Reading Text in the Wild from Compressed Images

Leonardo Galteri *
University of Florence
leonardo.galteri@unifi.it

Dena Bazazian *
CVC, Barcelona
dbazazian@cvc.uab.es

Lorenzo Seidenari
University of Florence
lorenzo.seidenari@unifi.it

Marco Bertini
University of Florence
marco.bertini@unifi.it

Andrew D. Bagdanov
University of Florence
andrew.bagdanov@unifi.it

Anguelos Nicolaou
CVC, Barcelona
anguelos@cvc.uab.es

Dimosthenis Karatzas
CVC, Barcelona
dimos@cvc.uab.es

Alberto Del Bimbo
University of Florence
alberto.delbimbo@unifi.it

Abstract

Reading text in the wild is gaining attention in the computer vision community. Images captured in the wild are almost always compressed to varying degrees, depending on application context, and this compression introduces artifacts that distort image content into the captured images. In this paper we investigate the impact these compression artifacts have on text localization and recognition in the wild. We also propose a deep Convolutional Neural Network (CNN) that can eliminate text-specific compression artifacts and which leads to an improvement in text recognition. Experimental results on the ICDAR-Challenge4 dataset demonstrate that compression artifacts have a significant impact on text localization and recognition and that our approach yields an improvement in both – especially at high compression rates.

1. Introduction

An extremely desirable feature of wearable vision systems is the ability to interpret text present in the observed scene. Reading text in the wild is of paramount importance to help visually impaired people navigating complex areas, such as streets, shopping malls and airports. An interesting scenario is multi-lingual visual reading, which enables real-time text translation. Reading text is a challenging task which is usually composed of two steps. Similarly to object detection, text reading consists of localizing text patches and then recognizing their content. Accurately performing both tasks is usually possible using computationally demanding deep Convolutional Neural Networks (CNNs). This demand in computation power conflicts with real-time wearable system requirements, unless images can be processed remotely. Unfortunately streaming images may present difficulties in narrow bandwidth situations. Moreover, wireless cameras systems, especially in the case of battery operated ones, may need to limit power consumption reducing the energy cost of image transmission applying strong compression.

Since user experience is also affected by image quality, compression algorithms are designed to reduce perceptual quality loss, according to some model of the human visual system. In fact, when compressing images several artifacts appear. These artifacts are due to the different types of lossy compressions used. Considering JPEG, the most common algorithm used nowadays, these artifacts are due to the chroma subsampling (i.e. dropping some color information of the original image) and the quantization of the DCT coefficients; these effects can be observed also in MPEG compressed videos, that is basically based the same schema with the addition of motion compensation and coding. Indeed, compression artifacts do reduce the performance of text recognition algorithms, affecting both localization and recognition.

Deep convolutional neural networks (DCNN) have become the basic approach for many computer vision tasks[21, 32, 25] and are of course the state-of-the art technique for text recognition [1, 17]. However, imperceptible pixel variations are known to alter image classification results, as shown by Goodfellow *et al*. [11]. The authors of this work computed adversarial examples by adding a tensor computed in a way to steer the classifier decision. These adversarial images are perceptually identical to the human eye

*These authors contributed equally to this work.

but the network they were made for will output a mistaken classification result with high confidence. Therefore there is compelling evidence that even small changes in images can indeed impair DCNN recognition capability. These results lead us to believe that *compression artifacts* will also have a negative impact on recognition results.

In this paper, we analyze issues related to end-to-end text recognition in the wild in the presence of compression artifacts. We show that both localization and recognition are affected by image compression and we propose a solution to improve text recognition performance in the presence of compression artifacts. We show that it is possible to learn a deep convolutional neural network that removes image artifacts and improves end-to-end text recognition in the wild. Adding this network does not require to change the compression pipeline, nor to re-train the text detection network. In Figure 1 we illustrate the types of compression artifacts our system is able to remove.

Figure 1: Examples of compression artifact removal. Odd rows: compressed images with compression artifacts; even rows: results of the proposed system. Best viewed in color and zoomed in.

2. Related work

The problem of image enhancement in wearable vision has not been addressed yet, therefore in the following we review the current state of the art in text recognition in the wild and image restoration.

