



UNIVERSITÀ
DEGLI STUDI
FIRENZE

UNIVERSITÀ DEGLI STUDI DI FIRENZE
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE (DINFO)
CORSO DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE
CURRICULUM: INFORMATICA (SSD ING-INF/05)

DEEP LEARNING FOR DETECTION IN COMPRESSED VIDEOS AND IMAGES

Candidate

Leonardo Galteri

Supervisors

Prof. Alberto Del Bimbo

Prof. Marco Bertini

Dr. Lorenzo Seidenari

PhD Coordinator

Prof. Luigi Chisci

CICLO XXX, 2014-2017

Università degli Studi di Firenze, Dipartimento di Ingegneria
dell'Informazione (DINFO).

Thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Information Engineering. Copyright © 2017 by
Leonardo Galteri.

A Lorenzo e Marco

Contents

Contents	v
1 Introduction	1
1.1 The objective	1
1.2 Contributions	3
2 Literature review	7
2.1 Object Detection	7
2.1.1 Bag-of-Words Based Approaches	8
2.1.2 Coarse to Fine and Boosted Approaches	8
2.1.3 Deformable Part Models Approaches	8
2.1.4 Deep Learning Approaches	8
2.2 Objectness	9
2.2.1 Spatial Objectness	10
2.2.2 Spatio-Temporal Objectness	11
2.3 Video Coding	12
2.3.1 Saliency Based Methods	13
2.4 Image Restoration	14
2.4.1 Processing Based Methods	14
2.4.2 Learning Based Methods	15
2.4.3 Other Image Transformation Tasks	15
3 Spatio-Temporal Closed-Loop Object Detection	17
3.1 Introduction	17
3.2 The proposed method	20
3.3 Experimental Evaluation	26
3.3.1 Spatio-temporal objectness performance	27
3.3.2 Detection performance on video	29

3.3.3	Execution speed	32
3.4	Conclusion	33
4	Video Compression for Object Detection Algorithms	39
4.1	Introduction	40
4.2	Extremely Fast Object Proposals	40
4.2.1	Visual feature	41
4.3	Coding with learned saliency	43
4.4	Experiments	46
4.4.1	Execution speed	46
4.4.2	Visual Quality	47
4.5	Conclusions	52
5	Removing Compression Artifacts with Generative Adversarial Networks Enhances Image Semantics	55
5.1	Introduction	56
5.2	Compression Artifacts	59
5.3	Methodology	60
5.3.1	Generative Network	63
5.3.2	Loss Functions for Direct Supervision	63
5.3.3	Generative Adversarial Artifact Removal	65
5.4	Experiments	67
5.4.1	Implementation Details	67
5.4.2	Dataset and Similarity Measures	68
5.4.3	Feature Maps Error	68
5.4.4	Comparison with State-of-the-Art	71
5.4.5	Object Detection	71
5.4.6	Segmentation Mask Proposal	79
5.4.7	Subjective evaluation	80
5.5	Conclusion	83
6	Reading Text in the Wild from Compressed Images	85
6.1	Introduction	85
6.2	Related Work	87
6.3	Methodology	88
6.3.1	Reading text in the wild	89
6.3.2	Restoring images with CNNs	91
6.4	Experiments	93

6.4.1	Text localization results	95
6.4.2	Text recognition results	95
6.4.3	End-to-end results	96
6.4.4	Qualitative results	97
6.5	Conclusion and future work	97
7	Conclusion	99
7.1	Summary of contribution	99
A	Publications	101
	Bibliography	103

Chapter 1

Introduction

In recent years we have witnessed a growing diffusion of digital devices such as mobile phones, cameras and surveillance systems capable to produce high resolution images and videos to be shared on the web. The substantial improvement of computation power of machines have contributed to an increasing employment of computer vision algorithms to data processing.

In this context, object detection represents one of the most important tasks and as such has received considerable attention from the research community. This ability to detect different classes of objects in videos and images has a remarkable importance in many kinds of applications such as video surveillance, human-computer interaction, autonomous driving, automated analysis of media content and automatic media tagging.

1.1 The objective

Object detection has been deeply explored by researchers in recent years, many approaches have been developed relying basically on the same concept. Supervised learning frameworks train detection models using labeled image or video datasets and, depending on the approach, different kinds of features are extracted from data, such as edges, contours orientations, colors and so on.

Recent advances in computer vision have shown that Convolutional Neural Networks (CNN) are able to learn rich feature representations directly from pixels. More specifically, they have been proven to outperform any other classifier in image recognition competitions like PASCAL VOC [37],

COCO [88] or Imagenet [115] the last few years. The reasons for CNNs success are mainly due to their ability to learn features much better than hand-crafted ones with location and slight transformation invariance. Figure 1.1 shows some examples from the COCO dataset, where detectors should be able to classify and localize several common classes like train, cows, monitors and boats.



Figure 1.1: Examples of object classes from COCO dataset.

CNN based detectors usually perform their task by evaluating a subset of locations in an image rather than using a sliding window approach. For this reason, approaches estimate the objectness measure have become increasingly popular in the last years, and this is due to the fact that they can be used as a pre-processing step for an object detector, therefore speeding up the overall execution time of the whole process.

One of the major challenges about this topic is that the majority of object proposals are suited just for still images. Therefore, one of the main targets of this dissertation is to elaborate a solution for generating better proposals exploiting the temporal coherence of frames.

Moreover, in this thesis we show that it is possible to exploit objectness to solve another kind of problem that typically harms the performance of

object detectors, that is the compression of images and videos. 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, like noise or small image structures, and higher frequency details tend to be eliminated. In Figure 1.2 two different compression quality factors on the same input are shown, we can easily notice that in the right-most sample many artifacts are present and details are heavily suppressed compared to the first one.



Figure 1.2: Examples of compression with different quality factor values: from left to right 90 and 15.

1.2 Contributions

We present in this thesis a set of approaches to improve the quality of detection and to overcome some problems due to compression of image data. The contribution of this dissertation is divided into two main branches.

The first part is related to the improvement of object proposal quality in video frames and it has the aim to generate high confident detection for video sequences. The interplay of detectors and proposal algorithms has not been fully analyzed and exploited up to now, although this is a very relevant problem for object detection in video sequences. For this reason, the quality of object proposals in the video domain has a remarkable importance to speed up the execution time of the algorithm and is likely to reduce the number of false positive outputs. In our approach, we show how to connect, in a closed-loop, detectors and object proposal generator functions exploiting the ordered and continuous nature of video sequences, so that detectors show a good performance using just a few proposals.

The second part of the thesis is related to improve the quality of detectors in compressed images and videos. To overcome the problem of compression in this kind of media, we have studied two different strategies. In the first place, we have developed an adaptive video coding approach based on a fast computation of saliency maps, in order to control the quality of frames so that automatic object detectors can still process the resulting video, improving their detection performance, by preserving the elements of the scene that are more likely to contain meaningful content. On the other hand, we show that using a CNN based approach for compression artifacts removal not only improves the performance of detectors in heavily corrupted images and videos, but also leads to more pleasant results for the human eye. We demonstrate that reconstructing very corrupted images gives benefits in both object detection and text in the wild recognition.

The rest of the thesis is organized as follows. We start with a review of the state of the art in object detection and object proposals in the first part of Chapter 2. This section builds a thorough background for the main theme of this thesis and in particular, we focus our attention on deep learning based approaches that represent the foundation of most of the study present in this work. In the remaining part of Chapter 2, we describe the literature related to image and video compression. In particular, we pay attention to approaches designed to remove compression artifacts.

Chapter 3 deals with the object proposals improvement in generic video sequences. The proposed closed-loop approach help detectors to provide a better output classification and localization due to the improved quality proposal windows and it's very efficient. We exploit the temporal coherence between video frames to combine proposal and detection boxes. In this way, an object detector needs very few proposal windows to elaborate its final decision, hence reducing the false positive rate and speeding up the whole process.

Chapter 4 deals with an adaptive video coding approach for computer vision-based systems. In this section, we show how to improve video compression for an object detection purpose by reducing the set of irrelevant information transmitted in the video stream, driving the codec to compress more the parts that do not contain semantically interesting objects and keeping the relevant regions mostly unaltered. We propose a very fast objectness estimator approach to build reasonable saliency maps that will be used by the codec to understand how to behave in the different parts of the frames.

In Chapter 5 we deal with the problem of data compression from another point of view. Our objective is to restore compressed images rather than to act directly on codecs before transmitting the data. We show that using a generative adversarial network to restore degraded images leads to very pleasant results and the performance of object detectors improves remarkably compared to classic image restoration methods.

Finally in Chapter 6 we describe another kind of application for image restoration. Compression artifacts critically affect even text localization and text recognition in the wild. We propose a residual convolutional network that is able to reduce those artifacts, in particular when dealing with very high compression rates. This approach leads to significant improvements in text localization, cropped text recognition and end-to-end text recognition.

Chapter 2

Literature review

In this chapter we review some recent works about the main topics of this dissertation. Most of the existing techniques which obtained state-of-the-art results for all these tasks are based on Deep Neural Networks. The first part of this chapter introduces the problems of object detection and objectness, while the second part deals with the problem of adaptive video coding. Finally, we review the literature related to the image restoration area.

2.1 Object Detection

One of the most important and challenging problems in computer vision is recognizing different objects and localizing them in image. The goal of object detection is to find the location and scale of all object instances that are present in an image, hence a detector should be able to determine all object instances of multiple classes regardless of scale, location, pose, occlusions and illumination.

Machine learning approaches aim to learn object representation models. With the advance of machine learning techniques, especially the powerful feature representations and classifiers, many recent approaches regarded object detection as a classification problem and have achieved significant improvements.

2.1.1 Bag-of-Words Based Approaches

The Bag-of-Words model is probably one of the most popular during the last decade. [99, 118, 123] This model is known to be pretty simple and efficient, also very resilient to background clutter. On the other hand, it has some issues related to those cases where two instances of the same object appear very close to each other. Furthermore, typically the localization of objects is not very accurate as in more complex methods.

2.1.2 Coarse to Fine and Boosted Approaches

Viola and Jones [133] describe a method for object localization that has become really popular in the scientific community. The research of objects is performed by shifting a template across space and scale extracting Haar like features. Then a cascade of classifier trained with Adaboost are used to reject image patches that do not belong to objects. Other approaches based on booster classifier include [43, 64, 84, 132, 141].

2.1.3 Deformable Part Models Approaches

There is a vast literature on deformable models for object detection based on deformable template models [20, 21] and part-based models [4, 22, 40, 83]. The key idea of this kind of approaches is considering objects as a collection of parts arranged in a deformable configuration, where each part identifies the properties of the local appearance of objects and the deformable framework is composed by spring-like connections between some pairs of parts. Recent works like [38, 39, 143] have become quite successful since a coarse-to-fine cascade model has been integrated in the framework to achieve a more efficient evaluation of the model.

2.1.4 Deep Learning Approaches

State of the art object detection is nowadays achieved by region based convolutional neural network methods [34, 47, 48, 112]. R-CNN pioneered this task by simply applying a pre-trained network to regions. Improved accuracy in detection is then achieved fine-tuning the network on object boxes and learning a bounding box regressor.

More recent approaches [47, 112] have applied a similar idea but avoiding a full computation of the convolutional feature for each region, sharing instead

a single image feature map for all the evaluated boxes.

Ren *et al.* [112] presented Faster R-CNN, an integrated approach of proposal and detection computation. Faster R-CNN adds a Region Proposal Network (RPN) to Fast R-CNN thus exploiting the same convolutional feature computation pipeline to compute proposals. This approach is efficient in terms of computation time since it avoids the burden of proposal generation from an external module, by sharing the features among RPN and Fast R-CNN detection.

Redmon *et al.* [111] presented a new approach considering object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A neural network predicts both bounding boxes and class probabilities directly from full images in one evaluation.

Liu *et al.* [90] proposed SSD, an approach where the output space of bounding boxes is discretized into a set of default boxes over different aspect ratios and scales per feature map location. Furthermore, predictions from multiple feature maps with different resolutions are concatenated to better handle objects of different sizes.

2.2 Objectness

The problem of quantifying how likely a part of an image is showing an object of some class is related to saliency detection. Works in this area typically aim at predicting salient points of human eye fixation [117] or modeling visual attention [10]. However, a detector may need to handle objects that are not visually conspicuous or that do not draw human gaze, thus an object proposal method should be able to deal also with objects that are not salient. Desirable properties for an object proposal method are:

- **High object detection rate / proposal recall:** to avoid discarding good candidate windows that are not processed by a specific object detector at a later stage.
- **High computational efficiency / low processing time:** to allow using the method in real-time applications or to effectively use it as pre-processing step in an object detection pipeline. This property is related to the number of candidate window proposals that are computed.
- **Good object generalization:** to detect a large number of different objects, so that proposals can be used with many different specific

object detectors.

- **Good cross-dataset generalization:** to maintain an acceptable detection rate on a testing dataset that is different from that of training, without need of retraining.
- **High repeatability:** to consistently propose windows on similar image content, despite image perturbation or changes, thus allowing to exploit proposals for a better training of object detectors [62].

Hosang *et al.* [62] have very recently presented a comparison of twelve object proposal methods for images, applying them to Pascal VOC 2007, MS COCO and ImageNet 2013 datasets, comparing some of these properties.

2.2.1 Spatial Objectness

These methods propose a relatively small number of proposals (e.g. 10^3 – 10^4) that should cover all the objects of an image, independently from their class. Typically they rely on low-level segmentation such as the method proposed by Felzenszwalb and Huttenlocher [41], or use their own segmentation algorithm.

Gu *et al.* [53] have presented a framework for object detection and segmentation that groups hierarchically segments to detect candidate objects, evaluating performance using the bounding boxes that encompass these regions.

The method proposed by Alexe *et al.* [2, 3] uses different cues such as multi-scale saliency, color contrast, edge density, superpixels segments, location and size of the proposal window, combining them in a Bayesian framework.

Enders *et al.* [31] generate a set of segmentations by performing graph cuts based on a seed region and a learned affinity function. Regions are ranked using structured learning based on a mix of a large number of cues.

Uijlings *et al.* [129] propose a method that requires no parameter learning, combining exhaustive search and segmentation in a data-driven selective search. The approach is based on hierarchical grouping of regions, using color, texture and region features. The work of Manén *et al.* [93] is similar in spirit to that of [129], but randomizing the merging process and learning the weights of the merging function.

Instead of following a hierarchical approach, the method proposed by Carreira and Sminchisescu [14] generates sets of overlapping segments, obtained solving a binary segmentation problem, initialized with different seeds. Segments are ranked by objectness using a trained regressor.

Differently from the methods reported above, the two methods proposed by Zitnick and Dollár [155], and Cheng *et al.* [19] do not use image segmentation.

The method of [155], called Edge Boxes, computes a scoring function in a sliding window fashion. Scoring is performed measuring the number of edges that exist in the box minus those that are members of contours that overlap the box's boundary.

The method of [19] is the fastest approach, as reported in the comparison of [63], and uses a simple linear classifier over edge features, that is trained and applied in a sliding window manner. The efficiency of the approach is due to the use of approximated features, binarized normed gradients that give the name (BING) of the method.

Following the Convolution Neural Network framework, a few objectness methods have been built on top of convolutional features. Multibox [34] approaches exploit a saliency based approach and after classifying an image they propose a few boxes per class on salient regions.

Different from the fully integrated approach of [112], DeepBox and DeepMask [106] learn to generate windows, or even masks with a deep convolutional architecture. These methods have a higher recall with respect to EdgeBoxes although they are more than an order of magnitude slower.

2.2.2 Spatio-Temporal Objectness

Objectness proposal in videos is typically cast as a problem of supervoxel segmentation, although supervoxel evaluation measures - such as those used in [142] - are reported as not being directly indicative of the performance of such methods when applied to spatio-temporal objectness proposal [104]. Van den Bergh *et al.* [131] have addressed the problem by tracking windows aligned with supervoxels, obtained from frame superpixel segmentation [130], over multiple frames using an online optimization; the proposed method runs at 30fps on a single modern CPU. Oneata *et al.* [104] follow a similar approach, in principle, by segmenting individual frames into a superpixel graph, then computing supervoxels through temporal hierarchical clustering. Spatio-temporal object detection proposals are based on super-

voxel segmentation, obtained using a version of the region growing method of Manén *et al.* [93] extended to the temporal domain.

Spatio-temporal objectness measures have been used to perform co-localization, i.e. spatial localization of common objects in a set of videos. Prest *et al.* [108] have proposed a fully automatic pipeline to learn object detectors from object proposals; segments of coherent motion are extracted from video shots, and spatio-temporal bounding boxes are fit to each segment, forming video “tubes” that are then used to train detectors, following a selection process based on objectness probability. The approach proposed by Joulin *et al.* [73] extends the method of image co-localization of [126] to videos, extending it with temporal terms and constraints, and solves efficiently the resulting quadratic problem applying the Frank-Wolfe algorithm. Unlike [108], the method does not use video tubes. Kwak *et al.* [81] address video object detection as a combination of two processes, i.e. object discovery and tracking, that complement each other. During discovery, regions containing similar objects are matched across different videos, while tracking associates prominent regions within each video. Motion statistics of individual regions and temporal consistency between consecutive regions are used to improve tracking and obtain the video tubes for object detection.

Recently, convolutional neural networks have been applied to the problem of video object detection. Tripathi *et al.* [128] have proposed a video object proposal method based on spatio-temporal edge contents, and a deep-learning based method for video object detection applied to clusters of these proposals. Class labels are propagated through streaming clusters of spatio-temporal consistent proposals, speeding up detection by $3\times$ with respect to per-frame detection. Kang *et al.* [74] have proposed a framework for video object detection based on CNNs that detect and track proposals. In a first stage video tubelets are proposed, combining object detection, to provide high-confidence anchors to the tracker, and tracking, to generate new proposals and to aggregate detections. In a second stage tubelets are classified and re-scored through spatial max-pooling and temporal convolution, for robust box-scoring and for incorporating temporal consistency.

2.3 Video Coding

Traditional adaptive video compression approaches do not consider the semantic content of video and instead adapt compression depending on the

requirements of the network or device used to deliver video to the end user. Semantic video compression, instead, alters the video by taking into account objects [65, 77] or a combination of objects and events [8], using pattern recognition techniques. Bagdanov *et al.* [5] have proposed to use image features correlated with downstream detector features, like corners and edges, to select frame areas to be maintained at higher visual quality and smoothing the rest. This approach has been applied, as pre-processing step to H.264 encoding, to surveillance videos. Videos compressed using [5] have a very low bitrate. Moreover the performance of pedestrian detection is better on videos compressed with this approach with respect to those compressed with H.264 at a similar bitrate.

Chen *et al.* [17] have addressed surveillance video coding, segmenting moving objects from background using low-rank and sparse decomposition, and coding the background with a few independent frames, based on their linear dependency, to further reduce their temporal redundancy. Guo *et al.* [54] follow a similar approach, but separating background from moving objects by learning a background dictionary, and coding motion together with the reconstruction coefficients of the background.

2.3.1 Saliency Based Methods

Visual saliency deals with properties and qualities of items that stand out with respect to their neighbors, and typically arises from contrasts in terms of color, luminance, motion, etc. Works in this area typically aim at predicting salient points of human eye fixation [117] or modeling visual attention [10]. Because of this relation with the human visual system and its attentional model, saliency has been used to identify parts of images or video frames that are to be compressed lightly, to preserve their visual content.

Use of saliency maps for video coding has been proposed by Gupta *et al.* [56], where low-level and high-level saliency features are combined and used to perform a non-uniform bit allocation over video frames. Since computation of these saliency maps is expensive the authors propose to use a shot-detection method to select a reduced number of frames for this computation. Hadizadeh and Bajić [57] have proposed to add a saliency distortion term in the rate distortion optimization (RDO) processing of H.264/AVC, to improve the coding quality of regions of interest (ROI). Saliency is computed with an extended version of the Itti-Koch-Niebur saliency model [68], with an improved temporal saliency that accounts for camera motion. The proposed

RDO is ~ 3 times slower than the standard approach. Li *et al.* [85] have proposed the use of saliency maps in H.265/HEVC coding to drive the quantization parameter of the coding units. An initial map is computed using graph-based visual saliency [12], then a Markovian algorithm concentrates saliency in a few locations, followed by a Gaussian filtering.

Differently from [17,54] the proposed approach can be applied to standard video coding such as H.264 or H.265. Differently from [57, 76, 85, 156] the proposed approach does not compute a saliency that attempts to mimic the human visual system; instead, our saliency exploits objectness measures in to order to obtain a better performance with object recognition algorithms.

2.4 Image Restoration

Removing compression artifacts has been addressed in the past. There is a vast literature of image restoration, targeting image compression artifacts. The vast majority of the approaches can be classified as processing based [15, 24, 42, 86, 140, 144, 149, 152] and a few ones are learning based [29, 94, 125, 137]. In the following we will briefly review image processing methods and learning based methods. We will also cover other works solving different image transformation tasks which are related to our problem. Finally we will state our contributions in relation to existing state of the art.

2.4.1 Processing Based Methods

This class of methods typically relies on information in the DCT domain. Foi *et al.* [42] developed SA-DCT, proposing to use clipped or attenuated DCT coefficients to reconstruct a local estimate of the image signal within an adaptive shape support. Yang *et al.* [144], apply a DCT-based lapped transform directly in the DCT domain, in order to remove the artifacts produced by quantization. Zhang *et al.* [152], fuse two predictions to estimate DCT coefficients of each block: one prediction is based on quantized values of coefficients and the other is computed from nonlocal blocks coefficients as a weighted average. Li *et al.* [86] eliminate artifacts due to contrast enhancement, decomposing images in structure and texture components, then eliminating the artifacts that are part of the texture component. Chang *et al.* [15] propose to find a sparse representation over a learned dictionary from a training images set, and use it to remove the block artifacts of JPEG com-

pression images. Dar *et al.* [24] propose to reduce artifacts by a regularized restoration of the original signal. The procedure is formulated as a regularized inverse-problem for estimating the original signal given its reconstructed form, and the nonlinear compression-decompression process is approximated by a linear operator, to obtain a tractable formulation. The main drawback of these methods is that they usually over-smooth the reconstructed image. Indeed it is hardly possible to add consistent details at higher frequencies with no semantic cues of the underlying image.

2.4.2 Learning Based Methods

Following the success of deep convolutional neural networks (DCNN), a learning driven paradigm has recently emerged in the artifact removal literature. The basic idea behind applying a DCNN to this task is to learn an image transformation function that given an input image will output a restored version. Training is performed by generating degraded versions of images which are used as samples for which the ground truth or target is the original image. The main advantage of learning based methods is that, since they are fed with a large amount of data they may estimate accurately an image manifold, allowing an approximated 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. Dong *et al.* [29] propose artifact reduction CNN (AR-CNN) which is 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 is designed following sparse coding pipelines. Svoboda *et al.* [125] report improved results by learning a feed-forward CNN to perform image restoration; differently from [29] the CNN layers have no specific functions but they combine residual learning, skip architecture and symmetric weight initialization to get a better reconstruction quality.

