Summer Training Report 2019

Text-Document Orientation Detection
Using convolutional neural networks

Company : ***Razorthink Technologies Pvt. Ltd.***

Address : 4, 17th Cross Rd, Siddanna Layout, Banashankari Stage II, Banashankari, Bengaluru, Karnataka.

Training Period : **25-05-2019 to 24-07-2019**

Submitted by:
Shivam Aggarwal
16103093 (CSE)
B2

Training In-charge:
Mr. Vamsi Krishna AV
(Product Manager)

Submitted to:

TABLE OF CONTENTS

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Mr. Sunil Kr. Vengalil** for their guidance and **Vamsi Krishna AV** constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & member of ***Razorthink Technologies Pvt. Ltd.*** for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

DECLARATION BY THE STUDENT

I the undersigned solemnly declare that the project report **Text-Document Orientation Detection Using convolutional neural networks** is based on my own work carried out during the course of our study under the supervision of **Mr. Vamsi Krishna AV.** I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

1. The work contained in the report is original and has been done by me under the general supervision of my supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Name : Shivam Aggarwal
Enrollment No. :16103093

ABOUT COMPANY

RAZORTHINK AIs FOR THE ENTERPRISE

The Razorthink Leadership team is comprised of seasoned leaders with expertise in Artificial Neural Networks, Deep Learning, Natural Language Processing, and advanced enterprise analytics as well as experience developing and deploying highly scalable enterprise systems. The team has worked with leading companies in energy, e-retail, financial services, manufacturing, pharmaceuticals, technology, telecommunications/mobile and transportation.

Razorthink, Inc. is an advanced software development company that is building an augmented brain for business. An artificial intelligence that is specifically designed to help business do business faster, better, and be more insightful.

Razorthink is helping to take this technology from the research lab and low level implementations to a platform that can be used to solve real problems. Our technology can be used to create models that can not only predict business outcomes, but dynamically adapt to underlying structural changes. It can find patterns and anomalies in massive sets of data, and can optimize a business process by solving complex multivariate problems. It can find insights and understanding that humans could find hard to comprehend.

INTRODUCTION

PROBLEM STATEMENT

Proper maintenance of source documentation is a key and today we have to deal with documents regularly in every aspect whether in the form of PDFs, images etc. For processing and retrieving useful information from documents is very crucial. With increase in craze for deep learning, artificial neural networks, various document analysis problems such as character recognition, layout analysis, and orientation detection of documents arises. And basis for all this starts with correctly oriented documents on which users can further work on. First step of every document processing is to correct its orientation. Lets say, a conventional OCR system needs correctly oriented pages of any document before recognition and cannot be applied directly. We aim to build a model which works on the problem of detecting correct orientation of disoriented documents, simply converting clockwise and anticlockwise oriented documents to normal position as shown in Fig. 1.

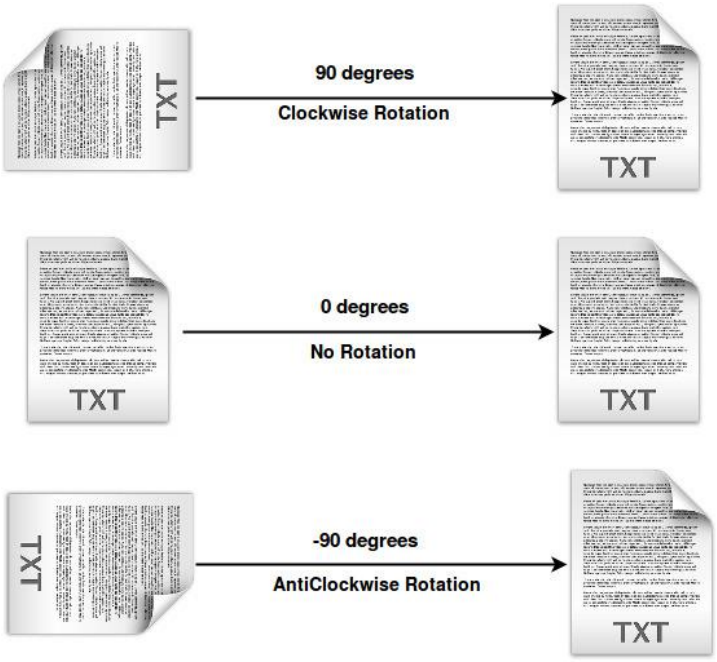


Fig.1. Describing the problem in left and aiming for solution shown in right.

