

DETECTION AND RECOGNITION OF TEXTS FEATURES FROM A TOPOGRAPHIC MAP USING DEEP LEARNING

Ashis Pradhan^{1*}, Shaikh Yusuf Niaz², Mohan P. Pradhan³, and Ratika Pradhan⁴

Received: July 08, 2021; Revised: September 18, 2021; Accepted: October 02, 2021

Abstract

The aim of this research is to digitize texts present on topographic maps. Digitization is critical for information processing, hence texts on topographic maps that are in pixel format are digitized. The above is achieved using deep learning technology. Detection and recognition deep learning models are leveraged for first detecting texts on topographic maps and then recognition the detected texts. Detection model is inspired by Faster Region Based Convolutional Neural Networks (Faster R-CNN) to achieve real time text detection using region proposal networks and recognition model is inspired by Convolutional Recurrent Neural Network (CRNN), which is an end to end trainable neural network to achieve image-based sequence recognition that is suitable for recognizing texts on character level. The models have been trained and validated on self-curated dataset of topographic maps. The digitized information may be very helpful for many GIS applications like automated construction of Digital Elevation Model, Digital Surface Model, etc.

Keywords: Digitization, topographic map, faster R-CNN, convolutional recurrent neural network, text detection and text recognition

Introduction

Information is a weapon in present time. Retrieving and manipulating data has become crucial in any business or academic domain. Although the above tasks could be done by human, but on a large scale would cost time and human power. Detecting and recognizing texts on an image could be done by human and stored in alphanumeric form in a software that would be machine interpret-able and hence manipulate-able. To remove the human intervention, intelligent algorithms are developed

that leverage machines to detect and recognize texts and store the alphanumeric or digitized form in a software automatically. The purpose of developing intelligent algorithms here is because it is almost not possible to retrieve information from images using a set of rules that are not known to human itself. In addition, detecting a text from a topographic map and recognize it is a challenging task as these features overlap with other topographic features. Some samples of topographic map containing text

¹ Department of CSE, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Sikkim, India. E-mail: ashis.p@smit.smu.edu.in, (Orchid ID 0000-0002-9111-6257)

² Department of CSE, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Sikkim, India. E-mail: yusufmitb@gmail.com

³ Department of Computer Application, Sikkim University, Sikkim, India. E-mail: mppradhan@cus.ac.in

⁴ Department of CSE, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Sikkim, India. E-mail: ratika.p@smit.smu.edu.in

* Corresponding author

features is shown in Figure 1. Bridging the gap between information in pixel format that is not machine understandable to digitized format that is machine understandable or manipulate-able has been a challenging computer vision task for a while. With the advent of deep learning and gradual increase in computational power researchers have developed algorithms that can do the above.

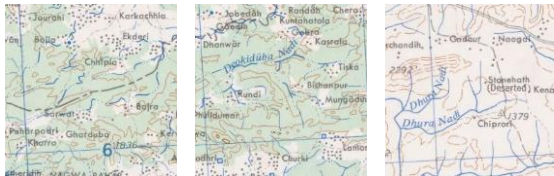


Figure 1. Some samples of Topographic Map

Traditional statistical machine learning algorithms lacked accuracy and required handcrafted pipeline for doing image processing. Convolutional neural networks gained popularity due to the fact that the kernel values are learned rather than relying on handcrafted kernels for extracting features and thereby dropping the requirement of handcrafted kernel pipeline.

This paper focuses on retrieving information like text from topographic maps that could benefit geographic information systems like aviation, planning, disaster management, etc. On a lower level, the robust methodology for text detection and recognition present in a topographic map is proposed and bring the information to digitized form that can be easily manipulated and used efficiently. To achieve this, texts present on topographic maps is detected or localized and then the localized region is further processed to identify the text present in that region. Detection Model is inspired by Faster R-CNN (Ren *et al.*, 2016) that incorporates region proposal network on Fast R-CNN (Girshick, 2015) to achieve real time text detection. Recognition model is inspired by CRNN (Shi *et al.*, 2015) which is an end to end trainable neural network to achieve image-based sequence recognition that is suitable for recognizing texts on character level.

