Neural Distributed Image Compression Using Common Information

Nitish Mital[1,*], Ezgi Özyılkan[1,2,†], Ali Garjani[1,2,‡], and Deniz Gündüz[*]

[*] *Dept. of Electrical and Electronics Engineering, Imperial College London*
[†] *Dept. of Electrical and Computer Engineering, New York University*
[‡] *Section of Mathematics, EPFL*
{*n.mital, d.gunduz*}*@imperial.ac.uk, eo2135@nyu.edu, ali.garjani@epfl.ch*

Abstract

We present a novel deep neural network (DNN) architecture for compressing an image when a correlated image is available as side information only at the decoder, a special case of the well-known distributed source coding (DSC) problem in information theory. In particular, we consider a pair of stereo images, which generally have high correlation with each other due to overlapping fields of view, and assume that one image of the pair is to be compressed and transmitted, while the other image is available only at the decoder. In the proposed architecture, the encoder maps the input image to a latent space, quantizes the latent representation, and compresses it using entropy coding. The decoder is trained to extract the common information between the input image and the correlated image, using only the latter. The received latent representation and the locally generated common information are passed through a decoder network to obtain an enhanced reconstruction of the input image. The common information provides a succinct representation of the relevant information at the receiver. We train and demonstrate the effectiveness of the proposed approach on the KITTI and Cityscape datasets of stereo image pairs. Our results show that the proposed architecture is capable of exploiting the decoder-only side information, and outperforms previous work on stereo image compression with decoder side information.

1 Introduction

Data compression is a fundamental and well-studied problem in engineering, and is commonly formulated with the goal of designing codes with minimal average code length for a given data ensemble. Shannon showed that the entropy is a fundamental bound in lossless data compression when multiple independent samples of the information source can be compressed jointly while allowing arbitrarily small probability of error. The design of entropy codes approaching the Shannon limit relies on modeling the probability distribution of the data ensemble. On the other hand, continuous-valued data (such as vectors of image pixel intensities) must be first quantized to a finite set of discrete values, which introduces error. In this context, known

[1] Contributed equally to this work.

[2] At the time of this work, E. Özyılkan and A. Garjani were with the Dept. of Electrical and Electronics Engineering, Imperial College London, and Dept. of Computer Engineering, Sharif University of Technology, respectively.

This work received funding from the European Research Council (ERC) through Starting Grant BEACON (no. 677854) and from the UK EPSRC (grant no. EP/T023600/1).

2375-0359/22/$31.00 ©2022 IEEE
DOI 10.1109/DCC52660.2022.00026

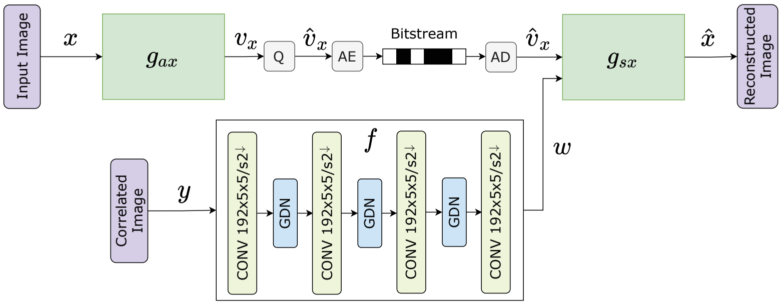


Figure 1: The proposed network architecture. Block Q corresponds to a uniform quantizer, while blocks AE and AD to arithmetic encoder and decoder, respectively.

as the lossy compression problem, one must trade-off two competing costs: the entropy of the discretized representation (rate) and the error arising from quantization (distortion). In the case of lossy compression, the fundamental performance bound is characterized by the information theoretic rate-distortion curve. In practice, lossy compression is a challenging problem, and codes are designed for specific information sources, e.g., image, audio, video. In lossy image compression, practical codes typically follow a two-step approach, where a linear transform is followed by quantization and lossless compression using entropy coding (e.g., JPEG, JPEG 2000). Recently, deep neural network (DNN) aided data-driven image compression algorithms have received significant research interest [1–7], and achieved impressive performance results, outperforming classical methods, such as JPEG 2000 and BPG [8, 9].

In this work, we are interested in DNN-aided *distributed image compression*, where side information in the form of a correlated image is available only at the decoder. This scenario occurs, for example, in the case of a pair of stereo cameras, where two cameras capture images of a scene from different angles at the same moment. In this case, the two images are highly correlated due to overlapping fields of view. The two cameras do not communicate with each other, and therefore, cannot apply joint encoding of the images as in [10]. Assume that the left camera delivers its image (in a lossless fashion) to the destination, e.g., a central storage or processing unit. The right camera, instead of employing a standard image compression algorithm, should be able to benefit from the presence of a highly correlated image from the left camera, even though it does not have access to this image.