2.1. Text detection and recognition

Detecting and recognizing text in natural images has received considerable attention in the computer vision community. Comprehensive surveys for scene text detection and recognition are given in [41, 47]. Classical text detection approaches based on connected components and sliding windows [7, 4, 15, 27, 28, 29, 42] are fairly robust techniques. However, CNN classifiers have recently led to significant improvements [38, 13, 16, 17] with notable increase in accuracy compared to previous techniques.

Despite the immense success of CNN models for tasks such as character classification and word-spotting, once text regions are localized the problem of unconstrained text recognition still poses significant challenges. To this end, Jaderberg *et al*. [17] proposed to use a CNN able to recognize words from an extensive lexicon and generic object proposals. However employing generic object proposals is not optimal when text is to be detected, as demonstrated in [9]. Furthermore, the authors of [10] proposed instead a text-specific object proposal method based on generating a hierarchy of word hypotheses computed with a region grouping algorithm.

In addition, Fully Convolutional Networks (FCNs) [25] have recently attracted considerable attention from the robust reading community [46, 14, 12]. FCN-based methods replace fully-connected layers with convolutional layers which allows them to preserve coarse spatial information which is essential for text localization tasks. The authors of [44] integrated semantic labeling by FCN with MSER to provide a natural solution for handling text at arbitrary orientations. In parallel work [46] designed a character proposal network based on an FCN which simultaneously predicts "characterness" scores and refines the corresponding locations. The "characterness" score is used for proposal ranking. Moreover, in [1] the authors improved the text proposal pipeline by fusing FCN outputs and the TextProposals of [10] in order to achieve higher recall with a less time consumed.

Inspired by Fully-Convolutional Networks [25] and [30], [12] propose a text localization network as an extreme variant of Hough voting. Moreover, [34] and [46] employed an FCN model in order to detect text orientation in natural scene images. Despite the significant

achievements of recent research on general object detection [30, 31, 32, 24], these methods are not appropriate for localizing text regions for several reasons. Typically the bounding box of a word/text line has much larger aspect ratio than common objects. TextBoxes [23] re-purposes the SSD detector [24] for word-wise text localization. Furthermore [37] follows the idea of Region Proposal Networks [32] and proposes a Connectionist Text Proposal Network which improves accuracy for text localization tasks and also is compatible with multiple scales, aspects, and languages.

In this paper we exploit the efficient, high recall text localization pipeline from [1]. We concentrate on analyzing the effect image compression artifacts have on localization and end-to-end scene text recognition in the wild.

2.2. Image restoration

Removing compression artifacts has been addressed in the past. The vast majority of previous works can be classified as processing-based [8, 39, 40, 45, 22, 3, 43, 5], while a few recent works are learning-based [6, 36, 26]. Processing-based methods rely only on the information of the image to be improved. They typically address artifacts introduced by JPEG compression, and thus usually work in the DCT domain.

Learning-based methods have recently been proposed following the successful application of deep Convolutional Neural Networks (DCNNs) to many multimedia and vision tasks. DCNNs are used to learn an image transformation function that, given an input image, will output a restored version. Starting from a set of original images, used either as ground truth or target, sets of degraded images are generated and used as a training set. Since it is possible to feed these learning-based methods with a large amount of data, they have the advantage that they can accurately estimate an image manifold, allowing an approximate inversion of the compression function. This manifold is also aware of image semantics and does not rely solely on DCT coefficient values or other statistical image properties – and thus can be applied to any compression algorithm.

Dong *et al*. [6] proposed an artifact reduction CNN (AR-CNN) based on their super-resolution CNN (SRCNN); both models share a common structure: a feature extraction layer, a feature enhancement layer, a non-linear mapping, and a reconstruction layer. The structure closely follows a sparse coding pipeline. Svoboda *et al*. [36] reported improved results by learning a feed-forward CNN; the CNN layers combine residual learning, skip architecture, and symmetric weight initialization for better reconstruction quality. Differently from [6], they do not have any specific function. Cavigelli *et al*. [2] proposed a 12-layer convolutional network with hierarchical skip connections and a multi-scale loss function, obtaining some improvement in objective perceptual quality metrics over AR-CNN.