Related Work

Detection

Deep Convolutional Neural Networks (DCNN) have been applied on computer vision tasks like detection, segmentation and classification of objects (Krizhevsky *et al.*, 2012; Girshick *et al.*, 2014). Overfeat (Sermanet *et al.*, 2014) runs sliding

windows of different scales on an image and classifies a region using an classifier and regress the bounding box coordinates. This method is computationally expensive and time consuming. Region based Convolutional Neural Network or R-CNN (Girshick *et al.*, 2014) incorporated region based proposal technique for object detection which further lead to a family of networks like Fast R-CNN and Faster R-CNN (Girshick *et al.*, 2014; Girshick, 2015). R-CNN (Girshick *et al.*, 2014) applies selective search (Uijlings *et al.*, 2013) on an image and proposes 2,000 regions that are passed through a Convolutional Neural Network (CNN) for feature extraction and evaluated for classification and bounding box regression. Selective search is not a part of the neural network. Graph based image segmentation method initially hypothesize set of regions which are then hierarchically grouped based on similarity measures like color, texture, size and region-filling. Bottleneck of R-CNN (Girshick *et al.*, 2014) is in computing feature maps of each 2,000 region proposals. Fast R-CNN (Girshick, 2015) compensates the bottleneck of R-CNN (Girshick *et al.*, 2014) by first generating feature map of the image through a CNN and a separate pipeline that does selective search on the image to produce 2,000 region proposals that are mapped to the feature map using Region of Interest (ROI) pooling (Girshick, 2015). Fast R-CNN (Girshick, 2015) reduces test time of an image by R-CNN (Girshick *et al.*, 2014) on a magnitude of 10. Faster R-CNN (Ren *et al.*, 2016) is an advancement on Fast R-CNN (Girshick, 2015) which incorporates Region Proposal Network (RPN) that replaces selective search pipeline that was offline i.e. not trainable to reduce computation time while testing. A RPN (Ren *et al.*, 2016) that gives region proposals is a fully connected neural network that does binary classification of whether a region is object or not and hence assigns confidence or objectness score. The regression end of the fully connected layer predicts offset values of the proposals from the ground truth region. Prior to the fully connected layer the RPN also has CNN as it takes feature maps as input (Ren *et al.*, 2016). The network is alternatively trained in a manner that CNN layers of RPN and detector network are shared (Ren *et al.*, 2016). Detector networks are not bound to just region proposal based networks. Dense sampling based networks (Liu *et al.*, 2016; Redmon *et al.*, 2016; 2017; 2018) are one stage detection network that detect objects over dense sampling of possible locations.

Text detection on topographic map could be carried out by any above network but the choice of network is crucial. Dense sampling based detector networks can fall short of accuracy but are much

faster than region based proposal detector networks (Huang *et al.*, 2017) where parameters like object size, input image resolution, number of proposals, memory, GPU speed that contribute towards accuracy and speed, was considered. Huang *et al.* (2017) concluded that Faster R-CNN (Ren *et al.*, 2016) using Inception Resnet (Szegedy *et al.*, 2016) as feature extractor with 300 region proposals attained highest mean average precision (mAP). Therefore, for detecting text on topographic maps Faster R-CNN architecture is adopted.

Recognition

Text by virtue is a sequential data where every character contributes to the context. CNNs predict only a single label to an input and also expect the input of a fixed size. Thus, sequential data will require a network that predicts a series of labels over time and also which can accept inputs of variable lengths. DCNN were heavily used to recognize texts, where (Wang *et al.*, 2012; Bissacco *et al.*, 2013) used the idea of detecting characters in a text and segmenting each to be classified against a DCNN model trained in on characters. Also another plausible method would be to recognize texts by classifying the entire text image over a DCNN that was implemented by (Jaderberg *et al.*, 2014). In Wang *et al.* (2012); Bissacco *et al.* (2013) and Jaderberg *et al.* (2014) implementations would require a strong classifier that would accurately predict a class of a character. The models would likely fail to generalize on classifying texts belonging to vocabulary of huge number of characters. Jaderberg *et al.* (2014) highlighted network for predicting text for a language that would require training the DCNN on an entire set of vocabulary which is computationally expensive as well would suffer accuracy. Therefore, from analysis it was noted that DCNN are not a viable option for text recognition. Classifications tasks on sequential data are commonly done using Recurrent Neural Networks (RNN). A RNN can accommodate dependence between inputs and variability in size of input. In case of image containing text the RNN expects to accept features of the image sequences. The extraction of features for image feature sequenced inputs had been handcrafted using image processing techniques as adopted by (Graves *et al.*, 2009; Su and Lu, 2014). Convolutional Recurrent Neural Networks (CRNN) (Shi *et al.*, 2015) replaces the offline image processing task of extracting features for image feature sequenced input with CNNs hence makes the entire network end to end trainable.

