Handwritten Uyghur Character Recognition Using Convolutional Neural Networks

Wujiahemaiti Simayi, Mayire Ibrayim, and Askar Hamdulla(✉)

Institute of Information Science and Engineering, Xinjiang University, Urumqi, China
askar@xju.edu.cn

Abstract. Handwritten Uyghur character recognition researches up to date have been based on traditional pattern recognition techniques that highly relies on handcrafted features. The similarity between character forms has been hindering the extraction of robust features. This paper proposed the deep learning based self-learned features to recognize 128 handwritten Uyghur characters forms. The first-hand online handwritten trajectory is first preprocessed and converted to a centralized binary image as input to the implemented deep neural network model. In experiments, the convolutional neural network models with 4, 5 and 8 convolutional layers are studied to get higher recognition accuracy. All models are trained implementing the same dropout regularization. The models 8 convolutional layers on 48 * 48 converted character images produced as high as 94.65% average accuracy on a test set of 10,240 handwritten character samples.

Keywords: Handwritten character recognition · Uyghur character forms · Size adjusting · Convolutional neural network

1 Introduction

Handwritten character recognition is one of the most typical implementations of pattern recognition [1]. Handwriting recognition has two main branches including online and offline handwriting recognition [2]. Recognizing the traced pen-tip trajectory is called online handwriting recognition, whereas offline handwriting recognition categorizes the already formed handwritten shape images. Online handwritten samples contain exact pen-tip trajectory with spatial and temporal information. A same handwritten object is perhaps formed in various sizes, orders and angles. Violation to the standard writing rules occurs very often in the handwriting process. For example, a stroke at the beginning part of a shape probably will be written at last, or a stroke is perhaps started without completing the previous one etc. These kinds of randomness decrease the advantage of temporal information [3–5].

Handwritten Uyghur character recognition studies have been based on traditional pattern recognition methods. This paper presents the first application of convolutional neural networks to recognize 128 Uyghur character forms. Different from the traditional methods, features are automatically learned and extracted during the model training [6].

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 370, pp. 613–624, 2021.
https://doi.org/10.1007/978-3-030-72795-6_49

The remaining content of the paper is given in the following sections. Section 2 gives a brief introduction to Uyghur characters and some related studies. Preprocessing techniques applied on the raw online samples before feeding them to the model is specified in Sect. 3. Section 4 introduces the convolutional neural network architecture implemented in this paper. The model training process and the experiment results are provided and discussed in Sect. 5. Finally, Sect. 6 draws conclusion.

2 Related Work on Uyghur Character Recognition

2.1 Uyghur Characters

Table 1. Uyghur character forms

End	Mid	Begin	Single	Rep	No.	End	Mid	Begin	Single	Rep	No.
ـڭ	ـڭـ	ڭـ	ڭ	ڭ	20	ـئا			ئا		
ـل	ـلـ	لـ	ل	ل	21	ـا			ا	ا	1
ـم	ـمـ	مـ	م	م	22	ـئە			ئە		
ـن	ـنـ	نـ	ن	ن	23	ـە			ە	ە	2
ـھ	ـھـ	ھـ	ھ	ھ	24	ـب	ـبـ	بـ	ب	ب	3
ـئو			ئو			ـپ	ـپـ	پـ	پ	پ	4
ـو			و	و	25	ـت	ـتـ	تـ	ت	ت	5
ـئۇ			ئۇ			ـج	ـجـ	جـ	ج	ج	6
ـۇ			ۇ	ۇ	26	ـچ	ـچـ	چـ	چ	چ	7
ـئۆ			ئۆ			ـخ	ـخـ	خـ	خ	خ	8
ـۆ			ۆ	ۆ	27	ـد			د	د	9
ـئۈ			ئۈ			ـر			ر	ر	10
ـۈ			ۈ	ۈ	28	ـز			ز	ز	11
ـۋ			ۋ	ۋ	29	ـژ			ژ	ژ	12
ـئې	ـئېـ	ئېـ	ئې			ـس	ـسـ	سـ	س	س	13
ـې	ـېـ	ېـ	ې	ې	30	ـش	ـشـ	شـ	ش	ش	14
ـئى	ـئىـ	ئىـ	ئى			ـغ	ـغـ	غـ	غ	غ	15
ـى	ـىـ	ىـ	ى	ى	31	ـف	ـفـ	فـ	ف	ف	16
ـي	ـيـ	يـ	ي	ي	32	ـق	ـقـ	قـ	ق	ق	17
	ـئـ	ئـ		ئ	33	ـك	ـكـ	كـ	ك	ك	18
ـلا			لا	لا	34	ـگ	ـگـ	گـ	گ	گ	19

Modern Uyghur script inherits alphabetic writing style that regular combination of characters generates words and further sentences according to the morphological and lexical rules. It is written from right to left and from top to down orientation. Table 1 provides the total Uyghur character forms [9].