The benefit of decoder-only side information in compression goes back to Slepian and Wolf's seminal work on distributed lossless compression [11]. They showed that a compression rate equal to the entropy of the source conditioned on the side information is necessary and sufficient for lossless compression, which is, interestingly, the same rate when the side information is available to both the encoder and the decoder. This was later extended by Wyner and Ziv to lossy compression with decoder side information in [12], which provides unbounded gains in rate. However, there is a non-zero rate loss in general compared to having the side information both at the encoder and the decoder.

Related works: Previous works [13, 14] study neural distributed source coding. In [13], the authors exploit high spatial correlations between pairs of stereo images,

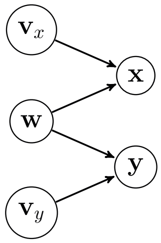


Figure 2: Graphical model.

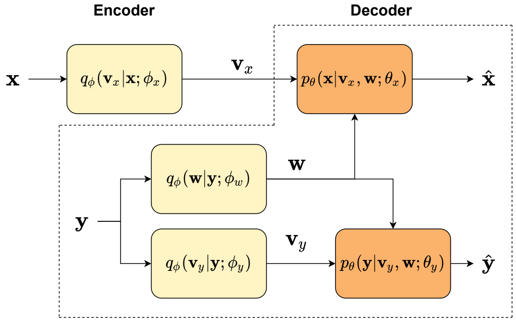

Encoder
Decoder
$\mathbf{x}$
$q_\phi(\mathbf{v}_x|\mathbf{x};\phi_x)$
$\mathbf{v}_x$
$p_\theta(\mathbf{x}|\mathbf{v}_x,\mathbf{w};\theta_x)$
$\hat{\mathbf{x}}$
$\mathbf{y}$
$q_\phi(\mathbf{w}|\mathbf{y};\phi_w)$
$\mathbf{w}$
$q_\phi(\mathbf{v}_y|\mathbf{y};\phi_y)$
$\mathbf{v}_y$
$p_\theta(\mathbf{y}|\mathbf{v}_y,\mathbf{w};\theta_y)$
$\hat{\mathbf{y}}$

Figure 3: DSC architecture.

having significantly overlapping fields of view, by finding similar patches in a intermediate reconstructed image and the side information image, and then using these to refine the reconstructed image. The paper [14] considers a different setup where the bottom half of the image of a face is used as side information for reconstructing the upper half of the face. In [15], the authors propose a framework for distributed compression of correlated sources, and joint decoding, by employing a recurrent autoencoder architecture which processes the residual content over repeated multiple iterations in order to achieve better reconstruction performance.

In this paper, we propose a novel neural architecture to perform lossy compression of an image assuming that its stereo pair is available at the decoder as side information, similarly to [13]. The novelty of the proposed method lies in employing the concept of common information to first learn the common features between the two correlated images. The common information generated by the decoder is then used along with the quantized latent representation of the original image conveyed by the encoder to reconstruct the original image. We show that our proposed method achieves a significantly better rate-distortion trade-off at low bit rates than the state-of-the-art single image compression algorithms, as well as a similar previous work on deep stereo image compression with decoder side information [13].

Wyner-Ziv compression: Let X and Y denote the source and side information variables, respectively, with joint distribution $p(x, y)$. The information theoretic fundamental limit in lossy compression is characterized by the *rate-distortion function*, given by $R^{WZ}_{X|Y}(d) = \inf I(X; V \mid Y)$, where $R^*(d)$ is the rate of compression (in bits per source sample) to achieve a given target average distortion d, between the input sequence $\mathbf{x} \in \mathbb{R}^n$ and its reconstruction $\hat{\mathbf{x}} \in \mathbb{R}^n$. The infimum is with respect to all auxiliary random variables V and reconstruction functions $f : \mathcal{Y} \times \mathcal{V} \to \hat{\mathcal{X}}$ that satisfy: i) V and Y are conditionally independent given X, that is, $V - X - Y$ form a Markov chain; ii) $\mathbb{E}\left[D(X, f(V, Y))\right] \leq d$.