The CRNN (Shi *et al.*, 2015) is adopted for doing text recognition on topographic map. The three component architecture with CNN for feature extraction, RNN for predicting labels per frame of input feature sequence and a top transcription layer to unify the prediction labels into a single label. All three layers are well justified for carrying out image-based sequence recognition. The CNN layer encodes representations of a text image. The input size of CNN is fixed, however the character representations of variable length input image sequence are not lost since representations are encoded depth wise through collection of feature maps hence making it invariant of the fact of fixed input size. The feature sequence is passed through a deep bi-directional RNN i.e. a stack of bidirectional Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997; Gers *et al.*, 2002; Graves *et al.*, 2013) which stores contexts of past predictions in both directions thereby making it more suitable for image based text recognition. RNNs suffer from vanishing gradient descent problem and cannot store long range contexts hence LSTMs are used (Bengio *et al.*, 1994).

Experimental Results

Text Detection

Dataset for text detection was curated by cropping 255×255 resolution 200 images out of topographic maps available in digital format. Along with images the dataset contains annotations of texts on an image i.e. the minimum and maximum (x, y) coordinates that describe position of texts in the image which follows the standard COCO dataset format for detection. Texts annotated included numbers and words. Biasness in dataset is inherently not present since we just have one class to predict i.e. texts and no text is left unannotated.

The detection model trained is a Faster R-CNN (Ren *et al.*, 2016) with resnet-50/101-feature pyramid network (R-50-FPN/R-101-FPN) (He *et al.*, 2015; Lin *et al.*, 2017) as the backbone of the detection model i.e. as the feature extractor. The entire model as a whole was adopted from Detectron2 with the

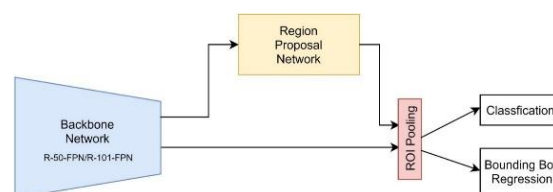


Figure 2. Detector pipeline

backbone of the network being pretrained on Imagenet Database (Deng *et al.*, 2009) with over 1,000 object classes including texts. Figure 2 is a high level pipeline of the detector model. Since the backbone is already pretrained, hence just fine tuning it with the dataset curated was enough. Fine tuning the feature extractors along with the object classification and bounding box regression ends was part of the training. The alternate training method of sharing CNN weights between the RPN and the detector network (Ren *et al.*, 2016) is taken care by the Detectron2 framework. The model is trained on 180 images and validated on 20 images. Table 1 displays the network and hyper parameters adjusted and experimented to get maximum average precision (AP) on the validation set or otherwise avoid overfitting of data. Optimizer used was Stochastic Gradient Descent (SGD), Cross Entropy Loss for classifier and Smooth L1 Loss for the bounding box regression as loss functions.

Table 2 displays network vs metrics. The inference in Table 2 is according to the COCO metric evaluation standard for detection. Other than AP on validation set, it is also relevant to calculate the total loss and false negative classification on the training set. Faster R-CNN (Ren *et al.*, 2016) incorporates multitask loss hence total loss is the combination of loss on object classifier and bounding box regression ends. False negative metric gives the idea of how many regions in actual were texts were not classified as texts. Figures 3, 4, 5, and 6 are the total loss vs epoch, false negative score vs epoch for Faster RCNN R-50-FPN and Faster RCNN R-101-FPN networks, respectively. The oscillating value of total loss seen in Figure 5 could be avoided by scheduling learning rate to decay for consequent epochs and is left for further experimentation. Training time with R-50-FPN (50 layers) took around 14 min and with R-101-FPN (101 layers) took around 35 min on a 12GB NVIDIA Tesla K80 GPU with 4 workers. Faster

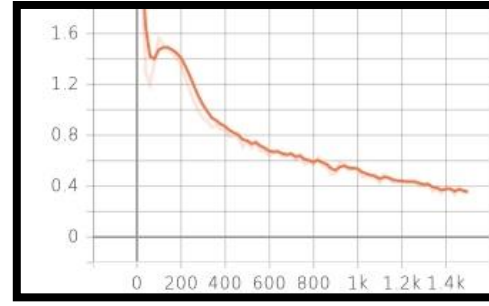


Figure 3. Total loss Vs epoch (Faster RCNN + R-50-FPN)

