Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

An empirical study of extracting embedded text from digital images

Emad Shafie *

Department of Computer and Applied Science, Applied College, Umm Al-Qura University, Mecca, Saudi Arabia

ARTICLE INFO

Article history:
Received 13 December 2022
Received in revised form
2 April 2023
Accepted 6 April 2023

Keywords:
Convoluted neural networks
Deep learning
Long short-term memory
Digital images
Text detection
Embedded information

ABSTRACT

The utilization of images as a means of transferring information is a widespread technique employed to circumvent simple detection functions that primarily focus on analyzing textual content rather than conducting thorough file examinations. This study investigates the efficacy of deep learning models in detecting embedded information within digital images. The data used for analysis was acquired from a secondary source and underwent comprehensive preprocessing. Feature extraction, sequence labeling, and predictive model training were performed using CRNN, CNN, and RNN models. Two specific models were trained and tested in this research: 1) CNN, RNN-LSTM with the Adam optimizer, and 2) CNN, RNN-GRU with the RAdam optimizer for text detection. The findings reveal that Model #1 achieved the highest F1-score during testing, with a score of 98.37% for text detection and 96.73% for word detection. The second model obtained an F1-score of 94.84% and 93.05% for text and word detection, respectively. Model #1 exhibited a word detection accuracy of 98.38% and a text detection accuracy of 96.47%. These findings indicate that the first model outperformed the second model, suggesting that employing RNN-LSTM and the Adam optimizer made a positive impact. Therefore, utilizing deep learning tools and emerging technologies is crucial for extracting textual information and analyzing visual data. In summary, this study concludes that deep learning models can be relied upon to effectively detect textual information embedded within digital images.

1. Introduction

The detection of embedded information in images has garnered significant attention from data scientists worldwide. Textual information, being the most straightforward form of human literature, holds crucial importance. Extracting such information from media is advantageous as it saves time and enhances productivity (Nozari and Sadeghi, 2021). Additionally, this information can provide further details about a scene and improve an individual's understanding of the context depicted in an image. In applied computer vision, the availability of embedded digital information proves valuable for tasks like image-based tasks, language translation, and industrial automation. Furthermore, advanced applications such as robot navigation often rely on extracting information from digital images (Shah et al., 2022). Despite notable technological advancements and widespread commercial deployments, the process of detecting and identifying text in real-world scenarios remains challenging and time-consuming in the field of machine learning.

Before the widespread adoption of deep learning in the research community, manual feature engineering was the primary focus (Verdonck et al., 2021). Overcoming various challenges, including complex backgrounds, diverse text variations, sensitivity, and interference, has been instrumental in the development of this technology. Chaotic backgrounds pose difficulties as scenes can be depicted against a wide range of backdrops such as signs, walls, glass surfaces, or even in mid-air. Some backdrops, like flashing billboards, transparent glasses, or walls adorned with patterns or strips of text, are visually distracting and can hinder text extraction from natural environments.

Extracting text from scenes with diverse typography is significantly more challenging compared to textual information found in documents, which often follow consistent rules regarding orientation, fonts, sizes, and colors. In

* Corresponding Author.
Email Address: eashafie@uqu.edu.sa
https://doi.org/10.21833/ijaas.2023.06.006
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-2041-6380

contrast, icons and text in images can vary in shape, color, font, size, and orientation. In certain situations, the text may also be embellished with patterns and LEDs. The diverse characteristics of text and interference from the background noise can make distinguishing between different entities difficult. Thus, environmental disruptions pose challenges to text identification and recognition. Additionally, lighting conditions, blurring, limited resolution, and partial occlusion have been identified as forms of interference (Shah et al., 2022).

These challenges have motivated data scientists to engage in continuous research to mitigate their impact on the predictions made by artificial intelligence. This investigation explores the application of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to extract hidden information from digital images. The primary objective is to determine the capability of these technologies in accurately predicting embedded textual information by measuring letter accuracy. The paper is structured into several sections, including an introduction, literature review, methods, findings, and commentary.

