Text Document Orientation Detection Using Convolutional Neural Networks

Chapter · May 2021

DOI: 10.1007/978-981-33-4582-9_13

Dr Manju
Jaypee Institute of Information Technology

Text-Document Orientation Detection Using Convolutional Neural Networks

Shivam Aggarwal[1], Safal Singh Gaur[1], and Manju[1]

[1]Jaypee Institute of Information Technology, Noida, India

Abstract

Identifying the orientation of scanned text documents has been a key problem in today's world where every department of any cooperation is surrounded by documents in one or another way. In this paper, our emphasis is on the more challenging task of identifying and correcting the disorientation of general text documents back to normal orientation. Our work aims to solve the real-world problem of orientation detection of documents in PDF forms which can be later used in further document processing techniques. All further document processing tasks depend on detecting the correct orientation of the document. To do this, the convolutional neural network (CNN) is used which can learn salient features to predict the standard orientation of the images. Rather than the earlier research works which act mostly between the horizontal and vertical orientation of non-text documents only, our model is more robust and explainable as it works at page level with text documents. Also, we have accelerated to a different level with proper explanation and interpretability. The proposed approach runs progressively in real-time and, in this manner, can be applied to various organizations as well.

Index Terms

Text Documents, Convolutional Neural Network, Orientation Detection, Model Interpretability

INTRODUCTION

Proper maintenance of source document is a key aspect and today we have to deal with documents regularly in every aspect whether in the form of PDFs, images, etc. Retrieving useful information from documents is crucial. With an increasing interest in deep learning and artificial neural networks, various document analysis problems such as character recognition, layout analysis, and orientation detection of documents arise. To process or retrieve anything out of these documents, a correctly oriented documentation is required on which users can work further. A basic step of every document processing is to correct its original orientation if it is not in a proper format. For example, a conventional OCR system needs correctly oriented pages of any document before recognition and cannot be applied directly to any other orientation. Therefore, we aim to build a model that works on the problem of detecting the correct orientation of disoriented documents by converting clockwise and anticlockwise oriented documents to the normal position as shown in Fig. 1. In this figure, we present an automated way where we first detect the orientation of text at the page level and then automatically right the page direction dependent on the visual information (like text patches, lines). To do this, we take input as a PDF of documents with disoriented pages, and the expected output is a PDF with all correctly oriented pages which can be later used in various applications.

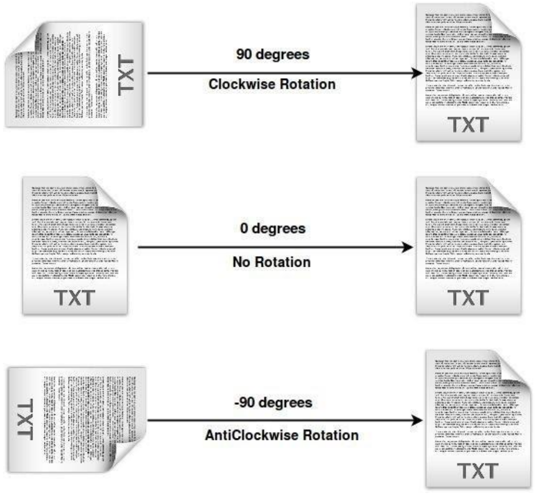


Fig.1. Description of the proposed model.

There is a lot of research works that aims to solve the above-stated problem statement [1]. However, most of these works done on detecting an orientation of non-text documents like family pictures, landscapes, nature scenes, etc. [3, 4, 6]. Most of the works are done to estimate head positioning using deep learning techniques [7,8]. To perform this task, there are inbuilt sensors inside the most recent cameras to modify the direction of pictures in 90-degree steps but, this function is usually not implemented due to a non-automated slow approach. To resolve this issue, various solutions were proposed by many researchers [2]. In our proposed work, we are not only focused on detecting the orientation of non-text documents instead we are interested in a more challenging task of text documents.

The task is challenging as few important features are contributing to finding out the direction of text as compared to scenic pictures which consist of various horizontal, vertical lines and shapes as features (contributing in identifying directions) [9]. Various traditional methods applied to solve the given problem at a very basic character-wise level which is slow for obvious reasons for a large number of characters over a single page [13]. Many other methods and approaches of computer vision are used to scan the text lines and classify the direction of text [17]. In such cases, the outcome relies upon subtle features of scanned documents that require a few comprehensions of the picture content. Over the most recent decade, profound convolutional systems have been demonstrated to be truly adept at learning such features [16, 18]. The major problem is the standard classification problem among different orientations of documents at the page level. Convolutional neural networks work in most satisfying ways of detecting good important features [10] and then use those feature vectors in solving classifying the direction of pages and then rotate page at corresponding anti-angles to make it normal [11].