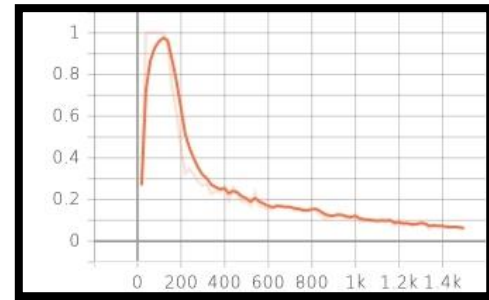


Figure 4. False Negative Vs Epoch (Faster RCNN + R-50-FPN)

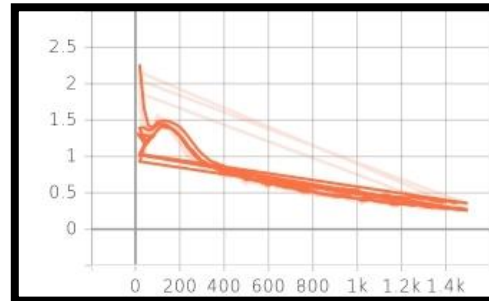


Figure 5. Total Loss Vs Epoch (Faster RCNN + R-101-FPN)

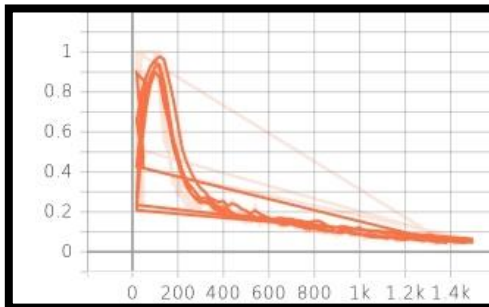


Figure 6. False Negative Vs Epoch (Faster RCNN + R-101-FPN)

Table 1. Detector network and hyper-parameters

Network	Backbone	Learning rate	Epoch	Region Proposals
Faster RCNN	R-50-FPN	0.001	1500	256
Faster RCNN	R-101-FPN	0.001	1500	256

Table 2. Average precision (IoU >= 0.7)

Network	Metrics				
	AP	AP50	AP75	APs	APm
Faster RCNN + R-50-FPN	76.04	98.95	94.08	76.12	76.30
Faster RCNN + R-101-FPN	79.61	98.99	97.89	80.30	78.77

RCNN + R-101-FPN returns best AP value of 79.6% against Intersection over Union (IoU) value ≥ 0.7 .

Text Recognition

Dataset for text recognition was curated by cropping out regions on topographic map only containing text (limited to words only). 500 images were collected and annotated. Annotation of an image is a single word containing combination of characters of the English alphabet. Texts in images with non-English alphabet characters and numbers are not considered for data collection.

Model of CRNN (Shi *et al.*, 2015) is adopted for recognizing text. Images of variable dimension are resized to 50×200 pixel. 450 images are used for training and remaining 50 for validation. A pretrained Resnet-18 (He *et al.*, 2015) without classifier head is used as the feature extractor layer. The feature sequence is broken down into a set number of frames i.e. the input size of the consequent RNN layer. A forward and a backward LSTM are stacked together to form a bidirectional LSTM which supplies contexts in both directions. This layer predicts a series of labels corresponding to per frame of feature sequence input. The LSTM has 256 hidden layers which is set as an hyper parameter while training. In between the feature extractor and sequence predictor layers is a map to sequence network as described in (Shi *et al.*, 2015) for turning feature sequence into feature maps for propagating error differentials from recurrent to convolutional layers. The top layer is a lexicon free based transcription layer that unifies per frame feature sequence prediction into a single label. The single label will be the recognition of text in image. Predicting a label sequence adopts a Connectionist Temporal Classification (CTC) loss as proposed in (Graves *et al.*, 2006). CTC basically maximizes the likelihood of a label sequence i.e. the ground truth conditioned over per frame predicted label. The network is designed to be end to end trainable. Error differentials in recurrent layers are calculated using Back Propagation through time (BPTT), in convolutional neural network layers

using back propagation and in transcription layer using a forward backward algorithm (Graves *et al.*, 2006). SGD is used as the optimizer with momentum and the objective function is the CTC loss (Graves *et al.*, 2006). Table 3 displays the best hyper parameters experimented over multiple