2 Image compression with side information

In this section, we describe our main contribution, where we propose a novel autoencoder architecture for image compression exploiting the side information, $\mathbf{y}$, available only at the decoder, to reconstruct the image $\mathbf{x}$. In the Wyner-Ziv characterization of the rate-distortion function, the encoder identifies and transmits the latent variable V, which is then combined with the side information to reconstruct the original image. This is the approach followed by [13]. In our architecture, we model the images $\mathbf{x}$ and $\mathbf{y}$ as being generated by the random variables $\mathbf{w}$, $\mathbf{v}_x$ and $\mathbf{v}_y$, according to the graphical model illustrated in Fig. 2, which also satisfy the Markov chains $\mathbf{v}_x - \mathbf{x} - \mathbf{y}$, $\mathbf{x} - \mathbf{w} - \mathbf{y}$ and $\mathbf{x} - \mathbf{y} - \mathbf{v}_y$. The variable $\mathbf{w}$ captures the common features between the two images, while the variables $\mathbf{v}_x$ and $\mathbf{v}_y$, which we call the private information variables for the respective images, capture those aspects of $\mathbf{x}$ and $\mathbf{y}$ that are not captured by the common variable $\mathbf{w}$. We expect that feeding only the common information to the combining function at the decoder, instead of all the side information, will help the encoder to identify and send only the information that is not available at the decoder through the side information, and simplify the task of the combining function.

In general, we have $R^{WZ}_{X|Y}(d) \geq R^{WZ}_{X|W}(d)$ (see the proof in Appendix 5.2 of [16]); that is, it is more beneficial to have W as side information at the decoder, as it is 'closer' to X in distribution. Of course, we cannot in general generate W reliably based only on Y, but we argue that even an approximate reconstruction of W allows the decoder to extract only the most relevant parts of the side information that are helpful in estimating X. Our experimental results confirm that the proposed approach helps in improving the reconstructed image quality.

Architecture: See Fig. 3 for the conceptual architecture of the distributed source encoder/decoder pair that we consider, and Fig. 1 for the implemented network architecture. The distribution $q_{\boldsymbol{\phi}}(\mathbf{v}_x \mid \mathbf{x}; \boldsymbol{\phi}_x)$ is learnt by a transform $\mathbf{g}_{ax}$ at the encoder, and $q_{\boldsymbol{\phi}}(\mathbf{v}_y \mid \mathbf{y}; \boldsymbol{\phi}_y)$ is learnt by a transform $\mathbf{g}_{ay}$ at the decoder. The encoder maps the image $\mathbf{x}$ to a latent representation $\mathbf{v}_x$ by applying the transform $\mathbf{g}_{ax}$ to it. The latent representation $\mathbf{v}_x$ is quantized to $\hat{\mathbf{v}}_x \in \mathbb{Z}^m$. Since the quantization step is a non-differentiable operation, which prevents end-to-end training, it is instead replaced by additive uniform random noise over $[-0.5, 0.5]$ during training (see [1]). Thus, $\mathbf{v}_x$ is perturbed by uniform noise during training to obtain $\tilde{\mathbf{v}}_x$, which approximates the quantized latents $\hat{\mathbf{v}}_x$. The decoder extracts the common information $\mathbf{w} = \mathbf{f}(\mathbf{y}; \boldsymbol{\phi}_f)$ between the images $\mathbf{x}$ and $\mathbf{y}$ by applying the transform $\mathbf{f}$, where $\boldsymbol{\phi}_f$ refers to the weights of the DNN. The transform $\mathbf{f}$ learns the marginal distribution $q_{\boldsymbol{\phi}}(\mathbf{w} \mid \mathbf{y}; \boldsymbol{\phi}_f)$. The decoder concatenates $\mathbf{w}$ to the received latent variable $\hat{\mathbf{v}}_x$, and reconstructs an estimate $\hat{\mathbf{x}} = \mathbf{g}_{sx}(\hat{\mathbf{v}}_x, \mathbf{w}; \boldsymbol{\theta}_x)$ of the image $\mathbf{x}$ by applying a transform $\mathbf{g}_{sx}$, which corresponds to the marginal decoder $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{v}_x, \mathbf{w}; \boldsymbol{\theta}_x)$. Simultaneously, the decoder learns to reconstruct the correlated image $\mathbf{y}$ by first mapping it to the latent representation $\mathbf{v}_y$ using a transform $\mathbf{g}_{ay}$, and then reconstructing an estimate $\hat{\mathbf{y}} = \mathbf{g}_{sy}(\mathbf{v}_y, \mathbf{w}; \boldsymbol{\theta}_y)$ by concatenating the common variable $\mathbf{w}$ to $\mathbf{v}_y$ and applying a transform $\mathbf{g}_{sy}$ to it. Note that the latent representation $\mathbf{v}_y$ is neither quantized nor perturbed with uniform noise. This is because the encoding and decoding of image $\mathbf{y}$ happen at the receiver side without being transmitted over the channel.

