
Single-task BKNet + SVM loss [31]	-	71.0	71.9	-	-
Our Multi-task BKNet (Config A1)	95.25 ± 0.43	68.10 ± 0.14	69.10 ± 0.57	93.33 ± 0.19	89.60 ± 0.22
Our Multi-task BKNet (Config A2)	95.56 ± 0.66	68.47 ± 0.33	69.40 ± 0.21	93.67 ± 0.26	90.50 ± 0.24
Our Multi-task BKNet (Config A3)	95.60 ± 0.41	70.43 ± 0.19	71.90 ± 0.36	93.70 ± 0.37	91.33 ± 0.42
Our Multi-task BKNet (Config A4)	$\mathbf{96.23 \pm 0.58}$	70.15 ± 0.19	71.62 ± 0.39	94.00 ± 0.24	$\mathbf{92.20 \pm 0.11}$
Our Multi-task BKNet (Config B1)	95.25 ± 0.44	68.60 ± 0.27	69.28 ± 0.41	95.25 ± 0.15	88.18 ± 0.26
Our Multi-task BKNet (Config B2)	95.13 ± 0.20	69.12 ± 0.18	69.40 ± 0.22	95.75 ± 0.18	88.68 ± 0.15
Our Multi-task BKNet (Config B3)	95.52 ± 0.37	70.63 ± 0.11	71.78 ± 0.08	95.95 ± 0.15	88.83 ± 0.18
Our Multi-task BKNet (Config B4)	95.70 ± 0.25	$\mathbf{71.03 \pm 0.11}$	$\mathbf{72.18 \pm 0.23}$	$\mathbf{96.33 \pm 0.16}$	89.34 ± 0.15
Our Multi-task ResNet ver1 (Config B4)	95.55 ± 0.28	70.09 ± 0.13	71.55 ± 0.19	96.03 ± 0.22	89.01 ± 0.18
Our Multi-task ResNet ver2 (Config B4)	95.30 ± 0.34	69.33 ± 0.31	71.27 ± 0.11	95.99 ± 0.14	88.88 ± 0.07

Multi-task ResNet frameworks.

The results of our Multi-task ResNet are also shown in Table 2. As one can see, our first version yields better results than the second version in all three tasks.

In smile detection task, the first version of multi-task ResNet achieves $95.55 \pm 0.28\%$ accuracy, while the second version achieves $95.30 \pm 0.34\%$ accuracy. With the same config B4, our Multi-task BKNet model achieves $95.70 \pm 0.25\%$ accuracy, which is slightly better then Multi-task ResNet.

In emotion recognition task, the accuracy of the first version of Multi-task ResNet is $70.09 \pm 0.13\%$ for public test set and $71.55 \pm 0.19\%$ for private test set. The accuracy of the second version is a little bit lower with $69.33 \pm 0.31\%$ and $71.27 \pm 0.11\%$ for public test set and private test set, respectively. In this task, both versions of Multi-task ResNet seem to clearly lose Multi-task BKNet, which obtains higher approximately 1% accuracy in each test set.

In gender classification task, both our variants of multi-task ResNet yield pretty good results, which compete with the results of of the multi-task BKNet model. The first variant achieves the accuracy of $96.03 \pm 0.22\%$ and $89.01 \pm 0.18\%$ for Wiki dataset and IMDB dataset, respectively. The second variant achieves the accuracy of $95.99 \pm 0.14\%$ for Wiki dataset and $88.88 \pm 0.07\%$ for IMDB dataset.

