SPECTRALLY CONSISTENT UNET FOR HIGH FIDELITY IMAGE TRANSFORMATIONS

Demetris Marnerides
University of Warwick, UK
Demetris.Marnerides@warwick.ac.uk

Thomas Bashford-Rogers
University of the West of England, UK
Tom.Bashford-Rogers@uwe.ac.uk

Kurt Debattista
University of Warwick, UK
K.Debattista@warwick.ac.uk

ABSTRACT

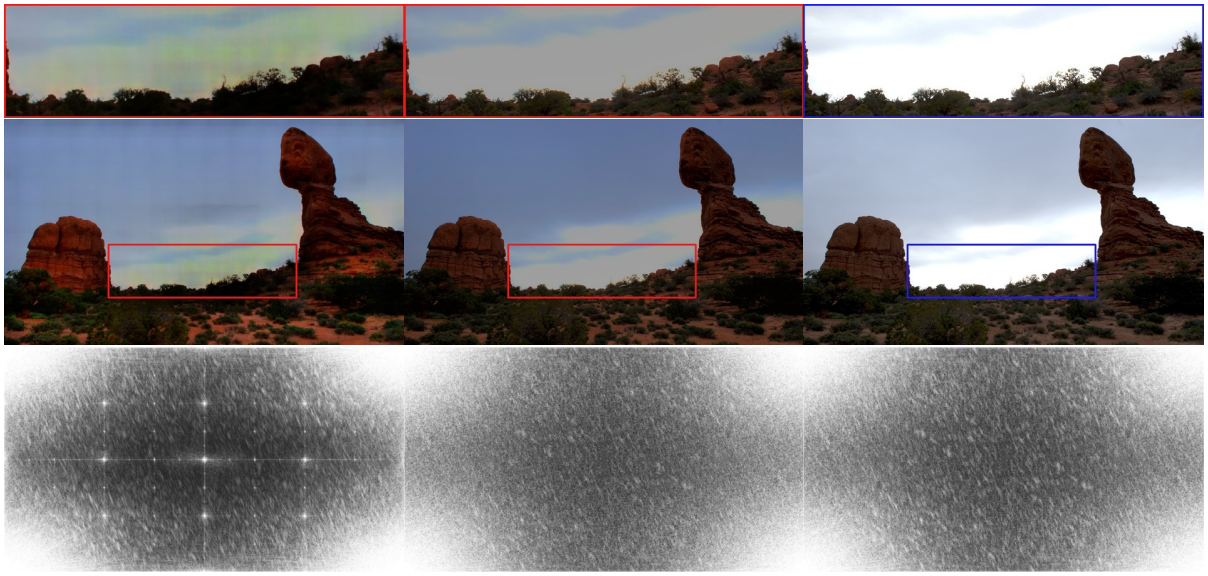
Convolutional Neural Networks (CNNs) are the current de-facto approach used for many imaging tasks due to their high learning capacity as well as their architectural qualities. The ubiquitous UNet architecture provides an efficient and multi-scale solution that combines local and global information. Despite the success of UNet architectures, the use of upsampling layers can cause checkerboard artefacts or blurring. In this work, a method for assessing the structural biases of UNets and the effects these have on the outputs is presented, characterising their impact in the Fourier domain. A new upsampling module is then proposed, based on a novel generalisation of the Guided Image Filter, that provides spectrally consistent outputs when used in a UNet architecture, forming the Guided UNet (GUNet). The GUNet architecture is evaluated quantitatively and qualitatively in an example application of dynamic range expansion for high dynamic range imaging. The proposed method provides higher fidelity results, while executing faster and consuming less memory than other dedicated architectures that avoid upsampling.

1 Introduction

Image transformation problems can be addressed using end-to-end training of Convolutional Neural Networks (CNNs). Such problems include colourisation [1], super-resolution [2] and dynamic range expansion [3, 4, 5, 6]. Solutions to these problems are required to be multi-scale, combining spatially local and global information, but must also be power and memory efficient, especially if they are to be used in real-time scenarios. The most popular CNN architecture for such problems is the ubiquitous UNet [7], which has been used extensively for image transformation problems [8, 3, 4, 5].

Despite the broadly positive results achieved using UNet architectures, it has been noted on multiple occasions [9, 10, 2, 6] that the upsampling layers can cause artefacts to appear in the output, especially the frequently used transposed convolutional layers which cause checkerboard-like patterns [11]. Figure 1(a) showcases an example of checkerboard artefacts arising from transposed convolutions for the application of inverse tone mapping [12]. These artefacts can be clearly seen in the spectral representation of the output image.

In this work, to better understand the effects that upsampling layers have on network predictions, an experiment investigating the structural properties of CNNs in the spatial frequency domain is first presented. The investigation compares the spectra of the outputs of multiple network configurations, showing the effects that the commonly used upsampling modules cause on the outputs and the pre-existing biases that UNets have architecturally. Subsequently, in order to resolve such issues, a novel module is introduced that generalises the Guided Image Filter (GIF) [13] to replace the standard upsampling and concatenation modules of the UNet architecture. The module is used within a UNet architecture to form what shall be termed a Guided UNet (GUNet).



(a) UNet (b) GUNet (c) Input

Figure 1: Comparison of outputs from a UNet architecture with transpose convolutions and the proposed spectrally consistent GUNet, for an example inverse tone mapping application. (row 1) Zoomed areas of interest. (row 2) Network outputs. (row 3) Corresponding frequency spectra highlighting the induced artefacts from UNet and a much better preserved spectrum from GUNet.

The proposed GUNet architecture attempts to alleviate the effects shown in the spectral investigation, minimising the structural biases of traditional UNets through guided feature upsampling, which preserves the spectrum of the input. In particular, the proposed architecture attempts to diminish the checkerboard artefacts and/or blurring that arises from using upsampling layers, be it transpose convolutions, nearest neighbour, or bilinear upsampling. While these artefacts are kept to a minimum, the efficiency of UNet architectures is maintained both in memory and computational speed.

In order to demonstrate the benefits of the GUNet architecture to a real-world application, the performance of the proposed architecture is evaluated for inverse tone mapping. Inverse tone mapping is the process of enabling high dynamic range (HDR) from low (sometimes called standard) dynamic range content (LDR). It has proven a popular application for deep learning solutions recently [3, 6, 4]. A number of these methods use UNets but require modifications to make them successful [3, 4, 14, 15], while others avoid using UNets because of the afore-mentioned issues and opted for dedicated architectures [6, 16]. However, ideally, a general purpose solution would be preferable to dedicated methods. The proposed GUNet, when applied to the inverse tone mapping problem, is shown to perform favourably compared to other state-of-the-art solutions while being faster and consuming less memory than the dedicated solutions.

In summary, the main contributions of this work are:

- A novel spectral analysis of the properties of UNets used in Imaging.
- A new feature upsampling method that improves the output image quality produced by UNet architectures.
- A trained GUNet architecture showcasing results for inverse tone mapping as an example application.

The following section presents an overview of related work. Section 3 is an investigation of the properties of CNNs in the Fourier domain, followed by the introduction of the Guided UNet in Section 4. Section 5 presents an application of the Guided UNet followed by some overall conclusions in Section 6.

2 Background and Related Work

Deep learning has been extensively used for image transformation problems, including image super-resolution and upsampling [17], inpainting/hallucination of missing information [18] and inverse tone mapping [3]. A variety of architectures have been presented for end-to-end image predictions, however the UNet architecture, which was first used for semantic segmentation [7], is the most commonly used.

