Deep Learning for American Sign Language Fingerspelling Recognition System

Huy B.D Nguyen and Hung Ngoc Do
School of Electrical Engineering, International University, Vietnam National University - Hochiminh City, Vietnam
Email: huy4512@gmail.com and dnhung@hcmiu.edu.vn

***Abstract*—Sign language has always been a major tool for communication among people with disabilities. In this paper, a sign language fingerspelling alphabet identification system would be developed by using image processing technique, supervised machine learning and deep learning. In particular, 24 alphabetical symbols are presented by several combinations of static gestures (excluding 2 motion gestures J and Z). Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP) features of each gesture will be extracted from training images. Then Multiclass Support Vector Machines (SVMs) will be applied to train these extracted data. Also, an end-to-end Convolutional Neural Network (CNN) architecture will be applied to the training dataset for comparison. After that, a further combination of CNN as feature descriptor and SVM produces an acceptable result. The Massey Dataset is implemented in the training and testing phases of the whole system.**

***Index Terms*—Deep learning, convolutional neural network, fingerspelling, American sign language.**

I. Introduction

Sign language is a complex combination of hand movements, body postures and facial expression; which is used to convey meaning among hearing and speech impaired people, as well as between people with disabilities and normal people. It is then realized that the design of a fast and highly accurate interpreting system plays a vital role, as it can facilitate the communication process for those in need, and further integrate them into the common social world. Through the capturing and recognition of hand gestures, it is possible to achieve deep analysis of human behavior, and further responds accordingly. Only by the implementation of a concrete sign language identification system this strenuous problem can be resolved.

Generally, hand gesture recognition is based on the following 2 approaches: vision based approach and non-vision based approach. In recent years, researches were conducted to develop hand gesture recognition systems have been more concentrated on vision based methods as they produce much less, to no restraint on the users, as compared to the previous non-vision based procedures which employ special gloves or sensors to directly collect data from hands [1].

The first step in hand gesture recognition with vision-based approach is preprocessing, in which skin segmentation is fundamental. The authors in [2] combined the skin-color model (RGB to YCbCr) with eight neighborhood method to segment out the hand region and erase noise. The skin color range can be adaptively changed based on different users, if motion detection and Gaussian Mixture Model were involved [3]. Also, Zuo proposed the use of an end-to-end network to detect human skin in complex background, which includes the integration of recurrent neural network (RNNs) layers into the fully convolutional neural networks (FCNs) [4]. Good feature extraction step is necessary to describe the relationship of the hand region pixels. In [2], the authors divided the hand region into several sections, calculated for the distribution of the farthest point on the perimeter to the centroid of the hand, and also, the area distribution of the hand region on multiple concentric arches clockwise. The fingertips can be found based on the combination of the K-curvature and convex hull algorithms [5]. Furthermore, the authors in [6] extracted scale-invariant features based on the ratio of length, width, perimeter, and area of the hand region. In [3], the authors implemented the SIFT algorithm with the seven Hu moments and Fourier descriptor to achieve the feature vector. In addition, as wavelet invariant moments can be sensitive to the change of area, so the 34-dimensional wavelet invariant moments were calculated in combination with hand contour features found with a Moore neighborhood algorithm to extract for the wanted features [7]. To further decrease the computational complexity, the authors in [2], [3] and [8] implement Principle Component Analysis (PCA) to reduce the dimensions of feature vectors. In the training and classification step, SVM and its variation such as Simplification SVM (SimpSVM) and Relevance Vector Machine (RVM) are employed [9]. Other methods include the use of Hidden Markov Models (HMM) [10], Artificial Neural Network (ANN) [5] and Madaline neural network [6].

Next, other sophisticated works related to hand gesture detection and pose estimation are reviewed. In [11], CNN is utilized to calculate the sixteen 3D hand joint locations, which are the solutions of the formulated regression problem. The input is the depth image detected using the contour-based approach. The output from the hand detection step is fed into the CNNs network with a PCA-layer in between to reduce the effect of large dimensionality. In [12], the authors developed a hybrid convolutional neural network plus auto-encoder model (Hybrid CNN-AE) to predict the articulation of a 3D hand gesture. The difference in this architecture compared with other models is that a denoising auto-encoder is inserted as an embedding layer. A new method to detect hands and their rotation in uncontrolled environments simultaneously was proposed in [13]. The architecture employs the backbone of faster R-CNN for local feature extraction and rectangular region proposals. Next comes the rotating network with the

purpose of calculating for the rotation of each region proposal, and a derotation layer to align the spatial feature map in corresponding with the previous rotating network. The already derotated feature map was then fed into the detecting network for hand detection.