The variables $\mathbf{w}$, $\mathbf{v}_x$ and $\mathbf{v}_y$ are modeled using a univariate non-parametric, fully factorized density function, similarly to the method proposed in [1, 2]. The joint distribution of the random variables, assuming the model illustrated in Fig. 2, is given by $p(\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{v}_x, \mathbf{v}_y) = p(\mathbf{w}) p(\mathbf{v}_x) p(\mathbf{v}_y) p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{w}, \mathbf{v}_x; \boldsymbol{\theta}_x) p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{w}, \mathbf{v}_y; \boldsymbol{\theta}_y)$, parameterized by $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$. To obtain tractable inference of latent variables, we introduce the following factored variational approximation of the posterior distribution: $q_{\boldsymbol{\phi}}(\mathbf{w}, \mathbf{v}_x, \mathbf{v}_y \mid \mathbf{x}, \mathbf{y}) = q_{\boldsymbol{\phi}}(\mathbf{v}_x \mid \mathbf{x}; \boldsymbol{\phi}_x) q_{\boldsymbol{\phi}}(\mathbf{w} \mid \mathbf{y}; \boldsymbol{\phi}_w) q_{\boldsymbol{\phi}}(\mathbf{v}_y \mid \mathbf{y}; \boldsymbol{\phi}_y)$, each of which is parameterized by a distinct DNN. The joint distribution of the latent variables can be estimated by minimizing the expectation of the Kullback-Liebler (KL) divergence between the approximate variational density $q_{\boldsymbol{\phi}}(\tilde{\mathbf{v}}_x, \mathbf{v}_y, \mathbf{w} \mid \mathbf{x}, \mathbf{y})$ and the true posterior distribution $p(\tilde{\mathbf{v}}_x, \mathbf{v}_y, \mathbf{w} \mid \mathbf{x}, \mathbf{y})$ over the data distribution $p(\mathbf{x}, \mathbf{y})$:

$$
\begin{aligned}
&D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\tilde{\mathbf{v}}_x, \mathbf{v}_y, \mathbf{w} \mid \mathbf{x}, \mathbf{y}) \mid\mid p(\tilde{\mathbf{v}}_x, \mathbf{v}_y, \mathbf{w} \mid \mathbf{x}, \mathbf{y}) \mid p(\mathbf{x}, \mathbf{y}) \right) \\
&\quad = \mathbb{E}_{\mathbf{x},\mathbf{y} \sim p(\mathbf{x},\mathbf{y})} D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\tilde{\mathbf{v}}_x, \mathbf{v}_y, \mathbf{w} \mid \mathbf{x}, \mathbf{y}) \mid\mid p(\tilde{\mathbf{v}}_x, \mathbf{v}_y, \mathbf{w} \mid \mathbf{x}, \mathbf{y}) \right) \\
&\quad = \mathbb{E}_{\mathbf{x},\mathbf{y} \sim p(\mathbf{x},\mathbf{y})} \mathbb{E}_{\tilde{\mathbf{v}}_x, \mathbf{v}_y, \mathbf{w} \sim q_{\boldsymbol{\phi}}} \Bigg[\Big(\log q_{\boldsymbol{\phi}}(\tilde{\mathbf{v}}_x \mid \mathbf{x}; \boldsymbol{\phi}_x) + \log q_{\boldsymbol{\phi}}(\mathbf{v}_y \mid \mathbf{y}; \boldsymbol{\phi}_y) + \log q_{\boldsymbol{\phi}}(\mathbf{w} \mid \mathbf{y}; \boldsymbol{\phi}_f) \Big) \qquad (1) \\
&- \Big(\underbrace{\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{w}, \tilde{\mathbf{v}}_x; \boldsymbol{\theta}_x)}_{D_x} + \underbrace{\log p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{w}, \mathbf{v}_y; \boldsymbol{\theta}_y)}_{D_y} + \underbrace{\log p(\mathbf{w})}_{R_w} + \underbrace{\log p(\tilde{\mathbf{v}}_x)}_{R_x} + \underbrace{\log p(\mathbf{v}_y)}_{R_y} \Big) \Bigg] + \text{const.},
\end{aligned}
$$

where the first term, $q_{\boldsymbol{\phi}}(\tilde{\mathbf{v}}_x \mid \mathbf{x}; \boldsymbol{\phi}_x)$, is a constant because the quantization noise has a uniform distribution with a constant width, and the second and third terms are zero because the inference model is deterministic for the common information and the latent representation of the correlated image, which implies that the associated conditional entropies are zero. The entropy terms R_w, R_x and R_y correspond to 'rate-like' terms for the common information, the quantized latent representation of the input image, and the latent representation of the correlated image, respectively. The distortion terms D_x and D_y may correspond to closed-form likelihoods, such as Gaussian distributions, in which case they can be interpreted as mean-squared error terms. For other perceptual distortion metrics, such as MS-SSIM, this is not the case in general. However, we retain the same form of the loss function, and train the DNN by minimizing the following loss function:

