Mobile Application for Monitoring Body Temperature from Facial Images Using Convolutional Neural Network and Support Vector Machine

Yufeng Zheng[a], Hongyu Wang[b], Yingguang Hao[b]

[a]Alcorn State University, Lorman, MS, USA

[b]Dalian University of Technology, Dalian, China

Emails: yzheng@alcorn.edu, whyu@dlut.edu.cn

ABSTRACT

Human body temperature is an important vital sign especially for health monitoring and exercise training. In this study, we propose a CNN plus support vector machine (SVM) approach (CNN-SVM) to estimate body temperature from a sequence of facial images. The sequence images could be from multiple shots or from video frames using a smartphone camera. First, the facial region is cropped out from a digital picture using a face detection algorithm, which can be implemented on the smartphone or at cloud side. Second, normalize the batch of facial images, and extract the facial features using a pretrained CNN model. Lastly, train a body temperature prediction model with the CNN features using a multiclass SVM classifier. The feature extraction and classification are performed in the cloud side with GPU acceleration and the predicted temperature is then sent back to the mobile app for display. We have a facial sequence database from 144 subjects. There are 12-18 shots of facial images taken from each subject. We selected AlexNet, ResNet-50, VGG-19, or Inception-ResNet-v2 models for feature extraction. The initial results show that the performance of the proposed method is very promising.

Keywords: Body temperature estimation, Convolutional neural network (CNN), Support vector machine (SVM), Facial image sequence, Mobile application.

1. INTRODUCTION

Human body temperature is an important vital sign yet easy to be measured. A number of diseases are characterized by a change in body temperature. A *fever* is the reaction to a disease-specific stimuli. The body changes its normal temperature to support the body's own defense mechanisms. Fever is the most common form of disease-related symptoms.

The normal human body temperature range is typically stated as 36.5–37.5 °C (97.7–99.5 °F).[1] However, human body temperature is variable and dependent upon one's sex, age, time of day, exertion level, health status (i.e. illness, menstrual cycle in females), the location in/on the body in which the measurement is being taken, the subject's state of consciousness (waking, sleeping or sedated), as well as emotional state. Body temperature is maintained within normal range by thermoregulation whereby the lowering or raising of temperature is triggered by the central nervous system.

There are various types of medical thermometers, as well as sites used for measurement. A typical way is to put a digital thermometer in one's mouth (oral temperature). This is relatively convenient in contrast with the measurement of axillary temperature or rectal temperature. However, all these measurements are done by contacting, which is not suitable for monitoring a crowd of contagious diseases.

New coronavirus (COVID-19) is quickly spreading over the world since December 2019. Clearly a contactless temperature measurement is desired. As shown in Fig. 1, a non-contact infrared forehead thermometer is an affordable hand-held device for individual temperature measurement, while a thermal camera system is viable (but expensive) for crowd monitoring in public sites (e.g., airports).

Mobile Multimedia/Image Processing, Security, and Applications 2020, edited by Sos S. Agaian, Vijayan K. Asari, Stephen P. DelMarco, Proc. of SPIE Vol. 11399, 113990B · © 2020 SPIE
CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2557856

Can we measure and monitor body temperature using a regular camera (like a smartphone camera)? The answer is yes with the power of deep learning neural network and big data. This is an affordable solution and no additional equipment to carry on since we always bring our smartphone anyway.

(a)

(b)

Fig. 1: Body temperature measurement tools: (a) Non-contact Infrared Forehead Thermometer; (b) Crowd temperature monitoring system using both visual and thermal camera .

In this study, we propose to extract facial features with a Convolutional Neural Network (CNN) model and train a support vector machine (SVM) classifier to predict temperature. Four representative CNN models (AlexNet, VGG-19, ResNet-50 and Inception-ResNet-v2) are selected, which have been trained on ImageNet[2] with excellent accuracies to classify nature images. CNN is a special type of multi-layer neural networks designed to recognize visual patterns directly from the pixel natural images, while SVM can efficiently perform a non-linear classification especially with a small dataset. The remainder of this paper is organized as follows. Section 2 describes preprocessing. Section 3 reviews the four CNN models. Section 4 presents the experimental results. Section 5 summarizes the paper.

2. PREPROCESSING AND CLASSIFICATION

The body temperature prediction model includes preprocessing (normalization, face detection, standardization), CNN feature extraction (depicted in Section 3), and SVM classification (or regression).

2.1 Image normalization and face detection

Image normalization (intensity scaling/stretching) is defined as

$$\mathbf{I}_{\mathrm{N}} = (\mathbf{I}_0 - I_{\mathrm{Min}})\frac{L_{\mathrm{Max}} - L_{\mathrm{Min}}}{I_{\mathrm{Max}} - I_{\mathrm{Min}}} + L_{\mathrm{Min}} \tag{1}$$

where $\mathbf{I}_N$ is the normalized image, $\mathbf{I}_0$ is the original image; I_{Min} and I_{Max} are the minimum and maximum pixel value in $\mathbf{I}_0$, while L_{Min} and L_{Max} are the desired minimum and maximum pixel value in $\mathbf{I}_N$. For example, we may select $L_{Min} = 0$ and $L_{Max} = 255$.