In this paper, we present three methods for fingerspelling alphabets recognition in which employ solely machine learning, deep learning, and a combination of both. The first system involves LBP and HOG descriptors for feature extraction and multi-class multi-kernel SVM classifiers, which is a classical machine learning method. In this system, HOG descriptor will be applied across the noise-filtered image, while LBP descriptor would be applied only on the high-frequency components (texture) of the original image. Then, the two feature vectors are concatenated for further processing. In the classification step, multiclass SVM will be implemented (1). The second model employs an end-to-end CNN architecture for training and classification (2). In the third method, trained CNN weights up to the fully connected layer in the second model (2) will be implemented as a feature extractor for a Linear-kernel SVM classifier (3). Throughout the three models, the Massey Dataset [14] is used for training and performance testing.

This paper is organized as follows. Section I introduces our motivation and several related researches. Section II describes our system model in which each processing stage will be explained in detail. The testing results and discussion are shown in section III. Finally, section VI concludes our paper and gives some recommendations.

II. System Description

Figure 1 shows the processing chart of the three alternative models. Training and testing data are obtained from the Massey Database [14], which includes 2524 images of static alphabetical hand gestures from a to z.

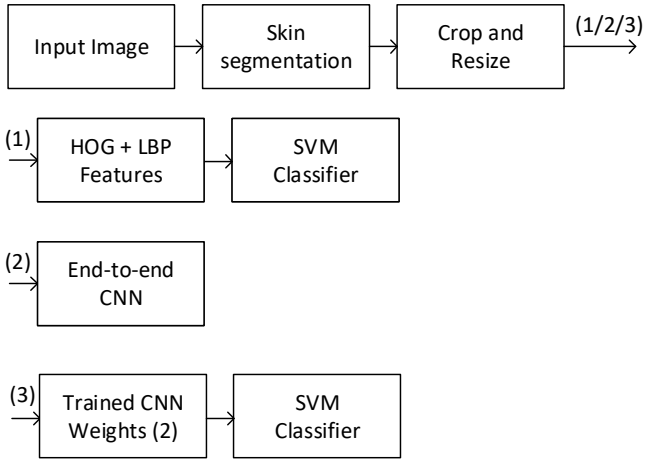

Fig. 1. System model

A. Histogram of Oriented Gradients

HOG is a feature descriptor that is usually combined with SVM classifier for detection and classification problems, with high performance rate. It was originally proposed to recognize pedestrians [15]. The primary idea is to detect sharp changes in their intensity and orientation. The magnitude (g) and orientation (θ) of the gradients are calculated as follows:

$$g = \sqrt{g_x^2 + g_y^2}, \qquad (1)$$
$$\theta = \arctan \frac{g_y}{g_x},$$

with g_x and g_y are the magnitude of gradients in horizontal and vertical direction. Then, the histogram of gradients is generated corresponds to the image cell. Finally, in order to make the gradients less sensitive to the overall lighting, normalization is applied to a block of cells. Figure 2 illustrates a particular result of applying HOG in an image.

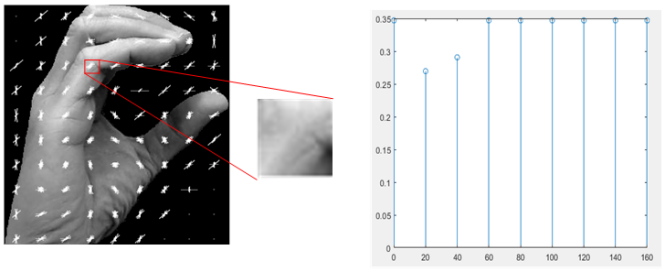

Fig. 2. Histogram of oriented gradients

B. Local Binary Patterns

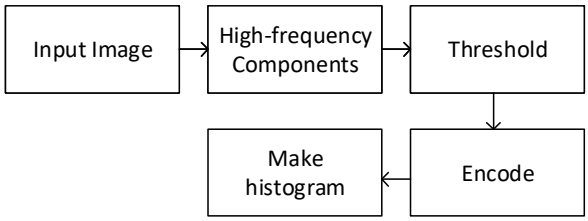

Fig. 3. LBP features processing chart

LBP is originally defined as a grayscale invariant texture operator, with outstanding power and advantage in computational complexity [16]. The detailed layer representing the texture of the hand gesture is extracted by using the processing chart in figure 3. The first step is to threshold a neighborhood of 8 pixels, with the threshold value being the value of the center pixel. After that, the binary values in 1 neighborhood of pixel are encoded as an 8-bit binary code, for example 10001001. The corresponding decimal value is 145. The encoding order must be the same image-wise. Finally, similar to HOG, the LBP histogram is obtained after normalizing block of cells. An example of LBP feature visualization is illustrated in figure 4.