As given in Table 1, there are 32 basic Uyghur characters containing 8 vowels (char.No.1–2, char.No.25–28, char.No.30–31) and 24 consonants. There are total 126 writing forms of 32 basic Uyghur characters. Each character has different writing forms according to character positions within a word or sub-word, such as initial, intermediate, ending and isolated forms. Furthermore, A special component character (char.No.33) and a compound character (char.No.34) are frequently used in both printing and handwriting. Each of them has two writing forms according to the position appeared in a word. To sum up, we have to handle with 130-character forms for handwritten Uyghur character recognition.

However, the isolated and ending forms of char.No.24 are very much similar to its beginning and intermediate forms respectively, not only in handwriting but also in printing fonts. Therefore, only the isolated and ending forms ھand ـھare considered during character data collection for char24. The neglected character forms are labeled red in Table 1. Therefore, actually collected handwritten character forms are 128 in this paper.

2.2 Previous Studies

Preliminary researches on Uyghur handwritten character recognition investigated the handwriting characteristics of Uyghur characters and proposed different recognition methods. Different feature extraction methods have been presented according to the structural and statistical property of handwritten character forms [6]. Due to the high similarity between the Uyghur character forms which are hardly differentiated without context information, some studies were conducted only on the isolated character forms. However, recognizing the total 128 character forms has great significance in implementations. In this paper, we only review the character recognition studies on the total 128 character forms.

Coded 8-directional features from divided 16 grids of handwritten character forms were applied to recognize 128 character forms in [7]. Template matching based on Euclidean distance was adopted using minimum distance classifier. 51200 online character samples were split into train and test sets with 1:1 ratio in the conducted experiments. Experiments recorded modest recognition result of 60% average accuracy.

Center distance feature proposed by [8] produced character recognition accuracy to 80% using minimum distance classifier. The train and test sets are arranged with 7:3 ratio, using dataset size of 51200 samples. Later on, average recognition accuracy was improved to 87% by modifying feature extraction method.

MQDF classifier on NCFE features made even better recognition results on 128 character forms. 89.08% average recognition accuracy were recorded using combination of different features [9]. The same dataset used in [7] and [8] was used in experiments and samples are split into train and test set with ratio of 7:3, too. Using the same dataset with the same splitting into train and test sets, the experiment results in [9] fairly demonstrates the superiority of the MQDF with NCFE feature for Uyghur handwritten character recognition.

Some comparative character recognition experiments using minimum distance classifier, MQDF and BP neural networks were conducted on the online character samples from 115 writers [10]. Online character samples from 60 writers were used for training while the remained 55 writers are arranged for testing. Directional chain codes were extracted as extracted feature for each of the Minimum distance classifier, MQDF and BP models. 83.36%, 87.77% and 84.73% average recognition accuracies are obtained for the compared classifiers, respectively. Combination of the classifiers gave further improved results of 90.88% accuracy, later.

Time division directional feature with weighted Naïve Bayes algorithm has received 93.15% average recognition accuracy [11]. The dataset used in experiments were collected from 102 writers who contributed 13056 online character samples. Among them, samples from 60 writers were arranged for training, samples from 42 writers were used for testing.

3 Preprocessing

Besides unique handwriting styles of each writer, the first-hand collected handwritten samples always vary in size, orientation as well as the location on the handwritten tablet screen. Preprocessing improves the performance of the recognition system if well manipulated [12]. Some preprocessing operations including point insertion and size adjustment are incorporated to make the binary images to be the model input.