Face detection is to localize the face in the image, which is an important step for successive process. Numerous techniques have been developed to detect faces in a single image, e.g., knowledge-based methods, feature invariant facial approaches, template matching methods, appearance-based methods. An excellent survey about face detection can be found in Reference [3]. Many methods use color information, i.e., identify the regions (skin-map) whose color is similar to the color of the skin, which can restrict the searching area significantly. One of the most successful algorithms in visual images is Viola-Jones algorithm[4] proposed in 2001.

Face detection methods are well developed and quickly mark multiple faces on a picture regardless of their sizes and backgrounds, utilizing spatial changes. The Viola-Jones (VJ) algorithm has become a very common method of object detection, including face detection. Viola and Jones proposed this algorithm as a machine learning approach for object detection with an emphasis on obtaining results rapidly and with high detection rates. The VJ method uses three important aspects. The first is an image representation structure called *integral images*,[5,6] wherein features are calculated by taking the sum of pixels within multiple rectangular areas. This is of course an extension of a method by Papageorgiou *et al.*[7]

Viola and Jones admit that the use of rectangles is rather primitive, but they allow for high computational *efficiency* in calculations.[5,6] This also lends well to the Adaboost algorithm that is used to learn features in the image. This extension of the Adaboost algorithm allows for the system to select a set of features and train the classifiers, a method first discussed by Freund and Schapire.[8] This learning algorithm allows the system to learn the differences between the integral representations of faces to those of the background.

The last contribution of the VJ method is the use of *cascading classifiers*. At each stage, the classifier either rejects the instance (represented as a sliding-window from a given test image) based on a given feature value or sends the instance down the tree for more processing. Initially, a large number of negative examples are eliminated. In order to significantly speed the system up, Viola and Jones avoid areas that are highly unlikely to contain the object. The image is tested with the first cascade, and the areas that do not conform to the first cascade are rejected and no longer processed. The areas that may contain the object are further processed until all of the classifiers are tested. The areas in the last classifier are likely to contain the object (face).

2.2 Facial image standardization

We apply general standardization all facial images.

$$\mathbf{I}'_N = \mathbf{I}_N - \mathbf{I}_M \,, \tag{2}$$

$$\mathbf{I}_S = (\mathbf{I}'_N - \mu)/\sigma \tag{3}$$

where $\mathbf{I}_N$ is the normalized facial image, $\mathbf{I}_M$ is the mean image of all faces in the dataset, and $\mathbf{I}'_N$ is their difference image. $\mathbf{I}_S$ is the standardized image; μ and σ denote the mean and standard deviation of the $\mathbf{I}'_N$ image, respectively.

2.3 Support vector machine model

Support-vector machine (SVM) can efficiently perform a non-linear classification especially with a small dataset.[9] SVMs are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two (or multiple) categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear.

A SVM constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space for classification or regression. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.[10] Whereas the original problem defined in a finite-dimensional space may not be linearly separable in that space. For this reason, the original finite-dimensional space can be mapped into a much higher-dimensional space, presumably making the separation easier in that space.

Multiclass SVM aims to assign labels to instances by using SVMs, where the labels are drawn from a finite set of several elements. The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems.[11]

3. CONVOLUTIONAL NEURAL NETWORKS

3.1 Convolutional neural network

Convolutional neural networks (CNNs) take a *biological inspiration* from the visual cortex that has small regions of cells that are sensitive to specific regions of the visual field. CCNs are sort of combination of biology, math and computer science, but these networks have been the most influential innovations in the field of computer vision and artificial intelligence (AI). 2012 was the first year that CNN grew to prominence as Alex Krizhevsky *et al.*[12] used an 8-layer CNN (5 conv., 3 fully-connected) to win that year's ImageNet competition (referred to as *AlexNet* thereafter), dropping the classification error record from 25.8% (in 2011) to 16.4% (in 2012), an astounding improvement at the time. Since then many applications have been developed using *deep learning*.

A common misconception in the deep learning community is that without huge amount of data, it is not possible to create effective deep learning models. While data is a critical part of creating the network, the idea of transfer learning has helped to lessen the data demands. *Transfer learning* is the process of taking a *pretrained* model (the weights and parameters of a network that has been trained on a large dataset) and *fine-tuning* the model with your small dataset. The idea is that the pretrained model will act as a *feature extractor*. In general, the last layer of the network is replaced with your classification layer (depending on how many classes in your problem). Then the network is trained normally (to get adapted to your dataset).

