Neural Network-based Efficient Measurement Method on Upside Down Orientation of a Digital Document

Yeji Shin, Youngone Cho, Hye Wook Kang, Jin-Ga Kang, Jin-Woo Jung

Department of Computer Science and Engineering, Dongguk University, 0620, Seoul, Korea

Abstract

As many digital documents are required in various environments, paper documents are digitized by scanner, fax, digital camera and specific software. In the case of a scanned document, we need to check whether the document is right sided or upside down because a person could put the paper upside down or take high computational burden even though they can recognize the document orientation in special case of upside down documents. The proposed artificial neural network method shows a high accuracy in time efficiency and high recognition accuracy for each word obtained through the OCR. In this paper, the proposed method is to handle documents immediately. For the small amounts of ascenders and descenders, these methods rely on counting the number of ascenders and descenders in the text to make decision on its orientation. Specifically, they defined coordinate value of the HOCR file with Beautifulsoup and parsed the 'bbox' of the OCR file with Beautifulsoup. In OCR-based algorithm, Tesseract OCR and ASIQ OCR are usually used to determine whether the document is upside down or right sided. The proposed method is a better and cost-efficient method compared with the previous OCR-based method.
2. OCR (Optical Character Recognition)-based measurement method

Let us assume there are n-lines in an image and each line consists of k-words. The W of each word is defined as the summation of minimum and maximum coordinate values (min.x, min.y, max.x, max.y) for each word obtained through the bbox tag of hOCR. L_j is defined as the summation of W_i for every words in each line.

\[ W_i = \text{min.x} + \text{min.y} + \text{max.x} + \text{max.y} \text{ for each word i} \]  \hspace{1cm} (1)

\[ L_j = W_1 + W_2 + \ldots + W_k \text{ for each line j} \]  \hspace{1cm} (2)

The document is determined as right sided if the value of L_j increases, but upside down if it decreases. The following is the pseudo code for this algorithm (Figure 1).

\[ \text{Input:} \]
\[ O \leftarrow \text{Set of W_i of all words line by line} \]
\[ \text{Output:} \]
\[ \text{(true: right sided / false: Upside down)} \leftarrow \text{state} \]

\[ \text{Initialization:} \]
\[ \text{state} \leftarrow \text{false} \]
\[ \text{increase} \leftarrow 0 \]
\[ \text{decrease} \leftarrow 0 \]

\[ \text{Begin} \]
\[ \text{for } i \text{ in len(lines) do} \]
\[ \text{sum} \leftarrow O[i][1] + O[i][2] + O[i][3] + O[i][4] \]
\[ \text{line_sum}[i] \leftarrow \text{sum} \]
\[ \text{for } i \text{ in len(line_sum) do} \]
\[ \text{if } \text{line_sum}[i] \leq \text{line_sum}[i+1] \text{ then} \]
\[ \text{increase} \leftarrow \text{increase} + 1 \]
\[ \text{else} \]
\[ \text{decrease} \leftarrow \text{decrease} + 1 \]
\[ \text{end if} \]
\[ \text{if } \text{increase} \geq \text{decrease} \text{ then} \]
\[ \text{state} \leftarrow \text{true} \]
\[ \text{end if} \]
\[ \text{End} \]

Figure 1. The pseudo code of OCR-based measurement algorithm

3. Artificial neural network (ANN)-based measurement method

3.1. Basic Idea

Many documents are written from left to right and top to bottom direction although titles, pictures, charts and so on may make exceptions. When the document can be separated as areas as shown in Figure 2, the ratio of the upper half non-background area to the lower half non-background area (defined as Top-to-bottom ratio), and the ratio of the left half non-background area to the right half non-background area (defined as Left-to-right ratio) may be able to distinguish whether the document is upside down or right sided. Actually, the Top-to-bottom ratio and the Left-to-right ratio of three different types of documents are calculated as shown in Table 1. And for these documents (not for all documents), when Top-to-bottom ratio is greater than 1 or the Left-to-right ratio is greater than 1, we can see that the orientation is right sided.