In this study, we aim to train a convolutional neural network to classify the orientation of an image of a document. Firstly, all the PDFs are converted to image format dataset which is the prerequisite

for training in the CNN model. This model is working for three positions or classes to classify into, clockwise (90 degrees), anticlockwise (-90 degrees), and normal position (0 degrees) from the vertical. Once the classification of the imaged page finds out, then the model simply rotated the PDF page corresponding to the classified image by anti-classified-angle.
Preparing ground-breaking, profound convolutional neural systems used to require a lot of labeled training data. Regularly this is a solid limitation as the assortment of such information can be intense. For the current work, be that as it may, preparing such informative (labeled) data can be produced effectively from any cooperation documentation. Any unlabeled set of images can be treated as a training set by manually labeling it. Training samples can be created by simply pivoting these pictures to the required degrees (- 90◦, 0◦, 90◦). Moreover, our proposed work tries to focus on model interpretability of the convolutional network via the proposed algorithm. Some guidelines enable one to see the internal working of neural networks [19,20]. By observing intermediate visualization of input image after every layer, produced results are semantically meaningful as depicted in the simulation section.

The rest of the paper is organized as follows: Section 2, discusses the related work on document orientation detection, and section 3 addresses the proposed methodology to solve the discussed problem. In section 4, discusses the model interpretability and gives a detailed description of the convolutional network used. Section 5 discusses the simulation outcomes and discusses the various outcomes achieved by the proposed model. Finally, in section 6 we conclude the proposed model accuracy and give direction on future work.

2. RELATED WORK

Orientation detection is a common challenge in document analysis and processing for certain applications. Most of the existing research works are proposed for non-text images and focuses on identifying the skew angles using various computer vision algorithms. Fischer et al. [1] applied CNN for solving the problem of orientation classification of general images only. In the proposed method, CNN predicts the angle of landscape photos without any preprocessing. It is also clear from this research that CNN can be used to feature out those areas responsible for different image orientations. However, authors in [1] did not provide any explanation that solidifies their CNN model further and robustness is poor for text images. Furhet, Jeungmin et al. [2] proposed a document orientation detection approach that detects document capturing moments to help users to correct the orientation errors. They captured the document orientation by tracking the gravity direction and gyroscope data and gave visual criticism of the construed direction for manual rectification. Wei et al. [3] used an interpolation artifacts approach implemented by applying a rotation to digital images. Be that as it may, this strategy likewise would not work for pictures that were not taken upstanding. H. S. Baird [4] defined an algorithm for detecting the page orientation (portrait/landscape) and the degree of skew for documents available as binary images. They used a new approach of Hough Transform and local analysis. Later, Solanki et al. [5] method are for printed images, make use of patterns of printer dots by analyzing them, and then estimate the rotation of images. Plainly, this strategy likewise not material to pictures taken with a computerized camera. In other words, the continuous-valued prediction of angles task has been decreased to discrete angles by confining the revolutions to multiples of 90. This issue can be understood as reasonable proficiently [6, 12]. With time a lot more researches happened over text documents too. Every one of these techniques exploits the extraordinary structure of text document images, for example, content design in lines and exact states of letters which is a comparatively harder task as there are fewer features to find out which are contributing to orientation detection. Chen et al. [13] were first to apply neural networks techniques to the recognition of document language type and acquired outcomes better than those of conventional strategies. They were using text lines as input data and feeding them to their CNN model to recognize the document properties but did not provide any model interpretability which solidifies their model simply to use it as a black box. Chen et al. [14] algorithm utilize recursive morphological transforms for text skew estimation. The automated experimental results indicate that the algorithm generates estimated text skew angles which are within 0.5/spl deg/ of the true text skew angles 99 percent of the time. Avila et al. [15] worked on a fast algorithm for orientation and skew detection for complex monochromatic document images, which is capable of detecting any document rotation at high precision. Sun, C. et al. [16] developed an algorithm that employs only the gradient information. Their results show that the statistics of the gradient orientation are enough to obtain the skew angle of a document image. The algorithm works on various document images containing text or may contain text with tables, graphics, or photographs. Yan et al. [17] worked on a model for gray-scale and color images as well as binary scanned images. In this work, the cross-correlation between two lines in the image with a fixed distance is calculated and the maximum distance is chosen to calculate the skew angle, later the image is rotated in the opposite direction for skew correction. Chen et al. [18] come up with a new deep learning language-based approach and orientation recognition in document analysis where CNN is used in determining features/ document properties. Here, authors have proposed a voting process to diminish the system scale and completely utilize the data of the content line.