2.4.3 Other Image Transformation Tasks

Similar approaches have been devised, to target different image transformation problems, such as image super-resolution [13, 23, 72, 82], style-transfer [45, 72] and image de-noising [150]. Zhang *et al.* [150] have recently addressed the problem of image denoising, proposing a denoising convolutional

neural networks (DnCNN) to eliminate Gaussian noise with unknown noise level and showing that residual learning (used in a single residual unit of the network) and batch normalization are beneficial for this task. The proposed network obtains promising results also on other denoising tasks such as super resolution and JPEG deblocking. Gatys *et al.* [45] have shown that optimizing a loss accounting for style similarity and content similarity it is possible to keep the semantic content of an image and alter its style, which is transferred from another source. Johnson *et al.* [72] propose a generative approach to solve style transfer, building on the approach of [45]. Their method improves in terms of performance with respect of [45], since the optimization is performed beforehand, for each style, it is possible to apply the transformation in real-time. Interestingly, with a slight variation on the learning, their method also can solve super-resolution. Kim *et al.* [78] use a deeper architecture [122] trained on residual images applying gradient clipping to speed-up learning. Bruna *et al.* [13] addressed super-resolution learning sufficient statistics for the high-frequency component using a CNN, Ledig *et al.* [82] used a deep residual convolutional generator network, trained in an adversarial fashion. Dahl *et al.* [23] propose to use a PixelCNN architecture for super-resolution task, applying it to magnification of 8×8 pixel images. Human evaluators have indicated that samples from this model look more photo realistic than a pixel-independent L2 regression baseline.

Chapter 3

Spatio-Temporal Closed-Loop Object Detection

Object detection is one of the most important tasks of computer vision. It is usually performed by evaluating a subset of the possible locations of an image that are more likely to contain the object of interest. Exhaustive approaches have now been superseded by object proposal methods. The interplay of detectors and proposal algorithms has not been fully analyzed and exploited up to now, although this is a very relevant problem for object detection in video sequences. We propose to connect, in a closed-loop, detectors and object proposal generator functions exploiting the ordered and continuous nature of video sequences. Different from tracking we only require a previous frame to improve both proposal and detection: no prediction based on local motion is performed, thus avoiding tracking errors. We obtain 3 to 4 points of improvement in mAP and a detection time that is lower than Faster R-CNN, which is the fastest CNN based generic object detector known at the moment.

3.1 Introduction

Object detection is one of the most important tasks of computer vision and as such has received considerable attention from the research community. Typically object detectors identify one or more bounding boxes in the image

containing an object and associate a category label to it. These detectors are specific for each class of objects, and for certain domains exist a vast literature of specialized methods, such as face detection [25, 96, 148] and pedestrian detection [28, 46].

In recent years the objectness measure, that quantifies how likely an image window is containing an object of any class [2], has become popular [3, 19, 32, 109, 129]. The popularity of objectness proposal methods lies in the fact that they can be used as a pre-processing step for object detection to speed up specific object detectors.

The idea is to determine a subset of all possible windows in an image with a high probability of containing an object, and feed them to specific object detectors. Object proposals algorithms perform two main operations: generate a set of bounding boxes and assign an objectness score to each box.

The window proposal step is typically much faster than the exhaustive evaluation of the object detector. Considering that a “sliding window” detector has typically to evaluate 10^6 windows, if it is possible to reduce this number to 10^3 – 10^4 , evaluating only these proposals, then the overall speed is greatly improved. In this sense objectness proposal methods can be related to cascade methods which perform a preliminary fast, although inaccurate, classification to discard the vast majority of unpromising proposals [62]. Reducing the search space of object bounding boxes has also the advantage of reducing the false positive rate of the object detector.

The great majority of methods for objectness proposal have dealt with images, while approaches to video objectness proposal are oriented toward segmentation in supervoxels [142], deriving objectness measures from the “tubes” of superpixels that form them [104, 131]. This process is often computationally expensive and requires to process the whole video.

In this chapter we present a novel and computationally efficient spatio-temporal objectness estimation method, that takes advantage of the temporal coherence of videos. The proposed method exploits the sequential nature of videos to improve the quality of proposals based on the available information on previous frames determined by detector outputs. We define this approach as closed-loop proposals, since we exploit not only the current frame visual feature but also the proposals evaluated on a previous frame. Integrating the output of objectness proposals with object detection, we obtain a higher detection rate when computing spatio-temporal objectness in videos and we also improve the detection running time.

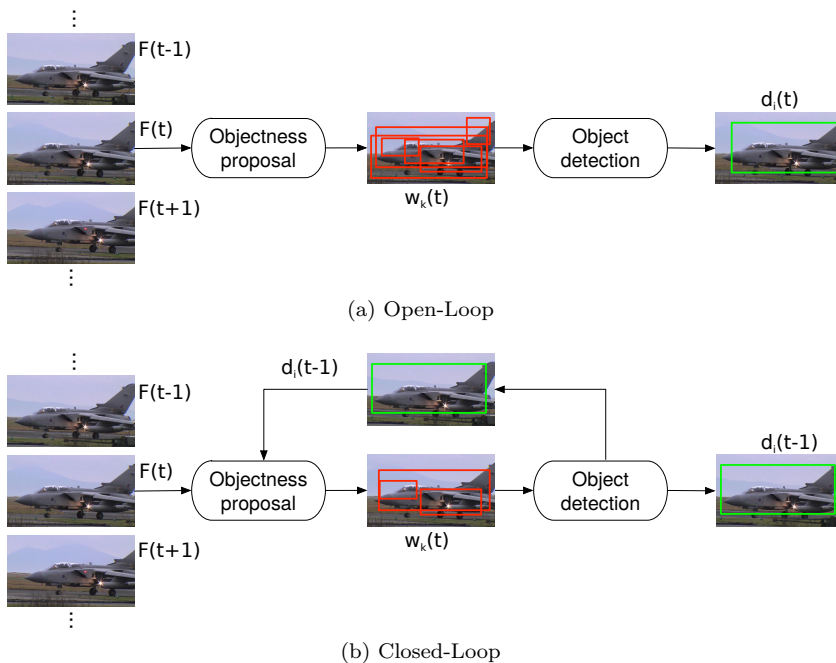


Figure 3.1: Schemes of: (a) typical objectness/detection pipeline; (b) our spatio-temporal objectness interaction. In our method window proposals are passed to the detector at time t and the detector output obtained at time $t - 1$ is fed back to the proposal algorithm to improve window ranking. This approach reduces the number of proposals w.r.t. typical pipeline.

We point out that our approach is different from tracking and is not based on any form of it. Object tracking, especially in the multi-target setting, is usually addressed using object detectors and some data association strategy that can be either causal [11] and non-causal [97]. In the proposed approach we exploit the temporal coherence of sequences causally, but we do not estimate motion of objects, either implicitly or explicitly. Moreover, our end goal differs from the one of tracking, that is to precisely locate an object instance in order to keep its identity correct as long as possible. Our goal is to enhance the quality of object proposals so to improve both detection quality and speed.

3.2 The proposed method

The method is based on the intuition that since objectness proposals are used as a pre-processing step followed by object detection, it is possible to exploit the joint statistics of window proposals and detections to compute spatio-temporal objectness in a video sequence, improving both detection rate and speed. Detection accuracy is improved by eliminating possible false detections, while processing speed is improved by selecting a reduced number of areas to be tested by the detector.

Typically window proposal methods require 10^3 windows to cover more than 90% of the objects shown in an image. In case objects are very small the number of proposals may become 10^4 . Considering video frame sequences, it is natural to use the detection of an object to improve the next proposal, since objects will likely be in about the same position in the next frame. Based on this consideration, we propose a feedback model accounting for spatio-temporal consistency of detections and window proposals over time, that re-ranks object proposals based on the overlap with detections and detector scores obtained in the previous frame. Using the outcome of a detector on a frame reduces the number, and improves the quality, of the proposals in a later frame. On the other hand those proposals are used to speed and improve the quality of detection in the following frame. In contrast to classical object detection pipelines, shown in Fig. 3.1a, our approach exploits previous frame detections to improve proposals. As shown in Fig. 3.1b, providing the detection as a feedback will allow to select a reduced number of higher quality proposals.

Given a video sequence with T frames, consider a set of object proposals

$$\mathcal{W} := \{w_1(1), \dots, w_P(1), \dots, w_1(T), \dots, w_P(T)\} \quad (3.1)$$

for the ease of notation we assume the proposal method computes a fixed amount of proposals P for each frame, but this is not a fixed requirement.

Considering the task of detecting objects from multiple classes, a set of models \mathcal{M} will be trained to output a vector of $|\mathcal{M}|$ scores for every window. A detector $C(F, w, \mathcal{M}) : \mathcal{F} \times \mathbb{R}^4 \rightarrow \mathbb{R}^{|\mathcal{M}|}$ is a function evaluating a proposal for a frame F according to some set of models \mathcal{M} and image features \mathcal{F} . Given a proposal $w_i(t)$ the detector C will associate it to a score vector $\mathbf{s}_i(t) \in \mathbb{R}^{|\mathcal{M}|}$.

Let \mathcal{D}_t be the set of scored proposals at time t defined by the tuples

$d_i(t) := \langle w_i(t), \mathbf{s}_i(t) \rangle$. The final set of detections $\overline{\mathcal{D}}_t$ is obtained preserving tuples d_i such that

$$\|\text{sign}(\mathbf{s}_i(t) - \tau_{\mathcal{M}})\|_1 > 0 \quad (3.2)$$

and performing non maximal suppression [47], where $\tau_{\mathcal{M}}$ is a model specific threshold vector on the soft-max per class output. To obtain detection windows useful for proposal re-ranking, we want to retain only the ones that have been assigned to at least one object class. This condition is ensured by the strict positivity of the L_1 -norm of the signs of thresholded classifier outputs vector as expressed by Eq. 3.2.

An object proposal method can be seen as a function, $P(w, F) : \mathcal{F} \times \mathbb{R}^4 \rightarrow \mathbb{R}$ evaluating the probability that a given window w in a frame F contains an object, independently from the object category, namely $p(\text{object}|w)$.

For a given frame at time t , our goal is to induce an ordering on set \mathcal{W}_t of proposals, exploiting information of previously evaluated ones $d(t-1) \in \mathcal{D}_{t-1}$, thus defining the ordered set $\widehat{\mathcal{P}}_t := \{\widehat{w}_1(t), \dots, \widehat{w}_P(t)\}$ such that

$$p(\text{object}|\widehat{w}_i(t)) > p(\text{object}|\widehat{w}_{i-1}(t)) \quad (3.3)$$

$$p(\text{object}|\widehat{w}_i(t)) > p(\text{object}|w_i(t)), i < \theta \quad (3.4)$$

The new ranking should keep the *objectness* property, defined by Eq. 3.3, meaning that highly ranked windows are more likely to contain an object than lowly ranked ones. According to Eq. 3.4, our re-ranked set $\widehat{\mathcal{P}}_t$ should have a better ranking than \mathcal{W}_t , meaning that, in the first θ windows, the probability of finding an object for the i -th window of our re-ranked set $\widehat{\mathcal{P}}_t$ is higher than for the same-rank window in \mathcal{W}_t .

We can define the likelihood of finding a generic object on the whole frame at time t as

$$\mathcal{L}_o = \sum_{i=1}^{|\mathcal{W}_t|} p(\text{object}|w_i) \quad (3.5)$$

and similarly

$$\widehat{\mathcal{L}}_o = \sum_{i=1}^{|\widehat{\mathcal{P}}_t|} p(\text{object}|\widehat{w}_i) \quad (3.6)$$

Considering that $\widehat{\mathcal{P}}_t$ is a re-ordered version of \mathcal{W}_t and that $|\mathcal{W}_t| = |\widehat{\mathcal{P}}_t|$, it is true that $\widehat{\mathcal{L}}_o = \mathcal{L}_o$. However, if Equation 3.3 and Equation 3.4 hold, a

more interesting result is obtained considering only a subset of the proposals; with the improved ranking we have that, for a $K < \theta$, in a truncated sum $\mathcal{L}_o^K = \sum_{i=1}^K p(\text{object}|w_i)$:

$$\widehat{\mathcal{L}}_o^K > \mathcal{L}_o^K. \quad (3.7)$$

This means that we can evaluate a set of lower cardinality K instead of the full proposal set without compromising the chance of finding the objects we are seeking with our classifier. Evaluating less proposals also means reducing the chance of finding false detections. This is an important benefit of our model that is useful to reduce the computational complexity and also to improve the accuracy of classifiers.

Since object detectors are trained to output a maximal score when the evaluated windows have high overlap with ground truth object windows, we can exploit detector scores as proxies of the probability of finding an object in the area occupied by an evaluated window w_i .

Therefore to obtain the new set of proposals \widehat{P} we link the detector and the proposal functions in a causal manner. Consider a set of N detections $d_i(t-1) \in \overline{\mathcal{D}}_{t-1}$, obtained from a frame at time $t-1$, and a set of proposals in frame at time t , it is possible to compute a spatio-temporal objectness at time t using for proposal window $w_k(t)$:

$$\widehat{o}_k(t) = o_k(t) + \alpha \sum_{m=1}^{|\mathcal{M}|} \sum_{i=1}^N \text{IoU}(w_k(t), d_i(t-1)) \cdot s_{im}(t-1) \quad (3.8)$$

where $o_k(t)$ represents the objectness score and

$$\text{IoU}(w, d) = \frac{\text{area}(w \cap d)}{\text{area}(w \cup d)} \quad (3.9)$$

is the overlap measure of the windows computed according to the PASCAL overlap criterion [35]. Term s_i is obtained via soft-max normalization therefore is comparable across classes without further calibration.

The IoU term makes sure that s_i can increase the objectness score of a proposal only if the detection window and the proposal window are overlapping, weighting the increase in objectness score by the overlap.

Finally, α is a parameter that weights the two parts of the function, and its optimal value is dependent on the dataset and the performance of the

proposal algorithm that is used. In the following experiments we tuned this parameter by cross-validation, maximizing detection rate with 1000 proposals (DET@1000) for each dataset and object detector used.

The function of Eq. 3.8, is composed by two parts:

- **Objectness measure.** The objectness score computed using a spatial objectness measure obtained from an object proposal algorithm such as BING or EdgeBoxes.
- **Feedback Term.** This term combines two terms via multiplication: *i)* the overlap measure $\text{IoU}(\cdot, \cdot)$ accounting for the fact that proposal windows that have larger overlap with detection windows are more likely to contain the objects detected in the next frame, and the higher the overlap the higher the probability of this; *ii)* the detection score s_{im} accounting for the fact that not all detection windows really contain objects, and this is more likely for windows with a low detector confidence score. Thus detection windows with higher detector confidence are to be weighted more, to rank higher the objectness windows that contain objects.

Using the spatio-temporal objectness measure of Eq. 3.8 allows to greatly reduce the number of object proposal windows.

The main differences of the proposed method with respect to previous approaches can be summarized as follows. Differently from the [81, 108] video object proposal methods, and from the video object detection methods of [74, 128], the proposed method does not perform any tracking although it is possible, in principle, to track the \hat{P} proposal windows to obtain video tubes. However, experimental results show that even without this additional processing it is possible to outperform the methods of [74, 81, 108] on two standard datasets. Differently from [73, 81, 108] the proposed method is supervised, as [74]. Differently from [128], that extends EdgeBoxes from image object proposals to videos exploiting temporal edge responses, the proposed method is based on image objectness measures, and the temporal aspect is included in Eq. 3.8. This allows to choose different proposal methods, e.g. depending on the needed speed or performance.

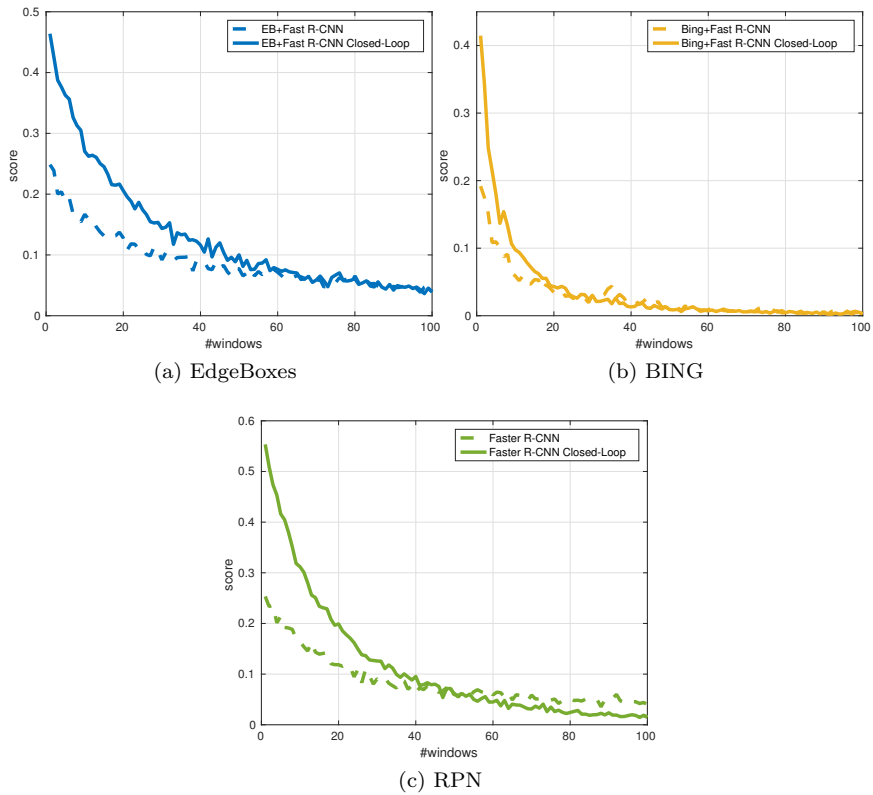


Figure 3.2: Average box detector score varying box rank on Youtube Objects. Proposals obtained with our method have higher scores in average and highly scored proposal have higher rank with respect to the baselines.

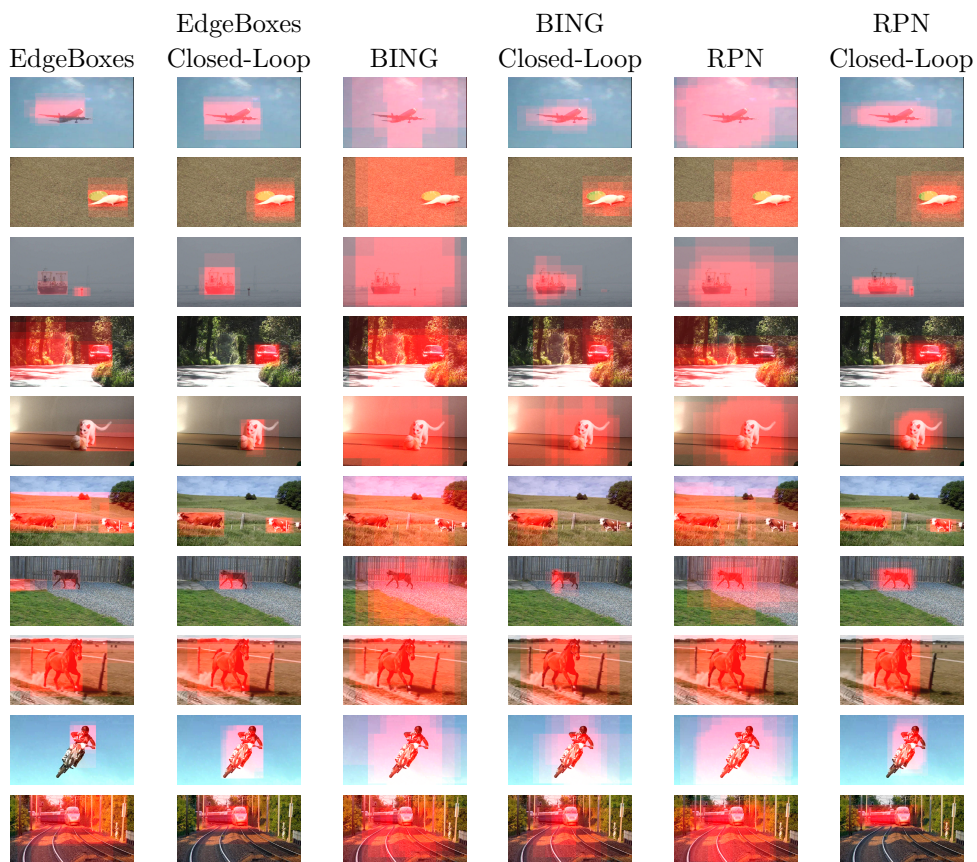


Figure 3.3: Sample frames from the 10 classes from YouTube Objects dataset with the 10 highest ranked boxes. Baselines are presented in odd columns and our improved closed-loop proposal on even columns. Each box is represented as an overlapping transparent red box on the image. Our closed-loop proposal are more concentrated and accurate with respect to baseline methods.

3.3 Experimental Evaluation

In the following experiments we evaluate the performance of the proposed method on videos, comparing it with three fast state-of-the-art methods – BING¹, Edge Boxes² and Region Proposal Networks used by Faster R-CNN³ – in terms of detection rate and speed. The method has been tested on the YouTube Objects dataset (YTO) [108], commonly used to test video object detection and proposal methods, and on the ILSVRC 2015 VID dataset [1], commonly used to test video object detection.

The YouTube Objects dataset (YTO) [108] contains 10 classes and consists between 9 and 24 videos for each class; to eliminate issues due to video compression artifacts 570,000 decompressed frames are provided. We report the results, in terms of localization metric (CorLoc) [26] that is typically used for evaluation on YTO; this experimental setup allows to compare the proposed method with the approaches of Prest *et al.* [108], Joulin *et al.* [73], Kwak *et al.* [81] and Kang *et al.* [74].

The ILSVRC 2015 VID dataset release used is the initial one, containing 30 object classes and consisting of 3 splits: a training set of 1952 fully-labeled video snippets with a length between 6 to 5213 frames per snippet; a validation set of 281 fully-labeled video snippets with a length between 11 to 2898 frames per snippet; a test set of 458 snippets whose ground truth annotation is not publicly available. We report the results, in terms of mean average precision (mAP), on the validation set; this experimental setup allows to compare the proposed method with the approach of Kang *et al.* [74].

The ILSVRC 2015 DET dataset comprises the fully annotated synsets from 200 basic level categories selected to provide various challenges such as object scale, level of image clutteriness and average number of object instances.

We used Fast R-CNN as object detector using the implementation from [112]. For the YouTube Objects dataset our model has been trained using the Faster R-CNN framework starting from the pre-trained network named VGG_CNN_M_1024 [16], fine-tuning both the classifier and the region pro-

¹We used the code publicly available at <http://mmcheng.net/bing/>

²We used the code publicly available at <http://research.microsoft.com/en-us/downloads/389109f6-b4e8-404c-84bf-239f7cbf4e3d/>

³We used the code publicly available at <https://github.com/rbgirshick/py-faster-rcnn>

posal net on PASCAL VOC 2007, since the YouTube Objects dataset object classes are a subset of the PASCAL VOC 2007 dataset.

For the ILSVRC 2015 VID dataset we trained the model using the pre-trained network named VGG.16 [122] as a starting point, fine-tuning both the classifier and the region proposal on the whole ILSVRC 2015 DET training set and some additional images from the training set of the ILSVRC 2015 VID dataset, choosing the ratio of 4 : 1 between DET and VID sets.

Faster R-CNN learns a Region Proposal Network (RPN) and an object detector which is architecturally equivalent to Fast R-CNN. Therefore the object detector weights are transferable to Fast R-CNN on which Faster R-CNN is based on. Indeed we used the same object detector weights in both frameworks. We refer to the detector as Faster R-CNN when we used Fast R-CNN and RPN as proposal sharing the weights, as referred by Ren *et al.* [112], and we refer to Fast R-CNN when proposals are computed externally.

3.3.1 Spatio-temporal objectness performance

In this set of experiments we evaluate the performance of the proposed spatio-temporal objectness method in terms of proposal correct localization.