In this paper we consider the specific case of compression artifact removal using CNNs for text recognition in natural scene images. We believe we are the first to explore the use of Deep CNNs for image restoration specifically in the context of text recognition and text localization. In the next section we describe the methodology of our text recognition and compressed image restoration pipelines.

3. Methodology

In this section we describe the general problem of compression artifacts in images of text, the problem of reading text in the wild, and our approach to removing compression artifacts from text images.

3.1. Compression artifacts and text

To understand the compression artifacts that may affect text elements in an image, we first review basic techniques used for image compression (e.g. in JPEG). Typically, in the first step the image is converted to the $YCrCb$ color space in order to separately handle luminance information (encoded in the Y component) and color information (encoded by Cr and Cb components). This is motivated by the fact that the human visual system discriminates brightness better than color; the separation enables spatial subsampling of Cr-Cb using different schemes like 4:2:0 (i.e. Y is never subsampled, Cb and Cr are subsampled every two pixels on alternating rows). Then finer details are eliminated; this is typically performed on image blocks composed by a few pixels. For example, in JPEG the downsampled pixels are split into 8×8 pixel blocks that are transformed using a Discrete Cosine Transform (DCT), to enable separate handling of low and high frequencies. The DCT coefficients are quantized, reducing the high frequency values, to obtain a vector of values that can be more easily compressed.

Considering these operations, the most common artifacts and distortions that affect text, that is characterized by a color and brightness contrast, and by having a sharp transition with respect to the background, are:

- **blurring**: this results from loss of high frequency signal components.
- **ringing**, i.e. introduction of spurious signal: this happens near sharp transitions in the image regions. It is due to the loss of high frequency components due to coarse quantization of high frequency components (e.g. DCT coefficients). This occurs also in wavelet-based JPEG-2000 compression and in MPEG compression. It is more annoying for human viewers than blurring [33].
- **color deviation**: due to the loss of color information due to subsampling. Since in videos several different

Figure 2: Examples of compression artifacts in text images. Top row: high quality image; bottom row: low quality compressed image. Ringing artifacts are visible on all letters, color deviation and blurring are more visible on the borders of the vertical strokes of P, I, N and T. Artifacts affect both versions of the image. Best viewed in color and zoomed in.

subsampling color schemes are used, e.g. in MPEG, DV and MJPEG, it is a common practice that superimposed captions use a 1 pixel border with high Y contrast, to reduce this effect.

Examples of these compression artifacts are shown in Figure 2, where details of high quality images are compared to those of low quality high compression images.

3.2. Reading text in the wild

In this work we use the pipeline of [1] to generate the text proposals as a prerequisite for text recognition. Afterwards, we apply the DictNet word classifier [17] to recognize the content of text regions. The pipeline of [1] is based on a Fully Convolutional Network for text detection and the TextProposals algorithm from [10].

3.2.1 Fully Convolutional Networks for text detection

We trained a Fully Convolutional Network (FCN) inspired by [25] for the task of text detection by fine-tuning a VGG16 network pre-trained on ImageNet [35]. Fine-tuning was performed for 1000 iterations using Caffe [18] on the ICDAR-Challenge4 training-set. Afterwards, we used the FCN to generate heatmaps indicating the degree of "textness" at each pixel in the original, compressed and reconstructed images of the ICDAR-Challenge4 test set. At this stage it was evident that the FCN was sensitive to details lost (and artifacts introduced) during the compression process. In Figure 3 we demonstrate the improvement of detecting text regions after reconstructing the compressed images.

3.2.2 The TextProposal algorithm

To generate candidate text regions we use the TextProposal algorithm of [10]. which generates the proposals based on clustering process over individual regions. In this approach the first phase over-segments the input image in order to obtain a set of connected components. Afterwards, it performs several bottom-up agglomeration processes. In the end, there is a ranking strategy for prioritizing each text proposal. We used the original TextProposals implementation of [10].[1]

Once we have the ranked list of TextProposals, we fuse the TextProposals with the FCN heatmaps described in the previous section in order to suppress false positive text proposals. As in [1], we sum the FCN probabilities in each TextProposal box and use a threshold of 0.14 to suppress boxes containing a sum total "textness" of less than this.