There are several other works too which are based on deep learning and various algorithms of image processing to solve the discussed problem. However, these approaches either work on text lines datasets or works on identifying the orientation of non-text data like simple landscape images. Building a new approach of CNN classifier using image dataset of documents is something we have researched to improve the existing orientation detection schemes to make them better. Our aims here are to enhance the speed and robustness along with accurate results by following a deep learning convolutional approach that can be used to extract document properties at the page level and helps in classifying the discrete angles.

3. PROPOSED METHODOLOGY

In this section, we will discuss the proposed methodology to solve the problem of text document orientation detection. Deep learning convolutional neural network is the proposed method based on the classification of the document. Here we classify the whole document in three categories namely, clockwise (90 degrees), anticlockwise (-90 degrees), normal (0 degrees), and change the orientation of the document for the respective degrees. With so much advancement in the field of image processing, the first basic deep learning model pops up in the mind is Convolutional Neural Networks (CNN). Image classification is one of the most important applications of CNN. Due to its simplicity and better results, it is widely used in image classification problems. It has three major parts which include the input layer to takes the input image to be classified, hidden layers that make it deeper and more efficient as compared to any other machine learning model, and finally output layer where the class is defined for the input image.

Most of the existing methods use text lines of the document as input to the CNN which further required a task to find out patches of text in a document and increase computational time and then work only for two orientations positive and reverse directions. In order to find a solution, there are some approaches in which we directly input text document as an image and corresponding orientation as label results in better and faster computations. Our model provides a complete end to end solutions where input is a wrongly oriented PDF of a text document in any form and output is correctly oriented pages in the same PDF format as depicted in the following Fig. 2.

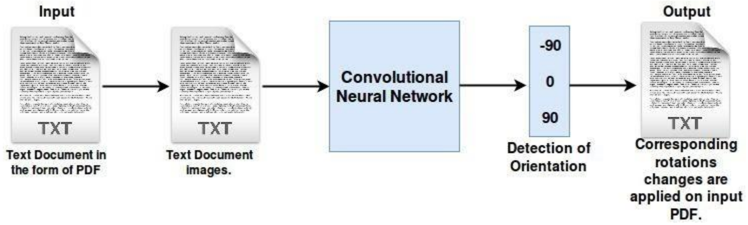

Input
TXT
Text Document in the form of PDF
TXT
Text Document images.
Convolutional Neural Network
-90
0
90
Detection of Orientation
Output
TXT
Corresponding rotations changes are applied on input PDF.

Fig. 2. Input is disoriented PDF; Output is correctly oriented PDF

Data Description

Preferably, for preparing a system to foresee orientation, one requires a dataset of characteristic pictures commented on how much their rotation angles deviate from the upstanding direction. We have used a dataset of text documents provided by some organization; the training dataset consists of 8618 JPEG images from different PDF documents where each document PDF is first converted to the image. These images are manually divided into 3 categories by rotating images where 2908 are -90-degree oriented images, 2927 are 0-degree oriented images and 2783 are +90-degree oriented images that are saved in the format like orientation.id.jpg to makes input-output label easier at the time of training. For example, 90.458.jpg refers to a 90-degree oriented JPEG image with unique id 458, 0.1563.jpg refers to a 0-degree oriented JPEG image with unique id 1563. In our proposed methodology, the CNN model has 5 Convolutional layers which have 32, 64, 128, 256, and 512 convolution kernels, and all kernels size are 33 with rectified linear unit activation function, and 5 max-pooling layers with a stride of 2 and size of 2x2. There is batch normalization after every convolution operation and dropout with a drop probability of 0.25 after each pooling layer. There is a flattening layer and two dense layers with one having an output shape of 512 and the last one with the shape of 3 as we have 3 classes (i.e.one for each). There is one SoftMax regression layer on the top of the Neural Network. The proposed model is shown in the following Fig.3.