The analysis of the behavior of our re-ranking process is shown in Fig. 3.2. We report the score of the detector on boxes of each rank, averaged over all frames and classes – we do not consider the scoring of detectors of classes different from the one present in the ground truth. This experiment shows that our boxes have a higher average detector score, meaning they are more precisely located on the object; moreover it can be seen how the highly scored boxes are all concentrated in the first 30-50 proposal while for the baseline methods they are more spread along the tail of the curve. A first qualitative glance at how our closed-loop spatio-temporal proposal improves over static baselines can be given in Fig. 3.3. It is clear, in this subset of frames, that our method increases the accuracy and quality of proposals generated by all baselines.

In Fig. 3.4 we evaluate the performance of proposals alone in terms of CorLoc on YTO. In this experiment we do not test if objects are correctly classified but only if proposal bounding boxes overlap with objects of any class. We compare all open-loop baselines and our closed-loop proposals with the method proposed by Oneata *et al.* [104]. The method of [104] has a performance close to BING, when using very few windows, but as the number

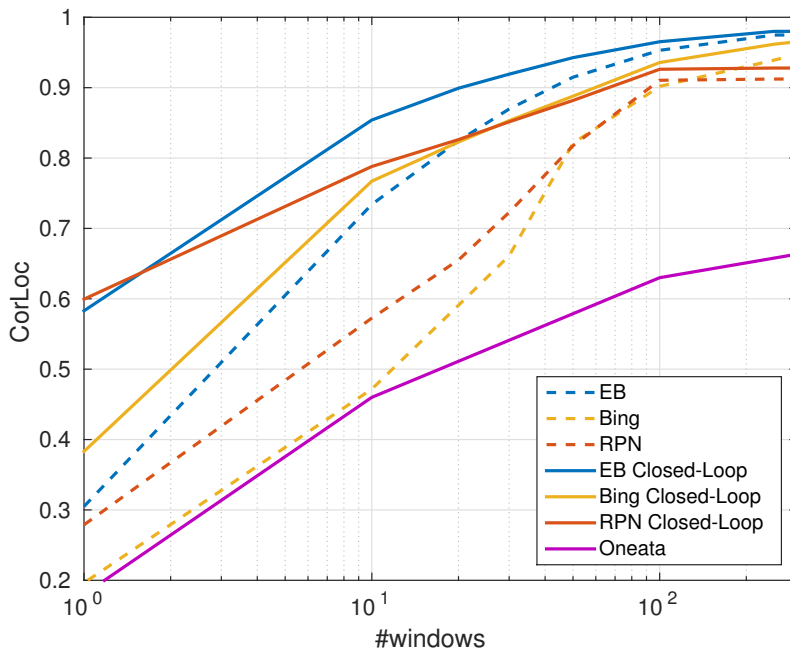


Figure 3.4: Trade-off between detection rate and number of window proposals for the YouTube Objects dataset. Comparison between the proposed method with temporal information using Fast R-CNN object detector, the proposed method without temporal information and the method of Oneata *et al.* [104]. The proposed spatio-temporal objectness measure greatly improves the performance w.r.t. image based objectness.











Method											Avg.
Prest <i>et al.</i> [108]	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5
Joulin <i>et al.</i> [73]	25.1	31.2	27.8	38.5	41.2	28.4	33.9	35.6	23.1	25.0	31.0
Kwak <i>et al.</i> [81]	56.5	66.4	58.0	76.8	39.9	69.3	50.4	56.3	53.0	31.0	55.7
Kang <i>et al.</i> [74]	94.1	69.7	88.2	79.3	76.6	18.6	89.6	89.0	87.3	75.3	76.8
RPN Closed-Loop	70.7	76.0	70.2	93.2	76.5	88.6	87.4	84.4	81.4	67.9	79.6
RPN	48.5	56.3	55.7	61.2	68.7	69.6	62.2	80.5	34.0	53.6	59.0
EdgeBoxes Closed-Loop	87.8	94.8	81.7	95.1	84.3	97.5	78.0	61.0	94.8	76.8	85.2
EdgeBoxes	71.9	72.9	75.6	86.4	52.2	91.1	79.5	62.3	74.2	71.4	73.8
BING Closed-Loop	71.1	87.5	54.2	90.3	80.0	92.4	89.0	85.7	79.4	69.6	79.9
BING	35.2	55.2	42.0	55.3	67.8	54.4	46.5	64.9	25.8	50.0	49.7

Table 3.1: Localization performances on the YTO dataset. We run all proposal methods with 10 windows per frame in the baseline and Closed-Loop (CL) version.

of window proposals increases this is reverted. Our closed-loop proposal ranking obtains very high recall even with few tens of windows compared with open-loop baselines. Note that even if proposal recall is predictive of detector accuracy [62] evaluating detectors on proposals is necessary to assess the final detection result. This analysis is reported in the following Sect. 3.3.2.

Moreover, it has to be noted that the method of [104] is dominated by the LDOF optical flow computation and roughly requires 15 seconds to process each frame, instead of the 0.16 required by EdgeBoxes, 0.017 required by BING and 0.006 by RPN. Note that RPN timing is reported on a high-end GPU (NVIDIA Titan X) while BING, EdgeBoxes and the timing from [104] are reported using a single-core implementation on a 3.6 GHz CPU.

In Tab. 3.1 we compare with previously published methods [73,74,81,108]. Our method is above the state-of-the art reported by Kang *et al.* [74]. Note that our method and the one by Kang *et al.* both use an algorithm trained with object class supervision, while [73,81,108] are unsupervised.

3.3.2 Detection performance on video

In the following set of experiments we evaluate the closed-loop object detector on videos. We perform several comparisons to assess the behavior of our technique using three state-of-the art proposals EdgeBoxes, BING and RPN. We focus on the former since it runs in under 200ms per frame and it obtains state-of-the art results in terms of recall and detection accuracy [62].

We also evaluate the quality of our approach using BING which is less performant in terms of recall and detection accuracy but has a much lower run-time; indeed BING proposals can be computed in less than 20ms on modern CPUs. Finally we test our strategy with Faster R-CNN, the fastest and most performing detector tested on still images [112].

First we assess the effect of the number of evaluated proposals on detection accuracy. In Figure 3.5 it is clear that even with a very low number of proposals, as low as 30 per frame, we can obtain a mAP figure that is similar or better than the open-loop baselines using one order more of proposals.

The best performing proposal method on YTO is EdgeBoxes. Faster R-CNN is the second best. BING performs the worst but is surprisingly close to Faster R-CNN. Note that our closed-loop detection improves all three open-loop baselines.

As it can be seen from Fig. 3.5 our method improves the detection accuracy on both datasets, reducing false positives and selecting a set of higher quality proposals for the detector down stream. In this experiment we show how reducing the set of windows to a very compact set, 30 windows per frame, we are able to perform as well or better than with the full set of non re-ranked windows with the further benefit of speeding up the computation.

Considering the curves in Fig. 3.2 the RPN proposal appears to be the best although in term of detection is outperformed by EdgeBoxes. This happens because EdgeBoxes provides a better recall covering a higher percentage of objects in frames as is measured in Fig. 3.4. Being EdgeBoxes dataset agnostic it is likely that RPN is suffering from overfitting with respect to PASCAL VOC 2007, on which it is trained. We believe that this behavior depends on the fact that the model used on YTO has not been tuned on video frames. Instead, on ILSVRC we trained the detectors using frames from the DET and the VID training subsets. We believe that this improved performance is due to the additional tuning of the CNN on this larger set of data which also comprises video frames.

In Tab. 3.2 we report a comparison on YTO of our closed-loop detector using 50 proposals computed from BING, EdgeBoxes and using Faster R-CNN with the respective baselines.

Our method obtains from 3 to 4 points increase in term of mean average precision. We improve on all classes except for “boat”, that is the hardest class to detect. In this case the detection feedback has not enough quality to obtain a good re-ranking of proposals, therefore the exhaustive proposal











Proposal											mAP
RPN Closed-Loop	68.3	72.7	44.3	88.8	58.3	60.2	71.5	69.1	77.3	58.6	66.9
RPN	58.6	63.6	47.2	85.3	53.4	60.8	67.1	65.5	67.5	52.3	62.1
EB Closed-Loop	72.7	81.3	58.6	90.5	64.8	63.0	65.3	62.5	79.7	66.0	70.4
EB	71.3	75.2	59.2	86.2	54.1	62.5	65.9	62.7	78.8	60.7	67.6
BING Closed-Loop	62.2	79.7	50.0	84.3	53.3	56.9	69.5	66.2	76.4	62.5	66.1
BING	56.6	74.9	51.3	82.6	53.3	61.0	66.7	65.2	68.4	59.9	64.0

Table 3.2: Comparison of open-loop and closed-loop proposals on YouTube Objects dataset using Fast R-CNN as a detector with 50 boxes. Using less or more boxes per frame resulted with worst or equal performance for all proposals in open- and closed-loop setting.















Method																		mean AP
Kang [74]	72.7	75.5	42.2	39.5	25.0	64.1	36.3	51.1	24.4	48.6	65.6	73.9	61.7	82.4	30.8	34.4		
RPN Closed-Loop	74.8	59.3	44.8	35.9	37.0	56.7	31.9	54.3	26.2	74.1	58.1	91.8	53.3	63.5	57.1	23.5		
RPN	61.8	55.4	42.8	26.9	35.4	56.5	23.8	52.2	26.6	71.9	46.9	92.3	51.0	76.4	57.3	24.8		
EB Closed-Loop	44.3	56.4	50.6	17.3	25.1	61.8	16.4	45.9	26.0	72.7	53.0	36.2	60.9	76.1	55.4	16.3		
EB	54.2	38.1	22.5	14.3	20.8	46.2	13.0	54.2	21.0	63.4	51.1	58.0	39.7	33.7	19.5	0.2		
BING Closed-Loop	29.1	35.9	37.4	23.2	22.5	46.1	15.6	35.1	16.3	54.6	58.2	44.7	50.4	72.1	49.5	9.6		
BING	16.2	36.2	29.3	18.5	16.5	42.0	11.2	31.9	9.5	45.7	57.0	30.6	46.2	62.9	22.6	3.3		
Kang [74]	54.2	1.6	61.0	36.6	19.7	55.0	38.9	2.6	42.8	54.6	66.1	69.2	26.5	68.6				47.5
RPN Closed-Loop	68.7	0.0	66.7	15.2	19.1	73.1	34.9	29.2	34.1	85.1	59.4	72.1	36.6	62.0	50.0			
RPN	68.2	0.0	61.0	14.5	20.6	64.3	37.6	3.3	34.0	86.6	59.8	73.1	35.9	57.9	47.3			
EB Closed-Loop	67.4	0.0	55.2	20.9	35.9	65.0	27.8	0.1	33.0	84.3	63.3	81.4	16.4	42.4	43.6			
EB	30.7	0.0	59.0	5.4	40.8	74.9	25.5	0.0	18.4	74.5	60.2	73.7	5.5	30.3	35.0			
BING Closed-Loop	60.4	0.0	52.7	8.6	29.0	49.9	3.6	0.3	28.0	68.2	41.4	62.7	12.8	34.7	35.1			
BING	56.4	0.0	48.9	3.1	26.0	47.8	2.6	1.7	15.0	66.6	28.4	56.0	5.7	24.9	28.8			

Table 3.3: Comparison of our method with Kang *et al.* [74] on ILSVRC VID dataset using 20 boxes per frame. Closed-Loop improves the map of RPN on 20, EB on 25 and BING on 28 out of 30 classes. Moreover our approach using RPN improves over the current state-of-the art.

evaluation may perform better.

Our method is able to increase the detection performance by reducing the amount of false positives per frame since it process a set of proposal with a high likelihood of containing an object.

Tab. 3.3 reports results of our method applied to RPN, EB and BING baselines on the validation set from ILSVRC 2015 VID using just 20 windows per frame. It can be observed that our closed-loop approach improves for most of the 30 classes. The only severe issues are on the “monkey” and “squirrel” classes. These classes are the most challenging and the detection quality is not adequate to provide any benefit in the loop. Interestingly we can boost the mAP on “squirrel” from 3.3 to 29.6 for RPN. Another chal-

lenging class is “lion”, on this class our method obtains a high improvement for EB and BING, while on RPN we have a similar result. Out of 30 classes, closed-loop improves RPN on 20, EB on 25 and BING on 28. Finally, our Faster R-CNN model (RPN) using closed-loop improves over Kang *et al.* [74] using just 20 window proposals per frame. In our preliminary experiments, training only on frames from DET reported a lower mAP, e.g. 41.0 for RPN closed-loop. We believe that the distribution of visual features in video, mostly because of blur and compression artifacts differs from the one in still images, and adding a small amount, i.e. a 4 : 1 ratio, of VID frames to the training set helps fine-tuning the CNN and the proposal network, and leads to an improvement of almost 10 mAP points.

Our algorithm is based mainly on the re-ranking process expressed in Eq. 3.8, where the only free parameter is α . We show how the value of α influences detection performance for different proposal algorithms and amount of evaluated windows in Figure 3.6. The alpha parameter appears to correlate negatively with the amount of windows evaluated. Our understanding of this behavior is that since the set of feedback windows $\overline{\mathcal{D}}_{t-1}$ is the signal from which we obtain our information, if this signal is weak the feedback term must compensate this lack of information. Finally the behavior of α depends on the distribution of objectness scores o_k which can differ quite significantly between the analyzed methods.

In real-time applications such as automotive or visual surveillance it is likely not possible to analyse a stream at 30 frames per second, therefore a certain frame drop will occur causing the video to be processed at a lower frame rate. We are interested in analysing the performance of our approach in this more realistic setting. To assess the behavior of a closed-loop proposal we test it dropping frames, meaning that instead of using the frame before the one to be analysed as a source for detection windows $d_i(t-1)$, we use $d_i(t-n)$, $n \in [2, 15]$.

In Fig. 3.7 we show how much detection accuracy of our method degrades if the source of detection windows is farther in time with respect to the current frame. It can be seen that our closed-loop method always performs better than its open-loop baseline.

3.3.3 Execution speed

In Tab. 3.4 we report timing and mAP for our proposed closed-loop object detection method compared with the open-loop baselines. Our closed-loop

Proposal	Detector	Time/frame	Speed-up	mAP	GPU
RPN Closed-Loop	Faster R-CNN	56 ms	34%	66.9	yes
RPN	Faster R-CNN	75 ms		67.0	yes
EB Closed-Loop	Fast R-CNN	206 ms	21%	70.3	no
EB	Fast R-CNN	250 ms		69.2	no
BING Closed-Loop	Fast R-CNN	63 ms	70%	65.6	no
BING	Fast R-CNN	107 ms		65.6	no

Table 3.4: Timing of our Closed-Loop proposals combined with Fast R-CNN detector, also compared with region proposal networks (RPN) and Faster R-CNN detector. The GPU flag indicates whether the proposal set is generated using GPU. Detection is always performed on GPU.

method is able to produce a significant speed-up without losing detection accuracy; for EdgeBoxes we even obtain a better mAP with our closed-loop proposal with respect to the open-loop baseline.

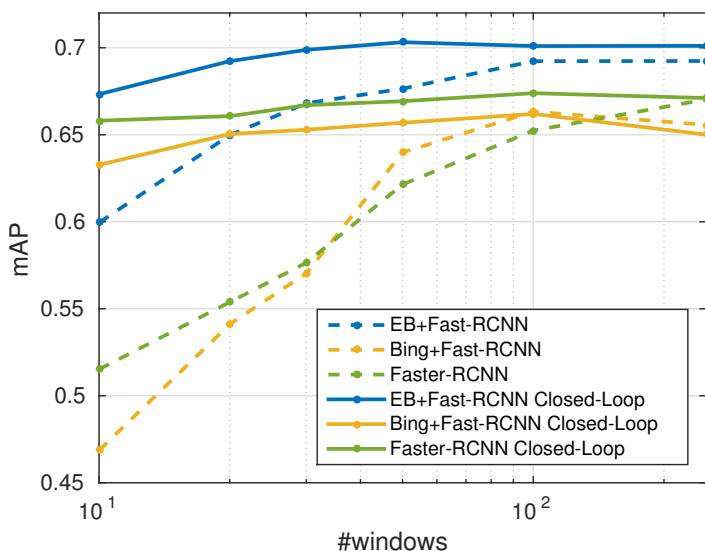
The gain in speed is higher for faster proposals since the full set of proposal has always to be computed before re-ranking and we can only reduce the amount of windows to be evaluated by the object detectors later in the pipeline.

3.4 Conclusion

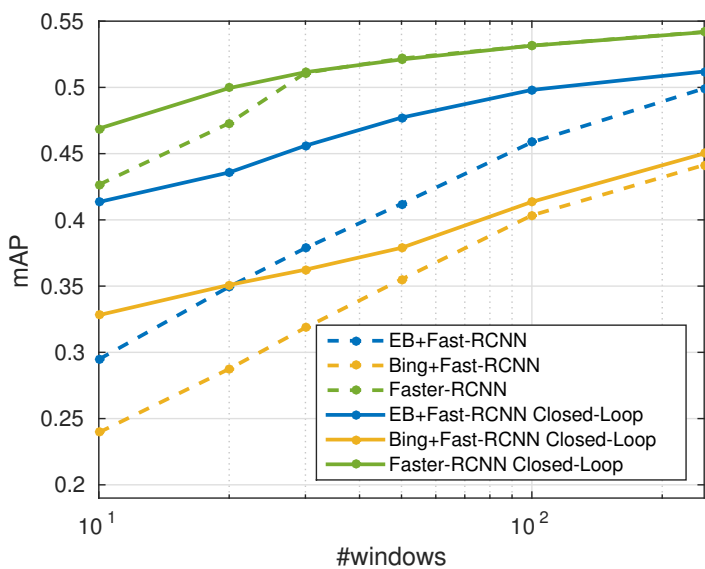
In this chapter we presented a novel closed-loop proposal strategy to be used on video sequences for object detection. Existing object proposal methods do not exploit the temporal ordering of frames. To the best of our knowledge we are the first to analyse and exploit the interplay between object detection and proposals. We show that our closed-loop strategy to generate proposals can improve speed and accuracy at the same time.

Our model is general and can be applied to any object detection pipeline on videos, which is based on window evaluation. We reported results using three state of the art object proposals in conjunction with Faster R-CNN, which is the fastest and most accurate object detector available. We measured a consistent improvement in proposal correct localization, detection accuracy and overall speed. The main limitation of our approach is constituted by the performance of the object detectors. If the open-loop detection quality is poor, the feedback can not provide any benefit.

Finally our method exploits the information of detectors in a causal manner and is robust to frame drop, thus providing ground for real-time applications.



(a) YouTube Objects



(b) ILSVRC 2015 VID

Figure 3.5: Detection accuracy with different proposals techniques and detectors on YouTube Objects and ILSVRC VID. Our Closed-Loop proposal improves mean average precision with respect to all baseline proposals. The gain is larger for a little amount of windows (10-50)

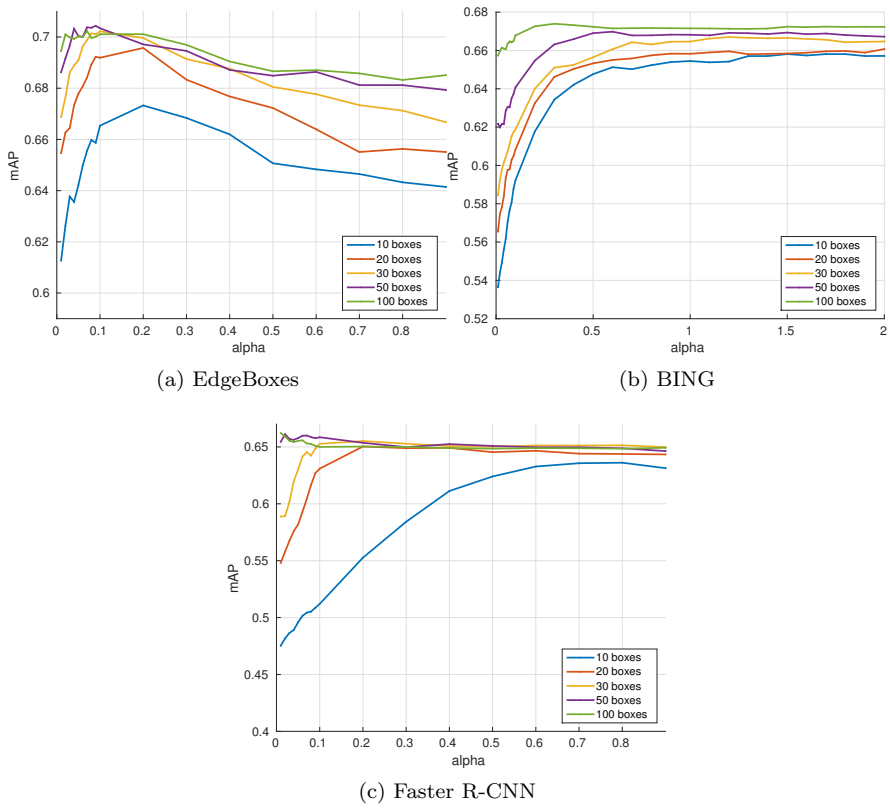


Figure 3.6: Effect of parameter α on detection accuracy for EdgeBoxes and BING varying the amount of proposals.

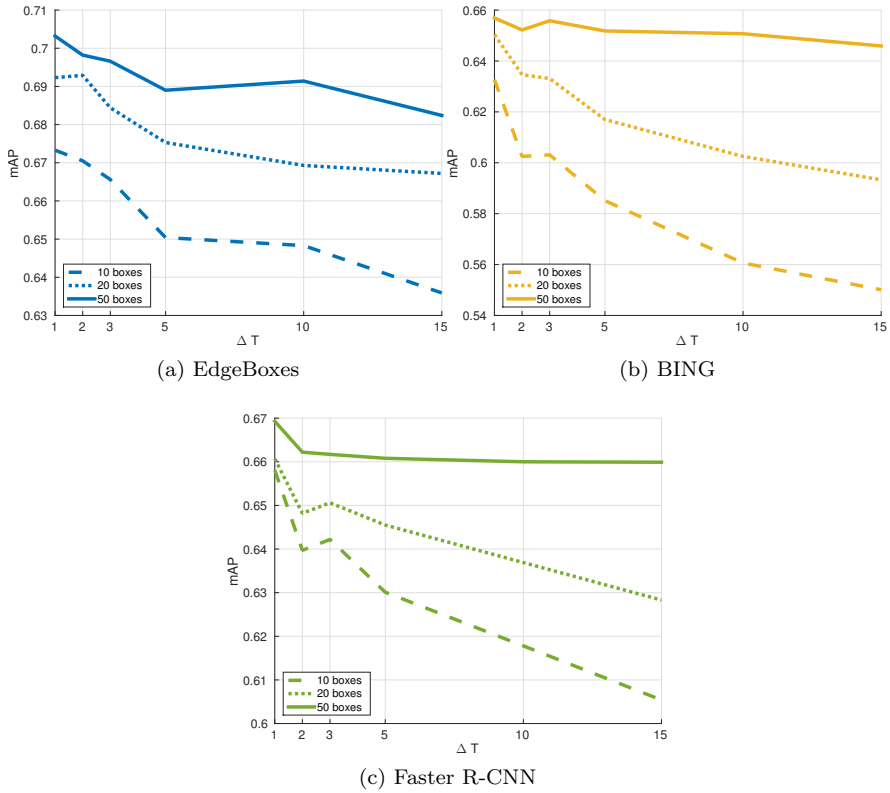


Figure 3.7: Mean Average Precision of Fast R-CNN with our spatio-temporal proposal varying the framerate. Full, dotted and dashed lines are referred to results obtained using the most relevant 50, 20 and 10 proposal respectively.