$$
L(\mathbf{g}_{ax}, \mathbf{g}_{sx}, \mathbf{g}_{ay}, \mathbf{g}_{sy}, \mathbf{f}) = (R_x + \lambda D_x) + \alpha \left(R_y + \lambda D_y \right) + \beta R_w, \qquad (2)
$$

where, for simplicity, we use the same weight λ for the distortion terms. Since our main objective is only to reconstruct $\mathbf{x}$, we introduce the hyperparameters α and β that determine how much importance is given to the reconstruction of the correlated image, and to the complexity of common information extracted by the decoder, respectively, which act as regularizers for the main objective. We also extend the above approach to use scale hyperpriors by using the model of [2] as the baseline.

Experimental setup: To compare the compression performance of our proposed model with the state-of-the-art, we conducted a number of experiments using the PyTorch framework. Our code is publicly available[1]. Fig. 1 illustrates the proposed DNN architecture for distributed source coding in detail. The transforms $\mathbf{g}_{ax}$ and $\mathbf{g}_{sx}$ have the same structure as those in [1]. These transforms are composed of convolutional layers, and linear (i.e., rectified linear unit) and nonlinear functions (i.e., generalized divisive normalization [GDN] and inverse generalized divisive normalization [IGDN]), which have been shown to be particularly suitable for density modelling in image compression [1]. Note that, we have omitted the section of the network that

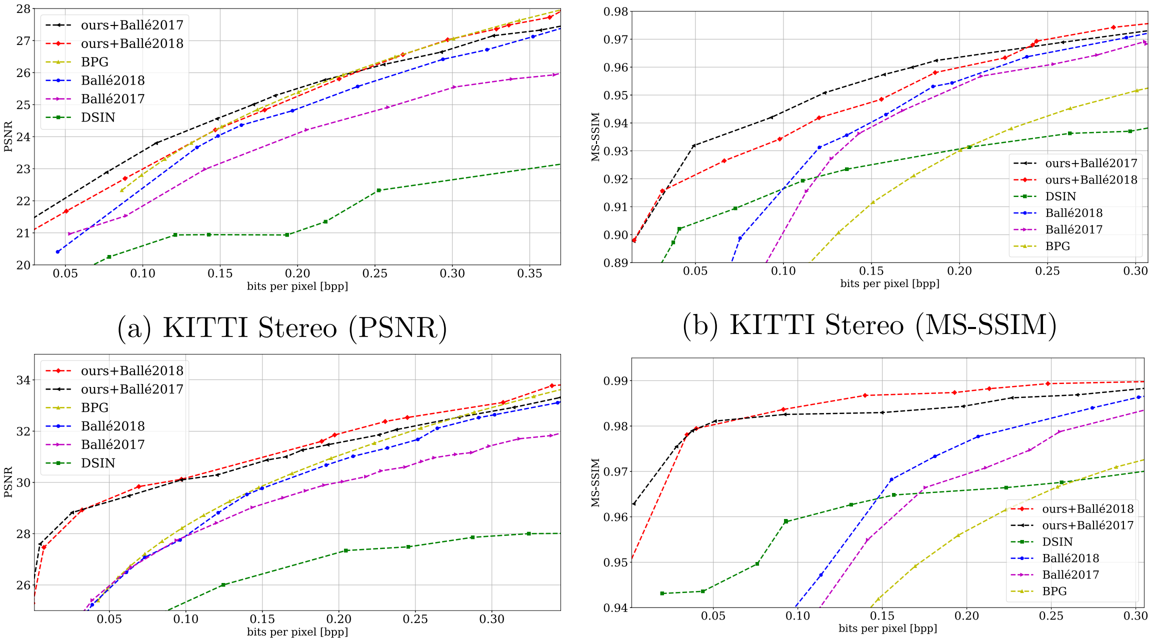
[1] Our code is available at: `https://github.com/ipc-lab/NDIC`

the decoder uses to reconstruct the correlated image during training in Fig. 1. We introduce the transform $\mathbf{f}$ as mentioned in Section 2.