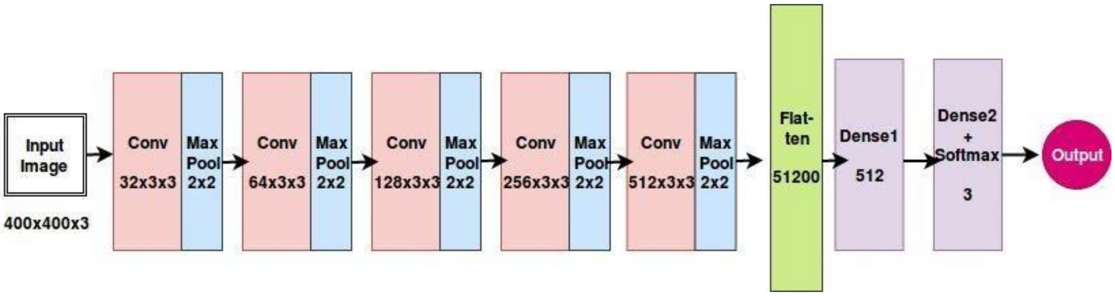


Fig.3. Block structure of Proposed Convolutional Neural Network.

RMSProp is used as an optimizer which is similar to gradient descent algorithm but with momentum for better and faster results. Cross-Entropy Loss is the loss function our model used in the case of multi-classes. The model uses callbacks provided by Keras at given stages of the training procedure. Early stopping is used to stop training when a monitored quantity has stopped improving with patience parameter (which defines a number of epochs with no improvement after which training will be stopped) equal to 10. Further, ReduceLROnPlateau is used with a function to reduce the learning rate when a metric has stopped improving. This callback monitors the quantity and if there is no improvement seen for a 'patience' number of epochs (i.e. set to 2), the learning rate is reduced to 0.00001.

Data augmentation is applied to the training dataset in order to expand the data for better performance and accuracy and to prevent overfitting. The augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images. Operations like rescaling, shearing, zooming and, width and height shifting are performed with batch-size of 32 and the number of epochs equal to 10.

4. MODEL INTERPRETABILITY

There are many research works exist where a solution is provided to a great extent but explaining the reason behind the working of the solution is still challenging. Model interpretability is the key as it is a foundation of the standards and procedures which are characterized. As correlation frequently doesn't rise to causality, a strong model comprehension is required with regards to settling on choices and clarifying them. Model interpretability is important to check what the model is doing is in accordance with what you expect and it permits making trust with the clients and facilitating the progress from manual to mechanized procedures.

Intermediate Feature visualization at every convolutional layer

With regard to profound neural systems, interpretability turns into somewhat extreme. Luckily, Convolutional Neural Networks (ConvNets or CNNs) have inputs (pictures) that are outwardly interpretable by people so we have different procedures for seeing how they work, what do they learn, and why they work in a given way. To make CNN from black box to white box, results are recorded in the form of images after multiple steps. Two algorithms are used in explaining model interpretability.