RELATED WORK

Orientation detection is a common challenge or task to research in document analysis and processing. There are many achievements and successes regarding this field. Li Chen et al. [1] first applied CNN to the recognition of document language type and orientation and obtained
results better than those of traditional methods. They were using text lines as input data and feeding them to their CNN model to recognize the document properties. But they did not provide any model interpretability which solidify their model,simply to use it as a black box. Fischer et al. [2] used CNN for solving the problem of orientation classification of general images but not text documents . CNN predicts the angle of landscape photos without any preprocessing step. It is clear from this research that CNN can be used to feature out those areas responsible for different image orientation. However, they also did not provide any explanation that solidify their CNN model further and robustness is poor for text images. Jeungmin et al. [3] proposed a document direction recognition method that detects document capturing moments to help users correct the orientation errors. They capture the document orientation by tracking the gravity direction and gyroscope data , and they provided visual feedback of the inferred orientation for manual correction. 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 new approach of Hough Transform and local analysis. There are several other work too based on Deep learning and various algorithms of image processing to solve the problem. However these approaches either work on textlines datasets or work on identifying orientation of non-text data, simply 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.

PROPOSED METHODOLOGY

CNN BASED APPROACH

Deep learning convolutional neural network is the proposed approach in solving the problem via classification of document in three categories, clockwise(90 degrees), anticlockwise(-90 degrees) and normal(0 degrees) and change the orientation of the document for the respective degrees. With so much advancements in the field of image processing, first basic deep learning model pops up in the mind is Convolutional Neural Networks - CNN. Image classification is among one of the most important applications of CNN. Due to its simplicity and better results, it is widely used in image classification problem. It has three major parts input layer which takes the input image to be classified, hidden layers which makes it more deeper and efficient as compared to any other machine learning model, and output layer at which class is defined for the input image.

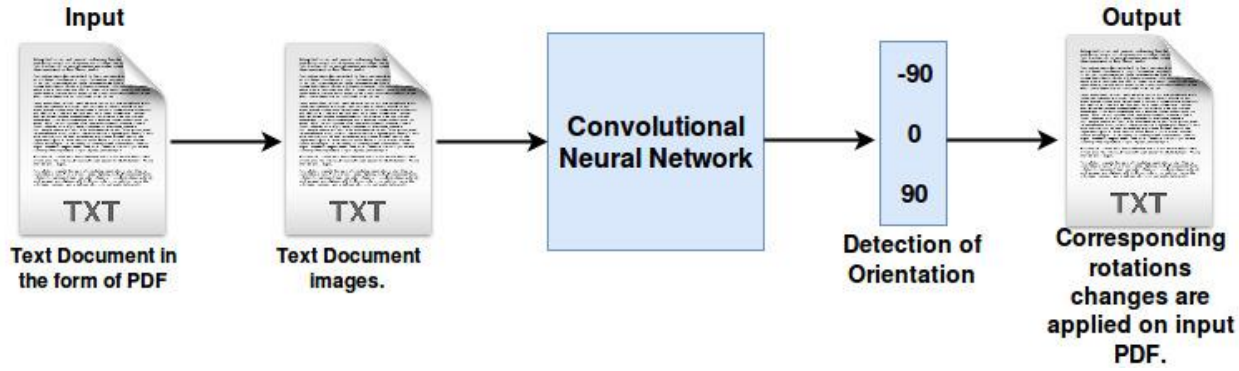


Fig.2. A flow chart of applied methodology. Input is disoriented PDF, Output is correctly oriented PDF.

Most existing methods use text lines of 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 secondly work only for two orientations positive and reverse directions. To solve this problem there are some methods in which we directly inputting text document as an image and corresponding orientation as label results in better and faster computations. Our model provides complete end to end requirements where input is a wrongly oriented PDF of text document in any form and output is correctly oriented pages in the same PDF format. As shown in Fig. 2.

DATASET DESCRIPTION

The training dataset consists of 8618 JPEG images from various different PDF documents i.e. each document PDF is first converted to image. These are manually divided into 3 categories by rotating images - 2908 are -90 degree oriented images, 2927 are 0 degree oriented images and 2783 are +90 degree oriented images. These are saved in the format like orientation.id.jpg which makes inputting 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.

The test dataset consists of new 1517 JPEG images which are extracted from completely different PDF documents from training dataset. And divided into 3 categories 515 for -90 degree orientation, 500 for zero degree orientation and 502 for +90 degree orientation.

While inputting in the model for training, images are resized into (400,400,3) i.e. height and width of 3 channel (RGB) is 400x400 respectively. Training deep learning neural network models on more data can result in more skillful models, and 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. Data augmentation is applied on training dataset in order to expand the data for better performance and accuracy. Operations like rescaling, shearing, zooming and, width and height shifting are performed with batch_size of 32 and number of epochs equal to 10.