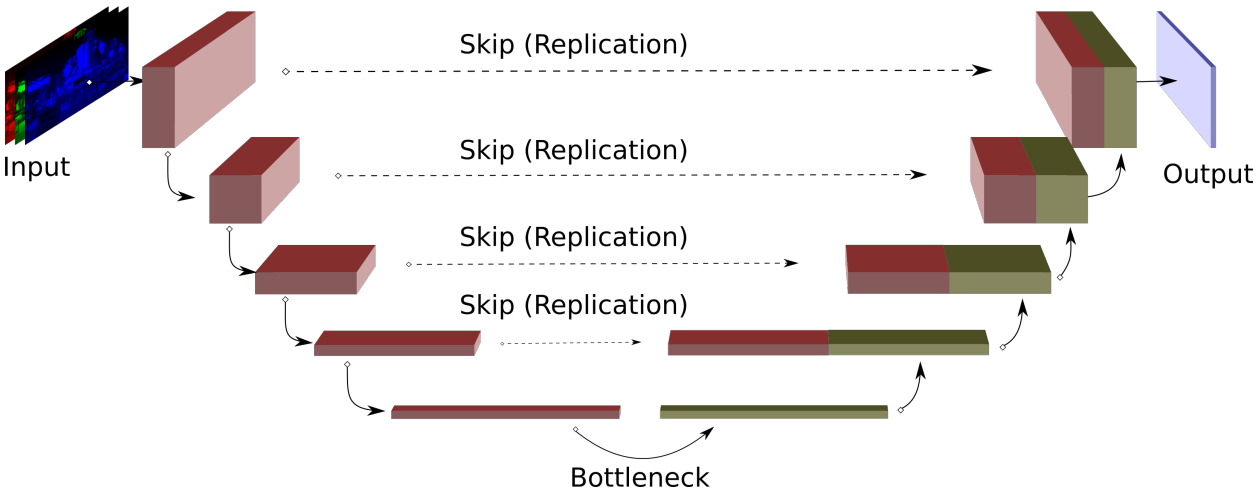


Figure 2: The UNet architecture with skip connections.

This section introduces the UNet architecture and discusses potential issues. It also presents recent development in inverse tone mapping as the sample application used to demonstrate the efficacy of the propsed GUNet method.

2.1 The UNet Architecture

The UNet architecture is based on autoencoder [19] networks and uses downsampling and subsequently upsampling layers for improved efficiency and expressiveness. The bulk of the computation is performed on lower resolutions and is thus faster and uses less memory. The encoder part of the architecture uses downsampling and convolutional layers to produce a low resolution encoding of the input, which is passed through a bottleneck and is then upsampled and processed by the decoder to produce the final output, as shown in Figure 2.

Downsampling is conducted using a variety of methods, including max-pooling, average-pooling or strided convolutions. Strided convolutions are equivalent to standard convolutions, but they instead skip pixels as they raster scan their input (standard convolutions are of stride 1) and are thus a "learnable" downsampling method. The lower resolution feature maps can be upsampled back to full resolution by the use of upsampling layers. Upsampling layers include the learnable transposed convolutions[1] [20], or more traditional upsampling algorithms like nearest neighbour or bilinear interpolation. Transposed convolutional layers are computationally equivalent to the operations performed when normal strided convolutional layers are backpropagated for the gradient computation.

Each layer of convolutions and downsampling in the architecture inevitably acts as a low pass filter, suppressing details and higher spatial frequencies present in the input[2]. For the higher frequencies to be propagated to the output, the encoder needs to learn to encode them in the intermediate features of the network.

Ronneberger et al. [7] introduced skip-connections to the autoencoder architecture to form the UNet architecture. Skip-connections concatenate the encoder with the decoder features at each level of the decoder, bypassing the lower levels of the network. This helps to better propagate details of the input to the output without having to learn to encode them, thus allowing for the lower levels to better encode global features. UNet architectures, are good at combining local and global scales since the receptive field of their lower levels is quite large and is progressively upsampled and combined with the more local scales in the decoder. Isola et al. [8] use this configuration to transform images between image domains using Generative Adversarial Networks (GANs) [21].

Despite the success of UNets, the use of upsampling layers in the decoder can cause checkerboard artefacts or blurring [11], depending on the type of layer used and the content it is applied on. Transposed convolutions are prone to checkerboard artefacts [9, 10], while non-learned upsampling methods can cause blurring, since they are based on pre-defined interpolation. In addition, upsampling layers cause information bleeding in low contrast areas, particularly ones close to sharp boundaries. Skip connections help alleviate some of these problems but are not sufficient as artefacts can be observed in multiple cases of fully trained UNet architectures.

[1] Sometimes incorrectly referred to as deconvolutional layers.

[2] This is unless the convolution filter only depends on the central pixel (i.e. all other filter values are zero), or it is of size 1×1, which does not alter the receptive field.

A number of methods have been presented that attempt to alleviate upsampling artefacts present in UNet architectures. Odena et al. [11] propose the use of "resize" convolutions, where the features are first upsampled using nearest-neighbour or bilinear upsampling and then convolution is applied. Wojna et al. [22] study variants of such configurations. Aitken et al. [23] propose a specialised initialisation scheme for transposed convolutional layers that correlates the kernel weights at initialisation. Sugawara et al. [2] propose a similar approach for use in super-resolution, which however correlates the kernel weights within the architecture and not at initialisation.

While these methods may alleviate artefacts, they do not effectively use the highly detailed information which already exists in the encoder, but rather try to recover it in the decoder. This approach can be prone to blurring, since the methods are based on pre-defined interpolation or correlation and do not provide robust results that preserve the input structure accurately. Section 3 presents a spectral investigation of the structure of CNNs and presents results on the properties of various CNN modules in the Fourier domain.

2.2 Inverse Tone Mapping

Section 5 presents results for using our approach for an example application of inverse tone mapping. Therefore, a discussion of the related work in this domain is presented for the rest of this section.

Multiple methods use UNet architecture variants for inverse tone mapping. A deep learning based method to predict HDR environment maps from captured LDR panoramas used for rendering is introduced by Zhang and Lalonde [14]. Their approach uses a UNet architecture with additive skip connections. Eilertsen et al. [3], use a UNet to predict values for saturated areas of badly exposed content for inverse tone mapping, to generate HDR images from LDR inputs. The UNet model focuses on inpainting small saturated areas, making UNet artefacts less apparent. Endo et al. [4] use a modified UNet architecture that predicts multiple exposures. These are then used to generate an HDR image using standard merging algorithms. The merging algorithms and postprocessing also potentially act like filters to improve image quality. Moriwaki et al. [15] propose a hybrid loss which combines a perceptual loss and the GAN framework to train a UNet architecture for inverse tone mapping.

Others use dedicated architectures for the same problem. Marnerides et al. [6] propose the ExpandNet architecture for inverse tone mapping which avoids upsampling altogether to alleviate artefacts. ExpandNet uses multiple branches and replication of a global feature vector, similarly to the colourisation network by Iizuka et al. [1], to fuse multiple scales together at a high resolution. Despite the good image quality that ExpandNet achieves for inverse tone mapping, improving in fidelity compared to UNet architectures and ColorNet, it is slower and uses more memory, since it mainly operates at high resolutions. More recently, Lee et al. [16] use a dedicated dilated CNN to infer multiple exposures from a single LDR image sequentially, using a chain-like structure, which are then fused.