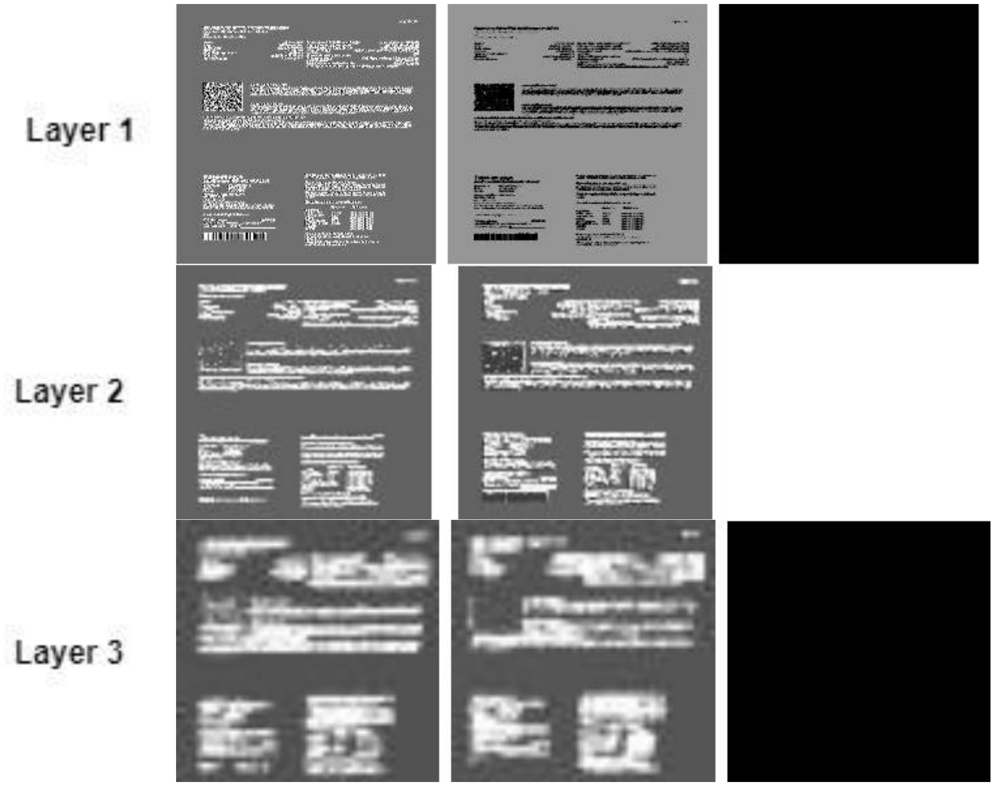
The first algorithm is to predict the correlation of every kernel with the input image and finding the kernel/filter which is contributing most in predicting the class. Since every kernel has different trained weights, every image contributes and establish different correlation which may help and plays an important role in predicting classes (positive correlations), or may not help in any form and therefore shows no impact towards classification score (zero correlated) or may have a negative impact on the score and contributing in the prediction of wrong class (negatively correlations). In this algorithm basically, every kernel or channel is replaced iteratively by setting its weight to all zeros and then observing the impact/changes by new classified score. Thus, the channel importance score can be computed by subtracting the normal score from the new classified score.

Channel imp score = new classified score – normal classified score

$$ChannelImpact = \begin{cases} \text{Positive,} & \text{if } channelimpscore > 0 \\ \text{No Impact,} & \text{if } channelimpscore = 0 \\ \text{Negative,} & \text{if } channelimpscore < 0 \end{cases}$$

If channel importance score comes out to be positive i.e. channel is important and has a positive impact on classification and channel corresponding to this score is said to be a positively correlated channel. On the other hand, if the channel importance score comes out to be negative i.e. channel has a negative impact on classification, the channel corresponding to this score is said to be a negatively correlated channel. Finally, if the channel importance score comes out to be zero i.e. channel has no impact on classification and channel corresponding to this score is said to be zero correlated channel.

The second algorithm is about visualizing an image after every set of layers. Featured images are saved after every set of 4 operations which includes convolutional, max pooling, batch normalization, and dropout for every kernel. The way after the first layer we obtained 32 images from 32 different kernels, after the second layer we obtained 64 images from 64 different kernels and so on. Fig 4. shows an example of results for an input image of zero-degrees.



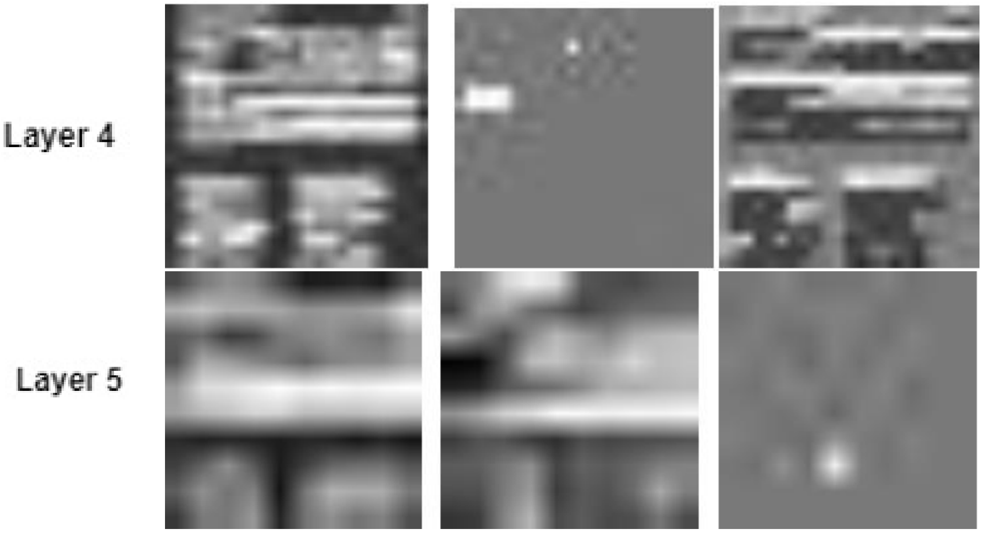


Image obtained from highly positive correlated filter

Image obtained from highly negative. correlated filter

Image obtained from zero correlated filter.