MODEL ARCHITECTURE

In our experiment, our CNN model has 5 Convolutional layers which have 32, 64, 128, 256 and 512 convolution kernels, and all kernels size are 3×3 with rectified linear unit activation function, and 5 max pooling layers with stride of 2 and size of 2x2. There are batch normalization after every convolution operation and dropout with drop probability of 0.25 after every pooling layer. There is flattening layer and two dense layers with one having output shape of 512 and last one with shape of 3 as we have 3 classes one for each. There are one Softmax regression layer on the top of Neural Network.The detailed network summary is shown below:

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 398, 398, 32)	896
batch_normalization_1 (Batch	(None, 398, 398, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 199, 199, 32)	0
dropout_1 (Dropout)	(None, 199, 199, 32)	0
conv2d_2 (Conv2D)	(None, 197, 197, 64)	18496
batch_normalization_2 (Batch	(None, 197, 197, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 98, 98, 64)	0
dropout_2 (Dropout)	(None, 98, 98, 64)	0
conv2d_3 (Conv2D)	(None, 96, 96, 128)	73856
batch_normalization_3 (Batch	(None, 96, 96, 128)	512
max_pooling2d_3 (MaxPooling2	(None, 48, 48, 128)	0
dropout_3 (Dropout)	(None, 48, 48, 128)	0
conv2d_4 (Conv2D)	(None, 46, 46, 256)	295168
batch_normalization_4 (Batch	(None, 46, 46, 256)	1024
max_pooling2d_4 (MaxPooling2	(None, 23, 23, 256)	0
dropout_4 (Dropout)	(None, 23, 23, 256)	0
conv2d_5 (Conv2D)	(None, 21, 21, 512)	1180160
batch_normalization_5 (Batch	(None, 21, 21, 512)	2048

max_pooling2d_5 (MaxPooling2	(None, 10, 10, 512)	0
dropout_5 (Dropout)	(None, 10, 10, 512)	0
flatten_1 (Flatten)	(None, 51200)	0
dense_1 (Dense)	(None, 512)	26214912
batch_normalization_6 (Batch	(None, 512)	2048
dropout_6 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 3)	1539

Total params: 27,791,043 Trainable params: 27,788,035 Non-trainable params: 3,008

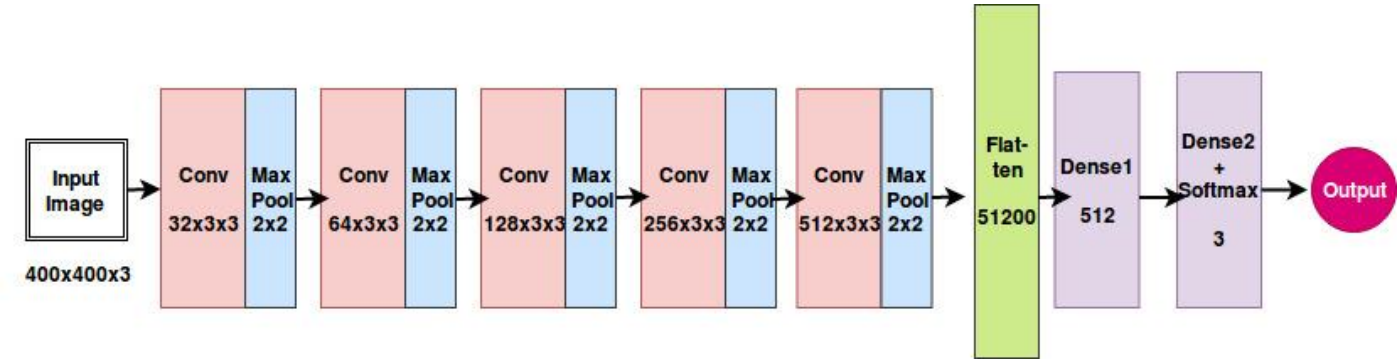


Fig.3. Block structure of Proposed Convolutional Neural Network.

RMSProp is used as 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 case of multiclasses. 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 number of epochs with no improvement after which training will be stopped.) equal to 10. ReduceLROnPlateau is used with a function to reduce learning rate when a metric has stopped improving. This callback monitors a quantity and if no improvement is seen for a 'patience' number of epochs i.e. set to 2, the learning rate is reduced upto 0.00001.