3.1 Point Insertion

In order to avoid generating extra points between strokes, this paper applies stroke-level point-insertion, which means the point insertion operation is applied only within a stroke and not between the strokes. Each neighbor point couples are checked to know if they need point insertion, by calculating Euclidian distance between them. Insertion is made between the neighbor points which have larger distance than a threshold value which is set 1 in this paper.

$$dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

$$x_i = x_1 + \frac{\Delta x}{N},\ y_i = y_1 + \frac{\Delta y}{N} \tag{2}$$

$P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ are the neighbor points that need point insertion between them. Δx and Δy are the horizontal and vertical distances calculated between the neighbor points. Euclidian distance between them is obtained using Eq. (1). The integer value of the distance is used as the number of points to be inserted, denoted as N. Coordinates of each insertion point is determined according to Eq. (2) where (x_i, y_i) are the coordinate values of the i th insertion point.

3.2 Coordinate Range Normalization

The original coordinate range refers to the size of screen window that handwriting is recorded. The ranges of horizontal and vertical axis coordinates of the collected raw online samples are [1, 255] in this paper. The original coordinate range is always much larger than needed in most cases. Normalizing original sample to smaller coordinate range will save time and storage for later operations. Coordinate values of points are adapted to new coordinate range using simple linear normalization as in Eq. (3).

$$x = \frac{w}{W} * X \ , \ y = \frac{h}{H} * Y \tag{3}$$

where (X, Y) are the original coordinate values in the original interval or screen size (W, H), while (x, y) are the normalized coordinate values in new interval or the normalized widow size (x, y).

3.3 Size Adjusting

There are many quite similar characters forms in Uyghur. Direct implementation of the common linear normalization methods easily loses overall shape information of original samples and influences the classification accuracy. To keep original handwritten character forms unchanged in the generated images, the handwritten trajectories are adjusted to the normalized coordinate ranges by the following steps:

a. Calculate the width ratio and height ratio between the handwritten shape and the normalized window (coordinate ranges) by Eqs. (4)–(5).

$$W_{ratio} = \frac{w}{\max(XX) - \min(XX)} \tag{4}$$

$$H_{ratio} = \frac{h}{\max(YY) - \min(YY)} \tag{5}$$

w and h are the width and height of the imaginary window of the normalized coordinate ranges.

b. Adjust sample shape to the window by linear size normalization using adjust ratio obtained by Eq. (6), Sample shape is first moved to the coordinate origin prior to multiplying the adjust ratio, so that adjusted shape will still be in the normalized coordinate range, as Eqs. (7)–(8).

$$adjust_{ratio} = \min(W_{ratio}, H_{ratio}) \tag{6}$$

$$x = (X - \min(XX)) * adjust_{ratio} \tag{7}$$

$$y = (y - \min(YY)) * adjust_{ratio} \tag{8}$$

Where (X, Y) and (x, y) are the point coordinate values before and after size adjusting. Thus, the original sample shape will be kept as before after size adjusting and make good use the space in the normalized window size (coordinate ranges).

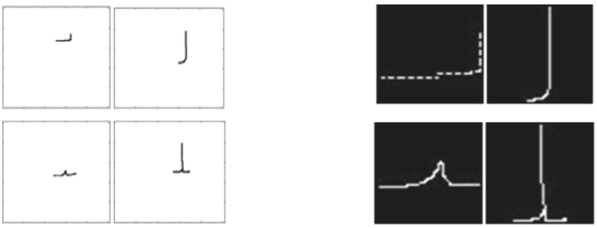
c. Centralize the sample trajectory to the normalized coordinate ranges using Eqs. (9)–(10).

$$x = X - (\frac{\max(XX) - \min(XX)}{2} - \frac{w}{2}) \tag{9}$$

$$y = Y - (\frac{\max(YY) - \min(YY)}{2} - \frac{h}{2}) \tag{10}$$

where $(\frac{\max(XX) - \min(XX)}{2}, \frac{\max(XX) - \min(XX)}{2})$ is the center of the sample shape, $(\frac{w}{2}, \frac{h}{2})$ the window center of the normalized coordinate ranges.

d. Convert the online trajectory to image. Since each point coordinate is clear to us, it is convenient to make binary images just marking black or white pixels at the corresponding positions in a matrix or window. Concerning the visibility for observation, background of the character sample images is set black while foreground is set white. The character forms shown in Fig. 1(a) are ones which easily lose their original shapes by careless use of linear size normalization. Implementing above described size adjusting kept the original shape of the samples, as shown in Fig. 1(b).