2. Literature review

Convoluted neural networks have been at the heart of detecting textual information in digital images. Huang and Xu (2019) used a U-shaped CNN architecture to take context information into account and recognize small text occurrences. In the study, the researchers directly regressed the vertical distances from a text pixel to the text boundaries without employing the typically used anchor method. Then the tiny local suggestions are joined during post-processing. Nagaoka et al. (2017), Li et al. (2021), and Wang et al. (2021) similarly employed CNN in detecting textual information from images. Li et al. (2021) obtained an F1-score of 86.5%, which was high enough to indicate the model's effectiveness.

Kang et al. (2022) were inspired by the inadequacies of pre-existing models, where they sometimes would fail to locate and correctly identify adjacent textual information. The researchers settled for a multi-level, which they called a multi-level residual feature pyramid network. The model produced precision=80.87%, recall=75.77%, f1-score=78.24%. Another study that considered the multi-level feature approach is the study of Wang et al. (2022). The researchers adopted this model to rid the inconsistencies in the information produced by multi-scale feature maps. The results were impressive, scoring precision=82.68%, recall=88.44%, and f-score=85.46%. Xiao et al. (2021) modeled scene text detection from a multi-directional point of view to alleviate low detection rates of some texts. Results revealed that the resultant model was an improvement upon YOLOv3. Specifically, the model scored precision=86.2%, recall=81.9%, and f1-score=84.0%.

The research by Li et al. (2021) and Raisi et al. (2020) was similar to Xiao et al. (2021) because the researchers used deep learning techniques to solve the problem of detecting multi-oriented text on digital images.

Some studies have engaged in scene text detection using the attention module to enhance model accuracy. Cao et al. (2020) applied this technique to detect multi-oriented text, and their findings suggested that including the attention module significantly increases the f1-score from 34.2% to 84.9%. Similarly, Wang et al. (2021) acknowledged that the attention mechanism significantly increases a model's performance. Their test produced recall=71.7%, precision=77.1%, f-measure f-score=74.3%. Research into the detection of curved text has also been rife.

Zhao et al. (2021) developed an instance-aware model while providing some features that analyze textual curvature. Their findings suggest that the accuracy scores increased significantly when the model accounted for curvature. The scores were precision=88.2%, recall=79.6%, and accuracy=83.7%. Another study that developed a deep learning model around the textual curvature problem is the study of Li et al. (2020). Results from their investigation were also impressive, as they reported that the maximum error was 0.953% when using different shooting angles. On the other hand, the minimum margin of error was reported as 0.295% when there was uniform illumination.

Huang et al. (2020) acknowledged that scene text detection for horizontal text is simple to decipher. However, text occurring in its natural environment may need some processing before applying predictive models. The text position correction module converts naturally occurring text into horizontal text. Afterward, the encoder-decoder module is applied to predict the text. The approach produced an accuracy metric of accuracy=96.5%.

3. Methodology

3.1. Data source

The research used secondary data from the research of Jaderberg et al. (2016). The source constructed a dataset comprising nine million images with their corresponding textual representation. All images are in grayscale with labels corresponding to the actual words adorning them. The researcher downloaded these 10 GB data from https://www.robots.ox.ac.uk. The text files contained in the downloaded dataset were annotation_text.txt, annotation_val.txt, and annotation_train.txt. The source constructed the dataset by using the VGG (Visual Geometry Group. It is a deep neural network with either 16 or 19 layers used in image classification and recognition. From the data, the researcher sampled 100,000, 6,000, and 7,500 images for training, validation, and testing, respectively.

3.2. Data preparation

The researcher began by putting the images into their respective folders so that programmed access could be made easier. The next step was to pull the text labels from the picture files and capitalize them before moving on to the next phase. After that, a.csv file was made, and the columns of that file matched the names of the photographs and the text labels that went along with them. The procedure was carried out many times for each of the test, val, and train datasets. After that, the researcher checked for picture sizes by using the OpenCV Python package. It was required to go through this procedure in order to enable remodeling of the width and height of the photos without causing any distortion to the quality or aspect ratio of the photographs. The objective was to adjust the height of each picture to fall somewhere between 30 and 31 inches.