MODEL INTERPRETABILITY

Intermediate Feature visualization at every convolutional layer

EXPERIMENT

To make CNN from black box to white box, results are recorded in the form of images after multiple steps. There are two algorithms which are used in this model interpretability. First is to predict the correlation of every kernel with input image and finding the kernel/filter which is contributing most in predicting the class. Since every kernel has different trained weights so every image contributes and establish different correlation whether they 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 score and contributing in 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.Therefore, computing the channel importance score by subtracting normal score from new classified score.

channel_importance_score = new_classified_score - normal_classified_score

1. If channel importance score comes out to be positive i.e. channel is important and have a positive impact on classification, channel corresponding to this score is said to be positively correlated channel.
2. If channel importance score comes out to be negative i.e. channel has a negative impact on classification, channel corresponding to this score is said to be negatively correlated channel.
3. If channel importance score comes out to be zero i.e. channel has no impact on classification, channel corresponding to this score is said to be zero correlated channel.

Second algorithm is about writing an image after every set of layers. Featured images are saved after every set of 4 operations, convolutional, max pooling, batch normalization and dropout for every kernel. Like after first layer we obtained 32 images from 32 different kernels, after second layer we obtained 64 images from 64 different kernels and so on.

INFERENCE

From the results observed, we can conclude that the model is considering text patches as features and we identify those patches with the bright part in our intermediate images. Normal orientation text is more in form of horizontal patches while for 90 and -90 these patches are tilted vertically which helps model in classification of horizontal and vertical text. So, by this model interpretability we can provide a satisfactory explanation for classifying horizontal and vertical test document images.

RESULTS

Input Image(0°)

After Layer1

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

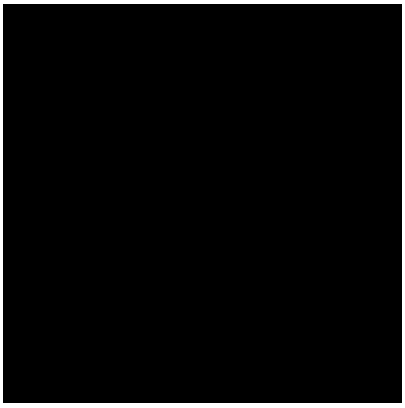


Image obtained from highly zero correlated filter.

After Layer2

Image obtained from highly positive correlated filter

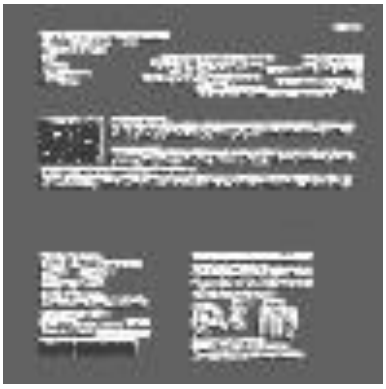


Image obtained from highly negative correlated filter.

No Image obtained from zero correlated filter.

After Layer3

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

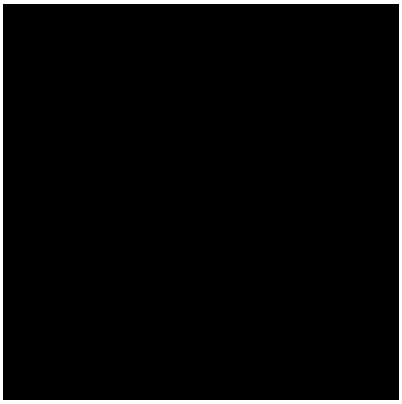


Image obtained from highly zero correlated filter.

After Layer4

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

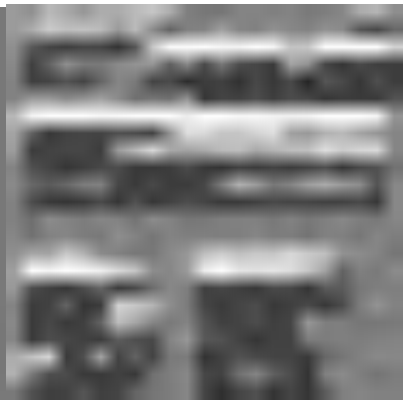


Image obtained from highly zero correlated filter.