Chapter 4

Video Compression for Object Detection Algorithms

Video compression algorithms have been designed aiming at pleasing human viewers, and are driven by video quality metrics that are designed to account for the capabilities of the human visual system. However, thanks to the advances in computer vision systems more and more videos are going to be watched by algorithms, e.g. implementing video surveillance systems or performing automatic video tagging. This chapter describes an adaptive video coding approach for computer vision-based systems. We show how to control the quality of video compression so that automatic object detectors can still process the resulting video, improving their detection performance, by preserving the elements of the scene that are more likely to contain meaningful content. Our approach is based on computation of saliency maps exploiting a fast objectness measure. The computational efficiency of this approach makes it usable in a real-time video coding pipeline. Experiments show that our technique outperforms standard H.265 in speed and coding efficiency, and can be applied to different types of video domains, from surveillance to web videos.

4.1 Introduction

Video streaming applications that require transmission of a high number of streams to some central server, have to deal with issues such as limited bandwidth channels or a bandwidth bottleneck on the server itself. Some examples are video surveillance networks, UHF video streaming networks connecting dash cams installed in police cars, or streams transmitted from wearable devices or smartphones. All these applications require to transmit videos with a reasonable high-quality for further processing by vision-based systems, e.g. to identify anomalous activities, detect and identify persons, and detect objects. A way to improve the video compression methods currently used is to reduce the amount of irrelevant information transmitted in the video stream, compressing more the parts that do not contain semantically interesting objects. Typically this is performed by computing visual saliency maps based on some model of the human visual system.

Differently from this approach our method is designed to compute a binary saliency map designed for computer vision-based systems, considering the case of a system that performs automatic object detection. To this end we propose a fast objectness measure, that quantifies how likely an image window is containing an object of any class [2]. An objectness saliency map is computed from window proposals, and it is combined with H.265 quantization parameter map. We propose a method to learn a map that indicates to the codec which parts of the frame are relevant for a computer vision algorithm, combining a semantic and a low-level cue, maintaining also the perceptual quality for human viewers. Detector performance improves especially for low bitrates ($\sim 1\text{k}-2\text{k}$ Kb/s), while for similar bitrates our compression method preserves perceptual quality of relevant regions better than standard codecs. An additional result is that the proposed method is much faster than standard video coding.

4.2 Extremely Fast Object Proposals

The method is based on the intuition that image structure can be represented effectively with corner response that is binary by definition and does not require any approximation.

Each proposal window, selected with a sliding window approach, is scored

using a linear objectness model \mathbf{w} :

$$s_l = \langle \mathbf{w}, \mathbf{b}_l \rangle \quad (4.1)$$

where s_l is the filter score and \mathbf{b}_l is the binary visual feature, and $l = (i, x, y)$ is the location, size and position of the window. The objectness score used is:

$$o_l = q_i \cdot s_l + r_i \quad (4.2)$$

and is computed on a reduced set of window proposals from each size i , obtained through non-maximal suppression. This accounts for the fact that certain sizes are less likely to contain object instances. Training process requires to train \mathbf{w} in a first stage and then q_i and r_i for each quantized dimension i in a second stage.

4.2.1 Visual feature

In this work we propose to use corners, that can be extracted robustly and with a very low computational cost, and are natively binary. In particular, we use the AGAST algorithm [92], that is faster than FAST [114] algorithm and adapts itself to content of an image while processing it, without requiring a specific training. The basic idea of this class of detectors is to use the center of a circular area to determine if the nearby pixels that are darker or lighter describe a segment, without having to evaluate all the pixels. Typically FAST and AGAST use a mask composed by 16 pixels around a central point, but it is possible to obtain acceptable results using masks of 12 and 8 pixels (Fig. 4.1). To avoid accessing all the mask pixels an *accelerated segment test* (AST) is performed, i.e. a pixel is considered a corner if there are at least S connected pixels in the circle that are darker or lighter of a threshold based on the value of the center. This criterion defines a threshold that can be used to change the detector sensitivity. This features are extremely fast to compute since they only require integer comparisons and the traversal of a decision tree which can be stopped as soon as one of the AST conditions fails to be satisfied.

Corners are extracted from different versions of the original image, obtained by resizing the original image with different aspect ratios [19], and are used to compute our features in 8×8 windows of these images. The computed maps are binary vectors $\mathbf{b} \in \{0, 1\}^{64}$ where a pixel has value 1 if it contains a corner and 0 otherwise. To efficiently evaluate the scoring function in Eq. 4.2, which depends on the dot product $\langle \mathbf{w}, \mathbf{b}_l \rangle$, Hare *et al.* [58]

have shown that the vector $\mathbf{w} \in \mathbb{R}^{64}$ can be approximated by iteratively projecting it to a base of N_w vectors \mathbf{a}_j :

$$\mathbf{w} \approx \sum_{j=1}^{N_w} \beta_j \mathbf{a}_j. \quad (4.3)$$

Finally the scoring function s_l in Eq. 4.2 can be efficiently computed using:

$$\langle w, b \rangle = \sum_{j=1}^{N_w} \beta_j \left(2 \langle \mathbf{a}_j^+, \mathbf{b}_l \rangle - |\mathbf{b}_l| \right) \quad (4.4)$$

that, after storing a and b as 64bit integers in memory, is evaluated very efficiently only with bitwise AND and bit counts. In practice we set $N_w = 2$ as in [58].

Computational cost

The time required to extract the low-level features used in the proposed approach, i.e. image corners, is dependent on the content itself of each image and on the threshold used by AGAST. In general it is slightly faster than computing the gradient. This phase is not dominant w.r.t. the overall execution time. The main improvements in terms of speed, compared to BING, are in the lower number of operations required to compute the features in the 8×8 windows, and the calculation of the objectness measure. Table 4.1 shows the average number of atomic operations required for the different steps of the method proposed in this chapter, compared to that of BING.

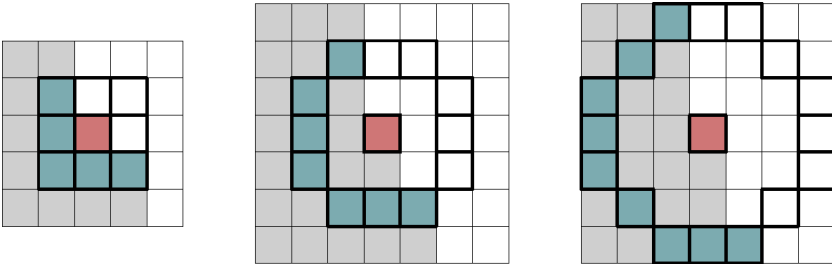


Figure 4.1: Masks of 8, 12 and 16 pixels used in AGAST.

		BITWISE			FLOAT		INT, BYTE	
		SHIFT	, &	CNT	+	×	+,-	min
Our method	AGAST	0	0	0	0	0	5-13	0
	Get feature	2	3	0	0	0	0	0
	Score	0	2	3	1	2	2	0
BING	Gradient	0	0	0	0	0	9	2
	Get feature	12	12	0	0	0	0	0
	Score	0	8	12	1	2	8	0

Table 4.1: Average number of atomic operations for computing objectness of each image window at different stages: calculate low-level features (gradient or AGAST), extract method features, and get objectness score. The proposed method requires less operations in each stage.

4.3 Coding with learned saliency

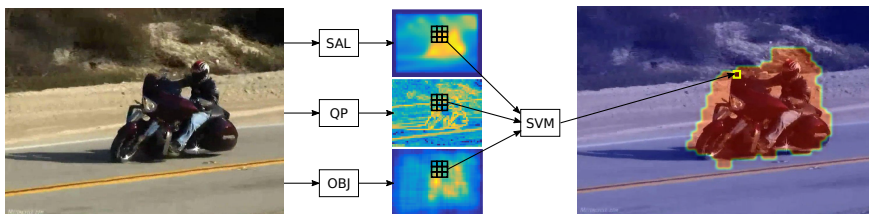


Figure 4.2: Our system pipeline. Binary saliency maps are predicted using [19, 113, 124] fused with our learned model. The final binary map is shown rightmost.

The goal of the proposed approach is to learn a saliency map that can drive compression of video frames in a way that is friendly for computer vision algorithms. The map is based on few features that are fast to compute, so to allow the application of the method to tasks that require real-time coding such as surveillance. Features are also related to the task of a computer vision-based system; in particular we have addressed the problem of object detection, that can be useful for both surveillance and automatic video tagging. This means that the map should indicate which part of the frame contains an object of interest for the algorithm. A secondary goal is to preserve visual quality, in terms of human visual system, for these objects.

Our method is based on modern video coding algorithms, specifically we use H.265, and the open source implementation x265 as practical reference implementation, but it can be applied also to H.264. Compression quality can be controlled with different strategies, e.g. using variable bitrate (VBR), constant bitrate (CBR) or with a constant rate factor (CRF); these approaches allocate the bits to different granularity levels of video, then reach the preallocated budget while encoding by adjusting the quantization parameter (QP) [9]. In particular, x265 trades distortion for bitrate, following different strategies, that typically adopt psycho-visual options that improve the perceived visual quality (e.g. favoring wrong motion over blur, or preserving the energy of the reconstructed patch). This approach is geared towards preserving the image appearance as much as possible. If the coded video has to be principally consumed by a machine we only need to keep the video quality good enough for the task at hand to be completed successfully. The video created with our approach is completely H.265 compliant and requires no changes in the decoder.

We propose to generate a binary saliency map, created according to the quadtree partitioning of the coding tree unit (CTU), indicating whether keeping the originally estimated QP or setting QP to the maximum value 51 (highest compression) [124]. We take into account *i)* our proposed objectness saliency map, *ii)* motion map, *iii)* visual saliency map and *iv)* QP map, and propose to combine them to predict a binary map. The objectness map is directly related to the task of the computer vision system that will consume the video, i.e. object detection, while motion and saliency map are used to weight how much visually conspicuous are the objects in the scene; finally, the QP map account for visual features such as texture and also for the secondary goal of the proposed approach, that is to obtain visually pleasant objects for the possible human viewers. Fig. 4.2 shows a schema of the proposed approach.

More formally, let us consider a video frame t , a patch of $N \times N$ pixels centered at location (x, y) , and $M_i, i = 1 \dots 4$ maps listed above. We define the operator $\mathcal{N}\mathcal{N}_N(x, y, M)$ that extracts the vector $[M_i(x - N, y - N) \dots M_i(x + N, y + N)]$ concatenating the values of a neighborhood of size $N \times N$ centered on (x, y) .

We learn a function $f(\mathbf{x}) \rightarrow [0, 1]$ using a RBF kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ setting γ and C by five-fold cross validation.

Positive samples are patches whose center belong to detected objects,

while negative patches are sampled from the background. Patch samples can be collected in an unsupervised manner by running an object detector with a high threshold on few frames; as an example on Town Centre we collect ~ 4000 samples using just 2 frames.

Each saliency map M_i is normalized to zero mean and unit variance.

In the following we review the maps, extracted with known algorithms, that we combine to create the proposed objectness-based saliency map.

Objectness Saliency Map Considering a set of object proposals $\mathcal{S}_k \subseteq \mathcal{S}$ we can define an objectness saliency map M_k by accumulating how many proposals of \mathcal{S}_k are present in each frame at each position. Let us denote \mathcal{S}_k^t the set of proposals \mathcal{S}_k at frame t . For every pixel $p = (x, y)$ of frame t , we compute the corresponding objectness saliency map value $M_k^t(p)$ as the count of proposals enclosing this position:

$$M_k^t = \sum_{s \in \mathcal{S}_k^t} \Psi_s \quad (4.5)$$

where for each proposals $s \in \mathcal{S}_k^t$ we define the function of pixels p in an image:

$$\Psi_s(p) = \begin{cases} 1 & \text{if } p \in s \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

The map $M(x, y, t)$ represents at any moment of the video, how much each pixel is relevant with respect to the set \mathcal{S}_k . The more proposals overlap in one position the more likely this pixel belongs to an object and is therefore relevant for video coding. Our fast objectness proposal measure has been used, because of its low computational cost that makes it amenable for inclusion in a real-time compression system.

Motion Map It is simply the difference, pixel by pixel, of two consecutive frames.

Visual Saliency Map To consider the visually salient elements of the scene, we have selected the Fast and Efficient Saliency (FES) map [113], based on estimating saliency of local feature contrast in a Bayesian framework. Again, its good computational performance make it usable in a real-time context.

QP Saliency Map This map is the quadtree representation of the QP values used in the CTU of the HEVC encoder. QP values are computed over 16×16 pixels by the x265 encoder. This saliency is related to the energy of each patch considered.

4.4 Experiments

4.4.1 Execution speed

To evaluate the influence of the different operations performed by the proposed method, presented in Sect. 4.2.1, we compare the execution time of the proposed method w.r.t. BING and Edge Boxes. Table 4.2 reports speed in terms of frames per second (FPS), while varying the training set, except for Edge Boxes that does not require it. Experiments have been performed on a laptop PC with Intel i5-3210M dual core CPU @ 2.5 GHz on the Town Center dataset. The difference in speed measured for the proposal methods trained with different training sets is due to the different number of window sizes used at test time, that is based on the number of windows of different aspect ratio and size seen during training. In fact, training a window proposal method on VOC 2007 will produce extremely generic proposals with a large variation in scale and aspect ratio, while training it on a dataset with smaller variations in scale and aspect ratio, such as Caltech5K, will produce less variations. Caltech30K, again contains a wider variation in scales than Caltech5k, therefore requiring the method to evaluate more windows at test time; of course this has as a beneficial effect on detection rate as can be seen in Figures 4.3 and 4.4. The proposed method is ~ 4 times faster than BING, and $\sim 20 - 40$ times faster than Edge Boxes.

Since the feature used in our method requires to determine a threshold to compute the corners, we have performed an analysis of the variation in terms of speed and detection rate while changing threshold values. If the threshold is too low then AGAST will perform less and less early rejections, thus slowing the feature computation stage of the method, while if the threshold is too high we can expect to have less proposals. An excessive number of features is also detrimental for window proposals, since many corners will not be really associated to objects.

Fig. 4.5 reports the speed, in terms of frames per second, while varying the threshold. Extracting very few corners, by increasing the thresholds,

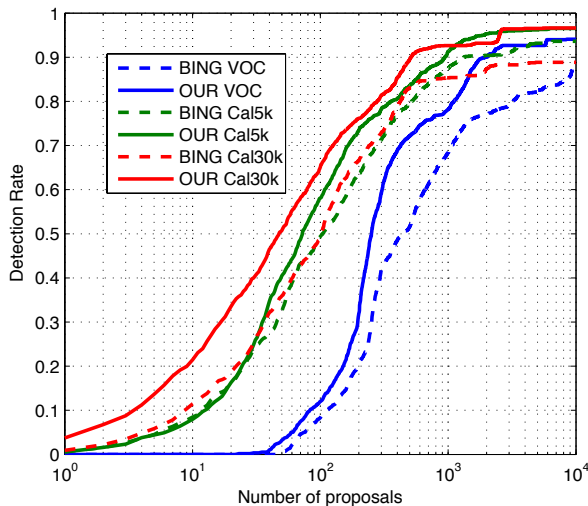


Figure 4.3: Trade-off between detection rate and number of window proposals for the Caltech dataset. Comparison between BING [19] and the proposed method, trained with PASCAL VOC 2007, Caltech 5000 frames and Caltech 30000 frames.

speeds up the process. Fig. 4.6 reports the detection rate at different speed – that depends on the threshold used in AGAST. This figure shows that high detection rates can be achieved within a relatively large range of threshold values. These figures have been obtained using the Town Center dataset.

4.4.2 Visual Quality

We evaluate structural similarity index (SSIM) [138], a visual quality metric that models the perception of compression artifacts, and the average log miss rate for a pedestrian detection scenario typical of surveillance videos. Furthermore to evaluate how the method generalizes we evaluated mean average precision (mAP) of generic object detectors on videos downloaded from YouTube. We compare these metrics with many baselines comprised the standard x265 codec for different bitrates.

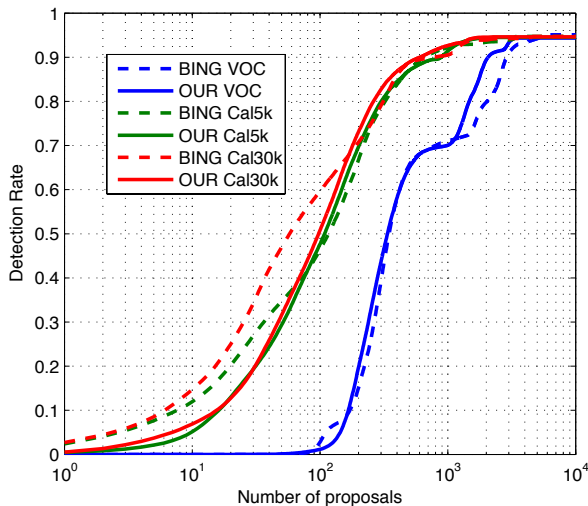


Figure 4.4: Trade-off between detection rate and number of window proposals for the Town Center dataset. Comparison between BING [19] and the proposed method, trained with PASCAL VOC 2007, Caltech 5000 frames and Caltech 30000 frames.

Surveillance Videos We used Town Centre [7] composed by a 5 minutes HD video, recorded at $1920 \times 1080 @ 25 \text{ fps}$ from a fixed camera, showing people walking in the street of a town, with 71,500 ground truth annotations of persons;

In the first experiment, we have compared the proposed method with a number of baselines. The original video has been compressed with the proposed method and with the baselines. We processed the resulting videos with the ACF pedestrian detector [27] and its detection results have been compared with the ground truth annotations. The baselines are: *i*) the x265 implementation of H.265 ; *ii*) a combination of QP map and motion map; *iii*) a combination of QP map with visual saliency computed with the method proposed by Walter and Koch [134]; *iv*) a combination of QP map, motion map and visual saliency by Walter and Koch; *v*) a combination of QP map, motion map and FES visual saliency.

It has to be noted that using the visual saliency of [134] results in a

Method	FPS
BING_VOC	17.22
Our_VOC	51.64
BING_Cal5K	24.38
Our_Cal5K	93.78
BING_Cal30K	19.20
Our_Cal30K	46.94
Edge Boxes	2.82

Table 4.2: Speed comparison of the proposed method w.r.t. BING; proposals are generated for the Town Center dataset, methods are trained with different datasets. The proposed method is ~ 4 times faster. Experiments have been carried on using an Intel i5-3210M CPU @ 2.5 GHz

computational cost that is higher than 150 ms; thus the combinations that use [134] are not suitable for applications that have constraints on processing time.

Fig. 4.7 reports the average log miss rate of the person detector; the lower the value the better the performance of the detector. The improvement with respect to the x265 baseline is due to the reduction of false detections, that are eliminated by the increased compression of non relevant parts of the frame. Comparison with the other baselines shows that adding the objectness-based map improves over other visual saliencies.

Fig. 4.8 evaluates the visual quality of the areas of interest for the detector, using the SSIM measure. The improvement obtained by the proposed approach is due to the fact that the encoder is able to better allocate the bits to the areas of interest, rather than distributing them also to the background. The comparison with the other saliency maps, that also beat the performance of standard x265, shows that the objectness measure better selects the elements of interest.

YouTube Videos For this experiment we used YouTube Objects [108], commonly used to test video object detection methods for different types of objects, evaluating the proposed approach on 10 classes of objects, from cats to trains, to assess the capability to generalize. Since the YouTube Objects dataset videos have typically been captured with non fixed cameras, the

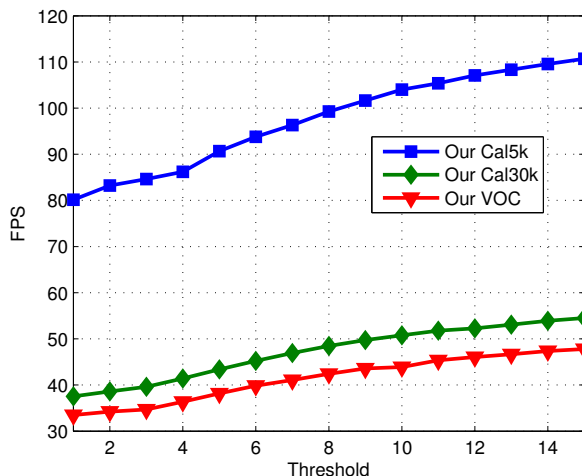


Figure 4.5: Trade-off between speed, expressed in terms of FPS, and AGAST threshold. The curves refer to different objectness methods trained with PASCAL VOC 2007, Caltech 5000 and Caltech 30000, tested on the Town Center dataset.

motion map is not useful, and it has not been used in this experiment.

Given the sparsity of ground truth annotations of YouTube Objects (usually just 1-2 frames per video), we have annotated 1500 objects. The method has been compared to a standard H.265 compression.

Object detections have been computed using Faster R-CNN [112], on the videos compressed with the proposed approach and with the baseline. We used the VGG-16 model, pretrained on Imagenet and fine-tuned on PASCAL VOC2007 [112]. No tuning of the detector has been performed on the YouTube Objects dataset.

Fig. 4.9 reports the mean average precision for the 10 object classes of YouTube object. Again, similarly to the previous case, the improvements of the detection can be attributed to the reduction of false positives.

Computational costs Another advantage of our method is in the reduction of computational costs in video encoding. In Figure 4.10 we show per-frame encoding time of our approach compared to H.265. We show the

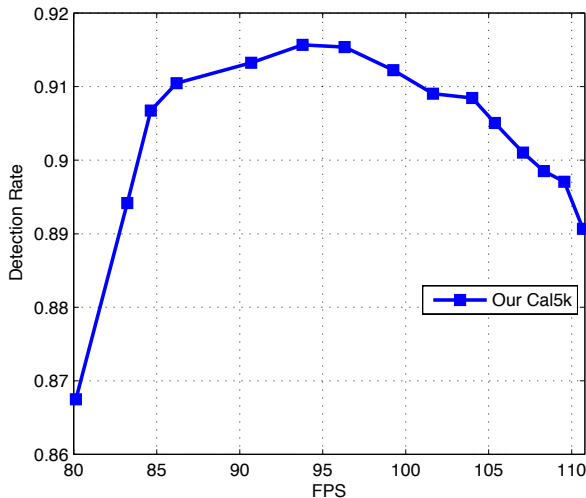


Figure 4.6: Trade-off between speed, expressed in terms of FPS, and detection rate @ 1000 proposals. The curve has been obtained testing a model trained with Caltech 5000 on the Town Center dataset.

timing for three presets of increasing quality and varying the CRF. As expected for both approaches a lower CRF implies more encoding time, setting on average higher QPs, thus leading to less sparse quantized coefficients for each CU.

This behavior is afflicting H.265 more than our adaptive coding algorithm. Frame blocks, predicted to be irrelevant, by our learned binary map, are quantized with the lowest quality. This mainly affects two aspects. Firstly the DCT coefficients will be mostly zero, leading to a less expensive coding later in the pipeline. Secondly, residues after block matching will be even more sparse and thus coded more efficiently.

Our method is more than two times faster than standard H.265 in coding video frames. This improvement is consistent for all H.265 presets.