The proposed guided filter upsampling module can be used complementary to all the UNet based methods as a drop in replacement for the upsampling modules, in order to improve the quality of the predictions.

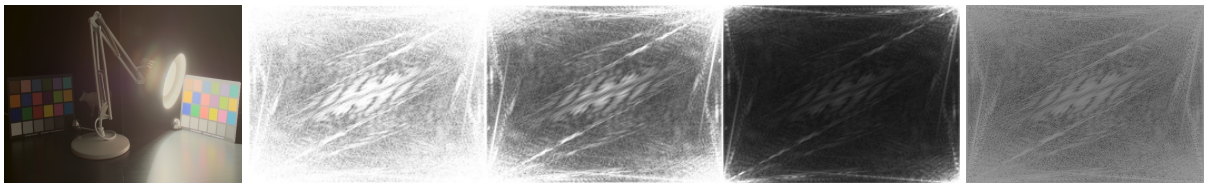
3 Spectral Properties of UNets

The frequently used UNet architecture uses downsampling and subsequently upsampling layers for improved efficiency and multi-scale operation, often causing quality issues. This section presents a method of studying the effect that upsampling layers have on UNet network predictions.

3.1 Motivation

The proposed spectral investigation method aims to identify and explain the sources of artefacts in UNet architectures. It has been suggested that the artefacts are due to the use of upsampling layers in the decoder [9, 10, 2]. The most frequent artefacts are checkerboard-like [11], arising from the use of transposed convolutional layers; see for example Figure 1(a) and Figure 9(a) in the leftmost column. These layers are equivalent to strided convolutions with fractional stride, or equivalently to strided convolutions but with their linear transformation matrices transposed.

Due to the regularly repeating nature of the checkerboard artefacts, it is hypothesised that such artefacts will consistently alter the output image spectrum due to the introduction or suppression of specific spatial frequencies. This can help identify and compare the effect of different modules but also provide a way to judge the structural properties of any alternative proposals. The following section gives an overview of the image Fourier domain and outlines the specific module configurations that are considered for the comparisons.



(a) Input (b) FFT Exposures (c) FFT TMO

Figure 3: Example of the spectrum of an input image (a) using the (fast) Fourier Transform in 2 dimensions. The spectrum can be HDR, independently of the input, so exposures are taken (b), or it is tone mapped (c) with a tone mapping operator (TMO) for better visualisation.

Table 1: Summary of the downsampling and upsampling modules used for the four UNet architectures.

Name	U1	U2	U3	U4
Encoder	Strided	Strided	Strided	Bilinear
Decoder	Transposed	Nearest	Bilinear	Bilinear

3.2 Method

The spectrum of the output of multiple networks is used as an evaluation of the structural bias of the model architecture. Images can be decomposed into a set of frequencies that fully characterise them. These frequencies are of two-dimensional discrete spatial waves of tone variation that occur over the image and can be analysed using the image spectrum computed using the discrete 2D Fourier transform. Given a greyscale input image I, its 2D, discrete Fourier transform (spectrum) S at spectral pixel location (u, v) is given by:

$$S_{u,v} = \frac{1}{\sqrt{HW}} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} I(h, w) e^{-2\pi i \left(\frac{hu}{H} + \frac{wv}{W}\right)} \tag{1}$$

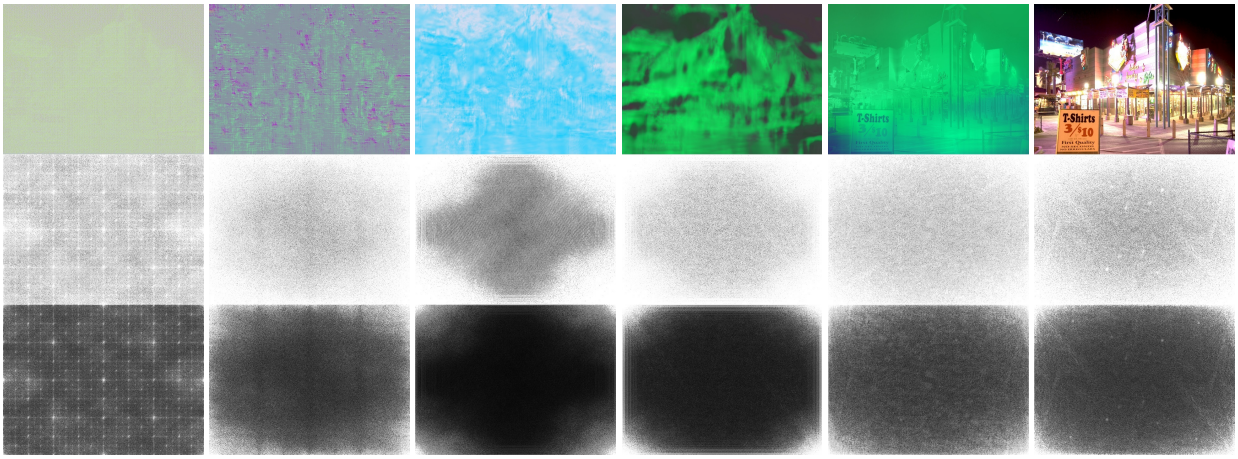
where H and W are the height and width of the image respectively.

Figure 3 shows an example of an image and its spectrum. The magnitude of the spectrum has a very high dynamic range (even for LDR images) thus it is tone mapped for displaying, either by exposure or via some other operator. Figure 3 shows multiple exposures and a tone mapped version. The rest of the figures in this section show only exposures of the spectrum magnitudes.

The spectrum is composed of complex values, consisting of a phase and magnitude. For the purposes of this work, the main focus will be on the spectrum magnitude, $|S_{u,v}|$ which is much more interpretable than the phase when it comes to describing artefacts. The three channels of coloured RGB images are calculated separately. The magnitude of the spectrum is radial from the centre, with high frequencies being more central[3]. Lower frequencies are towards the boundaries of the magnitude image. The brighter the pixel, the higher the magnitude of the corresponding frequency is in the original image. For example the regular pattern of the colour-checker boards in the input image of Figure 3 causes bright pixels to appear towards the centre of the spectrum.

Using spectral analysis, the structural properties of modules used in UNet architectures can be investigated by observing their effects on the spectrum of the output images. In particular, the effects of upsampling modules are investigated on the output image spectrum of UNet architectures. The upsampling modules under consideration are the transposed convolution, nearest neighbour and bilinear interpolation, which are the most commonly used in CNNs [11]. A total of four UNet architectures (U1 – U4) are presented and compared, initialised with random weights. Random weight initialisation is performed to isolate and outline structural biases. Three of the UNets use the three upsampling modules considered, one each, and make use of an encoder with strided convolutions. These provide the basic comparisons for the three upsampling modules under consideration. The fourth UNet uses a bilinear encoder with a bilinear decoder. This configuration aims to show the minimal difference that the encoder module choice makes. The four network configurations are summarised in Table 1.

[3]Usually high frequencies are depicted on the boundaries, but for better visualisation of the comparisons in this section they are depicted in the centre.



(a) U1 (b) U2 (c) U3 (d) U4 (e) GUNet (f) Input

Figure 4: Comparison of spectra from network outputs. The first row is the image domain, while the second and third rows are high and low exposures of the spectrum magnitude respectively.