$$s(x,y) = \begin{cases} 1, \text{if } g_p \geq g_e \\ 0, \text{ otherwise} \end{cases} \qquad (2)$$

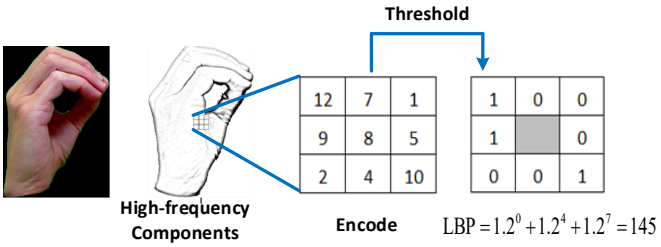

Fig. 4. LBP features visualization

C. Support Vector Machines

SVM is a supervised machine learning method, which was originally developed to solve the problem of linearly separable binary classification. SVM attempts to define an optimal decision boundary so that the margin between the two classes is maximized. Not all datasets are linearly separable or nearly linearly separable, so SVM is adjusted to develop a complex nonlinear classifier. Kernel methods map the original dataset to a higher dimensional space, in which the training dataset is separable, and methods like hard margin SVM or soft margin SVM could be applied to obtain a separating hyperplane. Figure 5 shows the result of applying kernel method to separate two data classes.

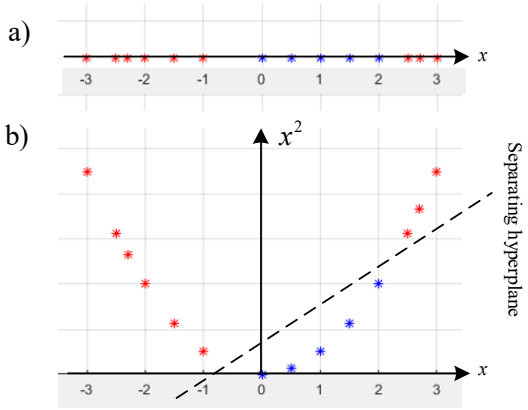

Fig. 5. a) 1D example linear classification does not work. b) Quadratic kernel maps the data into 2D and the separating hyperplane

In this paper, we applied three commonly used kernels including Radial Basis Function (RBF) kernel, Polynomial kernel and Linear Kernel for the multi-class SVM classifier. The function for RBF kernel is described in equation (3):

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma||\mathbf{x} - \mathbf{z}||_2^2), \quad \gamma > 0, \tag{3}$$

with x and z are the feature vectors in the input space. Polynomial kernel function is as follows:

$$K(\mathbf{x}, \mathbf{z}) = (r + \gamma \mathbf{x}^{\mathrm{T}} \mathbf{z})^d. \tag{4}$$

The Linear kernel function does not need any other parameters and can be expressed as in equation (5):

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^{\mathrm{T}} \mathbf{z} \tag{5}$$

D. Convolutional Neural Network Architecture

CNNs are a specialized kind of neural networks, which implement convolution, instead of general matrix multiplication, in at least one of their layers [17]. They have been popularly known to be among the first deep learning models gaining noticeable success.

In the second model (2), we propose a simple chain architecture which achieves reasonable results as in figure 6. Dropout layers are present to reduce overfitting. Data augmentation is applied to make the model more robust to variance in rotation, scaling and shearing. Data augmentation could also help lower the overfitting at a cheaper cost than collecting raw data. Figure 7 shows the training phase of CNN in this project.