The experiment results show that the Multi-task ResNet is slightly worse than the Multi-task BKNet in all tasks. The reason could be due to that ResNet with a pretty deep architecture and fairly large number of parameters tends to be over-complex w.r.t the mixing training data across the three tasks and leads to overfitting. Meanwhile, BKNet is quite smaller than ResNet, and is capable to fit the data

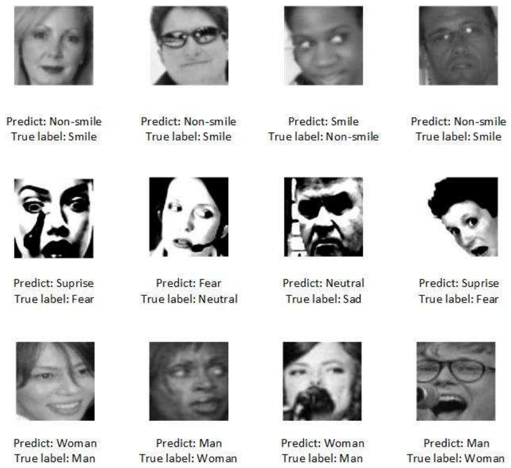


Figure 10: Some samples that our Multi-task BKNet gives wrong predictions.

better.

4.3.3 Speed performance comparison between different frameworks

In Table 3 and Table 4, we show the inference time and training time of three frameworks: Multi-task BKNet, Multi-task ResNet ver1 and Multi-task ResNet ver2 with Config B4 (from Table 1).

As one can see, the Multi-task ResNet ver2 acquires the fastest convergence. Despite a little longer in training time, Multi-task BKNet is significantly faster in inference in comparison with both versions of Multi-task ResNet. The fast inference with high accuracy make the Multi-task BKNet well suitable for real-time applications.

Table 3: Comparison of inference time between different frameworks

Framework	Inference time per image (sec)
Multi-task BKNet	0.02
Multi-task ResNet ver1	0.065
Multi-task ResNet ver2	0.071

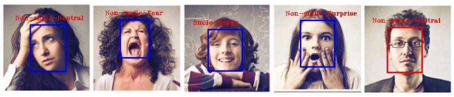


Figure 11: Some results of our Multi-task BKNet framework. The blue box corresponds to females and the red box corresponds to males.

5 Conclusion

In this paper, we propose effective multi-souce multi-task deep learning frameworks 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 frameworks achieve superior accuracy over recent state-of-the-art methods in all tasks. We also show that the smile detection task with few data largely benefit from the two other tasks with richer data.

In the future, we would like to exploit some new auxiliary losses to regulate the model learning process in order to improve the performance accuracy of neural networks in various computer vision tasks.

6 Acknowledgments

This research is funded by Hanoi University of Science and Technology under grant number T2016-LN-08.

References

[1] Challenges in respresentation learning: Facial expression recognition challenge, 2013.

[2] Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. `https://doi.org/10.1109/lsp.2016.2603342`.

[3] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. *Computer vision-eccv 2004*, pages 469–481, 2004.

[4] L. An, S. Yang, and B. Bhanu. Efficient smile detection by extreme learning machine. *Neurocomputing*, 149:354–363, 2015. `https://doi.org/10.1016/j.neucom.2014.04.072`.

[5] S. Baluja, H. A. Rowley, et al. Boosting sex identification performance. *International Journal of computer vision*, 71(1):111–119, 2007. `https://doi.org/10.1007/s11263-006-8910-9`.

[6] J. Chen, Q. Ou, Z. Chi, and H. Fu. Smile detection in the wild with deep convolutional neural networks. *Machine vision and applications*, 28(1-2):173–183, 2017. `https://doi.org/10.1007/s00138-016-0817-z`.

[7] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao. Wld: A robust local image descriptor. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1705–1720, 2010. `https://doi.org/10.1109/tpami.2009.155`.

Table 4: Comparison of training time between different frameworks

Framework	Number of epochs	Training time per epoch (min)	Total training time (min)
Multi-task BKNet	250	3.42	854
Multi-task ResNet ver1	100	8.12	817
Multi-task ResNet ver2	80	8.67	693

[8] T. F. Cootes, C. J. Taylor, et al. Statistical models of appearance for computer vision, 2004.

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

[10] O. Déniz, G. Bueno, J. Salido, and F. De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603, 2011. https://doi.org/10.1016/j.patrec.2011.01.004.