(a) samples in 256*256 (b) after size adjusting

Fig. 1. Binary images of character forms after preprocessing

4 Convolutional Neural Network Model

4.1 Convolutional Neural Networks-CNN

Convolutional Neural Networks-CNN is one the most popular deep neural network architectures and has been tremendously successful in practical applications [13]. CNN is given the first priority to tackle the pattern recognition problems with fixed number of classes. CNN has been providing very high accuracies for many computer vision tasks including handwriting recognition [14]. The main architecture of CNN consists convolution layer, pooling layer, fully connected layer and output layer for classification.

4.2 Model Architecture and Classification

We conducted some preliminary experiments using CNN models with 4, 5 and 8 layers to get higher recognition accuracy. Figure 2 shows the architecture of the convolutional networks with 8 convolutional layers, which produced highest recognition accuracy in this paper. It is consisted of 8 convolution layers and 4 pooling layers and 2 fully connected layer. Softmax layer with 128 neurons is fed with the flattened feature vector to perform classification.

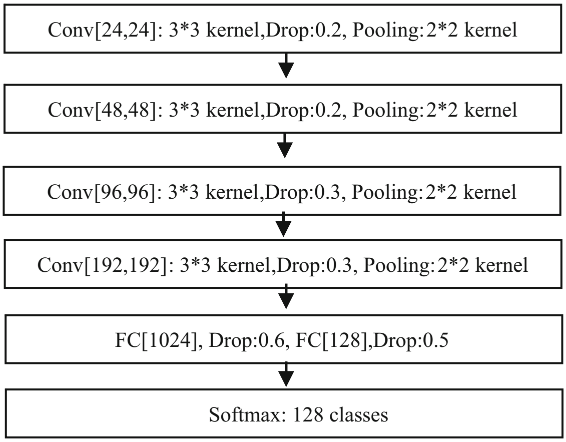


Fig. 2. The implemented CNN network structures with 8 conv layers (The number of neuron in a layer is given in brackets, for examples [24,24] means two successive layers with 24 and 24 neurouns, respectively)

All convolutional layers armed with ReLU and filter kernel size of 3 * 3 in our experiments. Lower layers of network are assumed to extract lower degree of feature-local feature, and higher layers correspond for higher level feature or global features. Keeping the image size same after convolution operation is an admirable effect of zero padding. Image or feature map size is kept same using zero padding at each convolution layer. Image size is decreased to its half via 2 * 2 max pooling, with stride of 2. All models implemented a few Fully connected layers to form more generalized features before softmax classification.

Softmax is one of the privileged classifiers with deep neural networks. It can be directly integrated to previous layers and produce a very good explainable classification results using Eq. (11).

$$P(y = i/x) = \frac{e^{x^T \cdot w_i}}{\sum_{k=1}^{K} e^{x^T \cdot w_i}} \tag{11}$$

With network parameters W, the normalized probability of the an input x to be class i in total K classes, the result at each output node reflects the normalized probability $P(y = i/x)$ of the input to be the corresponding classes. The number of output nodes is set by the number of classes and the node with the largest belonging probability is chosen to be the classification result y.

5 Experiments and Discussion

5.1 Dataset and Configuration

The experiments in this paper are based on a dataset of 51200 online handwritten character samples collected from 400 writers. Samples are stored in writer specific manner. 80% of the dataset, 40960 samples from 320 writers, is used for training. The remained 10240 samples from 80 writers, 20% of total samples, are used for testing. Randomly selected 10% portion of the train set is used as validation set. The online samples are converted to binary images, see Sect. 3, as input to CNN model.

In experiments, each model is trained using stochastic minibatch training algorithm with minibatch size of 128. Adadelta optimization algorithm for model training, with initial learning rate of 1. Dropout is applied to avoid overfitting. Training is stopped when no improvement on the validation set is seen in 10 continues epochs. The model is evaluated using the test set after the training stopped. The recognition performance of the models are evaluated using Character Error Rate in the experiments. Generality of model is reflected by the difference between the last training error and the test error.

5.2 Experiment Results

Table 2 gives the experiment records on the 48 * 48 character images with centralized 44 * 44 character forms. CNN model architectures with 4, 5 and 8 convolutional layers are compared.