3.3. Overall approach

Convolutional RNN, or CRNNs for short, were the models applied to solve the problem. It integrates Deep CNN, also known as DCNN, with RNN, resulting in a complete system for sequence recognition (Chang et al., 2021). One of the three components that comprise the model is the transcription layer. At this point, the convolutional and recurrent layers are the only two components of the network that are still intact. The system works such that each picture that is sent into the system is analyzed by the convolutional layers. The output of this process is the production of a feature sequence consistent with the input image. The subsequent step is creating a recurrent network, which is responsible for making frame-to-frame predictions. It is important for this process to be complete because it is the source of the model's accuracy (Al-Saffar et al., 2021). Although a CRNN consists of two separate network topologies (a DCNN and an RNN), it is feasible to train it concurrently using a single loss function. This attribute is owing to the hybrid nature of the CRNN model.

3.4. Modelling

The researcher trained two models to determine which would deliver more accurate results. The first model used convoluted neural networks (CNN) and a bi-directional long short-term memory as its preferred RNN. The RNN component was trained using the Adam optimizer. The second model was similar to the first, except that it considered Bi-directional GRU as its preferred RNN. Also, the researcher used RAdam and Adam to optimize the trained models. For this study, English alphabet words and numbers were the basis of model detection. The training stage involved putting the image, its labels, length, and label features before outputting the instance's Connectionist Temporal Classification (CTC) loss (Kang et al., 2021). The output is a measure of performance used to test RNNs, which in this case is the LSTM. Training of the model ran for 20 epochs with the option of early stopping. The validation process also involved similar procedures.

4. Results

4.1. Word and letter accuracy

Findings established that the word and letter accuracy scores obtained from running the models were acceptably high during validating and testing procedures. During the validation phase, Model #1 achieved a word accuracy score of 96.75%, while during the testing phase, it achieved 98.38%. On the other hand, Model #2 achieved a word accuracy of 92.13% and an accuracy of 94.88% throughout testing. Both models had testing accuracies that were greater than their validation accuracies, which is an indication of their strength when applied to the task of detecting unknown textual content.

The researcher was also concerned with another parameter, which was letter correctness. It was the overall average degree of accuracy with which the models accurately predicted each letter. Based on the findings presented in Table 1, it seemed that Model #1 performed better than Model #2. During the validation process, the first model had an accuracy rate of 94.93%, whereas the other model was only able to accurately predict all of the letters 92.72% of the time. During the testing phase, both Model #1 and Model #2 achieved letter accuracy scores of 96.47% and 93.05%, respectively. These findings represent an improvement over the previous phase.

Table 1: Word and letter accuracy statistics

Model	Data	Word accuracy	Letter accuracy
Model #1	Val	96.75%	94.93%
Model #1	Test	98.38%	96.47%
Model #2	Val	92.13%	92.72%
Model #2	Test	94.88%	93.05%

4.2. Text detection and matching metrics

Other metrics consulted to evaluate model performance were F1-score, precision, and recall. Table 2 summarizes these results. Whenever the model would attempt to read textual information from an image, there were two outcomes; either to detect it as text or not a text. After detecting it as a text, the next test was to match it correctly with its textual representation.

4.2.1. Text detection metrics

In this section, the goal was to determine the accuracy with which the models detected embedded in the image as text. Findings suggest that both models performed highly in detecting embedded information as text. Model #1's performance was higher, as it scored an F1-score of 98.37% in testing. Model #2's testing accuracy was 94.84%. Other

metrics are shown in Table 2. These findings imply that the models developed in this study are capable of telling textual from non-textual information with minimal chances of error. Nevertheless, Model #1 significantly outperforms Model #2.