Fig. 4. Intermediate visualization of an input image (0 degrees) after every layer sets.

Interpretations

The underlying layers (layer-1 and layer-2) hold the vast majority of the info picture's component. It would seem that the convolution channels are actuated at all aspects of the input image. Simply we can say that the model is considering text patches as features and we identify those patches with the bright part in our intermediate images. As we go deeper (layer-3 and layer-4), the highlights extricated by even most associated channels become outwardly less interpretable. One can see only bright horizontal and vertical lines. An instinct for this can be clarified that intermediate image is now abstracting endlessly visual data of the input image and attempting to change over it to the necessary yield grouping area since normal orientation text is more in form of horizontal patches while for 90 and -90 these patches are tilted vertically which helps model in the classification of horizontal and vertical text. They go about as data refining pipeline where the input image is being changed over to space which is visually less interpretable (by evacuating noises) however mathematically valuable for the convolutional network to settle on a decision from the yield classes in its last layer [20]. In the end, by this model interpretability method, one can provide a satisfactory explanation for classifying horizontal and vertical test document images.

5. SIMULATION RESULTS

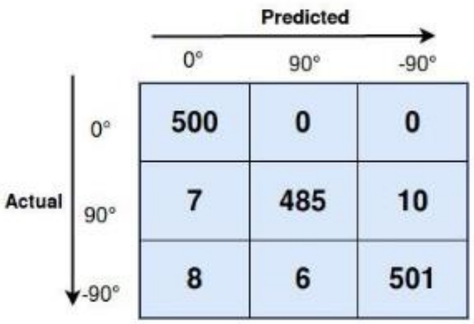
We have used a dataset of text documents provided by some organization; the training dataset consists of 8618 JPEG images from different PDF documents where each document PDF is first converted to the image. These images are manually divided into 3 categories by rotating images where 2908 are -90-degree oriented images, 2927 are 0-degree oriented images and 2783 are +90-degree oriented images that are saved in the format like orientation.id.jpg to makes input-output label easier at the time of training. For example, 90.458.jpg refers to a 90-degree oriented JPEG image with unique id 458, 0.1563.jpg refers to a 0-degree oriented JPEG image with unique id 1563.

Manually verified test set: The test dataset consists of new 1517 JPEG images which are extracted from completely different PDF documents from the training dataset. Similarly, like training dataset, divided into 3 categories where 515 for -90-degree orientation, 500 for zero-degree orientation, and 502 for +90-degree orientation. While providing input to the model for training, images are resized into (400, 400, 3) i.e. height and width of 3 channels (RGB) is 400x400 respectively. Experiments were run online using Kaggle kernels (a free platform to run Jupyter notebooks) using python version 3 on a computer system with AMD A8-7410 processor of 4GB RAM, running Windows version 10 Home.

While evaluating the test dataset of 1517 images, our model predicts 1486 with correct orientation while the other 31 images wrongly predicted. More clarity in results is provided by observing on what orientations our model fails by dividing wrong predicted images into 6 subsections as shown below.

1. Images with normal orientation (zero-degrees) predicted in 90-degrees category.
2. Images with normal orientation (zero-degrees) predicted in -90-degrees category.
3. Images with 90-degrees orientation predicted in 0-degrees category.
4. Images with 90-degrees orientation predicted in -90-degrees category.
5. Images with -90-degrees orientation predicted in 0-degrees category.
6. Images with -90-degrees orientation predicted in 90-degrees category.