[11] P. Ekman and E. L. Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997. https://doi.org/10.1093/acprof:oso/9780195179644.001.0001.

[12] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In *NIPS*, volume 1, page 2, 1990.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. https://doi.org/10.1109/cvpr.2016.90.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016. https://doi.org/10.1007/978-3-319-46493-0_38.

[15] X. He and P. Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017. https://doi.org/10.1109/cvpr.2017.243.

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

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

[19] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. https://doi.org/10.1109/5.726791.

[22] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–42, 2015. https://doi.org/10.1109/cvprw.2015.7301352.

[23] M. Liu, S. Li, S. Shan, and X. Chen. Enhancing expression recognition in the wild with unlabeled reference data. In *Asian Conference on Computer Vision*, pages 577–588. Springer, 2012. https://doi.org/10.1007/978-3-642-37444-9_45.

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

[25] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. In *International conference on image and signal processing*, pages 236–243. Springer, 2008. https://doi.org/10.1007/978-3-540-69905-7_27.

[26] A. J. O'toole, T. Vetter, N. F. Troje, and H. H. Bülthoff. Sex classification is better with three-dimensional head structure than with image intensity information. *Perception*, 26(1):75–84, 1997. https://doi.org/10.1068/p260075.

[27] R. Ranjan, V. M. Patel, and R. Chellappa. Hyper-Face: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1,

2017. https://doi.org/10.1109/tpami.2017.2781233.

[28] R. Rothe, R. Timofte, and L. Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–15, 2015. https://doi.org/10.1109/iccvw.2015.41.

[29] D. V. Sang, L. T. B. Cuong, and V. V. Thieu. Multi-task learning for smile detection, emotion recognition and gender classification. In *Proceedings of the Eighth International Symposium on Information and Communication Technology, Nha Trang City, Viet Nam, December 7-8, 2017*, pages 340–347, 2017. https://doi.org/10.1145/3155133.3155207.

[30] D. V. Sang, L. T. B. Cuong, and D. P. Thuan. Facial smile detection using convolutional neural networks. In *The 9th International Conference on Knowledge and Systems Engineering (KSE 2017)*, pages 138–143, 2017. https://doi.org/10.1109/kse.2017.8119448.

[31] D. V. Sang, N. V. Dat, and D. P. Thuan. Facial expression recognition using deep convolutional neural networks. In *The 9th International Conference on Knowledge and Systems Engineering (KSE 2017)*, pages 144–149, 2017. https://doi.org/10.1109/kse.2017.8119447.

[32] C. Shan. Smile detection by boosting pixel differences. *IEEE transactions on image processing*, 21(1):431–436, 2012. https://doi.org/10.1109/tip.2011.2161587.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. https://doi.org/10.1109/cvpr.2015.7298594.

[36] Y. Tang. Deep learning using support vector machines. *CoRR, abs/1306.0239*, 2, 2013.

[37] I. Ullah, M. Hussain, G. Muhammad, H. Aboalsamh, G. Bebis, and A. M. Mirza. Gender recognition from face images with local wld descriptor. In *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, pages 417–420. IEEE, 2012.

[38] H. Van Kuilenburg, M. Wiering, and M. Den Uyl. A model based method for automatic facial expression recognition. In *Proceedings of the 16th European Conference on Machine Learning (ECML'05)*, pages 194–205. Springer, 2005. https://doi.org/10.1007/11564096_22.

[39] P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. In *Advances in neural information processing systems*, pages 1311–1318, 2002.

[40] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017. https://doi.org/10.1109/cvpr.2017.634.

[41] S. Zagoruyko and N. Komodakis. Wide residual networks. In *Procedings of the British Machine Vision Conference 2016*. British Machine Vision Association, 2016. https://doi.org/10.5244/c.30.87.

[42] K. Zhang, Y. Huang, H. Wu, and L. Wang. Facial smile detection based on deep learning features. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, pages 534–538. IEEE, 2015. https://doi.org/10.1109/acpr.2015.7486560.