3.2.3 Text recognition

The main purpose of text recognition in this work is to demonstrate its sensitivity to compression artifacts and quantify how our CNN reconstruction approach helps compensate for them. For recognition, we use the state-of-the-art CNN DictNet word classifier of [17] to read the cropped words. The word classifier net [17] consists of five convolutional and three fully connected layers. The first two fully-connected layers have 4k units and the final fully-connected layer has the same number of units as number of words in the dictionary (90k words).

To evaluate text recognition independently of text localization, we perform a series of experiments on cropped text words from the ICDAR-Challenge4 test set. We feed the cropped original, compressed (at varying quality factors), and reconstructed images to the DictNet word classifier. To evaluate end-to-end text recognition performance, and thus to measure localization and recognition performance, we use FCN+TextProposals pipeline described above and feed all TextProposal boxes passing the threshold to the DictNet classifier.

3.3. Restoring images with CNNs

The general problem of image restoration, i.e. computing a recovered image I^{RQ} from a low quality image I^{LQ}, that

[1] http://github.com/lluisgomez/TextProposals

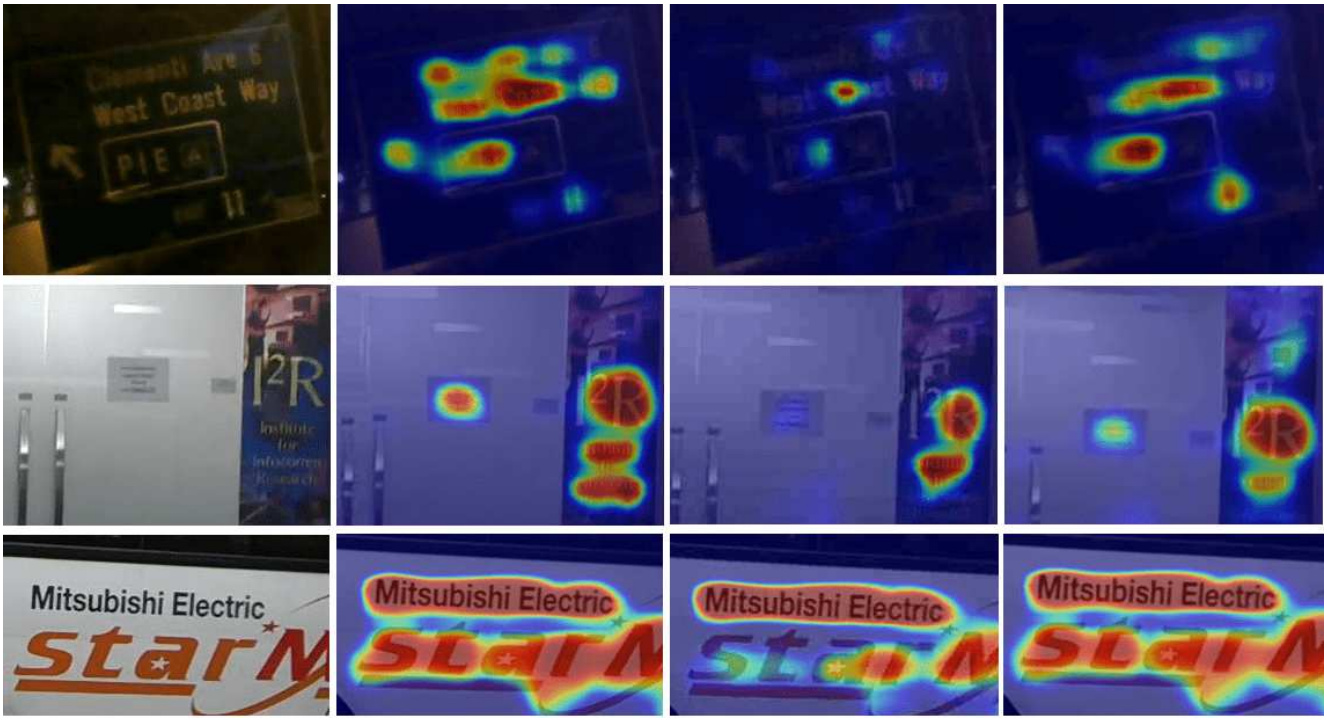

Figure 3: Improvement in text detection after reconstructing compressed images. In this figure we illustrate the original images and their corresponding heatmaps for the original, compressed, and reconstructed (in order, from left to right).