3.3 Implementation

All the UNet networks downsample five times in the encoder and upsample five times back to the original resolution in the decoder. The encoder and decoder are composed of features with sizes 8-16-32-64-128 each. Downsampling and upsampling is always performed with a factor of two. At each level the encoder features are concatenated with the corresponding features in the decoder using a skip connection. All convolutional layers have a kernel size of 4×4. This is to avoid the overlap issues when combining stride two convolutions with 3×3 kernels as described by Odena et al. [11]. The SELU activation is used [24], and the network weights are initialised using a normal distribution.

3.4 Results

Figure 4 shows the spectra of the inputs and the corresponding outputs for the architectures presented in Section 3.2. For the purpose of comparison the figure also includes the proposed GUNet architecture introduced in the next section. The first row depicts the input images and predictions in the spatial domain while the second and third rows are high and low exposures of the corresponding magnitudes of the spectra of row 1. The samples are of a single input image, depicted in column (f), used as input to the randomly initialised networks. The output image colour and structure is distorted due to the random initialisation and the structural properties the networks which are under investigation.

Column (a) shows the outcome for the U1 configuration which uses transposed convolutional layers for upsampling in the decoder. The checkerboard artefacts in the output image are a result of the transposed convolution and are translated into regular peaks of dominant frequencies in the Fourier domain. The central pixel in the spectrum image is one of the brightest and it corresponds to the smallest checkerboard patterns (2-pixel period) which are clearly visible in the output.

Column (b) shows results for the U2 configuration, where the transposed convolutions are replaced with nearest neighbour for upsampling followed by a convolution. In this case, the output tends to be more blurry which is also shown in the output spectrum where there are patterns of higher frequencies being suppressed thus appearing darker in the spectrum.

The U3 and U4 configuration results are shown in columns (c) and (d) respectively. Both these configurations use bilinear upsampling in the decoder, but the encoder uses strided convolutions for the first while using bilinear downsampling for the second. The higher frequencies are suppressed (the overall spectrum slices are darker than the input spectrum towards the center) but there is much improvement compared to the effects of the transposed convolution and nearest neighbour upsampling. Besides the suppressed high frequencies, the spectrum has minor inconsistencies/artefacts which now appear as vertical and horizontal lines at the borders of the spectrum, due to the bilinear algorithm's specific use of interpolation.

Models that contain such modules are structurally biased towards producing artefacts. These effects can possibly be diminished by constructing training losses that direct the network weights to counteract these structural biases. This can be hard (or impossible) for pixel-wise losses, for example the L_1 or L_2 norms, since they don't take into account

inter-pixel correlations which will inform the training procedure with respect to the spectrum. Column (e) of Figure 4 shows results using a randomly initialised GUNet, which is introduced in Section 4 and further discussed in Section 5.

3.5 Discussion

The models presented above exhibit persistent artefacts due to their architectures. Transpose convolutions in the decoder introduce high frequencies and favour some over the others, while nearest neighbour upsampling suppresses high frequencies. Bilinear upsampling produces better output spectra compared to nearest neighbour which severely suppresses specific frequencies, and the choice of encoder downsampling does not alter the result by much.

All the networks considered in this section have a weight configuration space highly populated with artefact producing points, that either promote or suppress specific frequencies. This does not mean that a subset of non artefact producing points (sets of weights) does not exist, nor that such a set is not reachable after sufficient training/fine-tuning or by using a loss which specifically aims to do so. However there are no good reasons to select an artefact-biased architecture to begin with. On the contrary, a less biased architecture can lead to improved results, since it must not un-learn existing biases. The GUNet architecture introduced in the next section is specifically designed to avoid such biases while maintaining the benefits of the traditional UNet architecture.

4 Guided UNet

This section introduces the GUNet model, which replaces the upsampling and concatenation modules of the UNet architecture to improve the spectral properties of the predictions while maintaining the performance and efficiency of UNets, that perform the bulk of the computations on lower resolutions. An overview of the proposed approach is presented in Section 4.1 followed by the presentation of the guided filter fusion module in Section 4.2 followed by the model architecture details in Section 4.

4.1 Overview

In order to improve image quality and produce high fidelity outputs using UNet type architectures, the high frequency detail that is gradually lost in the encoder must be preserved and transferred to the decoder. This was the initial motivation for the introduction of the skip connections in autoencoder networks to form the UNet architecture [7]. Despite its success, the skip connections do not alleviate artefacts, which are due to upsampling as shown in Section 3. A variety of imaging techniques have been introduced that can transfer detail from one image to another, for example the joint bilateral filter [25] or the GIF [13].

Such a technique can be used in this case to transfer detail from the encoder features directly to the decoder, without using skip connections. Most importantly, these techniques can leverage higher resolution guides, to not only transfer detail but to also upsample at the same time using the high resolution image as a guide. Thus, the proposed module uses the fast GIF [26] to jointly filter and upsample the decoder features, using the corresponding features in the encoder. The GIF is preferred over bilateral filtering due to its lower computational cost. The next section describes the proposed guided feature upsampling module in detail, followed by the description of a full GUNet architecture.

4.2 Guided Feature Upsampling

The proposed network makes use of a new module based on the GIF [13]. The GIF is a differentiable edge preserving filter which can also be used for guided upsampling [26]. When used in a decoder-encoder architecture, To filter a decoder feature z using the encoder feature y as a guidance, the resulting feature q is assumed to be a locally linear model of the guidance y, similarly to the GIF. For a spatial feature neighbourhood ω_k of (square) radius r, containing $N = (2n+1)^2$ pixels:

$$q_i = \bar{a}_k y_i + \bar{b}_k, \forall i \in \omega_k \tag{2}$$

with the constants (in ω_k) $\bar{a}_k$ and $\bar{b}_k$ given approximately by using linear ridge regression by:

$$\bar{a}_k = \frac{1}{N} \sum_{k \in \omega_i} \frac{\frac{1}{N} \sum_{i \in \omega_k} y_i z_i - \mu_k \bar{z}_k}{\sigma_k^2 + \epsilon} \tag{3}$$

$$\bar{b}_k = \frac{1}{N} \sum_{k \in \omega_i} \bar{z}_k - \bar{a}_k \mu_k \tag{4}$$