Let us take a look at the pretrained model with *ImageNet*, a dataset that contains 14 million images with over 1,000 classes.[2] The *lower layers* of the network will detect features like edges and curves. Most likely your network is going to need to detect curves and edges as well. Rather than training the whole network (i.e., all layers) through a random initialization of weights, it is more efficient and effective to use the weights of the pretrained model and focus on the more important layers (higher layers towards classification output) for training. If your dataset is quite different than ImageNet, then you would want to train more of higher layers.

3.2 AlexNet model

AlexNet[12] is a deep CNN for image classification that won the ILSVRC (ImageNet Large-Scale Visual Recognition Challenge)[13] 2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry. AlexNet has 8 weight layers as shown in Fig. 1. The first 5 are convolutional and the last 3 are fully connected layers. *Relu* (Rectified Linear Unit) is applied after every convolutional and fully connected layer. *Dropout* is applied before the first and the second fully connected layer. The input image size should be 227×227×3. The AlexNet has a total of 63M parameters.

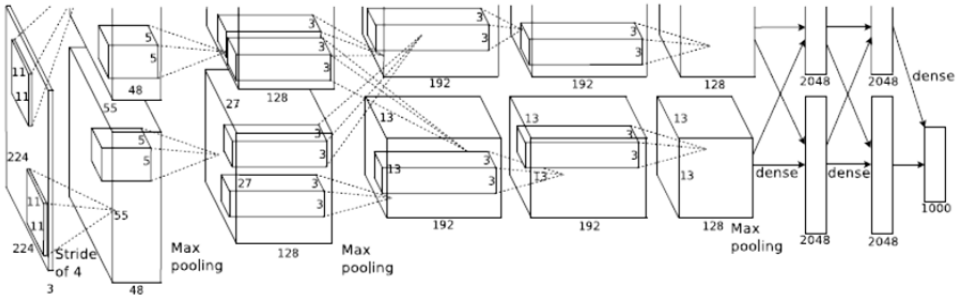


Fig. 1 The network architecture of AlexNet[12] (5 conv. + 3 FC layers): conv means convolution, FC means fully connected.

Though there are many network topologies that have emerged since with lot more layers, AlexNet was the first to make a breakthrough. The AlexNet (1) used *Relu* instead of *Tanh* to add non-linearity. It accelerates the speed by six times at the same accuracy; (2) used *dropout* instead of *regularization* to deal with overfitting. However the training time is doubled with the dropout rate of 0.5; and (3) overlapped *pooling* to reduce the size of network. It reduces the top-1 and top-5 error rates by 0.4% and 0.3%, respectively. The AlexNet was trained using batch *stochastic gradient descent*

(SGD), with specific values for *momentum* and weight decay. The training on two GTX 580 GPUs (for two sub-nets, up and bottom, as shown in Fig. 1) took for about six days. This was the first time a model performed so well on a historically difficult ImageNet dataset, which backed CNNs up with record breaking performance in the competition.

Though there are many network topologies that have emerged since with lot more layers, AlexNet was the first to make a breakthrough. The AlexNet (1) used *Relu* instead of *Tanh* to add non-linearity. It accelerates the speed by six times at the same accuracy; (2) used *dropout* instead of *regularization* to deal with overfitting. However the training time is doubled with the dropout rate of 0.5; and (3) overlapped *pooling* to reduce the size of network. It reduces the top-1 and top-5 error rates by 0.4% and 0.3%, respectively. The AlexNet was trained using batch *stochastic gradient descent* (SGD), with specific values for *momentum* and weight decay. The training on two GTX 580 GPUs (for two sub-nets, up and bottom, as shown in Fig. 1) took for about six days. This was the first time a model performed so well on a historically difficult ImageNet dataset, which backed CNNs up with record breaking performance in the competition.

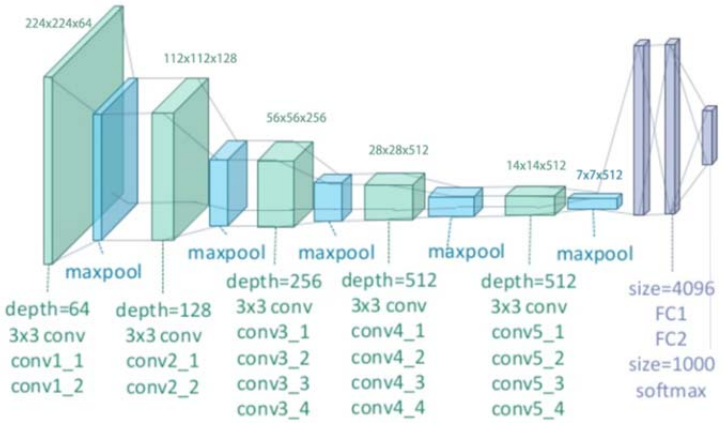


Fig. 2 The network architecture of VGG-19 model[14] (16 conv. + 3 FC layers): conv means convolution, FC means fully connected