The Confusion Matrix of results on the test set is shown below with an accuracy of **97.956%.**

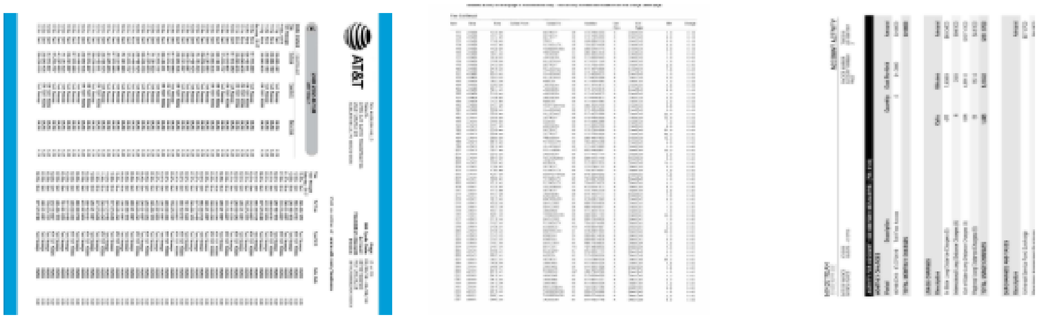


Predicted

Actual	0°	90°	-90°
0°	500	0	0
90°	7	485	10
-90°	8	6	501

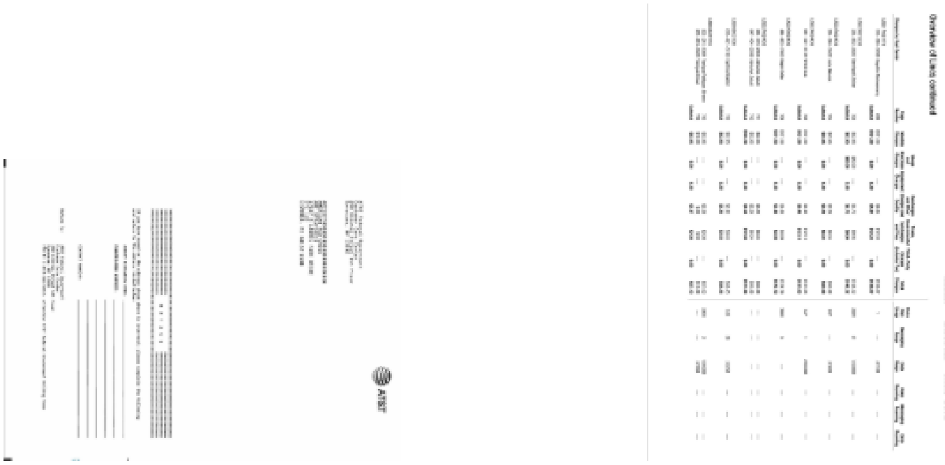
Confusion matrix

One can observe that there are zero wrongly predicted normal orientation images (zero-degrees) which is well described by our interpretability model which can provide a satisfactory explanation for classifying horizontal and vertical test document images. All the wrongly predicted images come either from clockwise or anticlockwise orientations as an explanation for model interpretability in case of 90 degrees and -90 degrees is still not crystal clear by a given algorithm. There are 17 wrongly predicted 90-degrees orientation images out of which 7 are predicted as in zero-degrees orientation and the other 10 are classified as -90-degrees orientation. And, there are 14 wrongly predicted -90-degrees orientation images out of which 8 are predicted as in zero-degrees orientation and the other 6 are classified as 90-degrees orientation. Also, there are some cases, like blank pages can be classified as any of three categories. Given below are some predictions results of both the cases wrongly and correctly predicted.



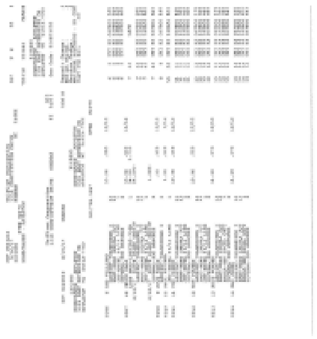
(a) (b) (c)

Examples of correctly predicted images,(a) -90° (b) 0° and (c) +90°.



(a) (b)

Examples of images with true orientation as -90° but predicted as +90°.



(a) (b)

Examples of images with true orientation as +90° but predicted as -90°.

Note: The second image is a blank page which indirectly considered as a true prediction.

(a) (b)

Examples of images with true orientation as +90° but predicted as 0°.

(a) (b)

Examples of images with true orientation as -90° but predicted as 0°.