Table 2. CNN training on 48 * 48 images

Num of Conv layers	Network architecture	Model size	#Epoch	Last train error	Test error
4 Conv	[C24-P2-d0.2]-[C48-P2-d0.2]-[C96-p2-D0.3]-[C192-P2-D0.3]-[FC256-D0.4]-[FC256-D0.4-FC128]-Softmax128	8.99M	45	4.28%	5.86%
5 Conv	[C24-P2-d0.2]-[C48-P2-d0.2]-[C96-p2-D0.3]-[C192-P2-D0.3]-[C384-p2-D0.4]-[FC256-D0.5]-[FC256-D0.5-FC128]-Softmax128	12.26M	68	4.36%	5.79%

(*continued*)

Table 2. (*continued*)

Num of Conv layers	Network architecture	Model size	#Epoch	Last train error	Test error
8 conv	[C24-C24-P2-d0.2]-[C48-C48-P2-d0.2]-[C96-C96-p2-D0.3]-[C192-C192-P2-D0.5]-[FC256-D0.5]-[FC256-D0.5-FC128]-Softmax128	14.1M	44	2.92%	5.35%

FC[24,24] means two successive convolutional layers with 24 and 24 neurouns, respectively. The implemented drop rate is noted D with a specific figure. P2 means the max pooling operation with kernel size of 2 * 2 and stride 2.

Recognition results of the CNN training experiments conducted in this paper are impressive that 4 and 5 conv layered CNN on 48 * 48 images reached as low as 5.86%, 5.79% average recognition error on 10240 test samples, which are equivalent to 94.14%, and 94.21% character recognition accuracy, respectively. Nevertheless, the training was not quite well, because the model performance on train set was not good enough due to the high drop rates implemented on layers.

Then, a deeper network which has 8 convolutional layers showed better training on train set than the other models in Table 2. The test error also reached to 5.35% which equals to 94.65% average accuracy. It is known that training neural networks requires many times of attempts and careful tuning. It is believed that the models used in this paper can be trained even better and get higher recognition accuracy.

5.3 Error Analysis

Most of the characters are recognized in high precision except only a few character forms. Some Uyghur character forms are quite simple in structure. However, the structural simplicity introduces higher similarity between different character forms and hinders to train a good classifier. The typical character forms of low precision are examined in more detail to find the most confused ones, as shown in Table 3 and Table 4.

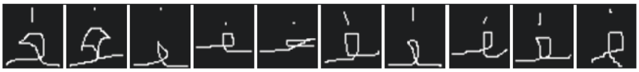
The character samples illustrated in Table 4 are very hard to make true differentiation from a confused couple or some other similar character forms. Human eye perhaps cannot come up with true answer to most of the samples shown in Table 4, such as in No. 1–4. Any careless handwriting makes character shapes which does not belong to the expected character class. For example, most of the character samples in No. 5 of Table are already moved to another character class (the predicted one) just by a tiny difference or ambiguity at the stroke end.

The handwritten Uyghur character recognition research has been conducting on self-collected datasets which are different in total volume, number of writers and sample qualification. It is unfair to compare the experiment results from different databases. Therefore, we only refer the results reported using the same dataset to compare, as shown in Table 5. We would like to share out dataset for research activities and will make it public, soon.

Table 3. Easily confused character forms

Char (label)	char	Total misclassified	precision	Most confused char(label)	char	Ratio in misclassied
51	غـ	34	57%	55	فـ	0.85
55	فـ	26	68%	51	غـ	0.79
65	ـگ	19	76%	66	گ	1
106	ې	20	75%	111	ي	0.91
111	ي	27	66%	106	ې	0.95
119	ى	22	72%	114	ى	0.83
124	ي	36	55%	121	ي	1

Table 4. Some misclassified character samples

No.	Truth	Pre-diction	Some misclassified character samples
1	غـ	فـ	
2	فـ	غـ	
3	ې	ي	
4	ى	ى	
5	گ	ـگ	

For fair comparison, the dataset is rearranged into train and test set with ratio of 7:3, like to the works in [12, 13]. The network with 8 convolutional layers is trained and tested on the new split dataset. Table 5 compares the recognition results on 128 Uyghur character forms using same dataset. The convolutional networks implemented in this paper achieved higher recognition accuracy than the previous reported results on this dataset.