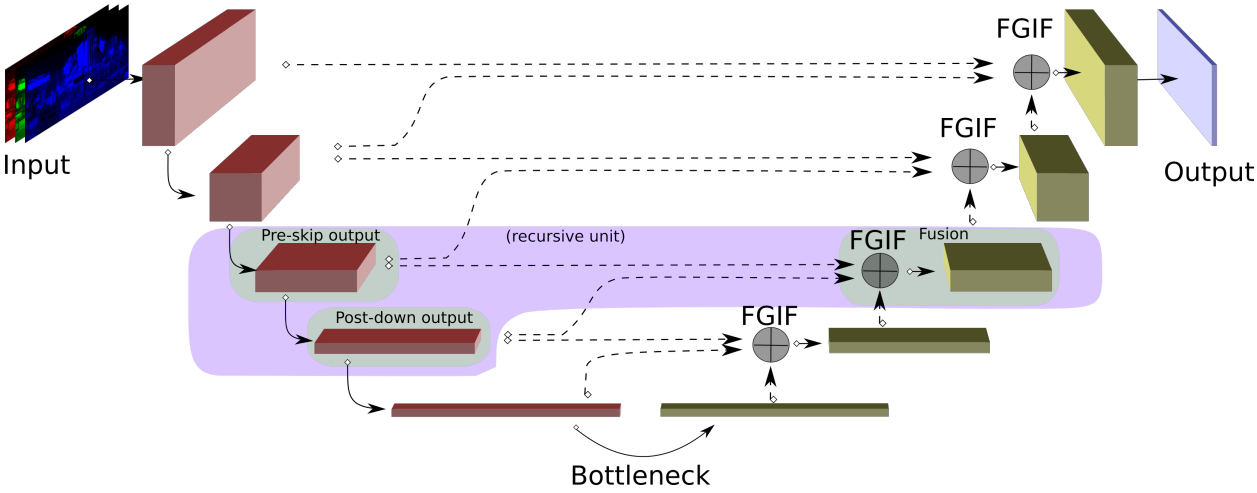


Figure 5: The Guided UNet architecture with four levels of downsampling. At each level the high resolution encoder features are used as a guide along with the low resolution version to upsample the decoder features. The purple section denotes the recursive unit that is implemented similarly on all levels. The unit is described in more detail in Figure 6

where μ_k and σ_k^2 are the mean and variance of the guidance feature in ω_k. ϵ is a regularisation parameter which penalises the effects of the guidance, by adjusting the value of $\bar{a}_k$. This approximation does not guarantee that $\bar{a}_k$ and $\bar{b}_k$ are constants in ω_k, however they still preserve large gradients (strong edges) from the encoder guidance feature:

$$\nabla q \approx \bar{a}_k \nabla y. \tag{5}$$

Similarly to the fast GIF implementation [26], which can be used for guided upsampling, y and z are the corresponding downsampled features of the architecture and are used to compute $\bar{a}_k^{\text{lr}}$ and $\bar{b}_k^{\text{lr}}$ on the lower resolution. The coefficients are then upsampled back to the higher resolution to form $\bar{a}_k^{\text{hr}}$ and $\bar{b}_k^{\text{hr}}$, using bilinear upsampling. The coefficients are then applied on the higher resolution guidance feature x to compute the final filtered decoder feature:

$$q_i = \bar{a}_k^{\text{hr}} x_i + \bar{b}_k^{\text{hr}}. \tag{6}$$

Guided feature upsampling combines the encoder and decoder features of the architecture and aims to guide its features at each upsampling stage in the output to be structurally similar to the corresponding feature set of the input features in the encoder. This is described in more detail within the context of the whole architecture in the next section.

A similar approach is followed by Wu et al. [27] where the deep guided filter is introduced and an analytic form of the derivative of the filter is derived, which is useful for implementing backpropagation for the GIF without using an automatic differentiation package, e.g. PyTorch. In the case of the deep guided filter, a network is used within a guided filter to model the mapping from the guide to the input, at a lower resolution and is trained end-to-end along with the filter from scratch. In the case of the GUNet architecture the opposite is proposed, where the guided feature upsampling module is used within an architecture, as many times as needed, to improve the decoder fidelity.

4.2.1 Motivation

As shown in Section 3, the upsampling modules in the decoder of UNet architectures cause artefacts to appear in the output image, which also distort the spectrum of the outputs, by introducing or suppressing specific frequencies. In the encoder part of the architecture, each level that downsamples from a higher resolution feature set to a lower resolution one, acts as a low pass filter, discarding high frequencies, which are the ones that need to be recovered in the decoder. Commonly used upsampling methods in the decoder are not able to fully recover these high frequencies, which however are still existent in the corresponding higher level in the encoder.

Applying the fast guided filter for upsampling the lower resolution decoder features of each level, using the higher level encoder features as the guidance "image", these frequencies can be successfully transferred to the upsampled features of the decoder. This is due to the nature of the guided filter to be edge preserving (Equation 5), hence preserving high frequency detail.

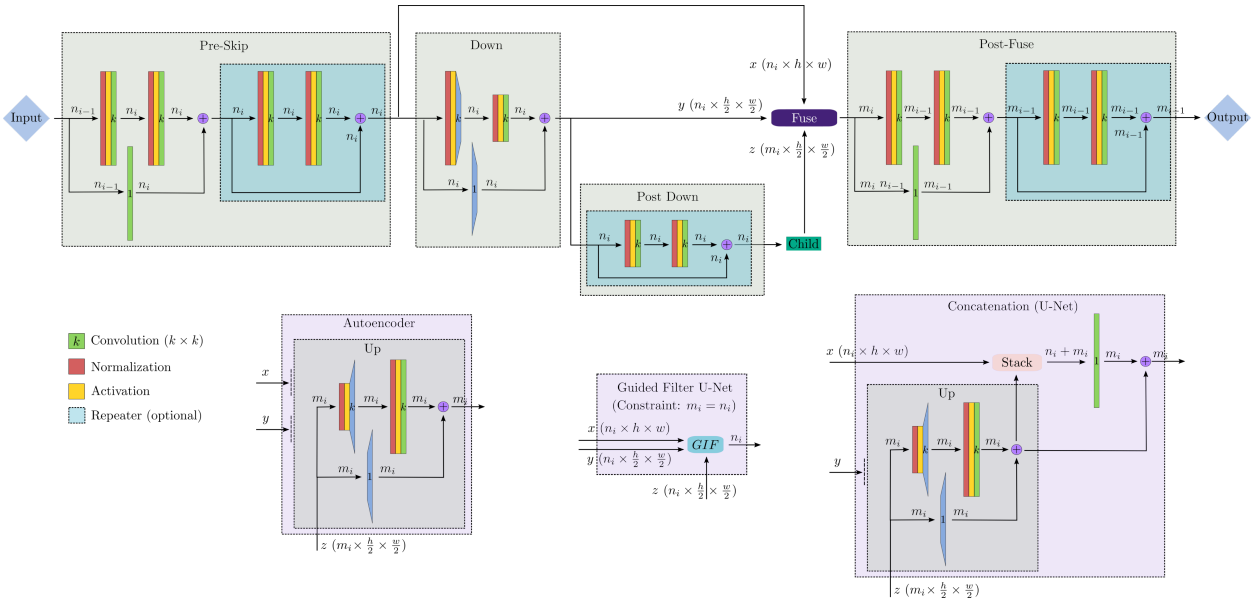


Figure 6: Slice of a single level from a generalised UNet module architecture. The fusion module (purple) blocks determine whether the network is an Autoencoder, a UNet or a GUNet. The slice is recursively used in the child (green) module. The Guided Filter module makes use of the fast guided filter to guide the feature upsampling in the decoder.

4.3 Model Architecture

Figure 5 shows a UNet architecture with a tagged slice at a particular level of the architecture, which can be thought of as a recursive element of the architecture. Figure 6 shows generalised version of the recursive element in detail. The input at that level is first pre-processed, with its feature size increased. The resulting features x are downsampled to form a lower resolution set y. y is used as the input to a child level of the same form, or the lowermost bottleneck module. The output of the child module, z can then be fused with x and y in one of the ways depicted in the purple boxes. The Pre-Skip, Down and Post-Down components are part of the encoder, while the fuse and post-fuse components are part of the decoder.