After Layer5

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

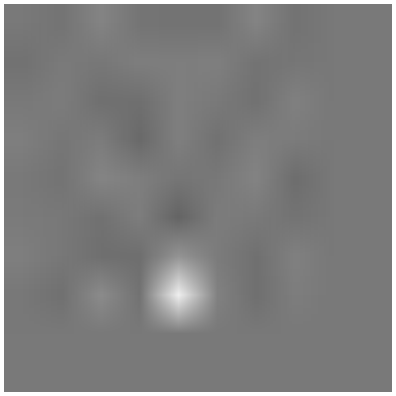
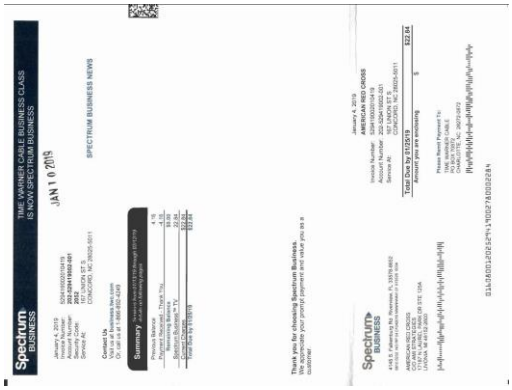


Image obtained from highly zero correlated filter.



Input Image(90°)

After Layer1

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

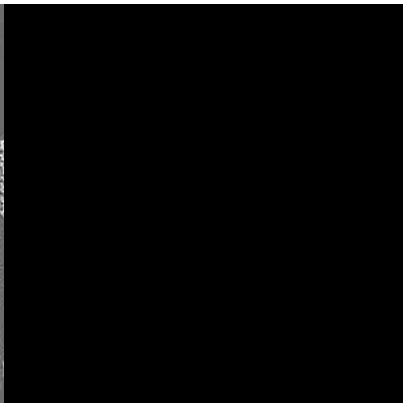


Image obtained from highly zero correlated filter.

After Layer2

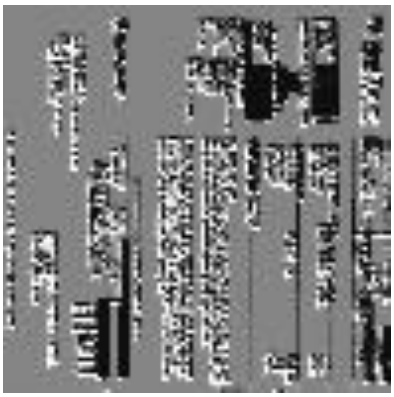


Image obtained from highly positive correlated filter

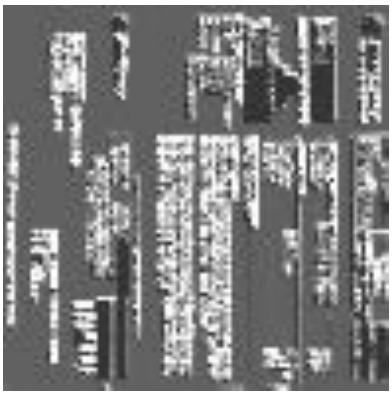


Image obtained from highly negative correlated filter.

No Image obtained from zero correlated filter.

After Layer3

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

No Image obtained from zero correlated filter.

After Layer4

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

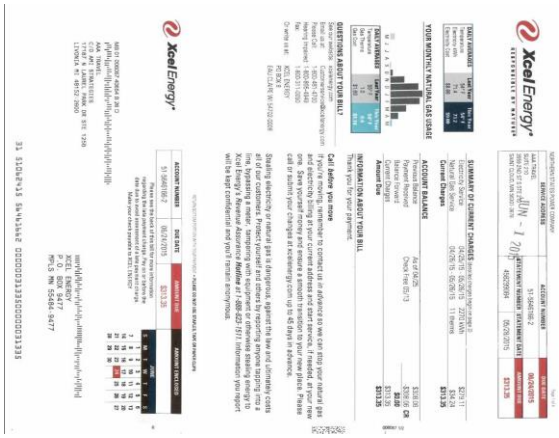
No Image obtained from zero correlated filter.

After Layer5

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

Blank Image obtained from zero correlated filter.



Input Image(-90°)

After Layer1

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

Image obtained from highly zero correlated filter.

After Layer2

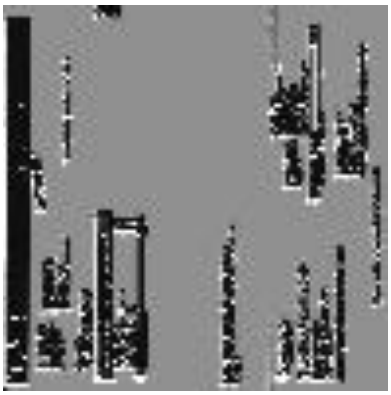


Image obtained from highly positive correlated filter

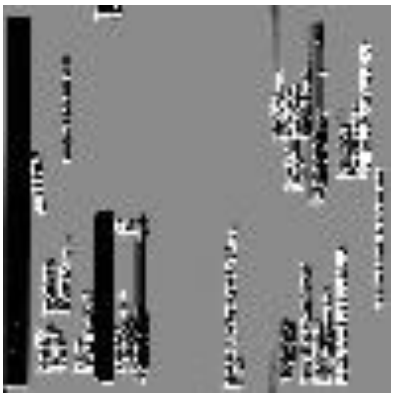


Image obtained from highly negative correlated filter.