in turn can be produced processing a high quality original image I^{HQ} so that $I^{LQ} = P(I^{HQ})$, can be divided in several different problems. If P is a "lossy" image compression algorithm, then the problem is to eliminate the compression artifacts introduced by the compression.

An image $I^{HQ} \in [0, 255]^{W \times H \times C}$ is processed by a compression algorithm A:

$$I^C = A\left(I^{HQ}, QF\right) \in [0, 255]^{W \times H \times C} \quad (1)$$

using some quality factor QF in the compression process.

Image transformation can be used to attempt to recover from image artifacts. To transform a compressed image into a version in which artifacts are removed or reduced, a function is applied pixelwise. Recent advances suggest that this task should be tackled by training a convolutional neural network from compressed and uncompressed image pairs.

3.3.1 Architecture

The full pipeline of the approach, both in training and testing phases is depicted in Figure 4 In this work we use a deep residual network composed of convolutional layers and ReLU non-linearities as activation function. Since the network performs a pixelwise transformation, the input and the output images have the same dimensions $W \times H \times C$ where W, H and C represent, respectively, width, height and the number of channels of the images. We use 5 residual blocks consisting of 2 convolutional layers, which have 3×3 kernels and 64 feature maps and padding of 1 pixel to maintain the same image size. The last part of the network is a convolutional layer with a *tanh* activation function.

Table 1: Our fully convolutional network architecture. In all our experiments we have used 5 residual blocks.

Layer	Feature Map Size
Input I^C	$W \times H \times C$
Convolution 3×3, ReLU	$W \times H \times 64$
Convolution 3×3, ReLU	$W \times H \times 64$
Element-Wise Sum	$W \times H \times 64$
...	...
Convolution 3×3, ReLU	$W \times H \times 64$
Convolution 3×3, ReLU	$W \times H \times 64$
Element-Wise Sum	$W \times H \times 64$
Convolution 3×3, ReLU	$W \times H \times C$
Output I^{RQ}	$W \times H \times C$

3.3.2 Training

Training is performed with direct supervision. The loss is computed as a function of the reconstructed image I^{RQ} and

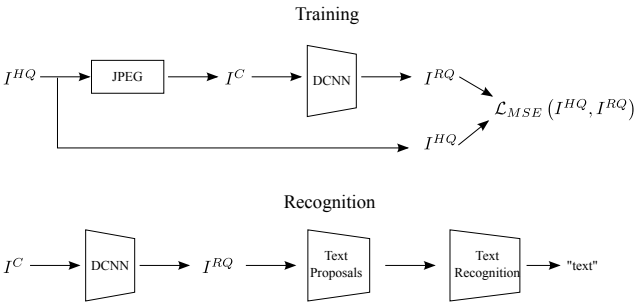

Figure 4: Diagram of our approach. Training is performed by minimizing MSE between reconstructed $\left(I^{RQ}\right)$ and high quality $\left(I^{HQ}\right)$ images. At test time we first remove artifacts from compressed images $\left(I^{C}\right)$ and then apply the two step process of localization and recognition.

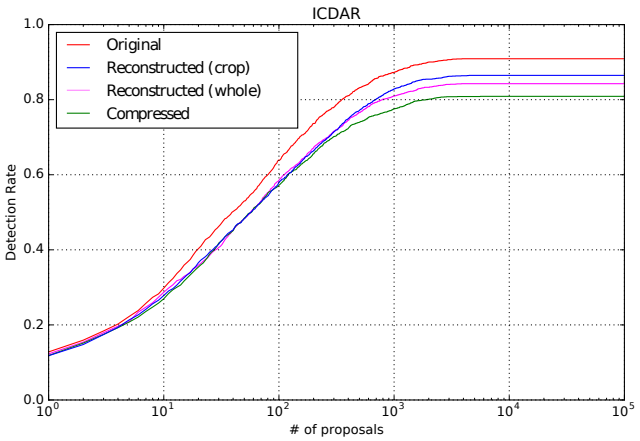

Figure 5: Detection rate (recall) at IoU 0.5 on the ICDAR-Challenge4 test images.