For the first set of experiments, we constructed our dataset from the KITTI Stereo 2012 dataset [17] and from the KITTI Stereo 2015 dataset [18, 19], consisting of unique 1578 stereo image pairs (i.e., a pair of images taken simultaneously by different cameras). We refer to this dataset as *KITTI Stereo*. We trained every model on 1576 image pairs, and we validated and tested every model on 790 image pairs from KITTI Stereo dataset. We also trained our models on the *Cityscape* dataset [20], consisting of 5000 stereo image pairs, out of which 2975 image pairs belong to the training dataset, 500 image pairs belong to the validation dataset, and 1525 image pairs belong to the test dataset. These datasets are designed to illustrate the calibrated and synchronized camera array use case. We evaluate our models using both the multi-scale structural similarity index measure (MS-SSIM), as well as the peak SNR (PSNR) based on the mean-squared error (MSE) distortion. The MS-SSIM has been widely reported to provide a better measure of human perception of distortion [21].

Training: We center-crop each 375×1242 image of the KITTI Stereo dataset to obtain a 370×740 image, and then downsample it to a 128×256 image. For Cityscape dataset, we directly downsample each image to 128×256. We train the baseline model with different values of λ to obtain points with different bit rates using the PSNR and MS-SSIM metrics for the reconstruction loss. We train the proposed model for 500K iterations, using randomly initialized network weights. Similarly to [1], we train our models using AMSGrad optimizer [22], with a learning rate of 1×10^{-4}. The learning rate is reduced by a factor of 10 whenever the decrease in the loss function stagnates, where the lower bound on the learning rate is set to $1\cdot10^{-7}$. We use a batch size of 1 because of the small size of the datasets under consideration. For comparison, we also train the model proposed in [13] which will be referred to as DSIN. We highlight that the results for DSIN reported here differ from those in [13] because we use smaller images in our experiments.

Experimental results: In this section, we discuss the performance of the proposed model and compare it with other alternatives (see Fig. 4). In addition to DSIN, we also consider BPG as well as DNN-aided compression schemes introduced in [1] and [2], which will be referred to as Ballé2017 and Ballé2018, respectively. Following [23], we employ 4:4:4 chroma format for BPG. We note that these schemes do not exploit the side information at the receiver. “ours + Ballé2017” and “ours + Ballé2018” refer to our proposed solutions with the models in [1] and [2] as the baselines, respectively. In Figs. 4a and 4c, we present the comparison in terms of the average PSNR. We observe that the proposed model is particularly effective at low bit rates. Note that DSIN does not perform well when optimized for MSE. We observe an even more stark improvement in performance when optimized with respect to MS-SSIM in Fig. 4b and 4d, where the models that do not use side information experience a sharp drop in performance at low bit rates, while DSIN and the proposed solution, which exploit the side information, experience a more graceful degradation of performance with decreasing bit rates. This illustrates the fact that the presence of side information is particularly useful at low bit rates, where even a very limited information allows reasonable reconstruction by exploiting the side information. We also



(a) KITTI Stereo (PSNR)

(b) KITTI Stereo (MS-SSIM)

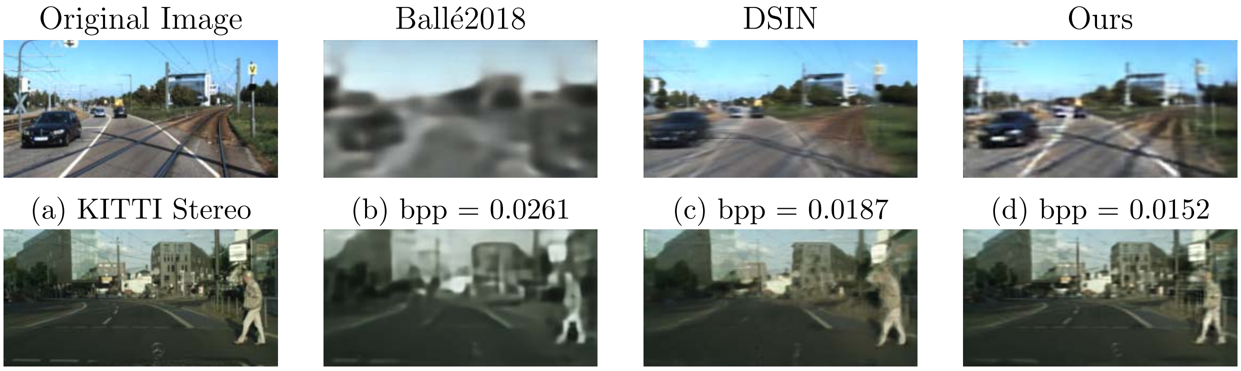
(c) Cityscape (PSNR)

(d) Cityscape (MS-SSIM)

Figure 4: Comparison of different models in terms of MSE and MS-SSIM metrics.

note that the proposed solution significantly improves the performance with respect to DSIN in experiments with both datasets.