Table 2: Text detection metrics

Model	Data	Recall	Precision	F1-score
Model #1	Val	96.25%	97.22%	96.73%
Model #1	Test	98.00%	98.74%	98.37%
Model #2	Val	91.00%	93.09%	92.04%
Model #2	Test	94.25%	95.44%	94.84%

4.2.2. Text prediction/matching metrics

In this section, the goal was to determine the accuracy with which the models matched the embedded textual information with its correct textual representation. It is one thing to tell textual from non-textual information and actually predicts the specific letter captured in an image. Findings suggest that the models performed well, especially on the precision metrics. Model #1 scored a precision of 94.00% in validation and 96.00% in testing. On the other hand, the sister model obtained 88.75% in validation. It implies that the models managed to correctly match at least 92% of the embedded letters to their true textual representation. Table 3 shows that their accuracy was also high, as it was 95.4% and 92.3% for Model #1 and Model #2, respectively.

Table 3: Text prediction metrics

Model	Data	Recall	Precision	F1-Score
Model #1	Val	94.00%	96.16%	95.07%
Model #1	Test	96.00%	97.46%	96.73%
Model #2	Val	88.75%	91.97%	90.33%
Model #2	Test	93.00%	93.70%	93.35%

4.3. Confusion matrix for the models

The researcher computed and produced a confusion matrix showing the average percentage of correct and wrong hits in Table 4 and Table 5. The values of the cells represent the average percentage of the predictions made in favor of the labels that correspond to the vertical and horizontal positions. For instance, the fact that the first cell's value is 98.00% indicates that the model properly identified embedded text since text can be deduced from that number. In the portion that compares the picture to the text, the first cell in this section suggests that the model accurately predicted words from the image 96.00% of the time.

Table 4: Confusion matrix for text detection

	Model #1			Model #2	
	TRUE	FALSE		TRUE	FALSE
TRUE	98.00%	2.00%	TRUE	94.25%	5.75%
FALSE	1.25%	98.75%	FALSE	4.50%	95.50%

Table 5: Confusion matrix for text prediction/matching

	Model #1			Model #2	
	TRUE	FALSE		TRUE	FALSE
TRUE	96.00%	4.00%	TRUE	93.00%	7.00%
FALSE	2.50%	97.50%	FALSE	6.25%	93.75%

5. Discussion

The study was successful in building models to detect textual information from images. Both models were able to recognize text from photos and accurately match it with its textual representation, according to the data, which suggested that both models generated statistically significant results. The first model used Convolutional Layers, LSTM Units for RNN, and an Adam optimizer in order to interpret information that was concealed inside pictures. This was the primary distinction between the two models. In order to achieve a similar objective as the first model, the second model used Convolutional Layers, GRU Units for RNN, and a RAdam optimizer. The multi-level technique that was used by Kang et al. (2022) in their study is virtually identical to the one that is being taken here. The F1 score that the referenced work achieved was 78.24%, which is lower than the score that the present research achieved, which was 96.73%. Wang et al. (2022) also used this multi-level technique, and the researchers there achieved a score of 85.46% on their F1-score. Their model continues to have metrics that are inferior to those of Model #1 despite their best efforts.

The investigation revealed that the first model outperformed the second model by a significant margin in all the associated metrics. It seems that integrating LSTM into the model was a brilliant decision, as it promoted performance. According to Levy and Schiller (2021), LSTM is an effective technique in deep learning because it ensures that the researcher has access to a wide variety of LSTM parameters, including learning rates, as well as input and output biases. As a result, there is no need for precise modifications. A benefit of using LSTMs is that the complexity required to update each weight is decreased to O(1), which is an improvement compared to that of Back Propagation Through Time (BPTT). Additionally, the Adam optimizer used in the first model was effective in promoting its performance, relative to the RAdam optimizer adopted in Model #2.