the original image I^{HQ}. Learning the transformation from compressed images to high quality ones requires training the weights and biases of the convolutional kernels. We minimize the Mean Squared Error (MSE) loss between the original uncompressed image and the network output:

$$\mathcal{L}_{MSE} = \left\| I^{HQ}_{x,y} - I^{RQ}_{x,y} \right\|_2 . \quad (2)$$

This loss is widely used in image restoration tasks and has been shown to be effective at reconstructing low-level details, such as edges and contours, that are very prominent in text patches.

The networks were trained on an NVIDIA Titan X GPU using patches from the ICDAR-Challenge 4 training set. All images were compressed with MATLAB JPEG compressor at 10, 20 and 30 QF. For the optimization process we used Adam [20] with momentum 0.9 and a learning rate of 10^{-4}. Training was performed for $50,000$ iterations.

For each mini-batch we sampled 8 random 48×48 patches without any data augmentation, using two different sampling strategies. In the first case, the network was fed with patches randomly selected from anywhere in the whole training image. In the second strategy we selected just the patches belonging to the text regions in order to specialize the network to reconstruct text degraded by the compression process.

4. Experiments

We used the ICDAR-Challenge 4 [19] as the benchmark dataset in our experiments[2]. This challenge focuses on incidental scene text, referring to scene text that appears in the scene without the user having taken any specific prior action to cause its appearance or to improve its positioning or quality in the frame. While focused scene text (explicitly photographed by the user) is the expected input for applications such as translation on demand, incidental scene text represents another wide range of applications linked to wearable cameras or massive urban captures where the acquisition process is difficult or undesirable to control. This challenge for the task of localization and end-to-end has 1000 images for training and a 500 images for testing that can be used for evaluation of specific tasks through submitting results online to the Robust Reading Competition portal. For the task of text recognition, which considers only the cropped words of scene images, there are 4468 images for training and 2077 images testing.

[2] https://www.rrc.cvc.uab.es

4.1. Text localization results

In this experiment we compare the ranked list of proposals from [1] on compressed, reconstructed and original images in order to demonstrate the improvement from our reconstruction CNN (with both sampling strategies). The comparison of text proposal on compressed and reconstructed images is shown in Figure 5. This plot shows the recall of text regions (at IoU 0.5) over a range of considered proposals.

These results show that compression has a significant effect on text box recall. We also see that both CNNs (cropped and whole image sampling) are able to improve recall performance – especially when about 1000 or more proposals are considered. We also see that cropped image sampling performs slightly better than whole image sampling. In all subsequent experiments we use the CNN trained with the *cropped* patch sampling strategy.

4.2. Text recognition results

In this experiment we consider cropped words from scene images. We compare the results of text recognition using the CNN word classifier of [17]. The main purpose of this experiment is to explore how compressed images affect

Table 2: Text recognition results on the ICDAR-Challenge4 dataset. We report the Correctly Recognized Words (CRW) and the Average Normalized Edit Distance (AED). All performance is measured case insensitive, and images were reconstructed using the CNN trained with the *cropped* patch sampling strategy.

	QF	CRW	AED
Original	-	49.16%	25.09%
JPEG	10	31.05%	38.50%
Reconstructed	10	**32.07%**	**37.61%**
JPEG	20	39.58%	31.28%
Reconstructed	20	**39.96%**	**31.14%**
JPEG	30	43.43%	28.35%
Reconstructed	30	**43.96%**	**28.30%**

Table 3: End-to-end results measured in Precision, Recall, and Hmean on the ICDAR-Challenge4 dataset. Images were reconstructed using the CNN trained with the *cropped* patch sampling strategy.