![Figure 2. Sketch of basic idea](image)

Table 1. Top-to-bottom ratios and Left-to-right ratios of three document samples

| Document | State         | Top-to-bottom ratio | Left-to-right ratio |
|----------|---------------|---------------------|---------------------|
| Doc. 1   | right sided   | 9.282392            | 7.849178            |
|          | upside down   | 0.107731            | 0.127402            |
| Doc. 2   | right sided   | 3.941437            | 1.16777             |
|          | upside down   | 0.253715            | 0.856333            |
| Doc. 3   | right sided   | 1.253945            | 1.446367            |
|          | upside down   | 0.797483            | 0.691387            |

However, experiments with 50 more documents with 10 different types have shown that if Top-to-bottom ratio and the Left-to-right ratio are less than 2, the distribution was scattered and not linearly separable as shown in Figure 3. So we have designed an algorithm based on ANN that can effectively and efficiently distinguish the upside down documents as shown in Section 3.2.

![Figure 3. Feature plot for upside down and right sided state](image)
3.2. Proposed algorithm

Artificial neural network (ANN)-based measurement method, it calculates the ratio of the upper half non-background area and the lower half non-background area. Similarly, it also calculates the ratio of the left half non-background area and the right half non-background area. These are used as feature values for training ANN to classify the upside down or not. And the recognition process of ANN gives the status of upside down automatically.

3.2.1 Feature extraction

We assumed that the characters are black and the algorithm for feature extraction is as follows:

Step1. Read the RGB color values of pixels from a scanned BMP image file for the given digital document.

Step2. Transform to the grayscale image and do thresholding with the threshold value, 33.

Step3. Calculate the number of black pixels at the upper half non-background area and the number of black pixels at the lower half non-background area. Subsequently, divide the number of black pixels in the top part by the number of black pixels in the bottom part like the below Eq. (3)

Top-to-bottom ratio = \frac{\text{Number of black pixels at the top}}{\text{Number of black pixels at the bottom}} \ldots (3)

Step4. Calculate the number of black pixels on the left half non-background area to the number of black pixels on the right half non-background area. Thereafter, the number of black pixels on the left is divided by the number of black pixels on the right like this Eq. (4)

Left-to-right ratio = \frac{\text{Number of black pixels on the left}}{\text{Number of black pixels on the Right}} \ldots (4)

Step5. Extract the input feature data for artificial neural network with the form of vector (‘Top-to-bottom ratio’, ‘Left-to-right ratio’, ‘right sided’ / ‘upside down’). 

3.2.2 Artificial neural network (ANN)

We have got the datasets of Top-to-bottom ratios and Left-to-right ratios from the section 3.2.1. To find appropriate structure of ANN, we tried 9 different structures by varying the number of hidden layers with 1, 2, 3 layers and the number of hidden nodes with 1, 3, 5 nodes as shown in table 2. By the results in Table 2, we decided the structure of ANN. The number of hidden layers is 1 and the number of hidden nodes are 5, whose structure shows the lowest error rate.

Table 2: Error rates of various ANN structures

| # of hidden nodes | 1 layer | 2 layers | 3 layers |
|-------------------|---------|----------|----------|
| 1 node            | 8.00    | 5.91     | 5.73     |
| 3 nodes           | 6.22    | 5.41     | 5.58     |
| 5 nodes           | 4.73    | 4.99     | 4.73     |

The artificial neural network system consisting of one hidden layer with five nodes calculates as in Figure 4.

Figure 4. Proposed ANN Structure

Figure 5 shows the min_thresh value of neuralnet function in R neuralnet library [13] according to the number of iterations to show the convergence of learning error. During the initial period up to 7,000 iterations, the min_thresh value decreases rapidly but the value becomes almost constant from 30,000 iterations.

Figure 5. Error plot of ANN

The final decision can be derived by applying the input feature data to the trained artificial neural network. The trained ANN will decide whether the given document is right sided or upside down. The following Figure 6 explains the pseudo code of the trained ANN-based measurement method.

4. Convolutional neural network (CNN)-based measurement method

To extract the pattern information of pixels that are laid out in the document, we have used convolutional neural network (CNN) because it has shown a high performance in understanding images [14, 15]. The used CNN model in this paper has four hidden layers with different number of nodes for each corresponding layer as shown in Table 3 which showed superior results compared to other models. Activation function and 2D max_pooling is applied followed by the first hidden layer. Then the output nodes of first hidden layer are fed into the second hidden layer. After applying
activation function and 2D max_pooling, the output is flattened and an activation function is applied for linear operation. Then in the third hidden layer, drop_out is done to avoid overfitting and to get rid of 30% of the total number of nodes for generalization. After that, the last hidden layer with 1026 nodes are fed forward to the output layer.