3.3 VGG-19 model

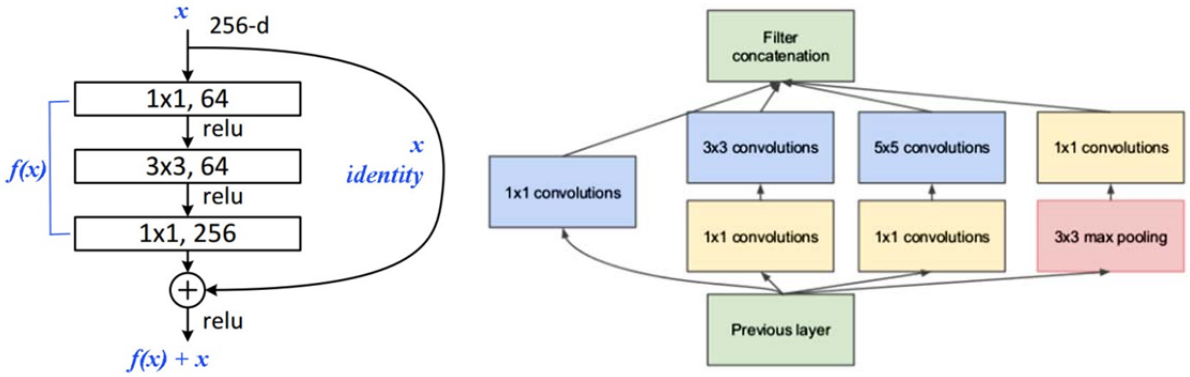
Simonyan and Zisserman[14] of the University of Oxford created a 19-layer (16 conv., 3 fully-connected) CNN that strictly used 3×3 filters with stride and pad of 1, along with 2×2 max-pooling layers with stride 2, called VGG-19 model.[14,15] Compared to AlexNet, the VGG-19 (see Fig. 2) is a *deeper* CNN with more layers. To reduce the number of parameters in such a deep network, it uses small 3×3 filters in all convolutional layers and best utilized with its 7.3% error rate. The VGG-19 model was not the winner of ILSVRC[13] 2014, however, the VGG Net is one of the most influential papers because it reinforced the notion that CNNs have to have a deep network of layers in order for this hierarchical representation of visual data to work. The motto for CNN architecture design is to keep it deep and keep it simple.

The VGG-19 model, a total of 143.7M parameters, was placed 2nd in classification and 1st in localization in ILSVRC 2014. The VGG-19 model is trained on more than a million images (ImageNet) and can classify images into 1000 object categories, for example, keyboard, mouse, pencil, and many animals. As a result, the model has learned rich feature representations for a wide range of images.

3.4 ResNet-50 and Inception-Resnet-v2 models

The ResNet-50 model[16] has 101 layers, which consists of 33 three-layer *residual* blocks (as shown in Fig. 3a) plus input and output layers. *Identity connections* learn incremental, or residual, representations, which creates a path for back-propagation. The identity layers gradually transform to complex. Such evolution occurs if the parameters for the $f(x)$ part (shown in Fig. 3a) begin at or near zero. The residual block helps overcome the hard the training problem in DeepNet (> 30 layers) due to vanishing gradients. The ResNet-101 model uses 3×3 filters with stride of 2, and 3×3 max-pooling layers with stride 2. The ResNet-101 was placed the 1st in ImageNet classification in ILSVRC 2015.

The Inception-v3[17,18] is a 42-layer CNN model including Inception Module A (see Fig. 3b), B, C. The inception model uses variant kernel size (in v1) to capture the features from variant object size and location, introduces batch (weight) normalization (in v2) and factorizing convolutions (in v3), and uses bottleneck layers (1×1) to avoid a parameter explosion. The Inception-v1 (GoogLeNet) was placed 1st in ImageNet classification in ILSVRC 2014. The Inception-v3 model uses 3×3, 5×5, 1×1 filters with stride 1-2, and 2×2 max-pooling layers with stride 2.



(a) (b)

Fig. 3 (a) A three-layer building block for ResNet-50, where the jump (*x* identity) connection and adder ('+') create a path for back-propagation. 256-d means 256 convolutional kernels; (b) Inception Module A in the Inception-v3 model: A large-size kernel is factored into variant small-size convolution kernels.

The combination of two of the most recent ideas: Residual connections[16] and the latest revised version of the Inception architecture[18]. In Ref. [16], it is argued that residual connections are inherently important for training very deep architectures. Since Inception networks tend to be very deep, it is natural to replace the filter concatenation stage of the Inception architecture with residual connections. This would allow Inception to reap the benefits of the residual approach while retaining its computational efficiency. Inception-ResNet-v2 is a hybrid Inception version with residual connections, which leads to dramatically improved recognition performance and training speed in contrast with the Inception architecture.[19]

3.5 CNN model summary

Four selected CNN models are summarized in Table 1 and more CNN models are visualized in Fig. 5. All CNN models take color images as inputs. All facial images are scaled to proper size according to each CNN model prior to training and testing. The pretrained CNN models are transferred from the ImageNet.