	QF	Precision	Recall	Hmean
Original	-	37.60 %	87.85 %	52.66 %
JPEG	10	25.57 %	87.19 %	39.54 %
Reconstructed	10	**28.74** %	**87.54** %	**43.28** %
JPEG	20	33.12 %	88.32 %	48.18 %
Reconstructed	20	**33.61** %	**88.69** %	**48.74** %
JPEG	30	**36.64** %	**87.88** %	**51.72** %
Reconstructed	30	36.59 %	87.76 %	51.65 %

text recognition independently of localization. The results of text recognition experiment are demonstrated in Table 2.

From these results we see that JPEG compression has a significant effect on word recognition. At high compression rates, our CNN improves both CRW and AED by about 1%. At lower compression rates the improvement is less significant, but our CNN for reconstruction still has a positive impact on performance.

4.3. End-to-end results

To perform a comprehensive experiment on compressed and reconstructed images we have also considered the end-to-end recognition task. This measures the overall improvement in localization and recognition for reconstructed images. For this experiment we only considered the top 2,000 proposals in the ranking list of each image set in order to accelerate the evaluation process. The results of our end-to-end experiment are given in Table 3.

Again, at high compression rates our network leads to significant improvement in all three metrics. We see that the combination of improved localization and improved recognition leads to much better end-to-end recognition results. However, at lower compression rates the improvement is less evident. The test images in the ICDAR-Challenge4 dataset are compressed to about QF 30, and this is why the improvement of our CNN saturates at this point as the performance of both JPEG and Reconstructed images approaches that on the Original images.

4.4. Qualitative results

In figure 6 we show some examples of compressed, reconstructed, and original images containing text. We see that compression does have a significant impact on text quality. Both CNNs (with cropped and whole image sampling) significantly improve the visual quality of text in the image.

5. Conclusion and future work

In this paper we explored the effect JPEG compression artifacts can have on text localization and recognition in the wild. Our experimental results demonstrate that JPEG compression has a significant effect on text localization and recognition. We also described a simple CNN architecture that is able to reconstruct compressed images and, especially at high compression rates, is able to improve text localization, cropped text recognition, and end-to-end text recognition results.

For future work we are interested in training our network using high-quality original images, since the ICDAR-Challenge4 images are already significantly compressed. We are also interested in training our CNN network for compressed image restoration on significantly more images than whose available in ICDAR-Challenge4. We expect both of these to significantly improve the impact our restoration has on text recognition.

Acknowledgments

This work was supported by the CERCA Programme of the Generalitat de Catalunya, the research project TIN2014-52072-P. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] D. Bazazian, R. Gomez, A. Nicolaou, L. Gomez, D. Karatzas, and A. Bagdanov. Improving text proposals for scene images with fully convolutional networks. In *DLPR workshop in conjunction with ICPR*, 2016. arxiv:1702.05089. 1, 2, 3, 4, 6

[2] L. Cavigelli, P. Hager, and L. Benini. Cas-cnn: A deep convolutional neural network for image compression artifact suppression. *arXiv preprint arXiv:1611.07233*, 2016. 3

[3] H. Chang, M. K. Ng, and T. Zeng. Reducing artifacts in JPEG decompression via a learned dictionary. *IEEE Transactions on Signal Processing*, 62(3):718–728, Feb 2014. 3

Figure 6: Examples of cropped text reconstruction. The leftmost column shows compressed versions of cropped text at QF 10, the second column is the reconstruction using the whole image sampling strategy, the third shows the reconstruction using the cropped text sampling strategy and the rightmost column is the ground truth.

[4] X. Chen and A. Yuille. Detecting and reading text in natural scenes. In *Proc. CVPR*, pages 366–373, 2004. 2

[5] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes. Post-processing of compressed images via sequential denoising. *IEEE Transactions on Image Processing*, 25(7):3044–3058, July 2016. 3

[6] C. Dong, Y. Deng, C. Change Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 576–584, 2015. 3

[7] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. CVPR*, pages 2963–2970, 2010. 2

[8] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Transactions on Image Processing*, 16(5):1395–1411, 2007. 3

[9] L. Gomez and D. Karatzas. Object proposals for text extraction in the wild. In *Proc. ICDAR*, pages 206–210, 2015. 2