\[
\text{Input :}
\begin{align*}
\text{TB} & \leftarrow \text{Top-to-bottom Ratio} \\
\text{LR} & \leftarrow \text{Left-to-right Ratio}
\end{align*}
\]

\[
\text{Output :}
\begin{align*}
\text{true: right sided} & \mid \text{false: Upside down} \\
& \leftarrow \text{Result}
\end{align*}
\]

\[
\text{Begin}
\begin{align*}
a & \leftarrow \text{sigmoid}(87.01576 - 183.38162*TB - 115.22165*LR) \\
b & \leftarrow \text{sigmoid}(0.04204546 + 0.52479163*TB + 0.63695455*LR) \\
c & \leftarrow \text{sigmoid}(-45.567399 + 5.410532*TB - 25.861325*LR) \\
d & \leftarrow \text{sigmoid}(-0.3730291 - 90.8726276*TB - 19.0923138*LR) \\
e & \leftarrow \text{sigmoid}(0.9640014 + 164.1678818*TB + 157.2681645*LR)
\end{align*}
\]

\[
\text{state} \leftarrow (1.144237 - 1.047634*a - 2.363419*b + 1.014042*c - 1.92740*0 + 2.205141*e)
\]

\[
\text{if state > 1.4 then} \\
\text{Result} \leftarrow \text{true}
\]

\[
\text{else} \\
\text{Result} \leftarrow \text{false}
\]

\[
\text{end if}
\]

\[
\text{End}
\]

Figure 6. The pseudo code of the trained ANN-based measurement Algorithm

Table 3. Used CNN structure

| Layer Type       | Output Shape | # of Params |
|------------------|--------------|-------------|
| conv2d_1 (Conv2D) | (None, 60, 60, 32) | 832         |
| activation_1 (Activation) | (None, 60, 60, 32) | 0           |
| max_pooling2d_1 (MaxPooling2) | (None, 30, 30, 32) | 0           |
| conv2d_2 (Conv2D) | (None, 30, 64, 30) | 51264       |
| activation_2 (Activation) | (None, 30, 64, 30) | 0           |
| max_pooling2d_2 (MaxPooling2) | (None, 15, 64, 15) | 0           |
| flatten_1 (Flatten) | (None, 14400) | 0           |
| activation_3 (Activation) | (None, 14400) | 0           |
| dense_1 (Dense) | (None, 512) | 7373312     |
| dropout_1 (Dropout) | (None, 512) | 0           |
| activation_4 (Activation) | (None, 512) | 0           |
| dense_2 (Dense) | (None, 2) | 1026        |
| activation_5 (Activation) | (None, 2) | 0           |

For CNN training process, adam is used for the optimizer which is known to give better results compared to SGD (stochastic gradient descent) optimizer [16]. The error rates are measured with categorical cross entropy and the results are shown as in Figure 7. As for Figure 7, the error rate decreases as the training progresses along with the number of epochs.

Figure 7. Categorical Cross Entropy per number of epochs in the trained CNN

5. Experimental Results and Analysis

5.1. Experimental Setup

There are many types of documents and we have categorized into 4 groups such as general document, picture-oriented document, free style document and miscellaneous. The general document is a formal document composed of almost all characters. Almost all previous researches [1]-[8] were done with this kind of documents. The picture-oriented document such as the fairy tale book contains not only characters but also some pictures in many pages. The free style document and miscellaneous have no formats or styles. Table 4 shows 4 categories and 10 types of documents in each category.

Table 4. 4 categories and 10 types of documents

| Category          | No. | Document Type       |
|-------------------|-----|---------------------|
| General document  | 1   | Culture book        |
|                   | 2   | Technical book      |
|                   | 3   | Paper               |
| Picture-oriented document | 4   | Document consisting of more than half of pictures |
| Free style document | 5   | Fairy tale book     |
|                   | 6   | Magazine            |
|                   | 7   | Leaflet             |
| Miscellaneous     | 8   | Comic book          |
|                   | 9   | Tabular document    |
|                   | 10  | Book cover          |

The general document and picture-oriented document have a basic format and a certain number of characters. Others are free in their character size, deployment of characters and the number of words in a document. Table 5 shows some examples of 10 types of documents.
Table 6. Result of Category 1 (General document)

| Document type | OCR-based method | ANN-based method | CNN-based method |
|---------------|------------------|------------------|------------------|
|               | Accuracy | Average Time (s) | Accuracy | Average Time (s) | Accuracy | Average Time (s) |
| 1             | 90%      | 2.580            | 90%      | 0.118            | 80%      | 1.049            |
| 2             | 80%      | 3.889            | 90%      | 0.071            | 90%      | 0.349            |
| 3             | 100%     | 4.700            | 90%      | 0.046            | 80%      | 0.368            |
| All           | 90%      | 3.723            | 90%      | 0.078            | 83%      | 0.589            |