Multi-directional scene detection played a critical role in building the models established in this study. While the researcher settled for a bi-directional approach, it still counts as a multi-directional Feature Fusion. Li et al. (2022) also adopted a bi-directional approach and their results were also promising. They obtained an F1-score of 83.6% and a recall of 78.2%. These metrics are lower than the current study's F1-score of 96.73% and a recall of 96.00%. Xiao et al. (2021) also modeled scene text identification from a multi-directional point of view in order to alleviate the poor detection rates of particular texts. The findings showed that the resulting model was an advancement over YOLOv3, which was the previous version. Their F1-score of 84.0% is testament to the effectiveness of a multi-directional approach to feature fusion. Furthermore,

it shows that deep learning techniques are applicable in detecting multi-oriented text on digital images.

The results have provided evidence that it is possible for artificial intelligence systems to read complex text on images regardless of whether the text is horizontal or twisted. Huang et al. (2020) admitted that scene text recognition for horizontal text is not difficult to understand. However, digesting the material in its natural setting can be necessary before applying predictive models to it. The text position correction tool will turn vertical text that was generated naturally into horizontal text. Following that, the encoder-decoder module is used so that the text may be predicted. Similarly, Li et al. (2020) found that it is difficult to deal with tiny targets and fix very imbalanced data, but most networks have a positive influence on the balancing of target samples in text identification. Huang et al. (2020) contended that the majority of the currently available deep learning models are able to address the issue of horizontal text recognition; nevertheless, the text that is seen in real scenes is often slanted and uneven, and there are still many difficulties that have not been solved.

6. Conclusion and future work

In conclusion, this study has demonstrated the efficacy of deep learning models in detecting embedded information within digital images. Two distinct models were employed in this research, specifically: 1) CNN, RNN-LSTM, Adam optimizer, and 2) CNN, RNN-GRU, RAdam optimizer, for the purpose of addressing the challenge of text detection from digital images. The findings indicate that the first model outperforms the second model in terms of efficiency, as evidenced by higher accuracy, f1-scores, and other relevant metrics. These nearly flawless scores surpass the outcomes reported in prior studies conducted on the same subject matter. Consequently, it is evident that technologies based on deep learning can be relied upon for analyzing visual data and extracting textual information. It is worth noting that the scope of our analysis was limited to textual content within digital photos, and no attempts were made to identify other types of objects or images. Nonetheless, this promising area of study holds potential for future leaders in the field. Furthermore, it is important to acknowledge that the experiment was conducted with a small sample size of models. Consequently, future researchers should consider constructing and evaluating a larger number of models to determine the most effective ones.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Al-Saffar A, Awang S, Al-Saiagh W, Al-Khaleefa AS, and Abed SA (2021). A sequential handwriting recognition model based on a dynamically configurable CRNN. Sensors, 21(21): 7306. https://doi.org/10.3390/s21217306 **PMid:34770612 PMCid:PMC8587523**

Cao Y, Ma S, and Pan H (2020). FDTA: Fully convolutional scene text detection with text attention. IEEE Access, 8(1): 155441-155449. https://doi.org/10.1109/ACCESS.2020.3018784

Chang L, Li D, Hameed MK, Yin Y, Huang D, and Niu Q (2021). Using a hybrid neural network model DCNN–LSTM for image-based nitrogen nutrition diagnosis in muskmelon. Horticulturae, 7(11): 489. https://doi.org/10.3390/horticulturae7110489

Huang C and Xu J (2019). An anchor-free oriented text detector with connectionist text proposal network. In the 11th Asian Conference on Machine Learning, PMLR, Nagoya, Japan, 101: 631-645. https://doi.org/10.1145/3318299.3318373

Huang Z, Lin J, Yang H, Wang H, Bai T, Liu Q, and Pang Y (2020). An algorithm based on text position correction and encoder-decoder network for text recognition in the scene image of visual sensors. Sensors, 20(10): 2942. https://doi.org/10.3390/s20102942 **PMid:32455941 PMCid:PMC7285298**