6. CONCLUSIONS

In this study, we have presented an approach dependent on convolutional networks that aims to a covert disorientated text document to its well-oriented form. Our model aims to solve the real-world problem of orientation detection of documents in PDF format which can be later used in further document processing techniques as the document processing tasks depend on detecting the correct orientation of the document. There were many related works already done on this problem of document orientation but none of them works at the page level. Also, we have accelerated to a different level with proper explanation. The proposed model achieves an accuracy of approx. 98%, based on deep learning 5-layer convolutional neural network. Next, this study also tried to explain model interpretability via a proposed algorithm of observing intermediate visualization of the image every layer. These produced semantically meaningful results. In future work, the network architecture will be optimized to run faster in real-time. Also, using interpretability results to analyze misclassified cases and propose a way to fix this either by creating more training data or other methods. Devise methods for model interpretability is to understand how 90-degrees and -90-degrees are distinguished by the model.

REFERENCES

1. Fischer, Philipp Dosovitskiy, Alexey Brox, Thomas, “Image Orientation Estimation with Convolutional Networks,” 9358. 368-378. 10.1007/978- 3-319-24947-6$_3$0, 2015.
2. J. Oh, W. Choi, J. Kim, and U. Lee, “Scanshot: Detecting document capture moments and correcting device orientation,” in ACM Conference on Human Factors in Computing Systems, pp. 953–956, 2015.
3. Wei, W., Wang, S., Zhang, X., Tang, Z., “Estimation of image rotation angle using interpolation-related spectral signatures with application to blind detection of image forgery,” Trans. Info. For. Sec. 5(3), 507–517, 2010.
4. H. S. Baird, “Measuring document image skew and orientation,” Proceedings of SPIE - The International Society for Optical Engineering, 1995.
5. Solanki, K., Madhow, U., Manjunath, B.S., Chandrasekaran, S., “Estimating and undoing rotation for print-scan resilient data hiding,” In: ICIP. pp. 39–42, 2004.
6. Vailaya, A., Zhang, H., Member, S., Yang, C., Liu, F.I., Jain, A.K., “Automatic image orientation detection. In: IEEE Transactions on Image Processing,” pp. 600–604, 2002.
7. Voit, M., Nickel, K., Stiefelhagen, R.: R., “Neural network-based head pose estimation and multi-view fusion” In: Proc. CLEAR Workshop, LNCS. pp. 299–304, 2006.
8. Peake, G.S., Tan, T.N., “A general algorithm for document skew angle estimation,” In: ICIP (2). pp. 230–233, 1997.
9. Pingali, G.S., Zhao, L., Carlbom, I., “Real-time head orientation estimation using neural networks,” In: ICIP. pp. 297–300, 2002.
10. Fefilatyev, S., Smarodzinava, V., Hall, L.O., Goldgof, D.B., “Horizon detection using machine learning techniques,” In: ICMLA. pp. 17–21, 2006.
11. Krizhevsky, A., Sutskever, I., Hinton, G.E., “Imagenet classification with deep convolutional neural networks,” In: NIPS. pp. 1106–1114, 2012.
12. Wang, Y.M., Zhang, H., “Detecting image orientation based on low-level visual content. Computer Vision and Image Understanding,” 93(3), 328–346, 2004.
13. L. Chen, S. Wang, W. Fan, J. Sun, and N. Satoshi, “Deep learning-based language and orientation recognition in document analysis,” In: International Conference on Document Analysis and Recognition, pp. 436–440, 2015.
14. Chen, S.S., Haralick, R.M., “An automatic algorithm for text skew estimation in document images using recursive morphological transforms,” In: ICIP. pp. 139–143,1994.
15. Avila, B.T., Lins, R.D., “A fast orientation and skew detection algorithm for monochromatic document images,” In: Proceedings of the 2005 ACM Symposium on Document Engineering. pp. 118–126, 2005.
16. Sun, C., Si, D., “Skew and slant correction for document images using gradient direction,” In: 4th International Conference Document Analysis and Recognition (ICDAR’97). pp. 142–146, 1997.
17. Yan, H., “Skew correction of document images using interline cross-correlation,” CVGIP: Graphical Model and Image Processing 55(6), 538–543, 1993.
18. R. Wang, S. Wang and J. Sun, “Offset Neural Network for Document Orientation Identification,” 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, 2018, pp. 269-274, 2018.
19. How convolutional neural networks see the world, https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html
20. Zeiler, M. D., Fergus, R., “Visualizing and understanding convolutional networks,” In Computer Vision, ECCV 2014 - 13th European Conference, Proceedings (PART 1 ed., pp. 818-833), 2014.