Table 7. Result of Category 2 (Picture-oriented document)

| Document type | OCR-based method | ANN-based method | CNN-based method |
|---------------|------------------|------------------|------------------|
|               | Accuracy | Average Time (s) | Accuracy | Average Time (s) | Accuracy | Average Time (s) |
| 4             | 90%      | 1.773            | 70%      | 0.050            | 50%      | 0.992            |
| 5             | 70%      | 2.140            | 80%      | 0.190            | 70%      | 1.883            |
| All           | 80%      | 1.957            | 75%      | 0.120            | 60%      | 1.438            |

Table 8. Result of Category 3 (Free style document)

| Document type | OCR-based method | ANN-based method | CNN-based method |
|---------------|------------------|------------------|------------------|
|               | Accuracy | Average Time (s) | Accuracy | Average Time (s) | Accuracy | Average Time (s) |
| 6             | 100%     | 3.480            | 40%      | 0.042            | 70%      | 0.430            |
| 7             | 70%      | 1.724            | 40%      | 0.033            | 90%      | 0.391            |
| All           | 85%      | 2.602            | 40%      | 0.038            | 80%      | 0.411            |

Table 9. Result of Category 4 (Miscellaneous document)

| Document type | OCR-based method | ANN-based method | CNN-based method |
|---------------|------------------|------------------|------------------|
|               | Accuracy | Average Time (s) | Accuracy | Average Time (s) | Accuracy | Average Time (s) |
| 8             | 90%      | 0.784            | 40%      | 0.030            | 40%      | 1.063            |
| 9             | 90%      | 2.209            | 50%      | 0.030            | 60%      | 0.758            |
| 10            | 100%     | 0.794            | 60%      | 0.026            | 50%      | 0.562            |
| All           | 93.3%    | 1.247            | 50%      | 0.029            | 50%      | 0.794            |

5.3. Discussion

In general documents (category 1 in Table 6) such as cultural books, technical books, and papers, both OCR-based algorithm and ANN-based algorithm have similar high accuracy in Table 6. But ANN-based algorithm shows tremendous time reduction. The ANN-based algorithm we proposed will have a high efficiency for checking a large number of documents automatically.

www.astesj.com
In picture-oriented documents (category 2 in Table 7) such as picture books and fairy tale books, OCR-based algorithm is a little better than ANN-based algorithm in accuracy (See Table 7). But the result is also seen that the time in ANN-based algorithm is shortened. This may be occurred by the reason that the result depends on our small number of experimental documents and it could be improved if we apply our algorithm with much more data. The thresholding method we used to be based on the fixed value. If the thresholding value can be adaptively changed as the given input image, then the performance could be enhanced.

In free style documents and miscellaneous documents (category 3 and 4 in Table 4) such as magazines, leaflets, comic books, tabular documents and book covers, OCR-based algorithm is better than ANN-based algorithm in accuracy (See Table 8 and 9). This may be occurred by the reason that there is no a formal document structure in the document type 6, 7, 8, 9. And, in case of book cover, there are more non-black characters than black characters. If the system can recognize all colors of characters exactly in high speed, then the performance could be enhanced.

The performance of ANN-based algorithm was also compared with that of CNN-based algorithm. Basically, the performance of CNN-based algorithm has less accuracy than ANN-based algorithm (See Table 6 and 7). This is based on the reason that the number of training data is limited to 50 documents for the comparison with other algorithms, and the image was resized into 64*64 pixels by the complex structure of the CNN-based algorithm and as a result, the character was very distorted and became small. In the case of category 3 (See Table 8), CNN-based algorithm has better accuracy than ANN-based algorithm. It may be from the reason that many of the documents in category 3 include more non-black characters than black characters.

6. Conclusion

As a result, OCR-based algorithm shows high accuracy by the power of character recognition, but it takes more time because all words should be recognized in advance. CNN-based algorithm shows very limited power especially for the document type 2 and 7 due to the limited number of training data and resized image.

In contrast, ANN-based algorithm calculates two feature values derived from the document structure and can determine the orientation of the document immediately. ANN-based algorithm can greatly reduce the time using small information and can obtain high accuracy using the power of artificial neural network.