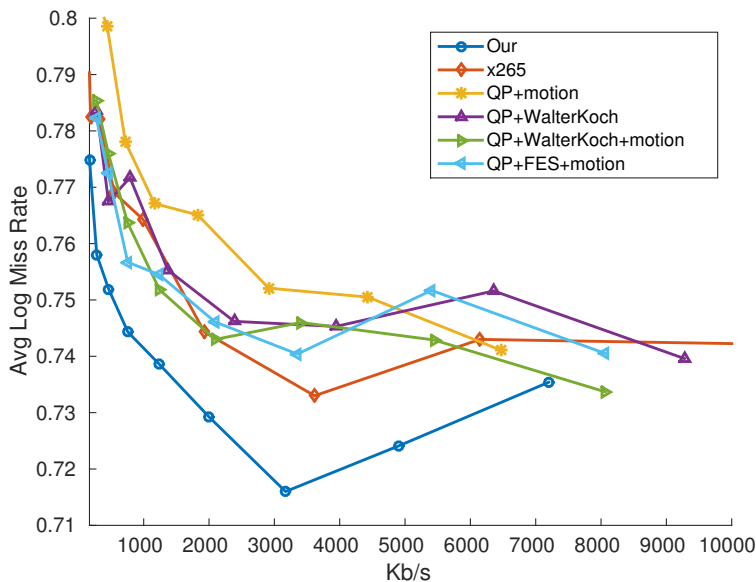


Figure 4.7: Detection Average Log Miss rate varying bitrate on Town Centre (lower is better).

4.5 Conclusions

We proposed a method for adaptive video coding based on a learned saliency. Our novel saliency drives video compression in order to preserve the appearance without damaging the performance of object detectors. Detector performance improves especially for low bitrates ($\sim 1\text{k}-2\text{k}$ Kb/s). We also show that for similar bitrates our compression method preserves perceptual quality of relevant regions better than standard codecs. Finally we also found that our codec is more than two times faster than standard H.265.

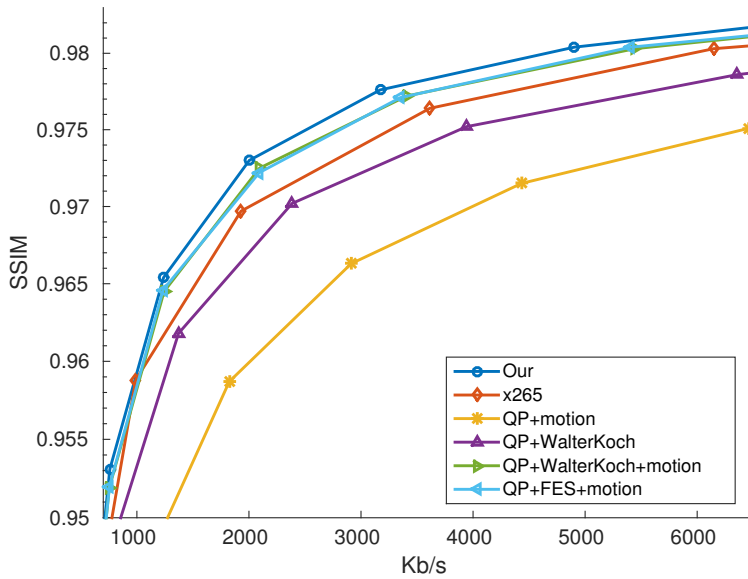


Figure 4.8: SSIM varying bitrate on Town Centre (higher is better).

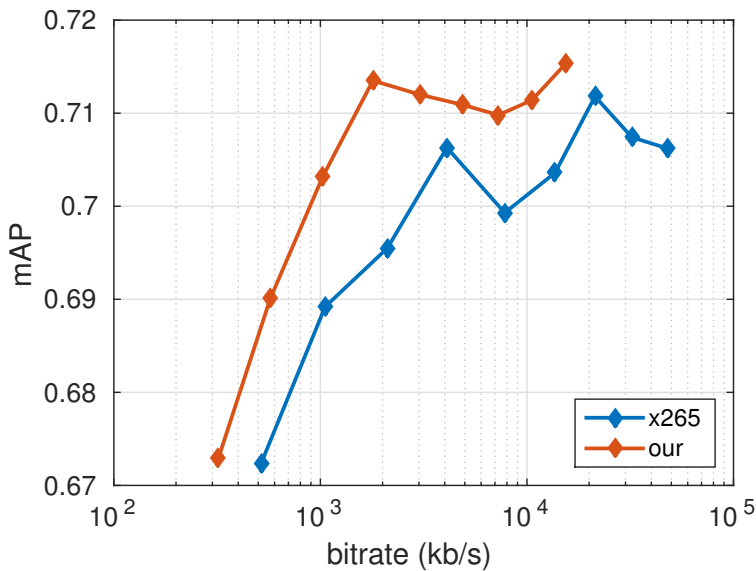


Figure 4.9: Mean average precision on YouTube Objects varying bitrate (higher is better).

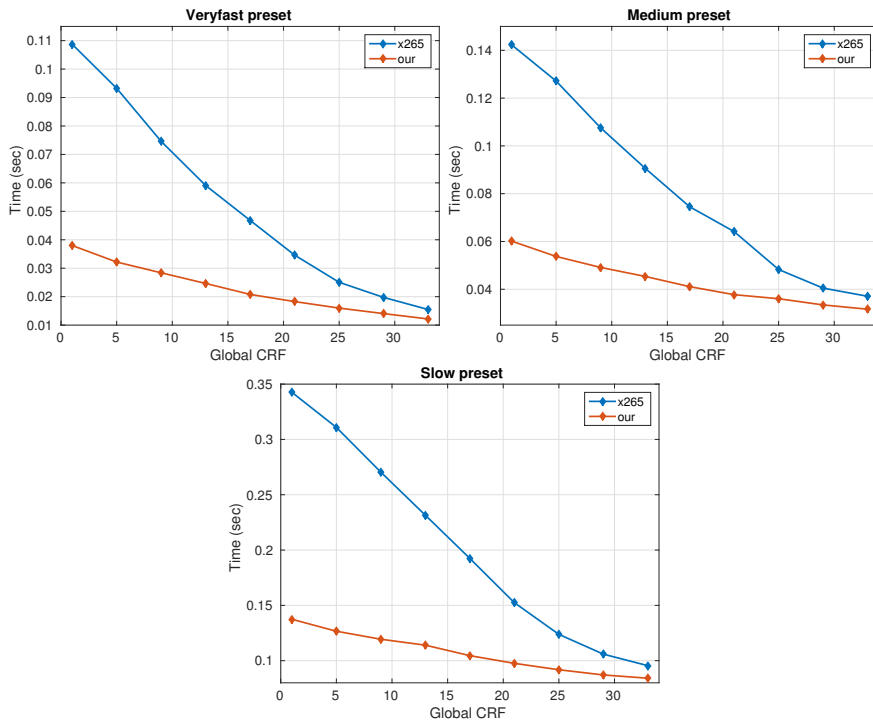


Figure 4.10: Coding time, per frame, of our codec compared to reference H.265. Our method is faster and depends less on the CRF (lower is better).

Chapter 5

Removing Compression Artifacts with Generative Adversarial Networks Enhances Image Semantics

Image compression is a need that arises in many circumstances. Unfortunately, whenever a lossy compression algorithm is used, artifacts will manifest. Image artifacts, caused by compression tend to eliminate higher frequency details and in certain cases may add noise or small image structures. There are two main drawbacks of this phenomenon. First, images appear much less pleasant to the human eye. Second, computer vision algorithms such as object detectors may fail to recognize categories. Removing such artifacts, means recovering the original image from a perturbed version of it. This means that one ideally should invert the compression process thorough a complicated non-linear image transformation. We propose an image transformation approach based on a feed-forward fully convolutional residual network model. We show how this model can be optimized traditionally using two losses as baselines. A structural similarity loss (SSIM) and a mean squared error loss (MSE). Finally, we reformulate the problem in a generative adversarial (GAN) framework. Our GAN is able to produce images with more photoreal-

istic details than MSE or SSIM based networks. We describe a novel training procedure based on sub-patches and devise a novel testing protocol to evaluate restored images quantitatively. We show that our approach can be used as a pre-processing step for object detection in case images are degraded by compression to a point that state-of-the art detectors fail. In this task, our GAN method obtains better performance than MSE or SSIM trained networks.

5.1 Introduction

Every day billions of images are shared on the web, and many more are produced and kept on private systems as mobile phones, cameras and surveillance systems. To practically store and transmit these images it is necessary to compress them, in order to reduce bandwidth and storage requirements. Apart from a few cases where compression has to be lossless, e.g. medical imaging or technical drawings, the algorithms used are lossy, i.e. they result in a more or less strong loss of content fidelity with respect to the original image data, to achieve a better compression ratio. A typical use case in which a high compression is desirable is that of web images, in which image files must be kept small to reduce web page latency and thus improve user experience. Another case is that of wireless cameras, in particular mobile and wearable ones, that may need to limit power consumption reducing the energy cost of image transmission applying strong compression. Also in tasks such as entertainment video streaming, like Netflix, there is need to reduce as much as possible the required bandwidth, to avoid network congestions and to reduce costs. 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 as shown in Fig. 5.1. 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.

So far, the problem of compression artifact removal has been treated

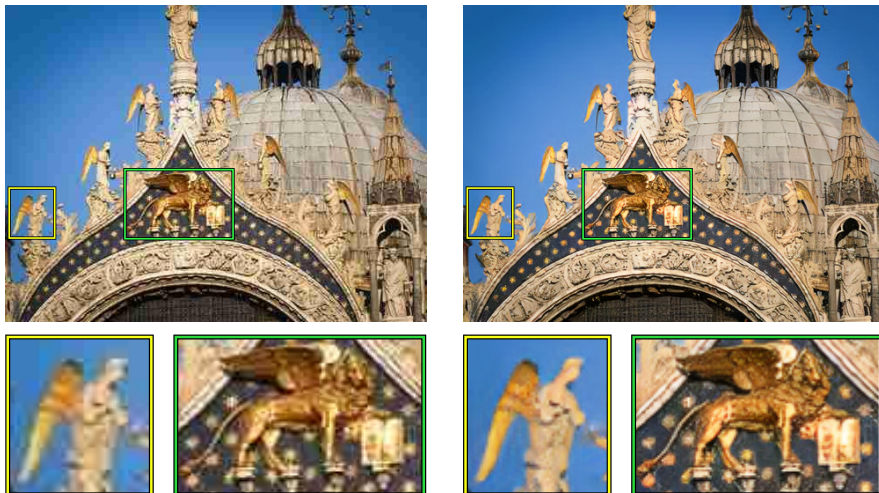


Figure 5.1: Left: A JPEG compressed image with two highlights of degraded regions. Right: our reconstruction where both regions are consistently sharper and most artifacts are removed. Best viewed in color on computer screen.

using many different techniques, from optimizing DCT coefficients [152] to adding additional knowledge about images or patch models [89]; however the very vast majority of the many works addressing the problem have not considered convolutional neural networks (CNN). To the best of our knowledge CNNs have been used recently to address artifact reduction only in two works [29, 125], while another work has addressed just image denoising [150]. These techniques have been successfully applied to a different problem of image reconstruction, that is super-resolution, to reconstruct images from low resolution, adding missing details to down-sampled images [82].

In this work we address the problem of artifact removal using convolutional neural networks. The proposed approach can be used as a post-processing technique applied to decompressed images, and thus can be applied to different compression algorithms such as JPEG, WebP, JPEG2000, intra-frame coding of H.264/AVC and H.265/HEVC. Compared to super resolution techniques, working on compressed images instead of down-sampled ones, is more practical, since it does not require to change the compression pipeline, that is typically hardware based, to subsample the image before its

coding; moreover, camera resolutions have increased during the latest years, a trend that we can expect to continue.

To evaluate the quality of reconstructed images, after artifact removal, there is need to evaluate both subjective and objective assessments. The former are important since most of the time a human will be the ultimate consumer of the compressed media. The latter are important since obtaining subjective evaluations is slow and costly, and the goal of objective metrics is to predict perceived image and video quality automatically. Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) are the most widely used objective image quality/distortion metrics. However, they have been criticized because they do not correlate well with perceived quality measurement [136]. Considering that the human visual system is highly adapted for extracting structural information from a scene, a framework for quality assessment based on the degradation of structural information, called Structural Similarity index (SSIM), has been introduced in [138]. Finally, we can expect that more and more viewers will be computer vision systems that automatically analyze media content, e.g. to interpret it in order to perform other processing. To consider also this scenario we have to assess the performance of computer vision algorithms when processing reconstructed images.

In this work we show how deep CNNs can be used to remove compression artifacts by directly optimizing SSIM on reconstructed images, showing how this approach leads to state-of-the-art result on several benchmarks. However, although SSIM is a better model for image quality than PSNR or MSE, it is still too simplistic and insufficient to capture the complexity of the human perceptual system. Therefore, to learn better reconstructive models, we rely on a Generative Adversarial Network where there is no need to specify a loss directly modeling image quality.

We have performed different types of experiments, to assess the diverse benefits of the different types of networks proposed in this chapter, using subjective and objective assessments. Firstly, we show that not only SSIM objective metric is improved, but also that performance of object detectors improve on highly compressed images; this is especially true for GAN artifact removal. Secondly, according to human viewers our GAN reconstruction has a higher fidelity to the uncompressed versions of images.

We make the following contributions. We define a deep convolutional residual generative network [59], that we train with two strategies. Similarly

to [125] our network is fully convolutional and is therefore able to restore images of any resolution. Differently from Svoboda *et al.* [125] we avoid MSE loss and we use a loss based on SSIM, this improves results perceptually. Nonetheless, as also happening in the super-resolution task, networks trained to optimize the MSE produce overly smoothed images; this behavior unfortunately is also present in our SSIM trained feed-forward network.

Generative adversarial networks [51], are instead capable of modeling complex multi-modal distributions and are therefore known to be able to generate sharper images. We propose an improved generator, trained in an adversarial framework. To the best of our knowledge we are the first proposing GANs to recover from compression artifacts. We use a conditional GAN [98], to allow the generator to better model the artifact removal task. An additional relevant novelty of this work is the idea of learning the discriminator over sub-patches of a single generated patch to reduce high frequency noise, such as mosquito noise, which instead arises when using a discriminator trained on the whole patch.

Finally, we propose a novel approach to evaluate the effectiveness of image restoration algorithm by analyzing the performance of object detectors on reconstructed images. We evaluate two tasks: object detection and object mask proposal generation. Our evaluation framework allows to obtain useful insights on the behavior of our method in conjunction with modern object detection frameworks.

5.2 Compression Artifacts

To understand the possible compression artifacts generated by a compression algorithm let us review the basic processing used in JPEG compression. First, the bitmap image is converted to the $YCrCb$ color space, to handle separately the luminance information Y and the color information encoded in the Cr and Cb components. Since the human visual system is able to discriminate the brightness of an image much more finely than its color information, the $Cr-Cb$ components are spatially subsampled. The next step is to split the downsampled pixels in the image into 8×8 pixel blocks, that are transformed using a Discrete Cosine Transform (DCT), to allow to handle separately 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. This is done using lossless techniques

such as RLE and Huffman coding.

Considering JPEG, the most common artifacts and distortions introduced 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 [120].
- texture deviation: due to the loss of fidelity in mid-frequency components, resulting in granular noise.
- blocking structures: this effect is due to the separation of the image in 8×8 pixel blocks, resulting in visible block edges at high compression rates. This happens also in MPEG video compression.
- posterizing, i.e. loss of color detail: depending on the color subsampling schema used, this results in loss of continuous tone gradation, that is transformed in abrupt changes..

Examples of these compression artifacts are shown in Fig. 5.2, where details of high quality images are compared to those of low quality high compression images.

5.3 Methodology

In the compression artifact removal task the aim is to reconstruct an image I^{RQ} from a compressed input image I^{LQ} . In this scenario, $I^{LQ} = A(I^{HQ})$ is the output image of a compression algorithm A with I^{HQ} as uncompressed input image. Typically compression algorithms work in the YCrCb color space (e.g. JPEG, H.264/AVC, H.265/HEVC), to separate luminance from chrominance information, and sub-sample chrominance, since the human visual system is less sensitive to its changes. For this reason, in the following, all images are converted to YCrCb and then processed.

We describe I^{RQ} , I^{LQ} and I^{HQ} by real valued tensors with dimensions $W \times H \times C$, where C is the number of image channels. Certain quality



Figure 5.2: Examples of compression artifacts. Top row: hi quality images; bottom row: low quality compressed images. *Left*: ringing artifacts (see chimney and roof); *Center*: texture deviation (see noise in black/yellow stripes); *Right*: posterization (see lack of smooth sky gradient); Blocking is visible in both left and right images. Best viewed in color and zoomed in.

metrics are evaluated using the luminance information only; in those cases all the images are transformed to gray-scale considering just the luminance channel Y and $C = 1$. Of course, when dealing with all the YCrCb channels $C = 3$.

An uncompressed image $I^{HQ} \in [0, 255]^{W \times H \times C}$ is compressed by:

$$I^{LQ} = A(I^{HQ}, QF) \in [0, 255]^{W \times H \times C} \quad (5.1)$$

using a compression function A , and using some quality factor QF in the compression process. The task of compression artifacts removal is to provide an inverse function $G \approx A_{QF}^{-1}$ reconstructing I^{HQ} from I^{LQ} :

$$G(I^{LQ}) = I^{RQ} \approx I^{HQ} \quad (5.2)$$

where we do not include the QF parameter in the reconstruction algorithm, since it is desirable that such function is independent from the compression function parameters.

To achieve this goal, we train a convolutional neural network $G(I^{LQ}; \theta_g)$ with $\theta_g = \{W_{1:K}; b_{1:K}\}$ the parameters representing weights and biases of the K layers of the network. Given N training images we optimize a custom loss function l_{AR} by solving:

$$\hat{\theta}_g = \arg \min_{\theta} \frac{1}{N} \sum_{n=1}^N l_{AR}(I^{HQ}, G(I^{LQ}, \theta_g)) \quad (5.3)$$

Removing compression artifacts can be seen as an image transformation problem, similarly to super-resolution and style-transfer. This category of tasks is conveniently addressed using generative approaches, i.e. learning a fully convolutional neural network (FCN) [91] able to output an improved version of some input. FCN architectures are extremely convenient in image processing since they perform local non-linear image transformations, and can be applied to images of any size. We exploit this property to speed-up the training process: since the artifacts we are interested in appear at small scales (close to the block size), we can learn from smaller patches, thus using larger batches.

We propose a generator architecture that can be trained with direct supervision or combined with a discriminator network to obtain a generative adversarial framework. In the following we detail the network architectures that we have used and the loss functions devised to optimize such networks in order to obtain high quality reconstructions.

5.3.1 Generative Network

In this work we use a deep residual generative network, which contains just blocks of convolutional layers and LeakyReLU non-linearities.

The architecture, shown in Fig. 5.3, is inspired by [59]. Specifically, we use convolutional layers with 3×3 kernels and 64 feature maps. Each convolutional layer is followed by a LeakyReLU activation. To reduce the overall number of parameters and to speed up the training time, we first use a convolution with stride 2 to obtain the feature maps half the original size, and finally we employ a nearest-neighbor upsampling as suggested in [103] to get the feature maps with original dimensions. We apply a padding of 1 pixel after every convolution, in order to keep the image size across the 15 residual blocks. We use replication as padding strategy in order to moderate border artifacts.

We add another convolutional layer after the upsampling layer to minimize potential artifacts generated by the upsampling process. The last layer is a simple convolutional layer with one feature map followed by a *tanh* activation function, in order to keep all the values of the reconstructed image in the range $[-1, 1]$ making the output image comparable with the input which is rescaled so to have values in the same range.

5.3.2 Loss Functions for Direct Supervision

In this section we deal with learning a generative network with a direct supervision, meaning that the loss is computed as a function of the reconstructed image I^{RQ} and the target original image I^{HQ} . Weights are updated with a classical backpropagation.

Pixel-wise MSE Loss

As a baseline we use the Mean Squared Error loss (MSE):

$$l_{MSE} = \frac{1}{WH} \sum_{x=1}^W \sum_{y=1}^H (I_{x,y}^{HQ} - I_{x,y}^{RQ})^2. \quad (5.4)$$

This loss is commonly used in image reconstruction and restoration tasks [29, 94, 125]. This kind of approach has shown to be effective to recover the low frequency details from a compressed image, but on the other hand most of the high frequency details are suppressed.

Generator Network

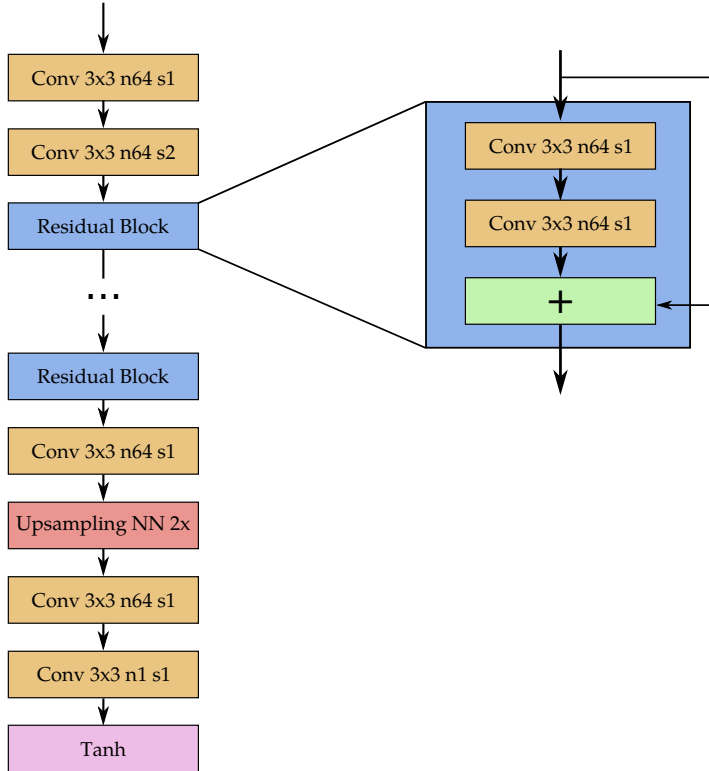


Figure 5.3: Architecture of Generator Network indicating with n the number of filters and with s the stride value for each Convolutional Layer.

SSIM Loss

The Structural Similarity (SSIM) [138] has been proposed as an alternative to MSE and Peak Signal-to-Noise Ration (PSNR) image similarity measures, which have both shown to be inconsistent with the human visual perception of image similarity. Given images I and J , SSIM is defined as follows:

$$SSIM(I, J) = \frac{(2\mu_I\mu_J + C_1)(2\sigma_{IJ} + C_2)}{(\mu_I^2 + \mu_J^2 + C_1)(\sigma_I^2 + \sigma_J^2 + C_2)} \tag{5.5}$$

We optimize the training of the network with respect to the structural

similarity between the uncompressed images and the reconstructed ones. Since the SSIM function is differentiable, we can define the SSIM loss as:

$$l_{SSIM} = -\frac{1}{WH} \sum_{x=1}^W \sum_{y=1}^H SSIM(I_{x,y}^{HQ}, I_{x,y}^{RQ}) \quad (5.6)$$

Note that we minimize $-SSIM(I^{HQ}, I^{RQ})$ instead of $1-SSIM(I^{HQ}, I^{RQ})$ since the gradient is equivalent.

5.3.3 Generative Adversarial Artifact Removal

The generative network architecture, defined in Sect. 5.3.1 can be used in an adversarial framework, if coupled with a discriminator. Adversarial training [51] is a recent approach that has shown remarkable performances to generate synthetic photo-realistic images in super-resolution tasks [82]. The aim is to encourage a generator network G to produce solutions that lay on the manifold of the real data by fooling a discriminative network D . The discriminator is trained to distinguish reconstructed patches I^{RQ} from the real ones I^{HQ} . To condition the generative network, we feed as positive examples $I^{HQ}|I^{LQ}$ and as negative examples $I^{RQ}|I^{LQ}$, where $|\cdot$ indicates channel-wise concatenation. For samples of size $N \times N \times C$ we discriminate samples of size $N \times N \times 2C$.