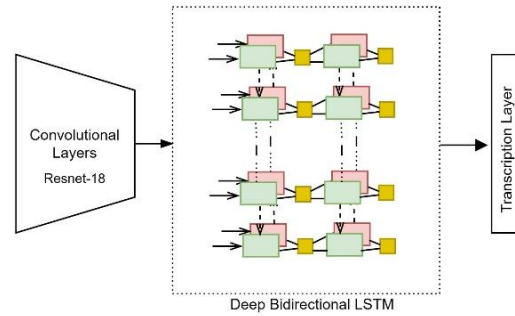


Figure 7. Recognizer pipeline

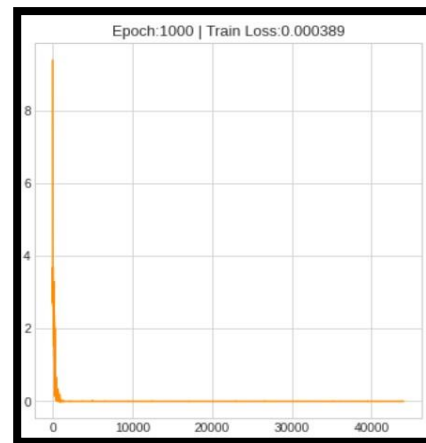


Figure 8. Training Loss vs (Epoch*Epoch Length) (CRNN)

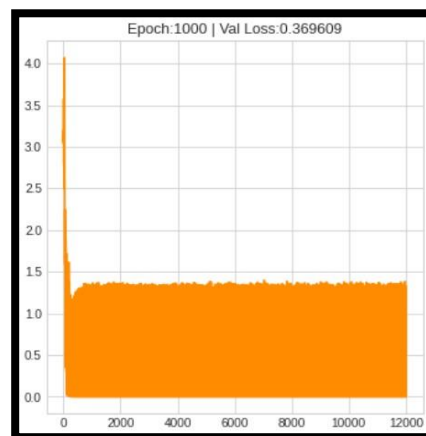


Figure 9. Validation Loss vs (Epoch*Epoch Length) (CRNN)

Table 3. CRNN Hyper-parameters

LR	E	WD	M	DO	CN	BD	HS
0.02	1,000	1e-5	0.9	0.1	5	True	256

Where, LR = Learning Rate, E = Epoch, ED = Weight Decay, M = Momentum, DO = Drop-out, CN = Clip Norm, BD = Bi-directional and HS= Hidden Size

Table 4. CRNN Accuracy

	Accuracy
Validation set	45%
Train set	60%

trainings to achieve maximum accuracy on validation set. The network is limited to English vocabulary of capital, small alphabets and a blank. Figure 7 shows the recognizer pipeline.

Table 4 displays the train and validation set accuracy. On training the network for more than 1,000 epochs the validation accuracy remained unchanged while the training loss plateaued, Figure 8. From Figure 9 the validation loss seemed to oscillate with no significant change. The cropped images from topographic maps were low resolution and required to be resized to higher dimension for training. The dataset contains low dimensional images having few characters which might get pixelated on resizing to the extent that receptiveness of characters might get lost and may work as wrong representations which further suffers accuracy of model. Increasing training data would possibly increase training and validation accuracy. Training time was around 30 min on 12GB NVIDIA Tesla K80 GPU with 2 workers.

Conclusions

With a detector and recognizer model trained and validated on datasets of topographic map, can be used to achieve the task of retrieving information from topographic maps and hence removing human interference in manual inspection and entry of data. For text detection or localization, an accuracy of 79.6% against Intersection over Union (IoU) value ≥ 0.7 is achieved using Faster RCNN+R-101-FPN. For text recognition, 60% accuracy was achieved for training set and 45% accuracy was achieved for validation set.

There remains a big challenge of producing a robust, high performance detector and especially a recognizer network. The current recognizer model is limited to just English alphabet and needs to be expanded to numbers and some non-English alphabet characters commonly seen on topographic maps of Indian region. The recognizer model has low validation as well as training accuracy which could be attributed to the low quality images in the recognition dataset. Data is driving these models hence having data that is engineered to correctly represent what a model must learn and in enough number is important and almost necessary. Though work on generalizing over small dataset is in research and also provably shown in (Olson *et al.*, 2018), currently state of the art deep learning models performs well on large amount of data. Specific to topographic maps, engineering a model for images containing texts with line strides of contours passing over and around could be heavily challenging. Wrong representations of characters could be

learned that would suffer on testing. The next most plausible step would be to create handcrafted image processing techniques to counter wrong representations and increase receptiveness of data to be learned. Fu *et al.* (2018) de-noises images with pepper like noise using CNNs. This inspires to research and work on removing strikes and unwanted lines from images using some variation of neural networks. Dataset for recognition was limited to just words in English alphabet but can be broadened to numbers. Detecting texts not horizontally inclined with non-horizontal bounding box is another sought out issues (Liao *et al.*, 2018).