In the case of autoencoders, z is upsampled and is not combined with x and y, thus relying only on the information encoded in the bottleneck, which might be useful in some applications, for example compression. The UNet architecture performs the same upsampling, but also combines the features x with the upsampled z and fuses them using a convolutional layer, usually of kernel size 1×1. The network also uses residual connections at various points to better propagate gradients. The specific configurations with regards to the ordering of the normalisation layers, activations and convolutions are adapted from the proposed sequence described by He et al. [28].

The GUNet architecture replaces the upsampling/fusion layer with the Fast Guided Filter. In this case, the features, x, serve as the high resolution guidance image, while the lower resolution features z are the filter input. The filter is applied separately on each feature channel. The low resolution input, y, and child output z are used as the guidance and input images, in Equation 3 and Equation 4, respectively, to compute $\bar{a}_k^{\text{lr}}$ and $\bar{b}_k^{\text{lr}}$. These coefficients are then upsampled using bilinear upsampling and applied on the guidance features x using Equation 6.

5 Application

This section presents the implementation details for training a GUNet architecture for inverse tone mapping as an example application. Inverse tone mapping is a good test for the fidelity of image transformation networks due to its extreme contrast in the output and its inverse nature.

5.1 Training

The GUNet architecture that was trained trained follows the design from Figure 6. Specifically, the encoder downsamples four times, with feature sizes 16, 32, 64 and 128, matched by the decoder. A kernel size of 3×3 is used

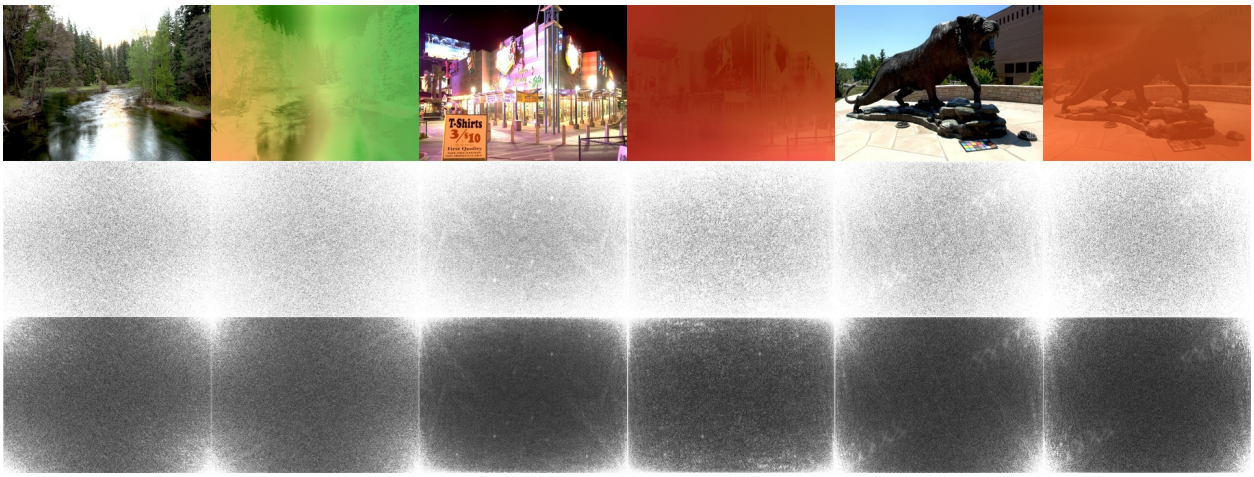


Figure 7: Spectrum for a randomly initialised GUNet. Odd columns are inputs, even columns are outputs. The outputs are randomised due to the random weight initialisation. The input structure is preserved well, even for an untrained network, something reflected by the high fidelity of the output spectrum.

except in the Pre-Skip and Post-Fuse modules which use 1×1 convolutions. There are no repeater units in any of the modules. The ReLU activation is used along with batch normalisation to help speed up training by using larger learning rates, without causing significant memory and performance issues. Residual connections are used to better propagate gradients to the bottleneck part which consists of four residual blocks of 128 features each containing two convolutional layers, exactly the same as a Post-Down module from Figure 6 with four repeater units.

The guided feature upsampling modules use $\epsilon = 0.001$. Lower epsilon values lead to stronger guidance, which is necessary for this problem, since the gradient structure needs to be preserved. The kernel size, r, is given by:

$$r = (\max(\lfloor \frac{w-1}{2} \rfloor, 1),\ \max(\lfloor \frac{h-1}{2} \rfloor, 1)) \tag{7}$$

where w and h are the spatial dimensions of the input features. This covers the entire feature at every level. It is worth noting that the size of the filter adapts to the size of the inputs at inference time to adjust for different image sizes.

The training and evaluation procedure is the same as in ExpandNet [6]. The dataset consists of 1,013 HDR images of different resolutions for training and 50 test images from the Fairchild Photographic Survey [29]. The LDR inputs are generated on-the-fly during training using four randomised tone mapping operators and exposures. The Adam optimiser is used with a learning rate of 0.0003 and a batch size of 32. The input LDR images are linearly mapped to the $[-0.5, 0.5]$ range. The loss optimised is the L1 and cosine similarity with $\lambda = 5$ as in Marnerides et al. [6]. The network is trained for approximately one week for a total of 720,000 iterations using an Nvidia 2070 SUPER GPU.

5.2 Results

This section presents a spectral comparison of GUNet architectures with the architectures considered in Section 3, followed by a presentation of quantitative and qualitative results obtained by training the GUNet architecture described in Section 5.1. Speed and memory comparisons are presented at the end of the section.

5.2.1 Spectral Comparison Without Training

An example of the spectrum of the output from an untrained GUNet is shown in Figure 7. The network is initialised with random weights, and has the same configurations as the networks in Section 3. It uses bilinear downsampling in the encoder and the guided feature upsampling module in the decoder. There are no apparent artefacts in the spectrum, which is mostly preserved as expected, even without training. Figure 4 compares all the methods presented in Section 3 and a GUNet with random weights. It is observed that GUNet best preserves the spectrum.

Table 2: Average values of the metrics for all methods and scenarios. Bold values indicate the best value.

Method	*scene-referred*							
	pu-SSIM		pu-MS-SSIM		pu-PSNR		HDR-VDP-2.2	
	opt	*cull*	*opt*	*cull*	*opt*	*cull*	*opt*	*cull*
LAN	0.72	0.72	0.78	0.64	22.21	17.15	39.01	30.47
AKY	0.72	0.72	0.78	0.64	22.70	17.08	39.11	30.75
MAS	0.75	0.72	0.80	0.63	23.29	16.87	38.98	30.59
BNT	0.70	0.74	0.73	0.66	19.56	18.91	37.63	32.03
KOV	0.75	0.75	0.80	0.68	25.31	18.60	38.71	31.92
HUO	0.74	0.75	0.78	0.64	19.71	16.27	38.04	29.95
REM	0.68	0.63	0.64	0.49	15.68	13.55	33.61	27.34
COL	0.58	0.63	0.69	0.69	23.21	22.08	31.23	29.74
UNT	0.68	0.77	0.71	0.70	20.52	19.66	34.88	34.65
EIL	0.72	0.52	0.78	0.53	22.90	17.92	39.06	28.14
EXP	0.74	0.81	0.79	**0.79**	**25.54**	**22.58**	39.27	**35.04**
GUN	**0.84**	**0.84**	**0.84**	**0.79**	23.54	22.17	**40.61**	33.41
	display-referred							
LAN	0.76	0.31	0.80	0.17	19.89	9.12	41.01	18.01
AKY	0.76	0.74	0.80	0.66	20.37	15.00	40.89	31.39
MAS	0.79	0.73	0.82	0.64	21.03	14.77	40.83	31.11
BNT	0.74	0.36	0.75	0.27	17.22	9.61	39.99	24.51
KOV	0.80	0.77	0.83	0.69	23.24	16.54	40.27	31.78
HUO	0.77	0.74	0.77	0.64	17.83	14.85	38.58	30.57
REM	0.66	0.59	0.59	0.46	14.60	12.81	33.74	27.96
COL	0.63	0.66	0.71	0.70	21.00	**19.99**	31.41	30.26
UNT	0.72	0.78	0.73	0.69	18.23	17.02	35.68	35.27
EIL	0.77	0.54	0.80	0.55	20.66	15.96	41.01	27.58
EXP	0.79	0.83	0.82	**0.79**	**23.43**	19.93	40.81	**36.21**
GUN	**0.85**	**0.84**	**0.84**	0.78	21.16	19.57	**41.93**	34.51