Discriminative Network

Our discriminator architecture uses convolutions without padding with single-pixel stride and uses LeakyReLU activation after each layer. Every two layers, except the last one, we double the filters. We do not use fully connected layers. Feature map size decreases as a sole effect of convolutions reaching unitary dimension at the last layer. A sigmoid is used as activation function. The architecture of the discriminator network is shown in Fig.5.4.

The set of weights ψ of the D network are learned by minimizing:

$$l_d = -\log(D_\psi(I^{HQ}|I^{LQ})) - \log(1 - D_\psi(I^{RQ}|I^{LQ})) \quad (5.7)$$

Discrimination is performed at the sub-patch level, as indicated in Fig. 5.4, this is motivated by the fact that compression algorithms decompose images

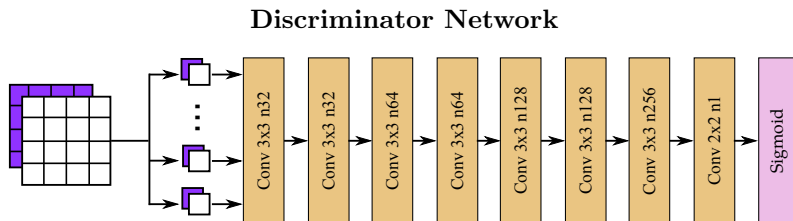


Figure 5.4: Architecture of Discriminator Networks indicating with n the number of filters for each Convolutional Layer. White squares indicate real (I^{HQ}) or generated patches (I^{RQ}), while purple ones are their respective compressed versions I^{LQ} .

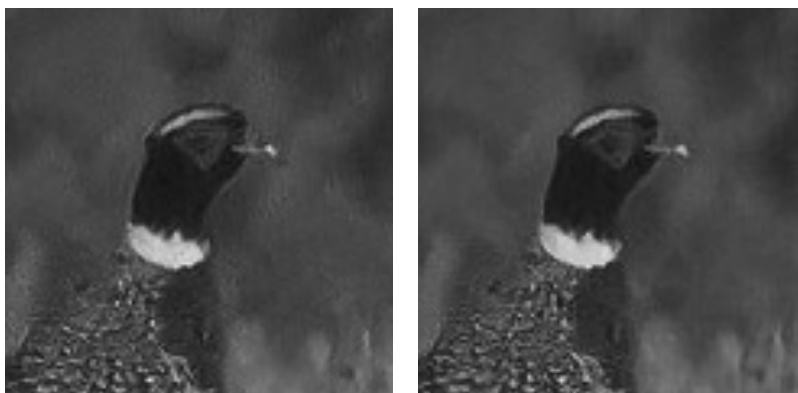


Figure 5.5: Left: reconstruction without sub-patch strategy. Right: our sub-patch strategy reduces mosquito noise and ringing artifacts.

into patches and thus artifacts are typically created within them. Since we want to encourage to generate images with realistic patches, I^{HQ} and I^{RQ} are partitioned into P patches of size 16×16 and then they are fed into the discriminative network. In Figure 5.5 it can be seen the beneficial effect of this approach in the reduction of mosquito noise.

Perceptual Loss

Following the contributions of Dosovitskiy and Brox [30], Johnson *et al.* [72], Bruna *et al.* [13] and Gatys *et al.* [44] we use a loss based on perceptual similarity in the adversarial training. The distance between the images is

not computed in image space: I^{HQ} and I^{RQ} are initially projected on a feature space by some differentiable function ϕ , then the Euclidean distance is computed between the feature representation of the two images:

$$l_P = \frac{1}{W_f H_f} \sum_{x=1}^{W_f} \sum_{y=1}^{H_f} \left(\phi(I^{HQ})_{x,y} - \phi(I^{RQ})_{x,y} \right)^2 \quad (5.8)$$

where W_f and H_f are respectively the width and the height of the feature maps. The model optimized with the perceptual loss generates reconstructed images that are not necessarily accurate according to the pixel-wise distance measure, but on the other hand the output will be more similar from a feature representation point of view.

Adversarial Patch Loss

In the present work we used the pre-trained VGG-19 model [122], extracting the feature maps obtained from the second convolution layer before the last max-pooling layer of the network. We train the generator using the following loss:

$$l_{AR} = l_P + \lambda l_{adv}. \quad (5.9)$$

Where l_{adv} is the standard adversarial loss:

$$l_{adv} = -\log(D_\psi(I^{RQ}|I^{LQ})) \quad (5.10)$$

clearly rewarding solutions that are able to “fool” the discriminator.

5.4 Experiments

5.4.1 Implementation Details

All the networks have been trained with a NVIDIA Titan X GPU using random patches from MS-COCO [88] training set. For each mini-batch we have sampled 16 random 128×128 patches, with flipping and rotation data augmentation. We compress images with MATLAB JPEG compressor at multiple QFs, to learn a more generic model. For the optimization process we used Adam [79] with momentum 0.9 and a learning rate of 10^{-4} . The training process have been carried on for 70,000 iterations. In order to

ensure the stability of the adversarial training we have followed the guidelines described in [116], performing the one-sided label smoothing for the discriminator training.

5.4.2 Dataset and Similarity Measures

We performed experiments on two commonly used datasets: LIVE1 [119] and the validation set of BSD500 [95] using JPEG as compression. For a fair comparison with the state-of-the-art methods, we report evaluation of PSNR, PSNR-B [146] and SSIM measures for the JPEG quality factors 10, 20, 30 and 40. We further evaluate perceptual similarity through a subjective study on BSD500. Finally we use PASCAL VOC2007 [36] and MS-COCO [88] to benchmark object detector performance for different reconstruction algorithms.

PASCAL VOC2007 is a long standing small scale benchmark for object detection, it comprises 20 classes for a total of roughly 11K images.

We use Microsoft Common Objects in Context (MS-COCO) [88] dataset, that contains 80 object classes and a total of more than 300K images. We used the 20.000 images in the *test-dev*.

5.4.3 Feature Maps Error

The first quantitative analysis we conduct is to understand if features computed using modern deep convolutional neural networks are affected, and how much, when an image is compressed. Another effect we quantify is how this variation in feature values is moderated if we apply artifact removal techniques. These results have been obtained running an object detection network based on Faster R-CNN [112] and comparing layer output for compressed and original images.

We run the following test, for every quality factor and method involved in our study, we compute the mean relative error of each layer:

$$\epsilon_l = \frac{|\phi_l(I^{RQ}) - \phi_l(I^{HQ})|}{\phi_l(I^{HQ})} \quad (5.11)$$

where $\phi_l(\cdot)$ are feature maps for layer l .

Results are reported in the plots of Fig. 5.6, that show the mean relative error, averaged over all layers, for different QF values. For higher QF values JPEG compression affects little, but noticeably feature map values. The

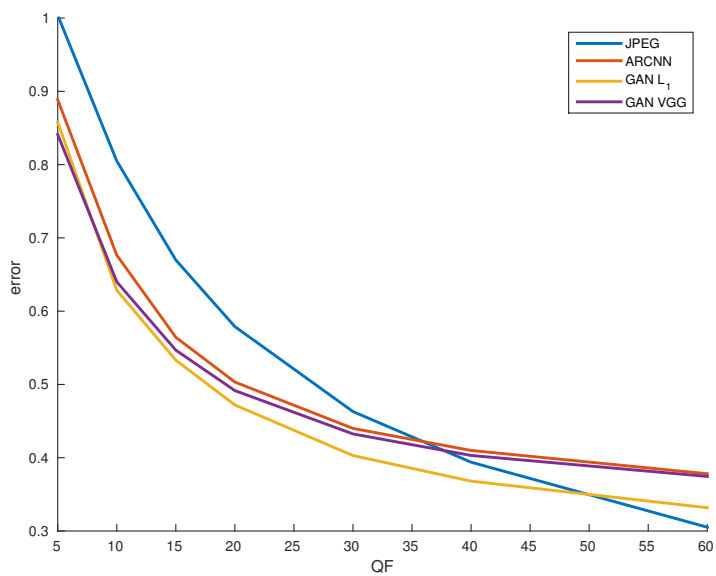


Figure 5.6: Mean relative error, averaged over all layers, for different QF and artifact removal techniques (the lower, the better). The proposed GAN restoration approach with L_1 loss obtains the smallest error; using VGG loss still improves over AR-CNN.

variation is closer to 30% for QF=60, and applying reconstruction methods on high quality images, as expected, does not produce any benefit. Clearly, when QF become smaller all reconstruction techniques help in generating images with feature maps closer to the original one, with GAN_{L_1} obtaining the best results and becoming effective from QF=50. The novel GAN approach obtains better results than AR-CNN also using VGG loss, but it is particularly effective when using L_1 loss for $QF \geq 20$.

In Fig. 5.7 we analyze the behavior for all feature maps, reporting the mean relative error for all the layers and different QF values. It is interesting to note that the first and last layers are less affected, while the ones that exhibit the most relative error are `conv3.2` and `conv4.2`. As also shown in Fig. 5.6, applying reconstruction is not beneficial for QF=60, while for other QF values it can be seen that the error is reduced for all layers, and specifically for the ones which are most affected. Notably, highest average relative errors can reach 100% ~ 150%.

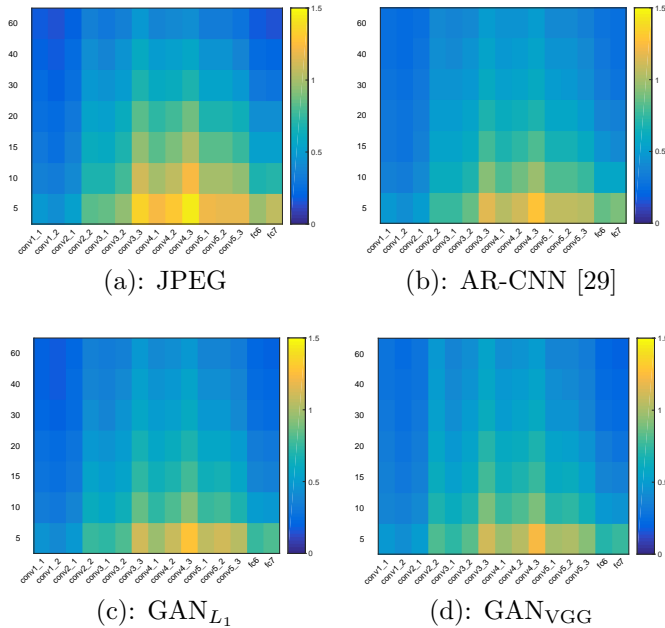


Figure 5.7: Mean relative error, for all layers, for different QF and artifact removal techniques (the lower, the better); the proposed GAN approach with L_1 loss obtains the least error.

5.4.4 Comparison with State-of-the-Art

We first evaluate the performance of our generative network trained without the adversarial approach, testing the effectiveness of our novel architecture and the benefits of SSIM loss in such training. For this comparison we have reported the results of our deep residual networks with skip connections trained with the baseline MSE loss and with the proposed SSIM loss. We compare our performance with the JPEG compression and three state-of-the-art approaches: SA-DCT [42], AR-CNN from Dong *et al.* [29] and the work described by Svoboda *et al.* [125]. In Table 5.1 are reported the results of our approaches on BSD500 and LIVE1 datasets compared to the other state-of-the-art methods for the JPEG restoration task. The results confirm that our method outperforms the other approaches for each quality measure. Specifically, we have a great improvement of PSNR and PSNR-B for the networks trained with the classic MSE loss, while as expected the SSIM measure improves a lot in every evaluation when the SSIM loss is chosen for training.

Regarding GAN, we can state that the performance is much lower than the standard approach from a quality index point of view. However, the generated images are perceptually more convincing for human viewers, as it will be shown in Sect. 5.4.7, in a subjective study. The combination of perceptual and adversarial loss is responsible of generating realistic textures rather than the smooth and poor detailed patches of the MSE/SSIM based approaches. In fact, MSE and SSIM metrics tend to evaluate better more conservative blurry averages over more photo realistic details, that could be added slightly displaced with respect to their original position, as observed also in super-resolution tasks [23].

5.4.5 Object Detection

We are interested in understanding how a machine trained object detector performs depending on the quality of an image, in term of compression artifacts. Compressed images are degraded, and object detection performance degrades, in some cases even dramatically when strong compression is applied. In this experiment we use Faster R-CNN [112] as detector and report results on different versions of PASCAL VOC2007; results are reported in Tab. 5.3. As an upper bound we report the mean average precision (mAP) on the original dataset. As a lower bound we report performance on images

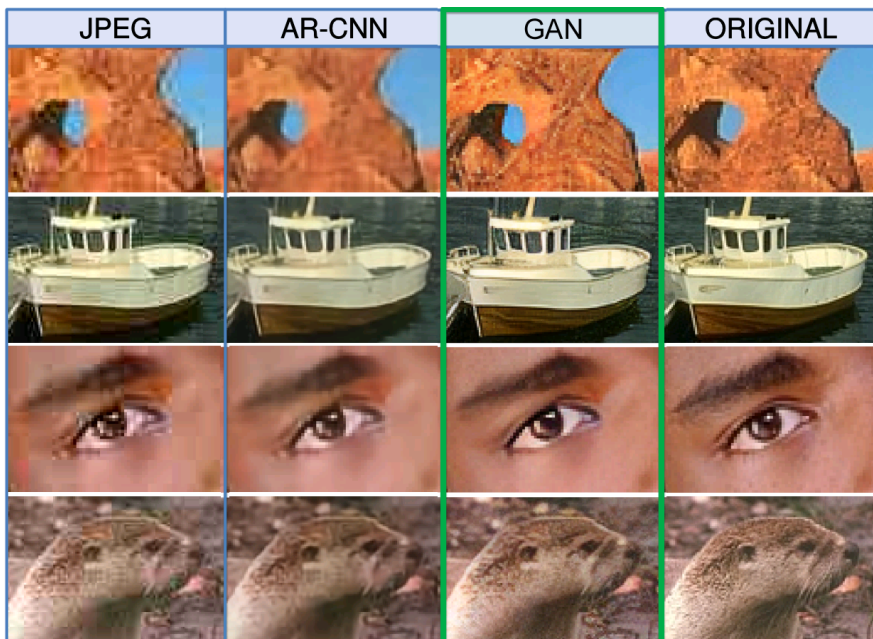


Figure 5.8: Qualitative results shown on two complex textured details. JPEG compression introduces severe blocking, ringing and color quantization artifacts. AR-CNN is able to slightly recover but produces a blurry result. Our reconstruction is hardly discernible from the original image.

Table 5.1: Average PSNR, PNSR-B and SSIM results on BDS500 and LIVE1. Evaluation using luminance.

QF	Method	LIVE1			BSD500		
		PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
10	JPEG	27.77	25.33	0.791	27.58	24.97	0.769
	SA-DCT [42]	28.65	28.01	0.809	-	-	-
	AR-CNN [29]	29.13	28.74	0.823	28.74	28.38	0.796
	L4 [125]	29.08	28.71	0.824	28.75	28.29	0.800
	Our MSE	29.45	29.10	0.834	29.03	28.61	0.807
	Our SSIM	28.94	28.46	0.840	28.52	27.93	0.816
	Our GAN	27.29	26.69	0.773	27.01	26.30	0.746
20	JPEG	30.07	27.57	0.868	29.72	26.97	0.852
	SA-DCT [42]	30.81	29.82	0.878	-	-	-
	AR-CNN [29]	31.40	30.69	0.890	30.80	30.08	0.868
	L4 [125]	31.42	30.83	0.890	30.90	30.13	0.871
	L8 [125]	31.51	30.92	0.891	30.99	30.19	0.872
	Our MSE	31.77	31.26	0.896	31.20	30.48	0.876
	Our SSIM	31.38	30.77	0.900	30.79	29.92	0.882
	Our GAN	28.35	28.10	0.817	28.07	27.76	0.794
30	JPEG	31.41	28.92	0.900	30.98	28.23	0.886
	SA-DCT [42]	32.08	30.92	0.908	-	-	-
	AR-CNN [29]	32.69	32.15	0.917	-	-	-
	Our MSE	33.15	32.51	0.922	32.44	31.41	0.906
	Our SSIM	32.87	32.09	0.925	32.15	30.97	0.909
	Our GAN	28.58	28.75	0.832	28.5	28.00	0.811
40	JPEG	32.35	29.96	0.917	31.88	29.14	0.906
	SA-DCT [42]	32.99	31.79	0.924	-	-	-
	AR-CNN [29]	33.63	33.12	0.931	-	-	-
	Our MSE	34.09	33.40	0.935	33.30	32.18	0.921
	Our SSIM	33.82	33.00	0.937	33.04	31.72	0.924
	Our GAN	28.99	28.84	0.837	28.61	28.20	0.815

compressed using JPEG with quality factor set to 20 ($6,7\times$ less bitrate). Then we benchmark object detection on reconstructed versions of the compressed images, comparing AR-CNN [29], our generative MSE and SSIM trained generators with the GAN. First of all, it must be noted that the decrease in the overall mAP measured on compressed images with respect to the upper bound is large: 14.2 points. AR-CNN, MSE and SSIM based generators are not recovering enough information yielding around 2.1, 2.4 and 2.5 points of improvements respectively. As can be observed in Table 5.3 our

Codec	Compressed	GAN
WebP	.601	.641
JPEG2000	.587	.616
BPG	.623	.644

Table 5.2: Object detection performance (mAP) of our method on other codecs on VOC2007 for similar bitrate.

GAN artifact removal restores the images in a much more effective manner yielding the best result increasing the performance by 7.4 points, just 6.8 points less than the upper bound.

Experiments on VOC2007

As a first experiment we test how our Patch Loss affects the artifact removal process. We run two object detection experiments. As a baseline we train our GAN using a Full Patch discriminator and we compare it with the discriminator described in Figure 5.4. Our GAN trained with a full patch discriminator obtains .605 mAP on VOC2007, while our sub-patch discriminator leads to .623 mAP. The Sub-Patch loss accounts for 1.8% mAP points, highlighting the importance of this novel method.

Smaller networks such as AR-CNN [29], are able to achieve reasonable, yet lower, results with respect to our approach. We therefore test a smaller GAN with 7 residual layers to see how much the depth of the network is relevant to obtain quality results. Our GAN recovers 8% mAP points while [29] only adds 3%, when dealing with object detection on compressed images. The smaller network gains 6% mAP point leading to .611, which is still better than [29] but worse than the full network, showing that, as noted for classification tasks [59, 122], network depth matters also for compression artifact removal and image restoration.

Our GAN artifact removal process recovers impressively on *cat* (+16.6), *cow* (+12.5), *dog* (+18.6) and *sheep* (+14.3), which are classes where the object is highly articulated and texture is the most informative cue. In these classes it can also be seen that MSE and SSIM generators are even deteriorating the performance, as a further confirmation that the absence of higher frequency components alters the recognition capability of an object detector. To assess the effect of color we report the use of GAN using only luminance (GAN-Y). Using l_P defined as in Eq. 5.8 is important, switching

to a simpler L1 loss (GAN-L1) we obtain much lower performance. Our GAN trained with a full patch discriminator obtains .605 mAP, while our sub-patch discriminator leads to .623 mAP, highlighting its importance.

In Fig. 5.9 we analyze the effects of different compression levels, changing the quality factor. GAN is able to recover details even for very aggressive compression rates, such as QF=10. In Fig. 5.9 it can be seen how GAN always outperform other restoration algorithms. The gap in performance is reduced when QF raises, e.g QF=40 (4, 3× less bitrate).

Finally, since there are many modern codecs available nowadays we also test our method for different codecs, which not always share artifact behavior with JPEG. In particular we considered WebP, JPEG2000 and BPG. We tuned all codecs to obtain the same average bitrate on the whole VOC2007 dataset of the respective JPEG codec using a QF of 20. Results are reported in Table 5.2, and show that our novel approach is effective also for all these compression algorithms.





















											
JPEG 20	0.587	0.692	0.516	0.434	0.350	0.673	0.71	0.559	0.334	0.559	0.579
AR-CNN [29]	0.641	0.686	0.523	0.413	0.367	0.702	0.742	0.530	0.363	0.574	0.607
MSE	0.647	0.696	0.512	0.406	0.409	0.713	0.750	0.542	0.386	0.546	0.614
Our SSIM	0.655	0.706	0.513	0.417	0.411	0.713	0.746	0.555	0.387	0.538	0.615
Our GAN-Y	0.657	0.696	0.547	0.461	0.354	0.719	0.708	0.673	0.380	0.653	0.605
Our GAN-L1	0.644	0.750	0.524	0.421	0.427	0.691	0.755	0.667	0.402	0.616	0.597
Our GAN	0.666	0.753	0.565	0.475	0.395	0.727	0.770	0.725	0.403	0.684	0.602
Original	0.698	0.788	0.692	0.559	0.488	0.769	0.798	0.858	0.487	0.762	0.637
										mAP	
JPEG 20	0.532	0.691	0.665	0.638	0.260	0.482	0.434	0.707	0.570	0.549	
AR-CNN [29]	0.581	0.724	0.661	0.658	0.313	0.499	0.526	0.712	0.578	0.570	
Our MSE	0.595	0.713	0.668	0.664	0.310	0.485	0.522	0.676	0.600	0.573	
Our SSIM	0.596	0.720	0.666	0.663	0.308	0.482	0.532	0.668	0.598	0.574	
Our GAN-Y	0.681	0.738	0.661	0.662	0.290	0.608	0.544	0.722	0.600	0.598	
Our GAN-L1	0.679	0.749	0.666	0.664	0.309	0.543	0.587	0.655	0.613	0.598	
Our GAN	0.718	0.753	0.707	0.670	0.303	0.625	0.586	0.712	0.611	0.623	
Original	0.790	0.802	0.757	0.763	0.376	0.683	0.672	0.777	0.667	0.691	

Table 5.3: Object detection performance measured as mean average precision (mAP) on PASCAL VOC2007 for different reconstruction algorithms. Bold numbers indicate best results among reconstruction approaches.

Experiments on MS-COCO

In Figure 5.10 we show how mean Average Precision (mAP) varies on the MS-COCO test set. When aggressive compression is used GAN_{L_1} and GAN_{VGG}

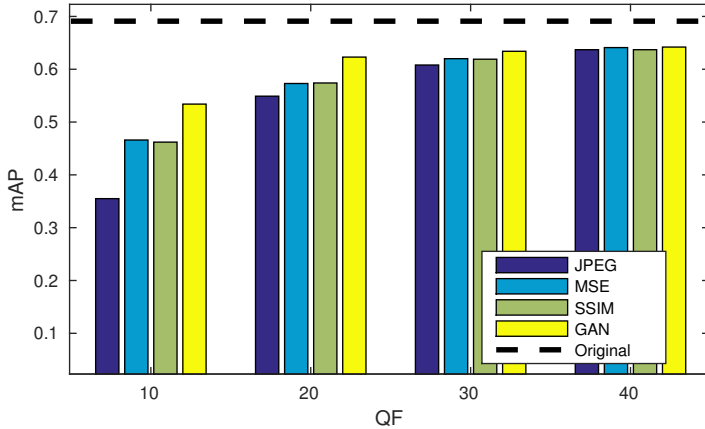


Figure 5.9: Mean average precision (mAP), for different Quality Factors (QF), and restoration approaches, on PASCAL VOC2007.

get the best results, while the simpler AR-CNN is less effective. For higher QF values we do not observe such difference, if AP is measured on all 80 classes. Interestingly, looking at classes separately we can see that for certain classes compression artifacts degrade more AP. This is shown in Tab. 5.4, where we report the 5 classes that obtain the highest and the lowest improvements in performance using GAN_{VGG} . It can be noticed that among the 5 classes that obtain the largest improvements there are several animals (e.g. *cat*, *dog*, *bear*, etc.): this is due to the reconstruction of finer details like fur obtained using the proposed GAN approach.