Table 1. Summary of four CNN models in term of accuracy, layers, parameters, input image, and feature dimension

CNN Model	AlexNet	VGG-19	ResNet-50	Inception-ResNet-v2
Top-1 Accuracy (on ImageNet)	57.1%	71.3%	75.1%	80.0%
Number of Layers	8	19	50	572
Number of Parameters	63M	143.7M	25.6M	56M
Input Image Size	227×227×3	224×224×3	224×224×3	299×299×3
Output Feature Dimension	4096	4096	2048	1536

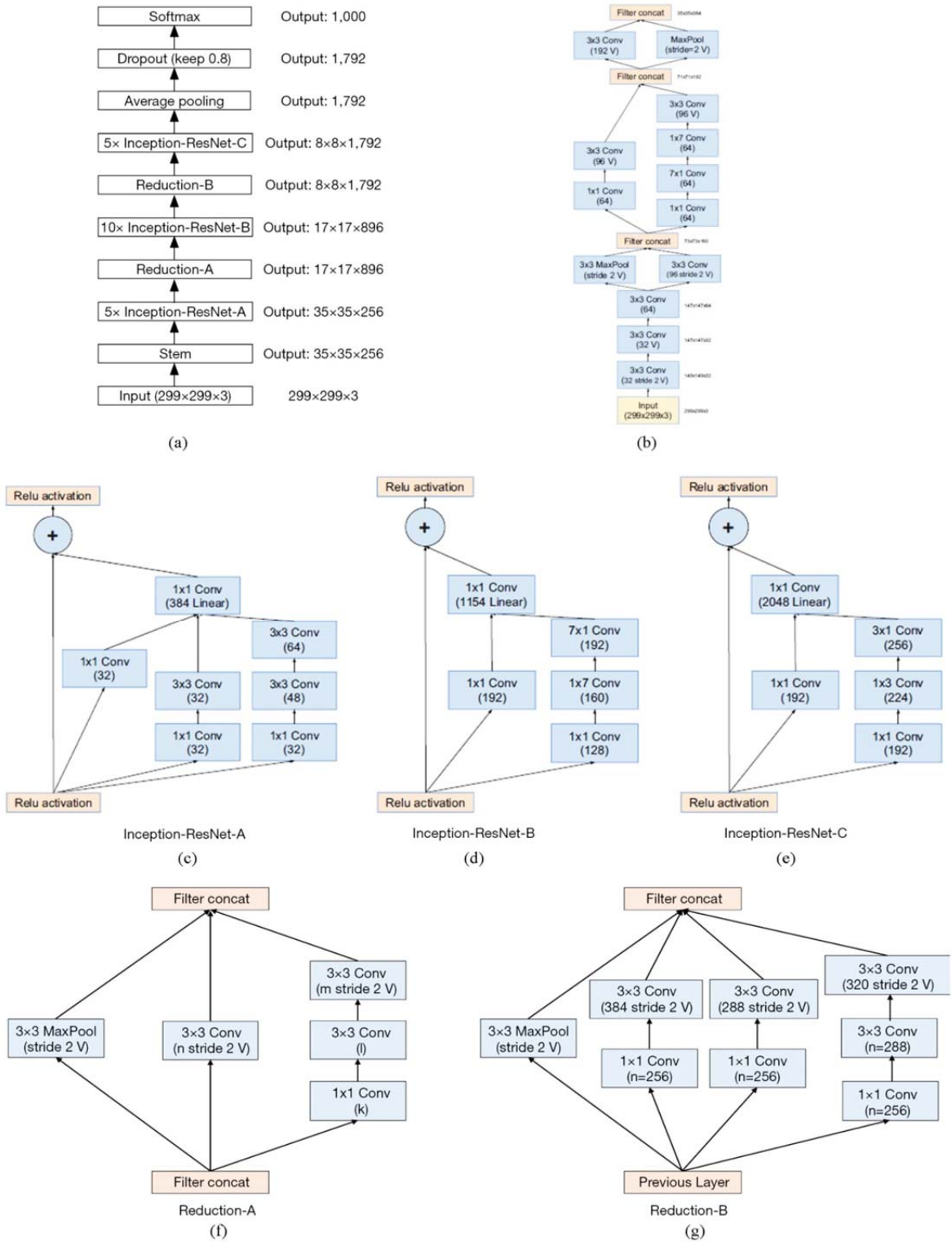


Fig. 4 Architecture of Inception-ResNet-v2[19]: (a) architecture (schema), (b) the stem, (c)-(e) interior inception (35 × 35, 17 × 17 and 8 × 8 grid) modules, (f)-(g) reduction modules.