[10] L. Gomez and D. Karatzas. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *Preprint submitted to Pattern Recognition*, 2016. arxiv:1604.02619. 2, 4

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015. 1

[12] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proc. CVPR*, pages 2315–2324, 2016. 2

[13] T. He, W. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural networks for scene text detection, 2015. arxiv:1510.03283. 2

[14] T. He, W. Huang, and J. Yao. Accurate text localization in natural image with cascaded convolutional text network,

2016. arxiv:1603.09423. 2

[15] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proc. ICCV*, pages 1241–1248, 2013. 2

[16] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *Proc. ECCV*, pages 497–511, 2014. 2

[17] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. *IJCV*, 116(1):1–20, 2016. 1, 2, 4, 6

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. ACM*, pages 675–678, 2014. 4

[19] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. Icdar 2015 robust reading competition. In *Proc. ICDAR*, pages 1156–1160, 2015. 6

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015. 6

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 1

[22] Y. Li, F. Guo, R. T. Tan, and M. S. Brown. A contrast enhancement framework with JPEG artifacts suppression. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 174–188, Cham, 2014. Springer International Publishing. 3

[23] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In *Proc. AAAI*, pages 4161–4167, 2017. 3

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. Ssd: Single shot multibox detector. In *Proc. ECCV*, pages 21–37, 2016. 3

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1, 2, 4

[26] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2802–2810, 2016. 3

[27] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Proc. ACCV*, pages 770–783, 2010. 2

[28] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. In *Proc. ICCV*, pages 97–104, 2013. 2

[29] Y. Pan, X. Hou, and C. Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Trans. on Image Processing*, 20(3):800–813, 2011. 2

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, pages 779–788, 2015. 2, 3

[31] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. computer vision and pattern recognition, 2016. arxiv:1612.08242. 3

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015. 1, 3

[33] M.-Y. Shen and C.-C. Kuo. Review of postprocessing techniques for compression artifact removal. *Journal of Visual Communication and Image Representation*, 9(1):2 – 14, 1998. 3

[34] B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments. In *Proc. CVPR*, 2017. 2

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 4

[36] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik. Compression artifacts removal using convolutional neural networks. *arXiv preprint arXiv:1605.00366*, 2016. 3

[37] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. In *Proc. ECCV*, pages 56–72, 2016. 3

[38] T. Wang, D. Wu, A. Coates, and A. Ng. End-to-end text recognition with convolutional neural networks. In *Proc. ICPR*, pages 3304–3308, 2012. 2

[39] T.-S. Wong, C. A. Bouman, I. Pollak, and Z. Fan. A document image model and estimation algorithm for optimized JPEG decompression. *IEEE Transactions on Image Processing*, 18(11):2518–2535, 2009. 3

[40] S. Yang, S. Kittitornkun, Y.-H. Hu, T. Q. Nguyen, and D. L. Tull. Blocking artifact free inverse discrete cosine transform. In *Proc. of International Conference on Image Processing (ICIP)*, volume 3, pages 869–872. IEEE, 2000. 3

[41] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. *IEEE Trans. on PAMI*, 37(7):1480–1500, 2015. 2

[42] X.-C. Yin, X. Yin, K. Huang, and H. Hao. Robust text detection in natural scene images. *IEEE Trans. on PAMI*, 36(5):970–983, 2014. 2

[43] J. Zhang, R. Xiong, C. Zhao, Y. Zhang, S. Ma, and W. Gao. CONCOLOR: Constrained non-convex low-rank model for image deblocking. *IEEE Transactions on Image Processing*, 25(3):1246–1259, March 2016. 3

[44] S. Zhang, M. ZLin, T. Chen, L. Jin, and L. Lin. Character proposal network for robust text extraction. In *Proc. ICASSP*, pages 2633–2637, 2016. 2

[45] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao. Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity. *IEEE Transactions on Image Processing*, 22(12):4613–4626, 2013. 3

[46] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and W. X. Bai. Multi-oriented text detection with fully convolutional networks. In *Proc. CVPR*, page 2016, 4159-4167. 2

[47] Y. Zhu, C. Yao, and X. Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 10(1):19–36, 2016. 2