Acknowledgement

I would like to thank Sikkim Manipal Institute of Technology, Sikkim Manipal University, Sikkim, India for giving me an opportunity to pursue my higher studies. In addition, I would also like to thank to my colleagues, friends and most importantly my families for consistent motivation, support and encouragement throughout the journey of this research.

References

- Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5:157-166.
- Bissacco, A., Cummins, M., Netzer, Y., and Neven, H. (2013). Photoocr: Reading text in uncontrolled conditions. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; 1-8 December, IEEE Xplore, Sydney, NSW, Australia, p. 785-792.
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition*; 20-25 June 2009, IEEE, Miami, FL, USA, p. 248-255.
- Fu, B., Zhao, X.Y., Li, X., and Ren, Y.G. (2018). A convolutional neural networks denoising approach for salt and pepper noise. *Multimedia Tools Appl.*, 78(21):30,707-30,721.
- Gers, F., Schraudolph, N., and Schmidhuber, J. (2002). Learning precise timing with lstm recurrent networks. *J. Mach. Learn. Res.*, 3:115-143.
- Girshick, R. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*; 7-13 December 2015, IEEE Xplore, Santiago, Chile, p. 1,440-1,448.
- Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*; 23-28 June 2014, IEEE Xplore, Columbus, USA, p. 580-587.
- Graves, A., Fernandez, S., and Gomez, F. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. *Proceedings of the International Conference on Machine Learning*; 25-29 June 2006, Association for Computing Machinery (ACM), New York, United States, p. 369-376.
- Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., and Schmidhuber, J. (2009). A novel connectionist system

- for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):855-868.
- Graves, A., Mohamed, A.R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing; 26-31 May 2013, IEEE, Vancouver, BC, Canada, p. 6,645-6,649.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 27-30 June 2016, IEEE, Las Vegas, NV, USA, p. 770-778.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comp.*, 9(8):1,735-1,780.
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. 2017 IEEE Conference on Computer Vision and Pattern Recognition; 21-26 July 2017, IEEE Computer Society, Honolulu, USA, p. 3,296-3,297.
- Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. *Int. J. Comp. Vis.*, 116(1):1-20.
- Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. 25th International Conference on Neural Information Processing Systems; 3-6 December, 2012 ACM Digital Library, Lake Tahoe, Nevada, p. 1,097-1,105.
- Liao, M., Shi, B. and Bai, X. (2018). Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, vol. 27(8):3,676-3,690.
- Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 21-26 July 2017, IEEE, Honolulu, USA, p. 936-944.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Scott R., Fu, C.Y., and Berg, A.C. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision: Lecture Notes in Computer Science*, 11-14 October 2016, Springer, Amsterdam, The Netherlands, p. 21-37.
- Olson, M., Wyner, A., and Berk, R. (2018). Modern neural networks generalize on small data sets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Redmon, J. and Farhadi A. (2017). Yolo9000: better, faster, stronger. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 21-26 July 2017, IEEE Xplore, Honolulu, USA, p. 6,517-6,525.
- Redmon, J. and Farhadi, A. (2018). YoloV3: An incremental improvement, Technical Report on Computer Vision and Pattern Recognition, Cornell University, arxiv: 1804.02767.
- Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition*; 27-30 June 2016, IEEE Xplore, Las Vegas, USA, p. 779-788.
- Ren, S., He, K., Girshick, B.R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks. Technical Report on Computer Vision and Pattern Recognition, Cornell University, earXiv: 1506.01497.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. Technical Report on Computer Vision and Pattern Recognition, Cornell University, arXiv:1312.6229.
- Shi, B., Bai X., and Yao C. (2015). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. Technical Report on Computer Vision and Pattern Recognition, Cornell University, arXiv:1507.05717.
- Su, B. and Lu, S. (2014). Accurate scene text recognition based on recurrent neural network. *ACCV 2014: Computer Vision - ACCV 2014*; 1-5 November 2014, Springer, Singapore, p 35-48.
- Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI Conference on Artificial Intelligence (AAAI-16)*; 12-17 February 2016, Arizona, USA.
- Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., and Smeulders, A.W.M. (2013). Selective search for object recognition. *Int. J. Comp. Vis.*, 104(2):154-171.
- Wang, T., Wu, D. J., Coates, A., and Ng, A.Y. (2012). End-to-end text recognition with convolutional neural networks. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*; 11-15 November 2012, Tsukuba Science City, Japan, p. 3,304-3,308.