Table 5. Comparison of the recognition results using the same data

Ref.	Ratio	Train set	Test set	Feature	Classifier	Accuracy
[12, 20]	7:3	280 writers, 280 * 128 samples	120 writers, 120 * 128 samples	Center distance feature+other features	Nearest neigbor	80%
[12, 20]	7:3	280 writers, 280 * 128 samples	120 writers, 120 * 128 samples	Modified center distance feature+other features	Nearest neigbor	87%
[13]	7:3	280 writers, 280 * 128 samples	120 writers, 120 * 128 samples	NCFE-8 direction +other features	MQDF	89.08%
This paper	7:3	280 writers, 280 * 128 samples	120 writers, 120 * 128 samples	Learned features	CNN-Softmax	**94.03%**
This paper	4:1	320 writers, 320 * 128 samples	80 writers, 80 * 128 samples	Learned features	CNN-Softmax	**94.65%**

6 Conclusion

This paper studied the Implementation of Deep learning on handwritten Uyghur character recognition for the first time. One of the most welcoming deep neural network architecture-CNN is applied using converted binary images of online handwritten samples as input. CNN models with 4, 5 and 8 convolutional layers on image size 48 * 48 gave 94.17%, 94.21% and 94.65% average recognition accuracy on the test set of 10240 samples. For fair comparison with the previous reported results, the dataset is re-split into train and test sets with the same splitting used in literature. The best performed CNN model in our paper achieved higher recognition accuracy than other reported results on the dataset. The character forms with low recognition precision are investigated in more detail. Further improving the handwritten Uyghur character recognition accuracy will be the main content in next work.

Acknowledgment. This work is supported by National Key Research and Development Plan of China (2017YFC0820603), Natural Science Youth Foundation of Universities in Autonomous Region (No. XJEDU2019Y007) and Natural Science Foundation of Xinjiang (No. 2020D01C045). The first author is very much grateful to the National Laboratory of Pattern Recognition of CASIA for providing the experimental environment.

References

1. Gao, Y., Jin, L., He, C., Zhou, G.: Handwriting character recognition as a service: a new handwriting recognition system based on cloud computing. In: 2011 International Conference on Document Analysis and Recognition, Beijing, pp. 885–889 (2011)
2. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: Online and offline handwritten Chinese character recognition: benchmarking on new databases. Pattern Recognit. **46**(1), 155–162 (2013)
3. Okamoto, M., Yamamoto, K.: Online handwriting character recognition method using directional and direction-change features. Int. J. Pattern Recognit. Artif. Intell. **13**(7), 1041–1059 (1999)

4. Alom, M.Z., Sidike, P., Hasan, M., et al.: Handwritten Bangla character recognition using the state-of-art deep convolutional neural networks. Comput. Intell. Neurosci. **2018**, 1–3 (2017)
5. Kurban, A., Mamat, H.: Beida FangZheng Uighur text to unicode text code codeconversion. J. Xinjiang Univ. (Nat. Sci. Ed.) **23**(3), 343–347 (2006)
6. Simayi, W., Ibrayim, M., Tursun, D., et al.: Survey on the features for recognition of on-line handwritten Uyghur characters. Int. J. Signal Process. Image Process. Pattern Recognit. **8**(3), 850–853 (2015)
7. Dawut, R.: Research on the key technologies of online handwritten Uyghur word recognition, M.S. thesis, Xinjiang University (2011)
8. Simayi, W.: Online Uyghur handwritten character recognition technology based on center distance feature, MS thesis, Xinjiang University (2014)
9. Ibrayim, M.: Key technologies for recognition of on-line handwritten Uyghur characters and words. Ph.D. thesis, Wuhan University (2013)
10. Dai, X.L.: Online handwritten Uyghur character and word recognition based on the mobile platform. MS thesis, Xidian University (2012)
11. Xu, Y.M.: A study of key techniques for Uighur handwriting recognition. Ph.D. dissertation, Xidian University (2014)
12. Liu, C.L.: Normalization-cooperated gradient feature extraction for handwritten character recognition. IEEE Trans. Pattern Anal. Mach. Intell. **29**(8), 1465 (2007)
13. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105. Curran Associates Inc. (2012)