No Image obtained from zero correlated filter.

After Layer3

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

No Image obtained from zero correlated filter.

After Layer4

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

No Image obtained from zero correlated filter.

After Layer5

Image obtained from highly positive correlated filter

Image obtained from highly negative correlated filter.

Blank Image obtained from zero correlated filter.

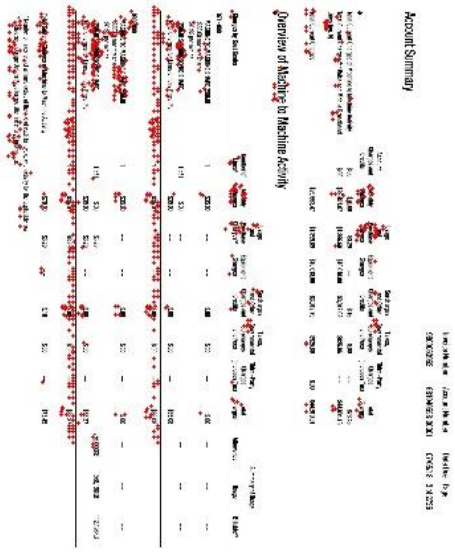
CNN Fixations Visualization

A simple yet powerful method that exploits feature dependencies between a pair of consecutive layers in a CNN to obtain discriminative pixel locations that guide its prediction. CNN Fixation are the discriminative image locations that guide the models prediction. This algorithm works by computing the discriminative locations at each layer starting from output layer [6].

ALGORITHM

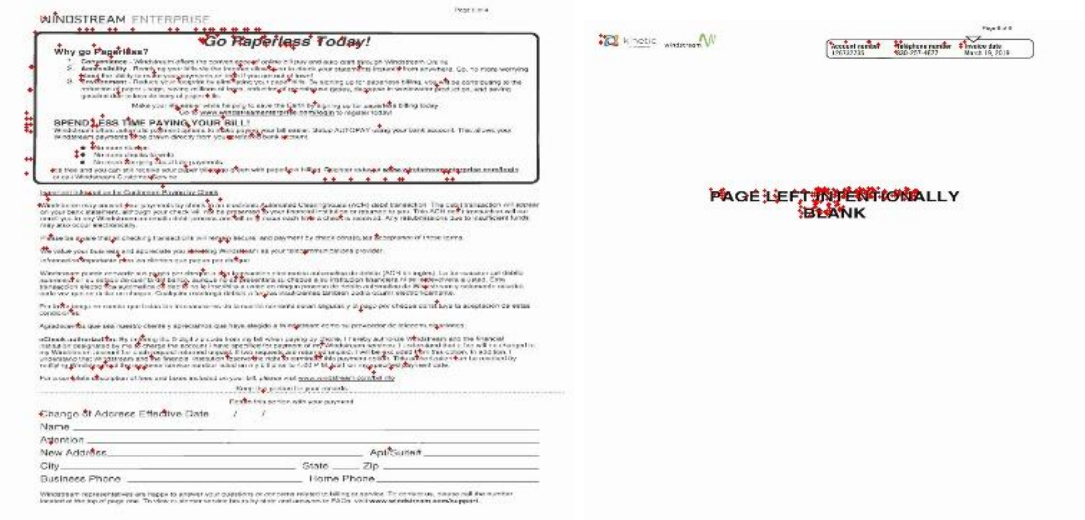
1. Get the output activation at each layer for the test image
2. The discriminative location at the final layer neuron corresponding the predicted class.
3. Starting from the final softmax layer, find the discriminative locations at each layer working back word till the input layer.
4. For fully connected layer threshold chosen is 0.1.
5. Pooling layer -> Find the location with highest activation within the receptive field in the previous layer
6. For convolutional layer , threshold is 0.5.
7. Choose the branch (skip or delta) with higher contributing activation.