We measure, for each class, how much the drop in average precision depends from image corruption. In Fig. 5.11 we show, for all the analyzed QF values, a scatter plot of ΔAP_c and $\bar{\epsilon}^c$ for each class c . Where

$$\Delta AP_c = \frac{AP_c^{HQ} - AP_c^{RQ}}{AP_c^{HQ}} \quad (5.12)$$

is the relative drop in average precision when detection is performed on original images (I^{HQ}) and restored images (I^{RQ}), with a special case of JPEG, when image reconstruction is not performed at all and

$$\bar{\epsilon}^c = \frac{1}{|L|} \sum_{l \in L} \epsilon_l^c \quad (5.13)$$

is the error averaged over the set of layers L for a class c . The lower the

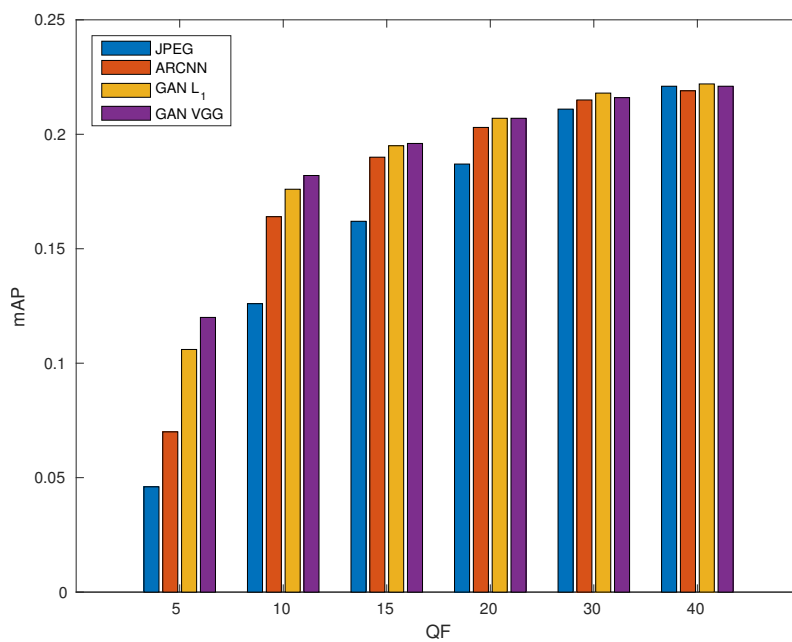


Figure 5.10: Mean Average Precision on MS-COCO varying the QF (the higher, the better). For aggressive compression rates GAN methods get the best results. For QFs higher than 30 variation is minimal.

	QF=5		QF=10		QF=15	
Highest 5 gains	pizza	0.249	cat	0.259	cat	0.203
	bear	0.215	bear	0.253	couch	0.128
	firehydrant	0.208	elephant	0.210	dog	0.116
	giraffe	0.201	dog	0.171	bear	0.113
	elephant	0.200	toilet	0.147	toilet	0.097
Lowest 5 gains	hairdrier	0.000	train	0.000	train	-0.006
	handbag	0.002	hairdrier	0.001	bus	-0.001
	toaster	0.004	toaster	0.004	hairdrier	0.000
	book	0.004	book	0.007	scissors	0.001
	spoon	0.009	handbag	0.008	carrot	0.002
	QF=20		QF=30		QF=40	
Highest 5 gains	cat	0.135	cat	0.053	tv	0.025
	couch	0.092	couch	0.036	cat	0.024
	bear	0.072	mouse	0.031	couch	0.015
	dog	0.067	toilet	0.028	mouse	0.014
	toilet	0.059	microwave	0.026	laptop	0.013
Lowest 5 gains	giraffe	-0.005	train	-0.016	train	-0.021
	keyboard	-0.004	bus	-0.014	firehydrant	-0.020
	baseballbat	-0.001	giraffe	-0.014	broccoli	-0.012
	train	-0.001	baseballbat	-0.013	elephant	-0.010
	bicycle	0.001	broccoli	-0.012	bear	-0.010

Table 5.4: Most and least affected classes in term of AP for different QF values when using GAN_{VGG} method to eliminate compression artifacts.

ΔAP_c , the better the performance of the classifier and of the reconstruction algorithm.

As shown in Fig. 5.11, there is an interesting correlation between feature map error and AP drop per class. Indeed, the error presented by feature maps, negatively affects performance in term of average precision, in case no reconstruction is applied. Interestingly when using our GAN based methods it can be seen that feature map error is still present, but with little correlation with ΔAP , even for extremely aggressive compression rates (e.g. QF=5, 10). This means that even if reconstructed images are different from original ones, their appearance, in term of semantic content understanding, is improved.

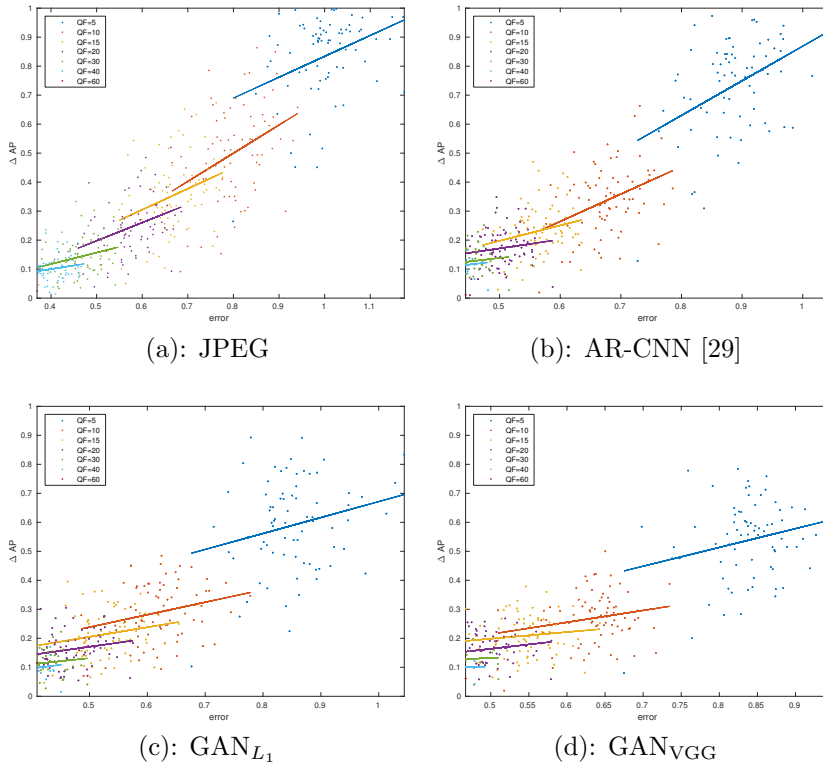


Figure 5.11: AP drop correlation with average error for different QF and methods. GAN based methods attain lowest error and AP drop.

5.4.6 Segmentation Mask Proposal

In this experiment we analyze the performance of the generation of mask proposals for an image. These proposals should precisely segment objects in a scene. Mask proposals can be used to derive bounding boxes to be fed to an object detector. Mask proposals, once evaluated by a classifier, can be used to label image pixels with categories. Differently from semantic segmentation, modern benchmarks evaluate not just the label correctness pixel-wise but also instance-wise, meaning that multiple people close-by should not be assigned a single “person” mask.

Method

Also in this experiment we use a recent method based on deep neural networks, i.e. SharpMask [107]. This approach is based on a previous method, proposed by the same authors named DeepMask [106], which learns to generate a binary mask jointly optimizing two logistic regression losses: a patch-wise object presence loss and a pixel-wise mask loss. Mask loss is inactive when an object is not present inside the patch. SharpMask proposes a refinement process able to improve 10-20% in object mask accuracy. Both methods use a pre-trained VGG-16 network to extract features.

We test SharpMask [107], with the same protocol described in Sect. 5.4.5. We measure performance in term of Average Recall for 10 proposals. This means that we average object recall over a set of intersection over union values, and report looking only at the first 10 proposals of every image (AR@10). Similarly to results reported in Sect. 5.4.5 we have GAN_{VGG} obtaining the best performance in recovering from artifacts. This behavior is consistent for all QFs. Images compressed with a QF higher than 40 exhibit little loss in AR@10.

5.4.7 Subjective evaluation

In this experiment we evaluate how images processed with the proposed methods are perceived by a viewer, comparing in particular how the SSIM loss and the GAN-based approaches preserve the details and quality of an image. We have recruited 10 viewers, a number that is considered enough for subjective image quality evaluation tests [139]; none of the viewers was familiar with image quality evaluation or the work presented in this chapter. Evaluation has been done following a DSIS (Double-Stimulus Impairment Scale) setup, created using VQone, a tool specifically designed for this type of experiments [102]: subjects evaluated the test image in comparison to the original image, and graded how similar is the test image to the original, using a continuous scale from 0 to 100, with no marked values to avoid choosing preferred numbers. We have randomly selected 50 images from the BSD500 dataset, containing different subjects, such as nature scenes, man-made objects, persons, animals, etc. For each original image both an image processed with the SSIM loss network and the GAN network have been shown, randomizing their order to avoid always showing one of the two approaches in the same order, and randomizing also the order of presentation of the tests

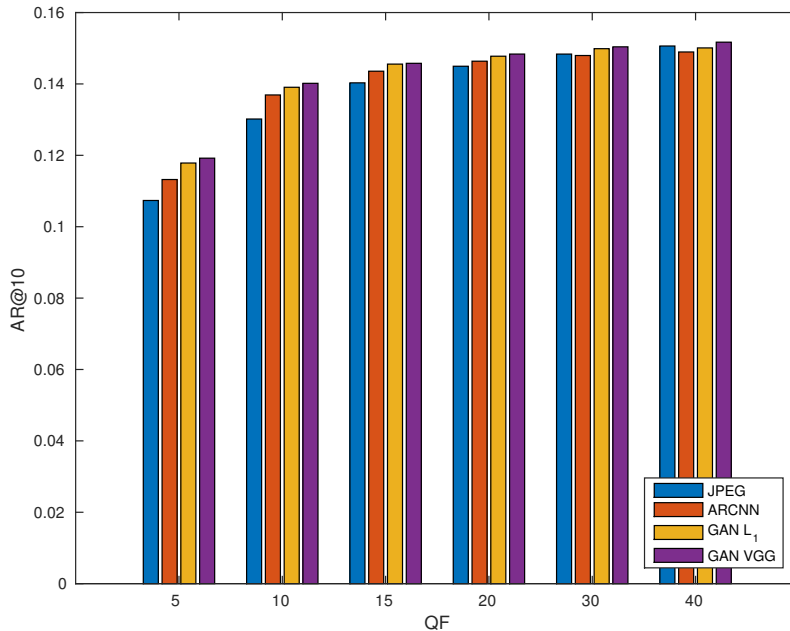


Figure 5.12: Average Recall for 10 proposals per image for different QF and methods. Performance at low QFs for GAN based methods is superior.

for each viewer. The number of 50 images has been selected to maintain the duration of each evaluation below half an hour, as suggested by ITU-R BT.500-13 recommendations [69] (typical duration was ~ 20 minutes). Overall 1,000 judgments have been collected and final results are reported in Table 5.5 as MOS (Mean Opinion Scores) with standard deviation. Results show that the GAN-based network is able to produce images that are perceived as more similar to the original image. A more detailed analysis of results is shown in Fig. 5.13, where for each image is reported its MOS with 95% confidence. It can be observed that in 90% of the cases the images restored with the GAN-based network are considered better than using the SSIM-based loss. Fig. 5.14 shows two examples, one where GAN performs better (see the texture on the elephant skin) and one of the few where SSIM performs better (see the faces).

Table 5.5: Subjective image quality evaluation in terms of Mean Opinion Score(MOS) on BSD500.

Method	MOS	std. dev.
Our SSIM	49.51	22.72
Our GAN	68.32	20.75

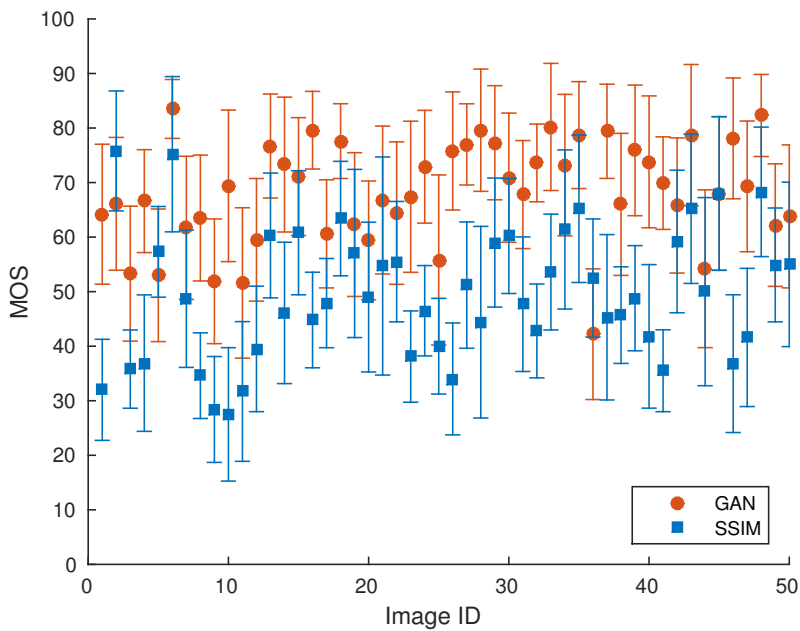


Figure 5.13: MOS values, with 0.95 confidence, for all the 50 images used in the subjective evaluation.

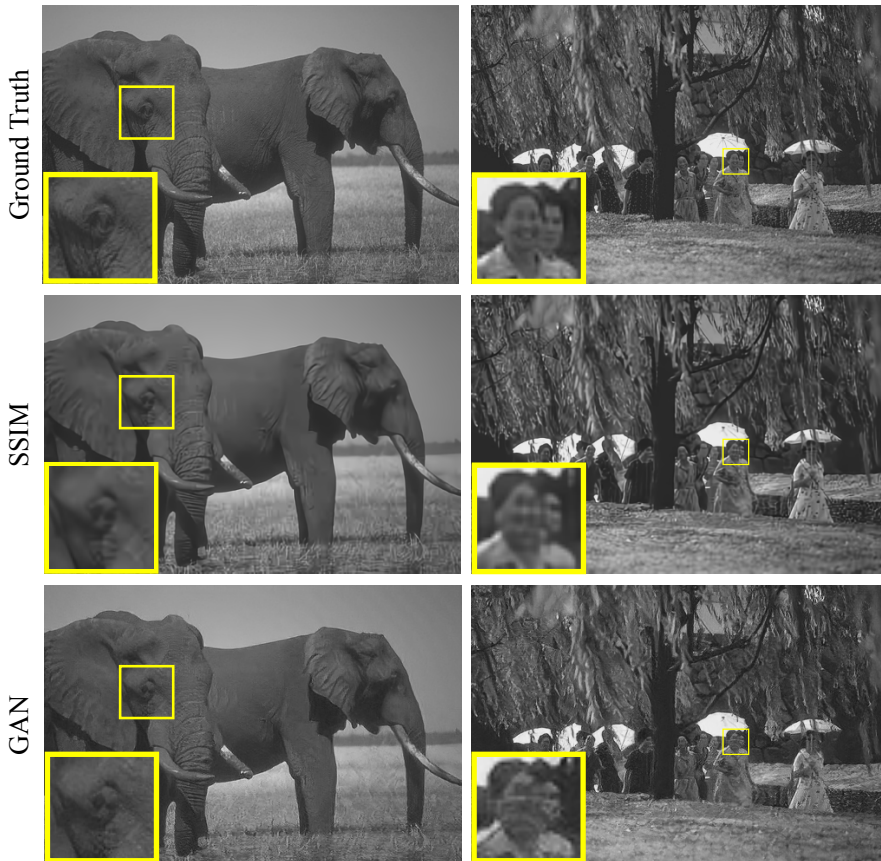


Figure 5.14: Samples of BSD500 validation set used in our subjective evaluation. Left column: best result for the GAN approach, right column: best result for the SSIM approach.

5.5 Conclusion

We have shown that it is possible to remove compression artifacts by transforming images with deep convolutional residual networks. Our baseline generative network trained using SSIM loss obtains state of the art results according to standard image similarity metrics. Nonetheless, images reconstructed as such appear blurry and missing details at higher frequencies. These details make images look less similar to the original ones for human viewers and harder to understand for object detectors. We therefore propose

a conditional Generative Adversarial framework which we train alternating full size patch generation with sub-patch discrimination.

We have extensively analyzed the behavior of deep CNN based algorithms when processing images that are compressed, evaluating results at different compression levels. As expected artifacts appearing even at moderately compression rates modify feature maps. This phenomenon is shown to correlate with errors in semantic tasks such as object detection and segmentation. We have shown a high drop in performance for classes where texture is an important cue and entities are deformable and articulated, such as cats and other animals.

Human evaluation and quantitative experiments in object detection show that our GAN generates images with finer consistent details and these details make a difference both for machines and humans.

Chapter 6

Reading Text in the Wild from Compressed Images

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.

6.1 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. Simi-

larly 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 [80, 91, 112] and are of course the state-of-the art technique for text recognition [6, 70]. However, imperceptible pixel variations are known to alter image classification results, as shown by Goodfellow *et al.* [52]. 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 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 chapter, 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



Figure 6.1: Examples of compression artifact removal. Odd columns: compressed images with compression artifacts; even columns: results of the proposed system. Best viewed in color and zoomed in.

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 6.1 we illustrate the types of compression artifacts our system is able to remove.

6.2 Related Work

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 [145, 154]. Classical text detection approaches based on connected components and sliding windows [18, 33, 66, 100, 101, 105, 147] are fairly robust techniques. However, CNN classifiers have recently led to significant improvements [60, 67, 70, 135] 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.* [70] 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 [49]. Furthermore, the authors of [50] 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) [91] have recently attracted considerable attention from the robust reading community [55, 61, 153]. 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 [151] integrated semantic labeling by FCN with MSER to provide a natural solution for handling text at arbitrary orientations. In parallel work [153] 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 [6] the authors improved the text proposal pipeline by fusing FCN outputs and the TextProposals of [50] in order to achieve higher recall with a less time consumed.

Inspired by Fully-Convolutional Networks [91] and [111], [55] propose a text localization network as an extreme variant of Hough voting. Moreover, [121] and [153] 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 [90, 110–112], 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 [87] re-purposes the SSD detector [90] for word-wise text localization. Furthermore [127] follows the idea of Region Proposal Networks [112] 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 chapter we exploit the efficient, high recall text localization pipeline from [6]. We concentrate on analyzing the effect image compression artifacts have on localization and end-to-end scene text recognition in the wild.

6.3 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.

6.3.1 Reading text in the wild

In this work we use the pipeline of [6] to generate the text proposals as a prerequisite for text recognition. Afterwards, we apply the DictNet word classifier [70] to recognize the content of text regions. The pipeline of [6] is based on a Fully Convolutional Network for text detection and the TextProposals algorithm from [50].

Fully Convolutional Networks for text detection

We trained a Fully Convolutional Network (FCN) inspired by [91] for the task of text detection by fine-tuning a VGG16 network pre-trained on ImageNet [122]. Fine-tuning was performed for 1000 iterations using Caffe [71] 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 6.2 we demonstrate the improvement of detecting text regions after reconstructing the compressed images.

The TextProposal algorithm

To generate candidate text regions we use the TextProposal algorithm of [50], 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 [50].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 [6], 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.

¹<http://github.com/lluigomez/TextProposals>

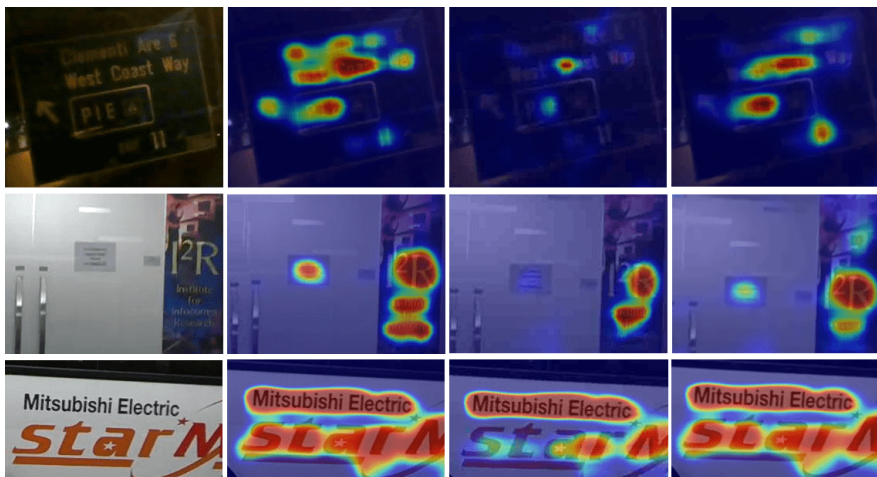


Figure 6.2: Improvement in text detection after reconstructing compressed images. In this figure we illustrate the original images and their corresponding heatmaps for the original, compressed, and reconstructed (in order, from left to right).

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 [70] to read the cropped words. The word classifier net [70] 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.

6.3.2 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 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 :

$$I^C = A(I^{HQ}, QF) \in [0, 255]^{W \times H \times C} \quad (6.1)$$

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.

Architecture

The full pipeline of the approach, both in training and testing phases is depicted in Figure 6.3 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.

Training

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

Table 6.1: 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}	$W \times H \times C$

and the network output:

$$\mathcal{L}_{MSE} = \|I_{x,y}^{HQ} - I_{x,y}^{RQ}\|_2. \quad (6.2)$$

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 [79] 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.

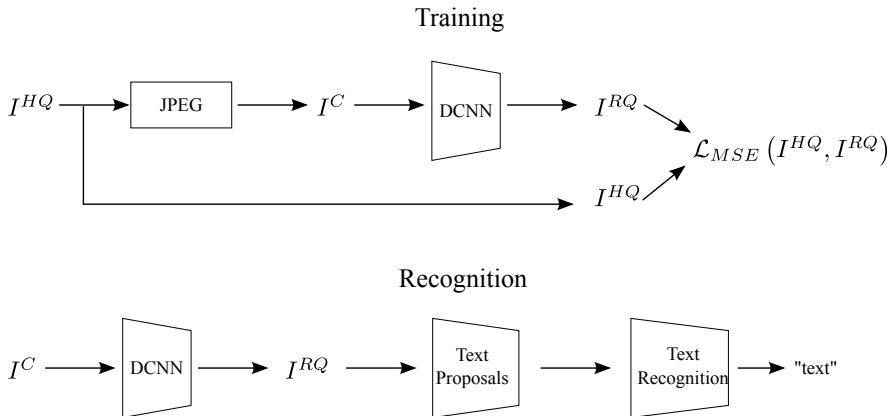


Figure 6.3: Diagram of our approach. Training is performed by minimizing MSE between reconstructed (I^{RQ}) and high quality (I^{HQ}) images. At test time we first remove artifacts from compressed images (I^C) and then apply the two step process of localization and recognition.

6.4 Experiments

We used the ICDAR-Challenge 4 [75] as the benchmark dataset in our experiments². 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.