We also present a visual comparison of the performance of the models proposed in Fig. 5. Notice that the Ballé2018 model fails to capture the colours at very low bit rates, as illustrated in Fig. 5, unlike DSIN and ours. Moreover, our model successfully captures the textures and details of colours and objects in the background, while DSIN has a blurring effect on the image. This is because the DSIN model operates by first reconstructing the image based on the compressed description provided by the encoder, then finding the offset of corresponding patches (using the "side information finder" block in their model) in the intermediate reconstructions of the input and correlated images when passed through their baseline autoencoder, and then using the corresponding patches from the original side information image to refine patches in the intermediate reconstruction of the input image. If the intermediate reconstructions are not of a high quality to begin with, it can cause the wrong patches to be recognized as the corresponding patches in the side information image. This causes distortions in the reconstructed image, leading to low MS-SSIM values. Our model operates instead by overlaying the content details sent by the encoder over the structural and texture details extracted from the side information. This is illustrated in Fig. 7. We generate the visualizations of the private and common information by passing them individually through the decoder while setting the other to zero. It is observed that the common information captures the colors and texture information, which explains why our model is able to capture them even at very low bit rates. We also note in Fig. 7 that as the bit rate is increased, the encoder is allowed to send more content


Original Image
Ballé2018
DSIN
Ours

(a) KITTI Stereo (b) bpp = 0.0261 (c) bpp = 0.0187 (d) bpp = 0.0152

(e) Cityscape (f) bpp = 0.0782 (g) bpp = 0.0460 (h) bpp = 0.0348

Figure 5: Visual comparison of different models trained for the MS-SSIM metric. “Ours” in the figures above refers to “ours + Ballé2017” model.

and structural information; and therefore, the decoder tends to extract less definite content details and more global color and texture details from the side information.

Ablation study: To study the impact of each component of the proposed model on the overall performance, we carry out an ablation study on the architecture of the decoder by varying the parameters α and β in Eq. (2), and compare the performances in Figs. 6 and 7. The default model sets $\alpha = \beta = 1$. By setting $\alpha = \beta = 0$, we remove the additional regularization terms, and we observe that this results in a slight decrease in the performance at low bit rate. Moreover, we observed an unpredictable behaviour when evaluating the DSIN model, which lacks the regularization terms R_w and $R_y + \lambda D_y$ that we have in Eq. (2). Having non-zero α and β provides the required regularization, and as a result, improves the performance. We also check whether the desired regularization can be obtained through only the rate of the common part, W, by setting $\alpha = 0$ and $\beta = 1$. However, we observe that the performance is even worse in this setting. We argue that by imposing the decoder to reconstruct its own side information under a rate and distortion penalty prevents W from being too far from Y, and hence, in a way, acts as another regularizer on the common part W. Making the value of β small, with $\alpha = 1$, is observed to provide an improvement over the default setting. This is illustrated in Fig. 7, where having $\alpha = 1$ and $\beta = 10^{-3}$ allows the decoder to generate more common information, and therefore, the encoder can

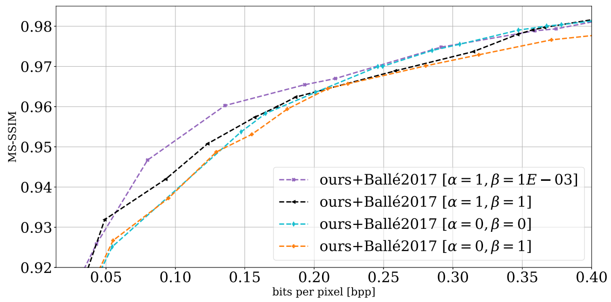
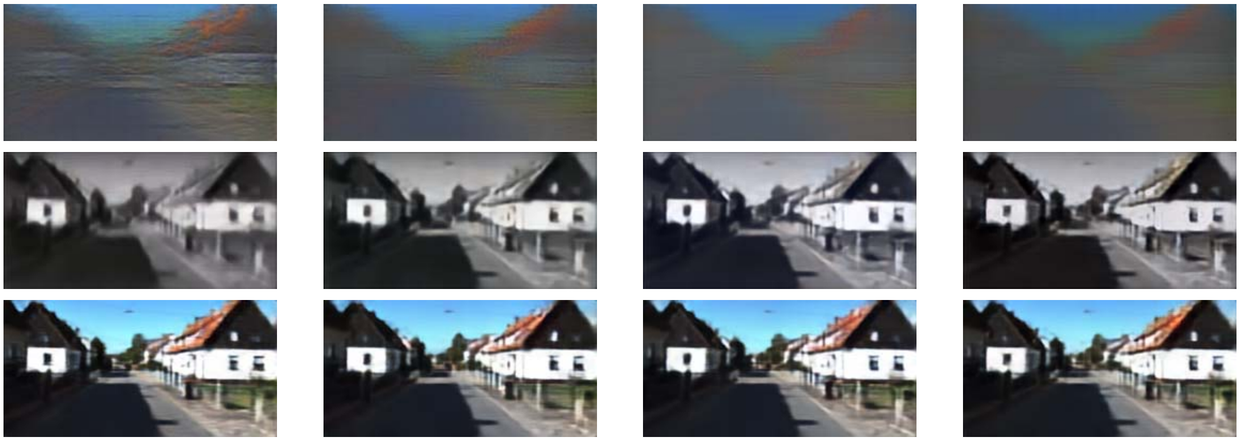