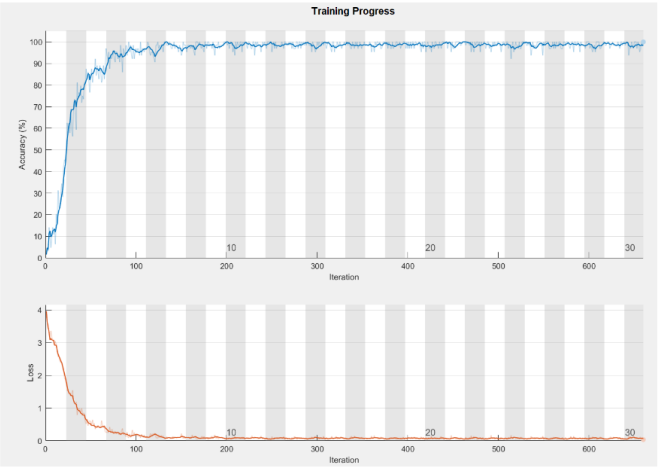

Fig. 7. CNN Training

E. CNN as feature extractor for SVM

Rich feature representations learned by the Convolutional Neural Network (CNN) could outperform classic feature descriptors such as LBP, HOG and SURF. In the third model, the trained weights of CNN layers are used as a feature extractor for a multiclass Linear SVM classifier. Experiments shows that the linear kernel in this case outperforms RBF and Polynomial kernels, which is appropriate since fully connected layer functions as a linear classifier in a CNN architecture. A more elaborated work involves fully replacing the last CNN layer with an SVM layer in the CNN architecture [18].

III. Results

Cross-validation is applied on the Massey Dataset to evaluate the performance of different methods (HOG-SVM, LBP-SVM, HOG-LBP-SVM, CNN, CNN-SVM). The first two schemes employ solely one feature descriptor (HOG or LBP) for multi-class SVM classifier. The third scheme uses the combination of both feature vectors for multi-class multi-kernel SVM classifier (RBF kernel for HOG and Polynomial Kernel for LBP). In the fourth scheme (CNN), the CNN architecture is utilized as an end-to-end system. The last method (CNN-SVM) applies the trained weights of the CNN as feature descriptors for multiclass Linear kernel SVM.

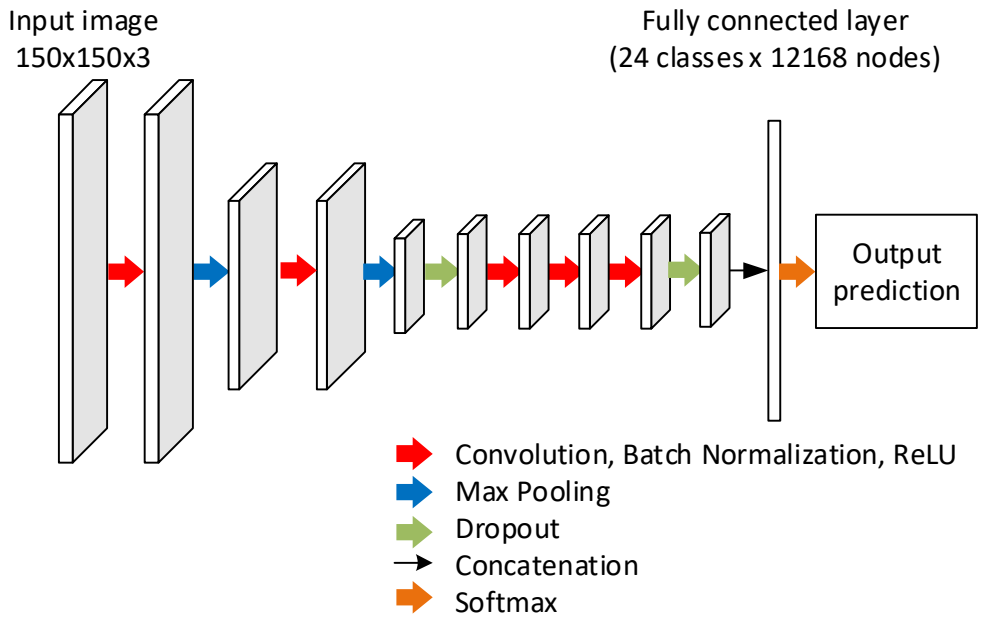

Fig. 6. CNN architecture

TABLE I
RECOGNITION RESULTS

Method	Recognition rate
HOG-SVM	97.49%
LBP-SVM	98.23%
HOG-LBP-SVM	98.36%
CNN	97.08%
CNN-SVM	98.30%

Table 1 compares the performance of five schemes. HOG and LBP as standalone features have acceptable recognition rates (97.49% and 98.23% respectively). The combination of the two features achieves the highest result (3rd scheme, 98.36%). Also, the performance rate of CNN-SVM (98.30%) proves that using CNN as a feature extractor is beneficial. Because CNN-SVM has the minimum classifying duration (much lower than HOG-LBP-SVM) and the model was developed to partially counter the effect of overfitting, it is the best scheme among the considered methods.