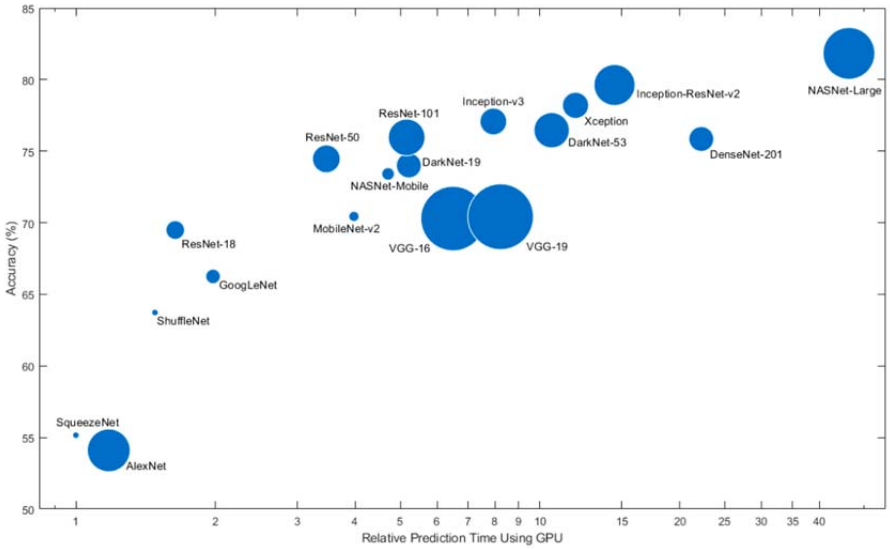
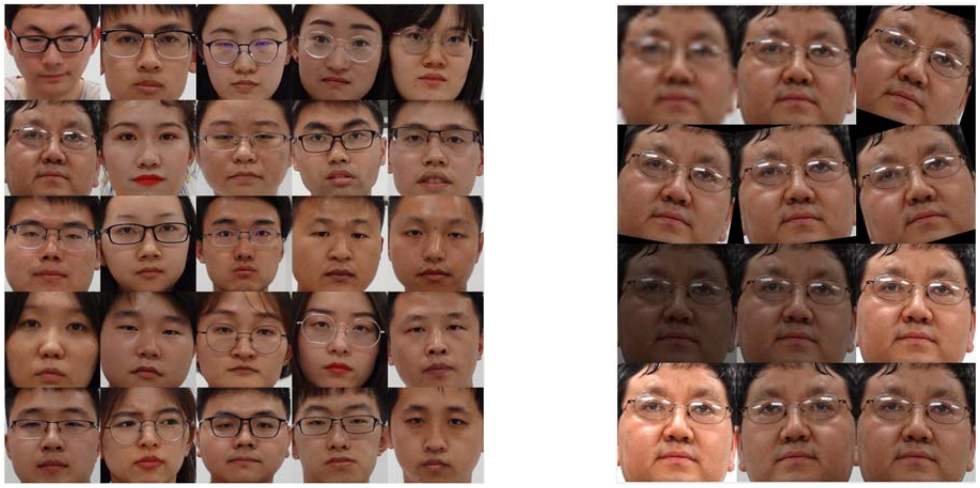


Fig. 5 Overview of CNN models: Top-1 accuracy (tested on ImageNet) vs. inference time (a single forward pass). The size of the blobs is proportional to the number of network parameters.

4. EXPERIMENTAL RESULTS

We have collected a facial sequence database from 144 subjects. From each subject three datasets were collected: (a) 12-18 shots of facial images taken using a Sony digital camera (a total of 2197 images, see Fig. 6a), (b) 4-8 images captured with a smartphone camera (a total of 385 images, see Fig. 7a), and (c) 3-4 video clips (25-40 seconds) recorded with the Sony camera. The body temperatures were measured using a hand-held infrared forehead thermometer (in degrees of Celsius, see Fig. 1a). Two measurements were taken at the beginning and end of experiment for each subject. The averaged temperature was used as ground truth.