Jaderberg M, Simonyan K, Vedaldi A, and Zisserman A (2016). Reading text in the wild with convolutional neural networks. International Journal of Computer Vision, 116: 1-20. https://doi.org/10.1007/s11263-015-0823-z

Kang J, Ibrayim M, and Hamdulla A (2022). MR-FPN: Multi-level residual feature pyramid text detection network based on self-attention environment. Sensors, 22(9): 3337. https://doi.org/10.3390/s22093337 **PMid:35591028 PMCid:PMC9102995**

Kang X, Huang H, Hu Y, and Huang Z (2021). Connectionist temporal classification loss for vector quantized variational autoencoder in zero-shot voice conversion. Digital Signal Processing, 116: 103110. https://doi.org/10.1016/j.dsp.2021.103110

Levy I and Schiller D (2021). Neural computations of threat. Trends in Cognitive Sciences, 25(2): 151-171. https://doi.org/10.1016/j.tics.2020.11.007 **PMid:33384214 PMCid:PMC8084636**

Li X, Liu J, Zhang G, Huang Y, Zheng Y, and Zhang S (2021). Learning to predict more accurate text instances for scene text detection. Neurocomputing, 449: 455-463. https://doi.org/10.1016/j.neucom.2021.04.035

Li Y, Silamu W, Wang Z, and Xu M (2022). Attention-based scene text detection on dual feature fusion. Sensors, 22(23): 9072. https://doi.org/10.3390/s22239072 **PMid:36501774 PMCid:PMC9739706**

Li Z, Zhou Y, Sheng Q, Chen K, and Huang J (2020). A high-robust automatic reading algorithm of pointer meters based on text detection. Sensors, 20(20): 5946. https://doi.org/10.3390/s20205946 **PMid:33096701 PMCid:PMC7589492**

Nagaoka Y, Miyazaki T, Sugaya Y, and Omachi S (2017). Text detection by faster R-CNN with multiple region proposal networks. In the 14th IAPR International Conference on Document Analysis and Recognition, IEEE, Kyoto, Japan, 6: 15-20. https://doi.org/10.1109/ICDAR.2017.343

Nozari H and Sadeghi ME (2021). Artificial intelligence and machine learning for real-world problems (A survey). International Journal of Innovation in Engineering, 1(3): 38-47. https://doi.org/10.59615/ijie.1.3.38

Raisi Z, Naiel MA, Fieguth P, Wardell S, and Zelek J (2020). Text detection and recognition in the wild: A review. ArXiv Preprint ArXiv:2006.04305. https://doi.org/10.48550/arXiv.2006.04305

Shah D, Osinski B, Ichter B, and Levine S (2023). LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action. ArXiv Preprint ArXiv:2207.04429. https://doi.org/10.48550/arXiv.2207.04429

Verdonck T, Baesens B, Óskarsdóttir M, and vanden Broucke S (2021). Special issue on feature engineering editorial. Machine Learning, 1-12. https://doi.org/10.1007/s10994-021-06042-2

Wang X, Zheng S, Zhang C, Li R, and Gui L (2021). R-YOLO: A real-time text detector for natural scenes with arbitrary rotation. Sensors, 21(3): 888. https://doi.org/10.3390/s21030888 **PMid:33525619 PMCid:PMC7865800**

Wang Y, Mamat H, Xu X, Aysa A, and Ubul K (2022). Scene Uyghur text detection based on fine-grained feature representation. Sensors, 22(12): 4372. https://doi.org/10.3390/s22124372 **PMid:35746154 PMCid:PMC9229707**

Xiao L, Zhou P, Xu K, and Zhao X (2021). Multi-directional scene text detection based on improved YOLOv3. Sensors, 21(14): 4870. https://doi.org/10.3390/s21144870 **PMid:34300607 PMCid:PMC8309843**

Zhao F, Shao S, Zhang L, and Wen Z (2021). A straightforward and efficient instance-aware curved text detector. Sensors, 21(6): 1945. https://doi.org/10.3390/s21061945 **PMid:33802093 PMCid:PMC8000375**