RESULT



CNN Fixation points in red for image in orientation predicted as (a) -90°

CNN Fixation points in red for image in orientation predicted as (a) +90°



CNN Fixation points in red for image in orientation predicted as (a) 0°

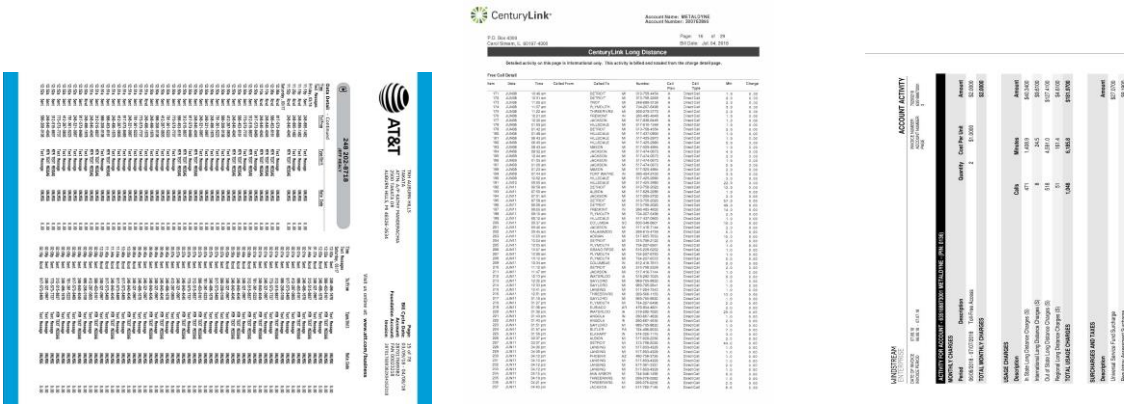
PREDICTION RESULTS

In a test dataset of 1517 images, our model predicts 1486 with correct orientation while 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.

Confusion Matrix of results on test set is shown below with accuracy of 97.956%.

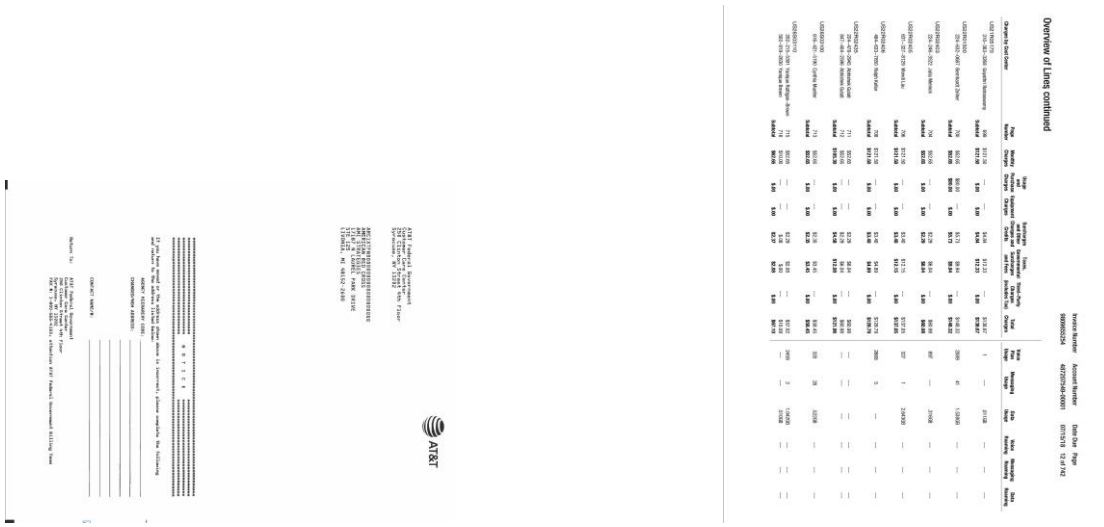
Actual \ Predicted	0°	90°	-90°
0°	500	0	0
90°	7	485	10
-90°	8	6	501

Confusion matrix



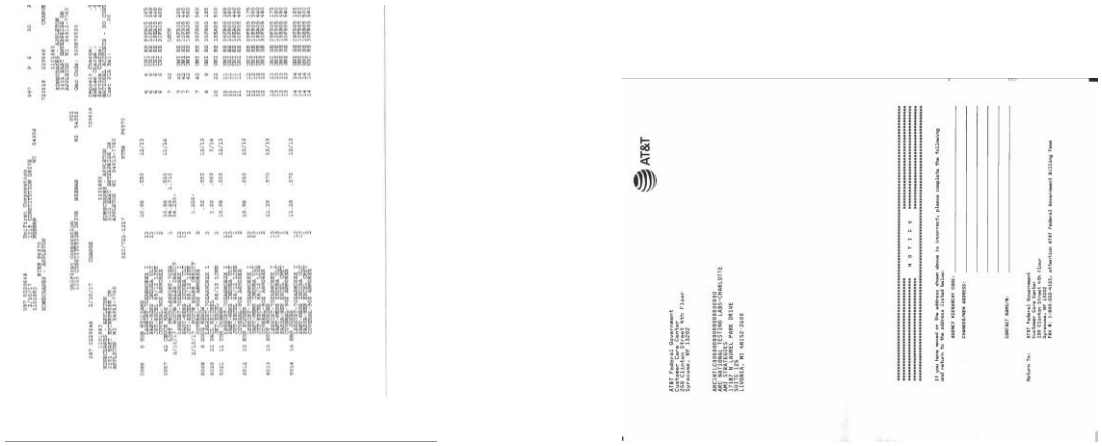
(a) (b) (c)