Additionally, we have conducted the experiments on the relatively free form of documents such as leaflets and magazines (category 3), comic books, tabular documents, and book covers (category 4). ANN-based algorithm determines the document structure immediately but the accuracy was significantly lower for these special cases.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

This research was partially supported by the MIST (Ministry of Science and ICT), Korea, under the national program for excellence in SW supervised by the IITP (Institute for Information & Communications Technology Promotion) (2016-0-00017), partially supported by the KIAT(Korea Institute for Advancement of Technology) grant funded by the Korea Government(MOTIE : Ministry of Trade Industry and Energy) (No. N0001884, HRD program for Embedded Software R&D) and partially supported by the KIAT(Korea Institute for Advancement of Technology) grant funded by the Korea Government (MSS : Ministry of SMEs and Startups). (No. S2755615, HRD program for Enterprise linkages R&D).

References

[1] S. Lu, C. L. Tan, “Automatic document orientation detection and categorization through document vectorization” in Proceedings of the 14th ACM international conference on Multimedia, Santa Barbara, CA, USA, 2006. http://doi.org/10.1145/1180639.1180673
[2] F. Hones, J. Lichter, “Layout extraction of mixed mode documents” Mach. Vision Appl. 7, 237–246, 1994.
[3] M. Ali, “An object/segment oriented skew-correction technique for document images” in Proceedings of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany, 1997. https://ieeexplore.ieee.org/document/620591/
[4] A. Hrishikesh, “A generic method for determining up/down orientation of text in roman and non-roman scripts” Pattern Recognition, 38(11), 2114-2131, 2005. https://doi.org/10.1016/j.patcog.2004.12.011
[5] S. J. Lu, J. Wang, C. L. Tan, “Fast and accurate detection of document skew and orientation” in Proceedings of the International Conference on Document Analysis and Recognition, Vol.2, 684–688, 2007. https://doi.org/10.1109/ICDAR.2007.4377002
[6] R. S. Caprari, “Algorithm for text page up/down orientation determination” Pattern Recognition Letters, 21, 311-317, 2000. https://doi.org/10.1016/S0167-8655(99)00161-0
[7] B. T. Avila, R. D. Lins, “A fast orientation and skew detection algorithm for monochromatic document images” in Proceedings of the ACM symposium on Document engineering, 118–126, 2005. https://doi.org/10.1145/1096601.1096631
[8] J. V. Beusekom, F. Shafait, T. M. Breuel, “Resolution independent skew and orientation detection for document images” in Proceedings of the SPIE Electronic Imaging: Document Recognition and Retrieval, Vol.7247, 72470K–72470K, 2009. https://doi.org/10.1117/12.807735
[9] M. Kim, S. Yim, Y. Lee, M. Kim, J.-W. Jung, “A study on automated checking for upside down printed materials based on Optical Character Recognition” in Proceedings of the 2018 International Conference on Fuzzy Theory and Its Applications (FUZZY), 2018. https://doi.org/10.1109/FUZZY.2018.8751060
[10] L. Richardson, Beautiful Soup Document Crunny. https://www.crummy.com/software/BeautifulSoup/bs3/documentation.html
[11] R. Smith, “An overview of the Tessera OCR engine” in Proceedings of the IEEE 9th International Conference on Document Analysis and Recognition (ICDAR 2007), Vol. 2, 2007. https://doi.org/10.1109/ICDAR.2007.4376991
[12] M. T. Hagan, H. B. Demuth, M. Beale, Neural Network Design, Boston Pws, 1996.
[13] R Documentation. https://www.rdocumentation.org/packages/neuralnet/versions/1.44/topics/neuralnet
[14] M. F. Aydogdu, M. F. Demirci, “Age Classification Using an Optimized CNN Architecture” in Proceedings of the International Conference on Computer and Data Analysis, Lakeland, FL, USA, 2017. https://doi.org/10.1145/3093241.3093277
[15] M. A. Shibli, P. Marques, E. Spiridon, “Artificial Intelligent Drone-Based Encrypted Machine Learning of Image Extraction Using Pretrained Convolutional Neural Network (CNN)” in Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality, Nagoya, Japan, 2018. https://doi.org/10.1145/3293663.3297155
[16] D. P. Kingma, J. Ba, “Adam: A Method for Stochastic Optimization” in Proceedings of the 3rd International Conference for Learning Representations, San Diego, 2015.
[17] ITU-R, Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios. https://www.itu.int/dms_pubrec/itu-r/rec/bt/BT.601-7-201103-I!!PDF-E.pdf