IV. CONCLUSION

In this paper, we proposed three models for hand gesture alphabetical recognition. In the first model, the combination of HOG, LBP features and multi-kernel multi-class SVM to are utilized to maximize the distinctive properties of each feature type. In particular, the feature-kernel pairs: HOG and RBF, LBP and Polynomial have the average recognition rate higher than two methods that use only one feature (98.36%). The results from the CNN and CNN-SVM models (97.08% and 98.30%) prove that by implementing CNN as a standalone feature extractor, better result could be obtained than using an end-to-end CNN architecture. Although the validation accuracy of the CNN-SVM model is lower than that of HOG-LBP-SVM model, it has a better chance to counter overfitting.

REFERENCES

[1] P. Premaratne, *Historical Development of Hand Gesture Recognition*, 1st ed. Singapore: Springer, March 2014.

[2] Y. Qiao, Z. Feng, X. Zhou, and X. Yang, "Principle component analysis based hand gesture recognition for Android phone using area features," in *the 2nd International Conference on Multimedia and Image Processing (ICMIP)*, March 2017, pp. 108–112.

[3] T. Y. Pan, L. Y. Lo, C. W. Yeh, J. W. Li, H. T. Liu, and M. C. Hu, "Real-time sign language recognition in complex background scene based on a hierarchical clustering classification method," in *the Second International Conference on Multimedia Big Data (BigMM)*, 2017, pp. 64–67.

[4] H. Zuo, H. Fan, E. Blasch, and H. Ling, "Combining convolutional and recurrent neural networks for human skin detection," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 289–293, March 2017.

[5] M. M. Islam, S. Siddiqua, and J. Afnan, "Real time hand gesture recognition using different algorithms based on American Sign Language," in *the IEEE International Conference on Imaging, Vision Pattern Recognition*, Feb 2017, pp. 1–6.

[6] S. Saha, R. Lahiri, A. Konar, and A. K. Nagar, "A novel approach to american sign language recognition using madaline neural network," in *the IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016, pp. 1–6.

[7] X. Liu, C. Li, and L. Tian, "Hand gesture recognition based on wavelet invariant moments," in *the IEEE International Symposium on Multimedia (ISM)*, Dec 2017, pp. 459–464.

[8] T. N. T. Huong, T. V. Huu, T. L. Xuan, and S. V. Van, "Static hand gesture recognition for Vietnamese sign language (VSL) using principle components analysis," in *the International Conference on Communications, Management and Telecommunications (ComManTel)*, Dec 2015, pp. 138–141.

[9] P. Q. Thang, N. T. Thuy, and H. T. Lam, "The svm, simpsvm and rvm on sign language recognition problem," in *the Seventh International Conference on Information Science and Technology (ICIST)*, April 2017, pp. 398–403.

[10] M. Hu, F. Shen, and J. Zhao, "Hidden Markov models based dynamic hand gesture recognition with incremental learning method," in *the International Joint Conference on Neural Networks (IJCNN)*, July 2014, pp. 3108–3115.

[11] N. Otberdout, L. Ballihi, and D. Aboutajdine, "Hand pose estimation based on deep learning depth map for hand gesture recognition," in *the Intelligent Systems and Computer Vision (ISCV)*, 2017, pp. 1–8.

[12] X. Fang and X. Lei, "Hand pose estimation on hybrid CNN-AE model," in *the IEEE International Conference on Information and Automation (ICIA)*, 2017, pp. 1018–1022.

[13] X. Deng, Y. Zhang, S. Yang, P. Tan, L. Chang, Y. Yuan, and H. Wang, "Joint hand detection and rotation estimation using CNN," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1018–1022, 2017.

[14] A. L. C. Barczak, N. H. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "A new 2D static hand gesture colour image dataset for ASL gestures," *Research Letters in the Information and Mathematical Sciences*, vol. 15, pp. 12–20, 2011.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[16] A. Hadid, "The local binary pattern approach and its applications to face analysis," in *the First Workshops on Image Processing Theory, Tools and Applications*, Nov 2008, pp. 1–9.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[18] Y. Tang, "Deep learning using linear support vector machines," 2015.