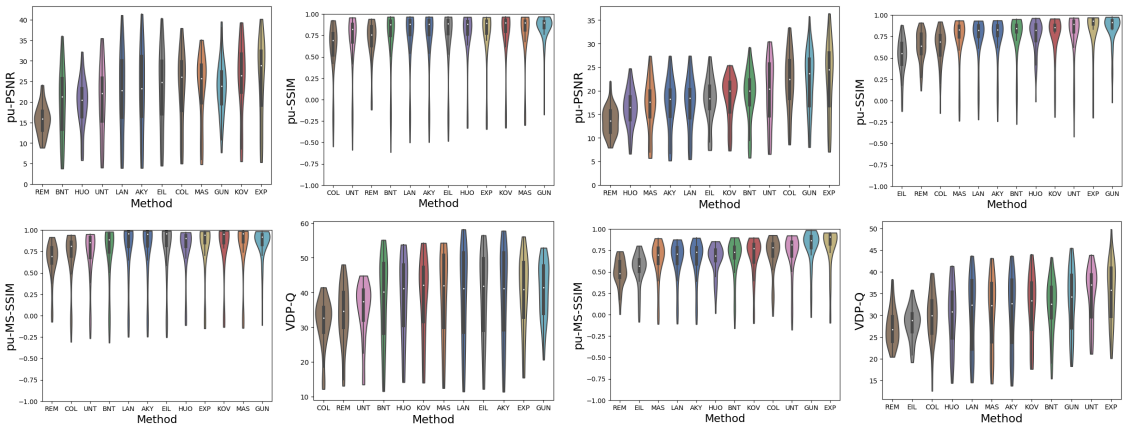
5.2.2 Quantitative

Quantitative comparisons between the trained GUNet (GUN) described in Section 5.1 and the inverse tone mapping methods by Landis [30] (LAN), Banterle et al. [12] (BNT), Akyüz et al. [31] (AKY), Rempel et al. [32] (REM), Masia et al. [33] (MAS), Kovaleski and Oliveira [34] (KOV) Huo et al. [35] (HUO), Eilertsen et al. [3] (EIL) and Marnerides et al. [6] (EXP) are presented. The metrics used are following the evaluation method of ExpandNet [6], which uses PU-PSNR, PU-SSIM, PU-MS-SSIM and HDR-VDP-2 for both *optimal* and *culling* (clipping of top and bottom 10% of values) exposures, and the *scene-referred* (scaling to original HDR image range) and *display-referred* (scaling to 1000 nits display range) settings. Table 2 shows the average test performance for all metrics and all scenarios, while Figure 8 shows the corresponding violin plots with the distributions.

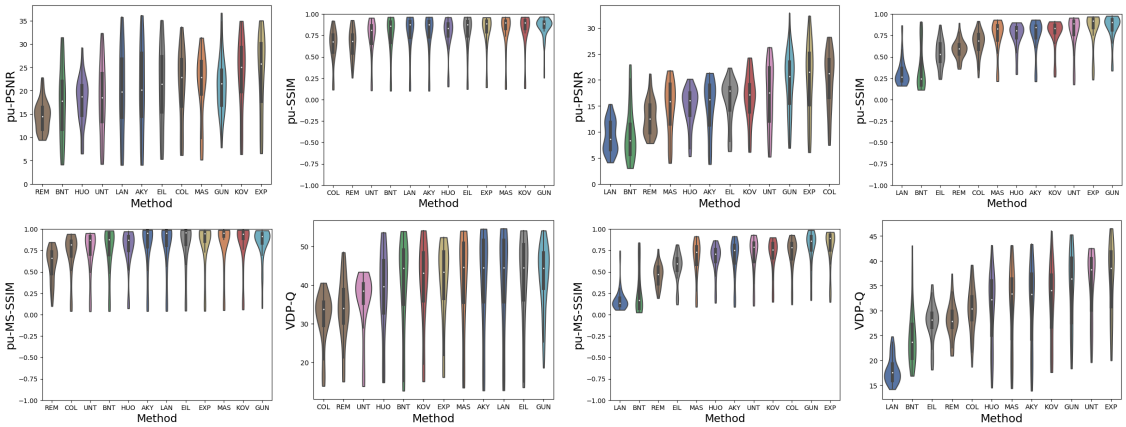
GUNet performs well, achieving the highest values in most cases of the optimal exposure setting. This is in line with its design, which relies on existing information regarding the structure and edges of the images to guide the inverse tone mapping in the result. Thus it also performs better on metrics which measure structural quality (SSIM, pu-SSIM and HDR-VDP-2) rather than just PSNR which is pixel-wise. In the culling exposure setting GUNet is on average the second best choice compared to ExpandNet.

5.2.3 Qualitative

To better showcase the importance of the architecture and the guidance, example images are presented for predictions from ExpandNet and GUNet in Figure 1 and Figure 9. The predicted HDR images produced using ExpandNet exhibit artefacts that are not completely removed, even though they are significantly reduced compared to other CNNs [6]. These artefacts likely correspond to the additional high frequencies observed in ExpandNet predictions in the fifth column of Figure 4. The predictions from GUNet are much smoother and handle high contrast areas and edges with greater fidelity.



(a) *scene-referred / optimal* (b) *scene-referred / culling*



(c) *display-referred / optimal* (d) *display-referred / culling*

Figure 8: Violin plots for *display-referred* HDR obtained from LDR via *culling* including GUNet.

5.2.4 Speed and Memory

The GUNet architecture is substantially faster than the ExpandNet architecture and scales better with resolution as can be observed in Figure 10. ExpandNet runs at an average of 172ms (5.81 frames per second) on an Nvidia P100 GPU for a Full HD (1920×1080) input and uses 4.1GB of memory. GUNet runs at an average of 82ms (12.2 frames per second), a more than double increase in speed, and uses 0.86GB of memory, an almost 80% decrease. ExpandNet could not be ran at 4K resolution (3840×2160) since it required more than the 16GB of memory of the GPU. It is worth noting that in this configuration, GUNet has 1.67 million trainable parameters, about three times more than ExpandNet's 0.457 million which means it can be more expressive and possibly handle a larger variety of scenarios using less memory and at a lower computational cost.