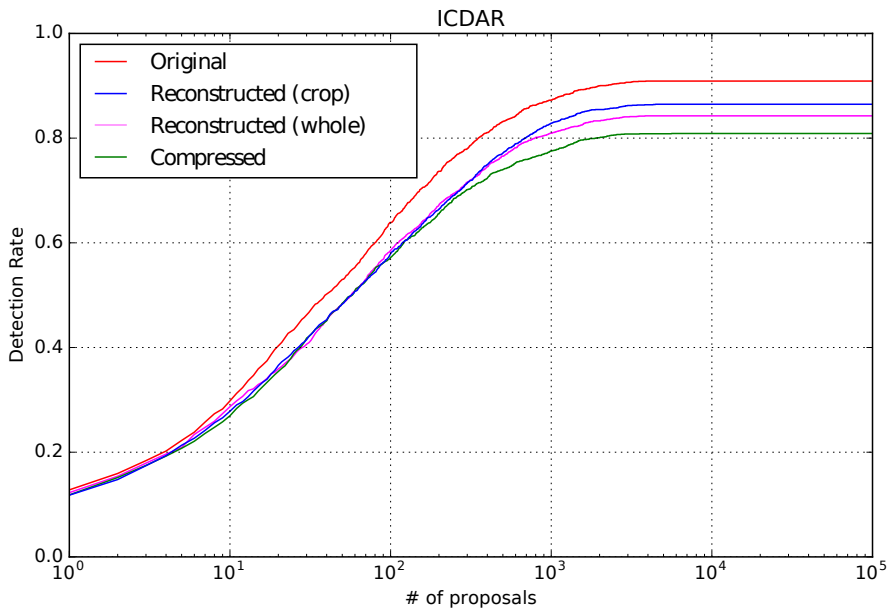


Figure 6.4: Detection rate (recall) at IoU 0.5 on the ICDAR-Challenge4 test images.

Table 6.2: 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%

6.4.1 Text localization results

In this experiment we compare the ranked list of proposals from [6] 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 6.4. 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.

6.4.2 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 [70]. The main purpose of this experiment is to explore how compressed images affect text recognition independently of localization. The results of text recognition experiment are demonstrated in Table 6.2.

²<https://www.rrc.cvc.uab.es>

Table 6.3: 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 %

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.

6.4.3 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 6.3.

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.



Figure 6.5: 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.

6.4.4 Qualitative results

In figure 6.5 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.

6.5 Conclusion and future work

In this chapter 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 those available in ICDAR-Challenge4. We expect both of these to significantly improve the impact our restoration has on text recognition.

Chapter 7

Conclusion

This thesis makes a contribution to the field of object detection. Designing strong end-to-end detectors is often a complex task, hence we have explored different solutions to overcome such difficulties. We have proposed methods and techniques for detection improvement both for objects and text in images and videos, focusing on generating better quality proposals in video sequences. Moreover, moving toward the compressed domain we have tackled the problem of detection from different points of view, obtaining promising results in both cases.

7.1 Summary of contribution

The major contributions are summarized below:

- In Chapter 3, we presented a novel method for quality improvement of window proposals in video sequences. We have studied and analyzed the interaction between object detections and proposals. We have shown that our approach is generic, every kind of proposal and detector can be plugged in for a video sequence. Our experiments report a significant improvement in window correct localization, detection accuracy and execution speed of the whole framework. Furthermore, we have proved that our approach is robust for video sequences in which frame drop is present since we process the detector output in a casual manner.
- In Chapter 4, we proposed an approach for semantic video coding by

learning saliency maps. Our novel saliency guides the codec to process a video frame depending on the importance of the region. To achieve such objective, we have developed a very fast object proposal method to select that critic parts of frames which quality has to be preserved. We have shown that, despite the relevant drop of the video sequences bit-rate, our approach not only does not affect the performance of object detectors, but we can even observe some cases of false positive reduction.

- In Chapter 5, we described a generative adversarial solution to reconstruct compressed images. We showed that our sub-patch discrimination approach is able to hallucinate high frequency details to make reconstructed images look similar to the original ones for the human eye and also objects are easier to be detected by automatic algorithms. We have also conducted an extensive analysis on the behavior of neural networks on compressed images. We observed that compression artifacts unavoidably modify feature maps, and this behavior correlates with precision errors in semantic tasks. We have proved that reconstructing degraded images with our approach leads to a significant reduction in the correlation of semantic and feature maps error.
- In Chapter 6, we have followed the previous research direction of reconstructing compressed images and we have shown that such artifacts influence meaningfully text localization and recognition tasks. We have implemented a simple residual neural network with a custom loss to restore the corrupted frames and we observed that the approach is able to improve text detector performance, in particular at very high compression rates.

Appendix A

Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.¹

International Journals

1. **L. Galteri**, L. Seidenari, M. Bertini and A. Del Bimbo, "Spatio-Temporal Closed-Loop Object Detection," in *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1253-1263, March 2017. [DOI: 10.1109/TIP.2017.2651367]

Submitted

1. **L. Galteri**, L. Seidenari, M. Bertini and A. Del Bimbo, "2017 Removing Compression Artifacts with Generative Adversarial Networks Enhances Image Semantics", *IEEE Transactions on Image Processing*, 2017. (Submitted)
2. **L. Galteri**, L. Seidenari, M. Bertini and A. Del Bimbo, "2017 Video Compression for Object Detection Algorithms", *Pattern recognition letters*, 2017. (Submitted)

International Conferences and Workshops

1. **L. Galteri**, L. Seidenari, M. Bertini and A. Del Bimbo. "Deep generative adversarial compression artifact removal", in *Proc. of International Conference on Computer Vision (ICCV)*, Venice (Italy), 2017.

¹The author's bibliometric indices are the following: H -index = 1, total number of citations = 2 (source: Google Scholar on Month October, 2017).

2. **L. Galteri**, D. Bazazian, L. Seidenari, M. Bertini, A. D. Bagdanov, A. Nicolau, D. Karatzas, A. Del Bimbo. “Reading Text in the Wild from Compressed Images”, in *Proc. of International Conference on Computer Vision Workshop (ICCVW)*, Venice (Italy), 2017.

Bibliography

- [1] “Imagenet object detection from video (VID) task dataset,” <http://image-net.org/challenges/LSVRC/2016/#vid>, 2016.
- [2] B. Alexe, T. Deselaers, and V. Ferrari, “What is an object?” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 73–80.
- [3] —, “Measuring the objectness of image windows,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 11, pp. 2189–2202, 2012.
- [4] Y. Amit and A. Trouvé, “Pop: Patchwork of parts models for object recognition,” *International Journal of Computer Vision*, vol. 75, no. 2, p. 267, 2007.
- [5] A. D. Bagdanov, M. Bertini, A. Del Bimbo, and L. Seidenari, “Adaptive video compression for video surveillance applications,” in *Proc. of ISM*, 2011.
- [6] 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.
- [7] B. Benfold and I. Reid, “Stable multi-target tracking in real-time surveillance video,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 3457–3464.
- [8] M. Bertini, A. Del Bimbo, A. Prati, and R. Cucchiara, “Semantic adaptation of sport videos with user-centred performance analysis,” *IEEE TMM*, vol. 8, no. 3, pp. 433–443, 2006.
- [9] T. Biatek, M. Raulet, J.-F. Travers, and O. Deforges, “Efficient quantization parameter estimation in hevc based on ρ -domain,” in *Proc. of IEEE EUSIPCO*, 2014.
- [10] A. Borji and L. Itti, “State-of-the-art in visual attention modeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 1, pp. 185–207, Jan 2013.

- [11] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33, no. 9, pp. 1820–1833, 2011.
- [12] B. Bross, W. Han, G. Sullivan, J. Ohm, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 9," JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Tech. Rep., Oct. 2012.
- [13] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," *CoRR*, vol. abs/1511.05666, 2015. [Online]. Available: <http://arxiv.org/abs/1511.05666>
- [14] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3241–3248.
- [15] H. Chang, M. K. Ng, and T. Zeng, "Reducing artifacts in JPEG decompression via a learned dictionary," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 718–728, Feb 2014.
- [16] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. of the British Machine Vision Conference (BMVC)*, 2014.
- [17] C. Chen, J. Cai, W. Lin, and G. Shi, "Surveillance video coding via low-rank and sparse decomposition," in *Proc. of ACM MM*, 2012. [Online]. Available: <http://doi.acm.org/10.1145/2393347.2396294>
- [18] X. Chen and A. Yuille., "Detecting and reading text in natural scenes," in *Proc. CVPR*, 2004, pp. 366–373.
- [19] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [20] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
- [21] J. Coughlan, A. Yuille, C. English, and D. Snow, "Efficient deformable template detection and localization without user initialization," *Computer Vision and Image Understanding*, vol. 78, no. 3, pp. 303–319, 2000.
- [22] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 10–17.

- [23] R. Dahl, M. Norouzi, and J. Shlens, “Pixel Recursive Super Resolution,” *ArXiv preprint arXiv:1702.00783*, Feb. 2017.
- [24] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes, “Postprocessing of compressed images via sequential denoising,” *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3044–3058, July 2016.
- [25] N. Degtyarev and O. Seredin, “Comparative testing of face detection algorithms,” in *Proc. of International Conference on Image and Signal Processing (ICISP)*, 2010, pp. 200–209. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13681-8_24
- [26] T. Deselaers, B. Alexe, and V. Ferrari, “Localizing objects while learning their appearance,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2010.
- [27] P. Dollar, R. Appel, S. Belongie, and P. Perona, “Fast feature pyramids for object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 36, no. 8, pp. 1532–1545, Aug 2014.
- [28] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 4, pp. 743–761, 2012.
- [29] C. Dong, Y. Deng, C. Change Loy, and X. Tang, “Compression artifacts reduction by a deep convolutional network,” in *Proc. of International Conference on Computer Vision (ICCV)*, 2015, pp. 576–584.
- [30] A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2016. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2016/DB16c>
- [31] I. Endres and D. Hoiem, “Category-independent object proposals with diverse ranking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 36, no. 2, pp. 222–234, Feb 2014.
- [32] —, “Category independent object proposals,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2010, pp. 575–588. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15555-0_42
- [33] B. Epshtein, E. Ofek, and Y. Wexler., “Detecting text in natural scenes with stroke width transform,” in *Proc. CVPR*, 2010, pp. 2963–2970.
- [34] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, “Scalable object detection using deep neural networks,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 2155–2162.
- [35] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes

- Challenge 2007 (VOC2007) Results,” <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>, 2008.
- [36] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.
- [37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [38] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, “Cascade object detection with deformable part models,” in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, 2010, pp. 2241–2248.
- [39] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [40] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.
- [41] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *International Journal of Computer Vision (IJCV)*, vol. 59, no. 2, pp. 167–181, 2004. [Online]. Available: <http://dx.doi.org/10.1023/B%3AVISI.0000022288.19776.77>
- [42] 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*, vol. 16, no. 5, pp. 1395–1411, 2007.
- [43] S. Gangaputra and D. Geman, “A unified stochastic model for detecting and tracking faces,” in *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*. IEEE, 2005, pp. 306–313.
- [44] L. A. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks,” *CoRR*, vol. abs/1505.07376, 2015. [Online]. Available: <http://arxiv.org/abs/1505.07376>
- [45] —, “Image style transfer using convolutional neural networks,” in *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2414–2423.
- [46] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, “Survey of pedestrian detection for advanced driver assistance systems,” *IEEE Transactions on*

- Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 7, pp. 1239–1258, 2010.
- [47] R. Girshick, “Fast R-CNN,” *arXiv preprint arXiv:1504.08083*, 2015.
- [48] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 580–587.
- [49] L. Gomez and D. Karatzas., “Object proposals for text extraction in the wild,” in *Proc. ICDAR*, 2015, pp. 206–210.
- [50] —, “Textproposals: a text-specific selective search algorithm for word spotting in the wild,” *Preprint submitted to Pattern Recognition*, 2016, arxiv:1604.02619.
- [51] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.
- [52] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. of International Conference on Learning Representations (ICLR)*, 2015.
- [53] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik, “Recognition using regions,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1030–1037.
- [54] X. Guo, S. Li, and X. Cao, “Motion matters: A novel framework for compressing surveillance videos,” in *Proc. of ACM MM*, 2013. [Online]. Available: <http://doi.acm.org/10.1145/2502081.2502145>
- [55] A. Gupta, A. Vedaldi, and A. Zisserman., “Synthetic data for text localisation in natural images,” in *Proc. CVPR*, 2016, pp. 2315–2324.
- [56] R. Gupta, M. T. Khanna, and S. Chaudhury, “Visual saliency guided video compression algorithm,” *Signal Processing: Image Communication*, vol. 28, no. 9, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0923596513000982>
- [57] H. Hadizadeh and I. Bajic, “Saliency-aware video compression,” *IEEE TIP*, vol. 23, no. 1, 2014.
- [58] S. Hare, A. Saffari, and P. Torr, “Efficient online structured output learning for keypoint-based object tracking,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [59] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

- [60] T. He, W. Huang, Y. Qiao, and J. Yao., “Text-attentional convolutional neural networks for scene text detection,” 2015, arxiv:1510.03283.
- [61] T. He, W. Huang, and J. Yao., “Accurate text localization in natural image with cascaded convolutional text network,” 2016, arxiv:1603.09423.
- [62] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, “What makes for effective detection proposals?” *arXiv preprint arXiv:1502.05082*, 2015.
- [63] J. Hosang, R. Benenson, and B. Schiele, “How good are detection proposals, really?” in *Proc. of the British Machine Vision Conference (BMVC)*, 2014.
- [64] C. Huang, H. Ai, Y. Li, and S. Lao, “High-performance rotation invariant multiview face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 671–686, 2007.
- [65] H.-J. Huang, X.-M. Zhang, and Z.-W. Xu, “Semantic video adaptation using a preprocessing method for mobile environment,” in *Proc. of IEEE CIT*, 2010.
- [66] 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*, 2013, pp. 1241–1248.
- [67] W. Huang, Y. Qiao, and X. Tang., “Robust scene text detection with convolution neural network induced msr trees,” in *Proc. ECCV*, 2014, pp. 497–511.
- [68] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE TPAMI*, vol. 20, no. 11, 1998.
- [69] *Rec. ITU-R BT.500-13 - Methodology for the subjective assessment of the quality of television pictures*, ITU, 2012.
- [70] M. Jaderberg, A. Vedaldi, and A. Zisserman., “Deep features for text spotting,” *IJCV*, vol. 116, no. 1, pp. 1–20, 2016.
- [71] 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*, 2014, pp. 675–678.
- [72] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2016, pp. 694–711.
- [73] A. Joulin, K. Tang, and L. Fei-Fei, “Efficient image and video co-localization with Frank-Wolfe algorithm,” in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 253–268.
- [74] K. Kang, W. Ouyang, H. Li, and X. Wang, “Object detection from video tubelets with convolutional neural networks,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

- [75] 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*, 2015, pp. 1156–1160.
- [76] H. Ke, H. Sun, L. Gao, and H. Wang, “A video image compression method based on visually salient features.” *JDIM*, vol. 12, no. 5, 2014.
- [77] C. Kim and J.-N. Hwang, “Fast and automatic video object segmentation and tracking for content-based applications,” *IEEE TCSVT*, vol. 12, no. 2, pp. 122–129, 2002.
- [78] J. Kim, J. Kwon Lee, and K. Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1646–1654.
- [79] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. of International Conference on Learning Representations (ICLR)*, 2015.
- [80] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2012.
- [81] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, “Unsupervised object discovery and tracking in video collections,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2015, pp. 3173–3181.
- [82] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” *CoRR*, vol. abs/1609.04802, 2016. [Online]. Available: <http://arxiv.org/abs/1609.04802>
- [83] B. Leibe, A. Leonardis, and B. Schiele, “Robust object detection with interleaved categorization and segmentation,” *International journal of computer vision*, vol. 77, no. 1-3, pp. 259–289, 2008.
- [84] S. Z. Li and Z. Zhang, “Floatboost learning and statistical face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112–1123, 2004.
- [85] Y. Li, W. Liao, J. Huang, D. He, and Z. Chen, “Saliency based perceptual HEVC,” in *Proc. of ICMEW*, 2014.
- [86] 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)*. Cham: Springer International Publishing, 2014, pp. 174–188. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10605-2_12

- [87] 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*, 2017, pp. 4161–4167.
- [88] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.
- [89] H. Liu, R. Xiong, J. Zhang, and W. Gao, “Image denoising via adaptive soft-thresholding based on non-local samples,” in *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [90] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [91] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.
- [92] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and generic corner detection based on the accelerated segment test,” in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 183–196.
- [93] S. Manen, M. Guillaumin, and L. V. Gool, “Prime object proposals with randomized Prim’s algorithm,” in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2536–2543.
- [94] 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)*, 2016, pp. 2802–2810.
- [95] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. of International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 2001, pp. 416–423.
- [96] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, “Face detection without bells and whistles,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2014, pp. 720–735.
- [97] A. Milan, S. Roth, and K. Schindler, “Continuous energy minimization for multitarget tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 36, no. 1, pp. 58–72, 2014.

- [98] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [99] J. Mutch and D. G. Lowe, “Object class recognition and localization using sparse features with limited receptive fields,” *International Journal of Computer Vision*, vol. 80, no. 1, pp. 45–57, 2008.
- [100] L. Neumann and J. Matas., “A method for text localization and recognition in real-world images,” in *Proc. ACCV*, 2010, pp. 770–783.
- [101] —, “Scene text localization and recognition with oriented stroke detection,” in *Proc. ICCV*, 2013, pp. 97–104.
- [102] M. Nuutinen, T. Virtanen, O. Rummukainen, and J. Häkkinen, “VQone MATLAB toolbox: A graphical experiment builder for image and video quality evaluations,” *Behavior Research Methods*, vol. 48, no. 1, pp. 138–150, 2016.
- [103] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and checkerboard artifacts,” *Distill*, 2016, <http://distill.pub/2016/deconv-checkerboard>.
- [104] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid, “Spatio-temporal object detection proposals,” in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 737–752.
- [105] Y. Pan, X. Hou, and C. Liu., “A hybrid approach to detect and localize texts in natural scene images,” *IEEE Trans. on Image Processing*, vol. 20, no. 3, pp. 800–813, 2011.
- [106] P. O. Pinheiro, R. Collobert, and P. Dollar, “Learning to segment object candidates,” in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 1981–1989.
- [107] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” in *Proc. of European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 75–91.
- [108] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, “Learning object class detectors from weakly annotated video,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 3282–3289.
- [109] E. Rahtu, J. Kannala, and M. Blaschko, “Learning a category independent object detection cascade,” in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 1052–1059.
- [110] J. Redmon and A. Farhadi., “Yolo9000: Better, faster, stronger. computer vision and pattern recognition,” 2016, [arxiv:1612.08242](https://arxiv.org/abs/1612.08242).
- [111] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

- [112] 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)*, 2015, pp. 91–99.
- [113] H. Rezazadegan Tavakoli, E. Rahtu, and J. Heikkilä, "Fast and efficient saliency detection using sparse sampling and kernel density estimation," in *Proc. of SCIA*, 2011. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21227-7_62
- [114] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. of European Conference on Computer Vision (ECCV)*, 2006, pp. 430–443.
- [115] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [116] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *CoRR*, vol. abs/1606.03498, 2016. [Online]. Available: <http://arxiv.org/abs/1606.03498>
- [117] B. Schauerte and R. Stiefelhagen, "Quaternion-based spectral saliency detection for eye fixation prediction," in *Proc. of European Conference on Computer Vision (ECCV)*, 2012, pp. 116–129.
- [118] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. Ieee, 2005, pp. 994–1000.
- [119] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik, "LIVE Image Quality Assessment Database Release 2," Apr. 2014.
- [120] M.-Y. Shen and C.-C. Kuo, "Review of postprocessing techniques for compression artifact removal," *Journal of Visual Communication and Image Representation*, vol. 9, no. 1, pp. 2 – 14, 1998.
- [121] B. Shi, X. Bai, and S. Belongie., "Detecting oriented text in natural images by linking segments," in *Proc. CVPR*, 2017.
- [122] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of International Conference on Learning Representations (ICLR)*, 2015.
- [123] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *null*. IEEE, 2003, p. 1470.
- [124] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE TCSVT*, vol. 22, no. 12, 2012.

- [125] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik, "Compression artifacts removal using convolutional neural networks," *arXiv preprint arXiv:1605.00366*, 2016.
- [126] K. Tang, A. Joulin, L. Li, and L. Fei-Fei, "Co-localization in real-world images," in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [127] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao., "Detecting text in natural image with connectionist text proposal network," in *Proc. ECCV*, 2016, pp. 56–72.
- [128] S. Tripathi, S. Belongie, Y. Hwang, and T. Nguyen, "Detecting temporally consistent objects in videos through object class label propagation," in *Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.
- [129] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision (IJCV)*, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s11263-013-0620-5>
- [130] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *Proc. of European Conference on Computer Vision (ECCV)*, 2012, pp. 13–26.
- [131] M. Van den Bergh, G. Roig, X. Boix, S. Manen, and L. Van Gool, "Online video SEEDS for temporal window objectness," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 377–384.
- [132] R. Verschae, J. Ruiz-del Solar, and M. Correa, "A unified learning framework for object detection and classification using nested cascades of boosted classifiers," *Machine Vision and Applications*, vol. 19, no. 2, pp. 85–103, 2008.
- [133] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of CVPR*, 2001, pp. 511–518.
- [134] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608006002152>
- [135] T. Wang, D. Wu, A. Coates, and A. Ng., "End-to-end text recognition with convolutional neural networks," in *Proc. ICPR*, 2012, pp. 3304–3308.
- [136] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?" in *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002.

- [137] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep dual-domain based fast restoration of JPEG-compressed images," in *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2764–2772.
- [138] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.
- [139] S. Winkler, "On the properties of subjective ratings in video quality experiments," in *Proc. of Quality of Multimedia Experience (QME)*, 2009.
- [140] 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*, vol. 18, no. 11, pp. 2518–2535, 2009.
- [141] B. Wu and R. Nevatia, "Cluster boosted tree classifier for multi-view, multi-pose object detection," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.
- [142] C. Xu and J. J. Corso, "Evaluation of super-voxel methods for early video processing," in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1202–1209.
- [143] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2497–2504.
- [144] 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)*, vol. 3. IEEE, 2000, pp. 869–872.
- [145] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. on PAMI*, vol. 37, no. 7, pp. 1480–1500, 2015.
- [146] C. Yim and A. C. Bovik, "Quality assessment of deblocked images," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 88–98, 2011.
- [147] X.-C. Yin, X. Yin, K. Huang, and H. Hao., "Robust text detection in natural scene images," *IEEE Trans. on PAMI*, vol. 36, no. 5, pp. 970–983, 2014.
- [148] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," Tech. rep., Microsoft Research, Tech. Rep., 2010.
- [149] 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*, vol. 25, no. 3, pp. 1246–1259, March 2016.
- [150] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, pp. 3142–3155, 2017.

-
- [151] S. Zhang, M. ZLin, T. Chen, L. Jin, and L. Lin., “Character proposal network for robust text extraction,” in *Proc. ICASSP*, 2016, pp. 2633–2637.
 - [152] 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*, vol. 22, no. 12, pp. 4613–4626, 2013.
 - [153] 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*, 4159–4167, p. 2016.
 - [154] Y. Zhu, C. Yao, and X. Bai., “Scene text detection and recognition: Recent advances and future trends,” *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.
 - [155] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2014, pp. 391–405.
 - [156] F. Zund, Y. Pritch, A. Sorkine-Hornung, S. Mangold, and T. Gross, “Content-aware compression using saliency-driven image retargeting,” in *Proc. of ICIP*, 2013.