Figure 6: Ablation study experiments for the MS-SSIM metric on KITTI Stereo.



(a) $\alpha = 1, \beta = 10^{-3}$
bpp = 0.1281

(b) $\alpha = 0, \beta = 0$
bpp = 0.1572

(c) $\alpha = 1, \beta = 1$
bpp = 0.1658

(d) $\alpha = 0, \beta = 1$
bpp = 0.1731

Figure 7: **Ablation study:** Effect of hyperparameters α and β on the common information (1^{st} row) and private information (2^{nd} row) decomposition, for a similar reconstruction quality (3^{rd} row).

send a lower quality private information image at a lower bit rate to achieve the same reconstruction quality. This suggests that β can be tuned to maximize the model's performance at different bit rates.

3 Conclusions

We presented a novel autoencoder for lossy image compression with decoder side information exploiting the common information between the image to be reconstructed and the side information. The encoder learns to send only input image specific information, like the content details, to the decoder, while common information, like texture and colors, are extracted by the decoder from the side information. We show that this approach allows good quality image reconstruction even at very low bit rates, improving significantly over both single image compression models, as well as previous work on image compression with side information. The loss function in Eq. (2) also provides a framework to extend this work to a setting where there are two distributed encoders having correlated images. Moreover, combining the side information with the received latent representation in the latent space provides a general framework to extend this work to task-aware image compression.

References

[1] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations (ICLR)*, 2017.

[2] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations (ICLR)*, 2018.

[3] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, “Full resolution image compression with recurrent neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] A. Cunningham L. Theis, W. Shi and F. Huszár, “Lossy image compression with compressive autoencoders,” `https://arxiv.org/abs/1703.00395`, 2017.

[5] D. Minnen, J. Ballé, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” in *Advances in Neural Information Processing Systems*, 2018.

[6] J. Lee, S. Cho, and S.K. Beack, “Context-adaptive entropy model for end-to-end optimized image compression,” in *International Conference on Learning Representations (ICLR)*, 2019.

[7] Y. Patel, S. Appalaraju, and R. Manmatha, “Saliency driven perceptual image compression,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

[8] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The jpeg 2000 still image compression standard,” *IEEE Signal Processing Magazine*, 2001.

[9] F. Bellard, “Bpg image format,” `https://bellard.org/bpg/`, 2014.

[10] J. Liu, S. Wang, and R. Urtasun, “Dsic: Deep stereo image compression,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[11] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, 1973.

[12] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Transactions on Information Theory*, 1976.

[13] S. Ayzik and S. Avidan, “Deep image compression using decoder side information,” in *European Conference on Computer Vision (ECCV)*, 2020.

[14] J. Whang, A. Acharya, H. Kim, and A. G. Dimakis, “Neural distributed source coding,” `https://arxiv.org/abs/2106.02797`, 2021.

[15] E. Diao, J. Ding, and V. Tarokh, “Drasic: Distributed recurrent autoencoder for scalable image compression,” in *Data Compression Conference (DCC)*, 2020.

[16] N. Mital, E. Ozyilkan, A. Garjani, and D. Gunduz, “Neural distributed image compression using common information,” `https://arxiv.org/abs/2106.11723`, 2021.

[17] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[18] M. Menze, C. Heipke, and A. Geiger, “Joint 3d estimation of vehicles and scene flow,” in *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[19] M. Menze, C. Heipke, and A. Geiger, “Object scene flow,” *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[20] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] Z. Wang, E. Simoncelli, and A. Bovik, “Multiscale structural similarity for image quality assessment,” in *Asilomar Conference on Signals, Systems and Computers*, 2003.

[22] S. J. Reddi, S. Kale, and S. Kumar, “On the convergence of adam and beyond,” in *International Conference on Learning Representations (ICLR)*, 2018.

[23] O. Rippel and L. Bourdev, “Real-time adaptive image compression,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.