6 Conclusion

Investigation of the structural integrity of CNNs in the Fourier domain can help assess the induced properties of CNN outputs. The use of the guided feature upsampling modules improves the prediction quality of UNet-like architectures by preserving the frequency spectrum. The proposed GUNet architecture can perform as well as other CNN architectures quantitatively and at the same time preserves the qualities of the UNet architecture, i.e. speed, memory, expressiveness and multi-scale structure. Even though the approach in this work preserves the structural integrity of images for end-to-end image transformations, GUNet is no silver bullet, and might not be the best choice for imaging problems that do not require guidance from the encoder features, for example style transfer, where the spectral structure of the input must be altered.

(a) UNet - GUNet - Input

(b) ExpandNet - GUNet - Input

Figure 9: Comparison of inverse tone mapping predictions from UNet, Expandnet, the proposed GUNet model and the input LDR image. (row 1) Exposures showcasing the whole dynamic range of the predictions. (row 2–3) Tone mapped outputs with zoomed areas of interest. (row 4) Corresponding frequency spectra. On the right (b) is an example where a dedicated non-UNet based architecture, that otherwise works well, produces some artefacts, which are more pronounced in the spectrum.

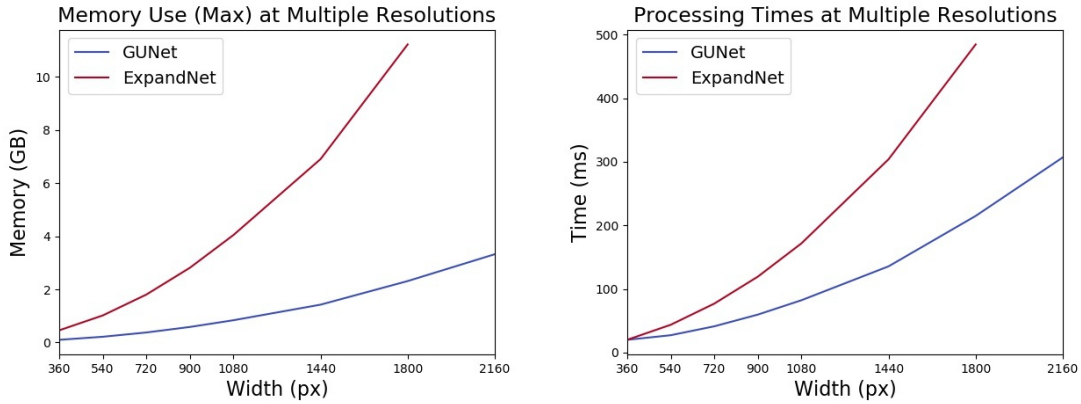


Figure 10: Performance comparison for GUNet and ExpandNet for memory consumption and prediction time at varying resolutions (16:9 ratio).

References

[1] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (TOG)*, 35(4):110:1—-110:11, 2016.

[2] Yusuke Sugawara, Sayaka Shiota, and Hitoshi Kiya. Super-Resolution using Convolutional Neural Networks without Any Checkerboard Artifacts. *IEEE International Conference on Image Processing (ICIP)*, pages 66—-70, jun 2018.

[3] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K Mantiuk, and Jonas Unger. HDR image reconstruction from a single exposure using deep CNNs. *ACM Transactions on Graphics (TOG)*, 36(6):178, 2017.

[4] Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. Deep reverse tone mapping. *ACM Transactions on Graphics (TOG)*, 36(6):1–10, 2017.

[5] Kyong Hwan Jin, Michael T. McCann, Emmanuel Froustey, and Michael Unser. Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, sep 2017.

[6] Demetris Marnerides, Thomas Bashford-Rogers, Jonathan Hatchett, and Kurt Debattista. ExpandNet: A Deep Convolutional Neural Network for High Dynamic Range Expansion from Low Dynamic Range Content. *Computer Graphics Forum*, 37(2):37–49, may 2018.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, abs/1505.0, 2015.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv*, page 16, 2016.

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation with Conditional Adversarial Nets. Supplementary Material, 2016.

[10] Jinsong Zhang and Jean-François Lalonde. Learning High Dynamic Range from Outdoor Panoramas. Supplementary Material, 2017.

[11] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and Checkerboard Artifacts. *Distill*, 2016.

[12] Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers. Inverse tone mapping. *Proceedings of GRAPHITE '06*, page 349, 2006.

[13] Kaiming He, Jian Sun, and Xiaoou Tang. Guided Image Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, jun 2013.

[14] J Zhang and J Lalonde. Learning High Dynamic Range from Outdoor Panoramas. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4529–4538, oct 2017.

[15] Kenta Moriwaki, Ryota Yoshihashi, Rei Kawakami, Shaodi You, and Takeshi Naemura. Hybrid Loss for Learning Single-Image-based HDR Reconstruction. *arXiv preprint arXiv:1812.07134*, 2018.

[16] S Lee, G H An, and S Kang. Deep Chain HDRI: Reconstructing a High Dynamic Range Image from a Single Low Dynamic Range Image. *IEEE Access*, 6:49913–49924, 2018.

[17] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(2):295–307, 2016.

[18] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (TOG)*, 36(4):107:1—-107:14, jul 2017.

[19] Jürgen Schmidhuber. Deep Learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:3431–3440, 2015.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[22] Zbigniew Wojna, Vittorio Ferrari, Sergio Guadarrama, Nathan Silberman, Liang-Chieh Chen, Alireza Fathi, and Jasper Uijlings. The devil is in the decoder. *arXiv preprint arXiv:1707.05847*, 2017.

[23] Andrew Aitken, Christian Ledig, Lucas Theis, Jose Caballero, Zehan Wang, and Wenzhe Shi. Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. jul 2017.

[24] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-Normalizing Neural Networks. *CoRR*, abs/1706.0, 2017.

[25] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (ToG)*, volume 26, page 96. ACM, 2007.

[26] Kaiming He and Jian Sun. Fast Guided Filter. may 2015.

[27] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Fast End-to-End Trainable Guided Filter. mar 2018.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. Technical report, 2016.

[29] Mark D Fairchild. The HDR photographic survey. *ingentaconnect.com*, 2007.

[30] Hayden Landis. Production-ready global illumination. *spherevfx.com*, 2002.

[31] Ahmet Oğuz Akyüz, Roland Fleming, Bernhard E Riecke, Erik Reinhard, and Heinrich H Bülthoff. Do HDR displays support LDR content?: a psychophysical evaluation. *ACM Transactions on Graphics (TOG)*, 26(3):38, 2007.

[32] Allan G Rempel, Matthew Trentacoste, Helge Seetzen, H David Young, Wolfgang Heidrich, Lorne Whitehead, and Greg Ward. LDR2HDR: On-the-Fly Reverse Tone Mapping of Legacy Video and Photographs. *ACM Transactions on Graphics (TOG)*, 26(3):39, 2007.

[33] Belen Masia, Sandra Agustin, Roland W Fleming, Olga Sorkine, and Diego Gutierrez. Evaluation of Reverse Tone Mapping Through Varying Exposure Conditions. *ACM Transactions on Graphics (TOG)*, 28(5):1–8, 2009.

[34] Rafael Pacheco Kovaleski and Manuel M Oliveira. High-Quality Reverse Tone Mapping for a Wide Range of Exposures. In *27th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 49–56. IEEE Computer Society, aug 2014.

[35] Yongqing Huo, Fan Yang, Le Dong, and Vincent Brost. Physiological inverse tone mapping based on retina response. *The Visual Computer*, 30:507–517, may 2014.