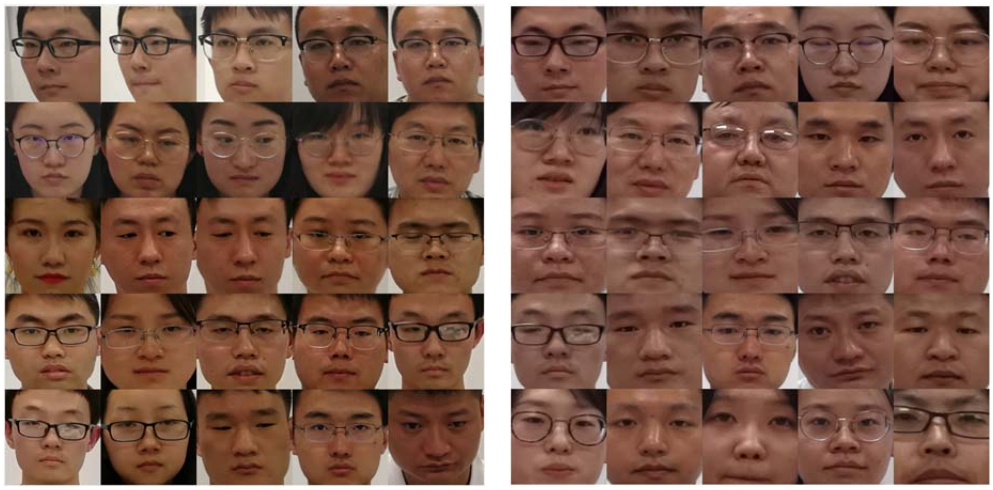
(a) (b)

Fig. 6 The example facial images for training: (a) Sony Camera pictures, (b) 12 augmented pictures by scaling, rotation, brightness changing, flipping and original.

All images were normalized (intensity stretched), face detected, and standardized. Each face image was augmented to 12 variations (see Fig. 6b) by 2 scales, 4 rotations, 4 brightness adjustments, 1 flipped and 1 original. Then facial features were extracted using four CNNs. A SVM model was trained utilizing the feature vectors from 90% of Sony pictures. The

SVM model was tested with three image sets (all with augmentation): (i) 10% of Sony pictures, (ii) all smartphone pictures (Fig. 7a), and (iii) thee frames extracted from each video clip (at time 5, 15, 20 second), a total of 1341 frame pictures (Fig. 7b).

The first section presents the temperature prediction performance with four CNN-SVM models. The time costs of CNN-SVM models are reported in the next section.



(a) (b)

Fig. 7 The example facial images for testing (inference) (a) from the Smartphone pictures, (b) from video frames taken by Sony Camera.

4.1 Body temperature prediction on three datasets

The performance of body temperature prediction is measured by prediction error and their descriptive statistics shown in Tables 2-4. The descriptive statistics include mean, standard deviation (SD), median, minimum and maximum (range = maximum - minimum).

$$\text{prediction error} = \text{prediction temperature} - \text{ground-truth temperature}. \quad (4)$$

Table 2. The prediction error of body temperature (in degrees of Celsius) varying with four CNN-SVM models: Trained on 90% Sony Camera pictures and Tested on 10% *Sony camera* pictures

Error \ CNN	AlexNet	VGG-19	ResNet-50	Inception-ResNet-v2
Mean	-0.000	-0.001	-0.000	-0.000
Standard Deviation	0.013	0.042	0.027	0.024
Median	0.000	0.000	0.000	0.000
Minimum, Maximum	-0.45, 0.25	-1.00, 0.65	-0.70, 0.50	-0.45, 0.65

The temperature prediction errors on 10% Sony pictures are presented in Table 2. The mean errors approach zero due to training and testing on the same dataset. AlexNet is the best in terms of all descriptive statistics (especially maximum error). Table 3 shows the prediction errors tested with the smartphone pictures. It seems ResNet-50 is the best according to standard deviation and error range. Temperature prediction error increases while tested with video frames, where VGG-19 and ResNet-50 perform very well. It is clear that video-frame images (Fig. 7b) are not as good as digital images (Fig. 6a) in terms of quality (e.g., signal to noise ratio, contrast).

Let us walk through one example in Table 3, where testing was conducted on smartphone pictures. Suppose we select ResNet-50-SVM model. The mean predication error is only -0.017 °C with SD = 0.168. With the assumption of normal distribution in all temperature predictions, 95% of prediction errors (2 SD) are within the range: [-0.353, 0.319] °C. This

is a reasonable good result considering the size of our datasets.

Table 3. The prediction error of body temperature (in degrees of Celsius) varying with four CNN-SVM models: Trained on 90% Sony Camera pictures and Tested on the *Smartphone* pictures

Error \ CNN	AlexNet	VGG-19	ResNet-50	Inception-ResNet-v2
Mean	-0.038	-0.002	-0.017	-0.039
Standard Deviation	0.178	0.185	0.168	0.210
Median	0.000	0.000	0.000	0.000
Minimum, Maximum	-1.00, 0.80	-0.90, 1.00	-0.85, 0.80	-1.00, 0.80

Table 4. The prediction error of body temperature (in degrees of Celsius) varying with four CNN-SVM models: Trained on 90% Sony Camera pictures and Tested on the *video frames*

Error \ CNN	AlexNet	VGG-19	ResNet-50	Inception-ResNet-v2
Mean	-0.158	-0.081	-0.091	-0.127
Standard Deviation	0.267	0.271	0.276	0.296
Median	-0.100	0.000	-0.050	-0.050
Minimum, Maximum	-1.00, 0.60	-1.00, 1.10	-1.00, 0.95	-1.00, 0.95

Overall, AlexNet and ResNet-50 performs very well and reliably, whereas VGG-19 and Inception-ResNet-v2 do not exhibit much advantage.