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



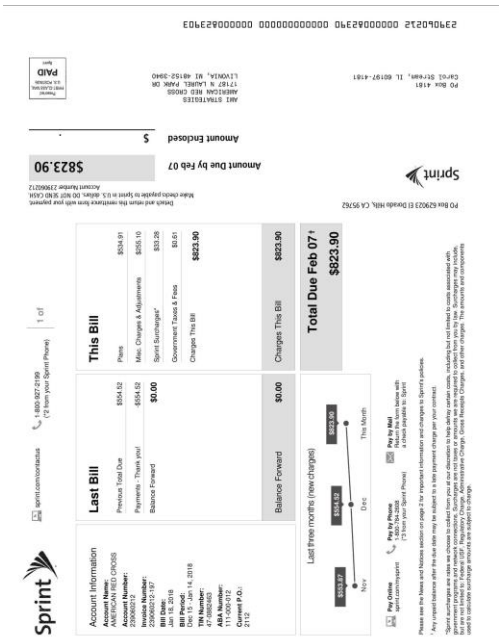
(a) (b)

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



(a) (b)

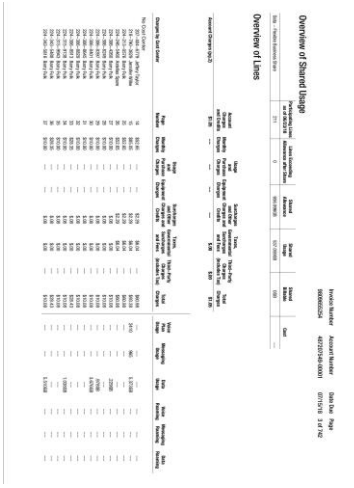
Examples of images with true orientation as +90° but predicted as -90°.



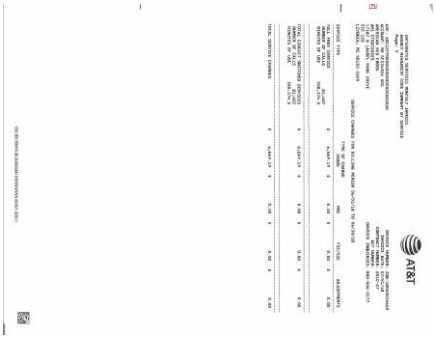
(a)

(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°.

CONCLUSION

Our model 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 depends on detecting the correct orientation of the document. There were many related work already done in this problem statement but none of them works at a page level. Also, we have accelerated to a different level with proper explanation. Proposed model achieves accuracy of approx. 98%, based on deep learning 5 layer convolutional neural network. Next, this study also tried to explain model interpretability via two algorithms. 1.) Observing intermediate visualization of image every layer and 2.) Observing the pixels responsible for predicting the class. These produced semantically meaningful results.

LIMITATIONS

1. Explanation for model interpretability in case of 90 degrees and -90 degrees is still not crystal clear by given algorithms.

FUTURE Work

1. Use interpretability results to analyse misclassified cases and propose a way to fix this either by creating more training data or other methods
2. Devise methods for model interpretability to understand how 90 degree and -90 are distinguished by the model.

REFERENCES

[1] 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, 2015, pp. 436–440.

[2] P. Fischer, A. Dosovitskiy, and T. Brox, Image Orientation Estimation with Convolutional Networks, 2015.

[3] 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, 2015, pp. 953–956.

[4] H. S. Baird, “Measuring document image skew and orientation,” Proceedings of SPIE - The International Society for Optical Engineering, 1995.

[5] R. Wang, S. Wang and J. Sun, "Offset Neural Network for Document Orientation Identification," *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, Vienna, 2018, pp. 269-274.

[6] K. R. Mopuri, U. Garg and R. Venkatesh Babu, "CNN Fixations: An Unraveling Approach to Visualize the Discriminative Image Regions," in *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2116-2125, May 2019.