4.2 Time costs of CNN models

The time costs of CNN feature extraction, SVM model training and testing are given in Table 5. It is clear that AlexNet took the longest time while Inception-ResNet-v2 is the fastest.

Facial feature extraction by four CNN models were run on a MSI GT73VR laptop with the following configuration: Intel i7-7820HK CPUs 2.9GHz, 16GB RAM, 1.25TB hard disk, 64-bit Windows 10; NVIDIA GeForce GTX 1070 Graphics Board with 8GB video memory (on board) and 1920 CUDA cores. Four CNN-SVM models were implemented and run in Matlab R2019a (Version 9.6), on a laptop computer, Apple MacBook Pro 2014, with the following configuration: Intel i7-4960HQ CPUs 2.6GHz, 16GB RAM, 1.0TB hard disk, 64-bit Windows 10; NVIDIA GeForce GT 750M Graphics Board with 2GB video memory (on board) and 384 CUDA cores.

Table 5. Time costs in CNN feature extraction, SVM model training and testing (inference latency) varying with four CNN-SVM models

Process \ CNN	AlexNet	VGG-19	ResNet-50	Inception-ResNet-v2
CNN Feature Extraction (ms/image)	1.17	12.69	6.87	22.93
SVM Training Time (sec)	118.7	127.9	82.6	74.8
SVM Testing Time (ms/image)	2.4	2.4	1.5	1.0

5. SUMMARY AND DISCUSSION

In this study, we proposed to use facial images and CNN-SVM model to predict body temperature. The facial images were taken using a digital camera and a smartphone. Facial features were extracted by AlexNet, VGG-19, ResNet-50 and Incpetion-ResNet-v2. Then a SVM model was trained for temperature prediction. Overall we suggest the ResNet-50-SVM model that provides low error (< 0.35 °C) and yet fast process. Our experimental results shed light on a new method for body temperature measurement and monitoring.

The facial images could be taken by a smartphone, which eliminates the need of any special device like infrared thermometer or thermal camera. This is an economic and convenient implementation for temperature monitoring especially for fever-related contagious diseases like COVID-19 virus.

With a large scale dataset, the accuracy and reliability of temperature prediction will be further improved. The inference latency may be reduced with highly configured hardware (GPUs). We are in the progress of extending out dataset.

6. REFERENCES

[1] Hutchison, James S.; et al., "Hypothermia therapy after traumatic brain injury in children". New England Journal of Medicine. 358 (23): 2447–2456 (2008).

[2] ImageNet, http://www.image-net.org.

[3] M. H. Yang, D. J. Kriegman, N. Ahuja, "Detecting faces in images: a survey," IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (1) 34-58 (2002).

[4] P. Viola , M. Jones, "Robust real-time object detection," International Journal of Computer Vision, (2001)

[5] Viola, P. and Jones, M., "Rapid Object detection using a boosted cascade of simple features." Proceedings of CVPR, vol. 1 pp. 511–518, (2001).

[6] Viola, P. and Jones, M., "Robust Real-time Object Detection," International Journal of Computer Vision, Vol. 57, Iss. 2, pp. 137–154, (2001).

[7] Papageorgiou, C., Oren, M., and Poggio, T., "A general framework for object detection," Sixth International Conference on Computer Vision, pp. 555-562, (1998).

[8] Freund, Y. and Schapire, R., "A decision theoretic generalization of on-line learning and an application to boosting," Computational Learning Theory: Eurocolt'95, pp 23-37, (1995).

[9] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery 2, 121-167, 1998.

[10] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" (Second ed.). New York: Springer. p. 134, (2008).

[11] Duan, Kai-Bo; Keerthi, S. Sathiya, "Which Is the Best Multiclass SVM Method? An Empirical Study", Multiple Classifier Systems. LNCS. 3541. pp. 278–285, (2005).

[12] Krizhevsky, A., Sutskever, I., Hinton, G.E., "Imagenet classification with deep convolutional neural networks," Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Pages 1097-1105, Lake Tahoe, Nevada, (2012).

[13] Russakovsky, O., Deng, J., Su, H., et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV). Vol. 115, Issue 3, pp. 211–252 (2015).

[14] Simonyan, K., Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv technical report, (2014).

[15] University of Oxford, Visual Geometry Group, http://www.robots.ox.ac.uk/~vgg/research/very_deep/.

[16] He, K., Zhang, X., Ren, S., & Sun, J., "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778, (2016).

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," CVPR, (2015).

[18] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z., "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818-2826, (2016).

[19] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alexander A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," AAAI 2017: 4278-4284, (2017).