Blind Visual Motif Removal from a Single Image

Amir Hertz   Sharon Fogel   Rana Hanocka   Raja Giryes   Daniel Cohen-Or

Tel Aviv University

Figure 1: Blind visual motif removal results on images unseen during training. Top: test images embedded with semi-transparent motifs. Bottom: our reconstructed results. Our network was trained on Latin characters, yet successfully identifies and removes the Hindi and Japanese characters (left three images). Similarly, the overlaid visual motifs on the right three images differ semantically from the motifs used during training.

Abstract

Many images shared over the web include overlaid objects, or visual motifs, *such as text, symbols or drawings, which add a description or decoration to the image. For example, decorative text that specifies where the image was taken, repeatedly appears across a variety of different images. Often, the reoccurring visual motif, is semantically similar, yet, differs in location, style and content (e.g., text placement, font and letters). This work proposes a deep learning based technique for* blind *removal of such objects. In the blind setting, the location and exact geometry of the motif are unknown. Our approach simultaneously estimates which pixels contain the visual motif, and synthesizes the underlying latent image. It is applied to a single input image, without any user assistance in specifying the location of the motif, achieving state-of-the-art results for blind removal of both opaque and semi-transparent visual motifs.*

1. Introduction

Images shared over the web often contain overlaid objects, or *visual motifs*, such as text, symbols or drawings, which can serve a variety of purposes. For example, adding university emblems or team logos can instill an image with pride, or, text with temperature information can add extra information to an image. In a subset of cases, motifs are used as a protection of image ownership, *e.g.,* with digital watermarks. However, in many cases, one may want to remove these visual motifs and reconstruct the original image without the existing occlusions.

The removal of these visual motifs and the recovery of a pristine image can be an extremely challenging task. The structure, size and location of these objects varies between different images, making them difficult to detect without user guidance or assumptions about the underlying image. Previous methods have relied on information about the location of the corrupted pixels to be restored [21, 20, 24, 36]. Dekel et al. [5] remove watermarks using large image collections, which contain the same watermark, as well as some minimal user guidance about the watermark location.

We present a method for completely *blind* visual motif removal. In the blind setting, the exact location, structure and size of these motifs is unknown. The generalization ability of our network is demonstrated by removing visual motifs that are not seen during training, *e.g.,* watermark re-

moval from real-world images. See Figures 1 and 8 for examples of visual motif removal.

Unlike previous approaches, our strategy does not require multiple images with the same object to be removed, or the exact location of the motif pixels. Our technique exploits the ability of deep neural networks to generalize from samples seen during training to unseen samples during test time, thereby enabling the removal of novel (*i.e.,* unseen) motifs. The network learns to separate the visual motif from the image, by first estimating which pixels contain the motif and subsequently reconstructing the latent image. In addition to estimating the binary motif matte and reconstructing the latent image, we also reconstruct the visual motif.

Decomposing the input image into both the visual motif and the background image enables the network to create a better reconstruction of the image. In our experiments, we demonstrate that images corrupted by different types of visual motifs such as text or semi-transparent drawings can be recovered successfully.

2. Related work

While blind removal of visual motifs is not addressed directly in the literature, there are related works in the watermark removal and blind image inpainting fields.

Watermark removal. Digital watermarks are commonly used to identify the copyright of images to prevent users from using images available online without consent. Though many techniques for creating watermarks are available [27], the visible watermarking process is usually composed of embedding a logo or text with varied opacity into a target image.

Watermark attack methods, *e.g.,* [29], strive to accurately remove watermarks from a host image. The robustness of a watermark to an attack is essential for protecting image copyrights. Watermark removal methods have been researched to test and improve the resilience of watermarks. Removing a watermark typically involves two steps: watermark detection and reconstruction of the underlying image content.

Most watermark removal methods try to reconstruct a single watermarked image relying on user guidance to identify the location of the watermark. Huang and Wu [11] obtain the watermark location from the user and then use image inpainting to restore the latent image. Pei and Zeng [21] use independent component analysis (ICA) to separate the source image from the watermark. Park et al. [20] find a color space transformation that separates the source image from the watermark. More recently, Dekel et al. [5] assume the existence of a collection of images with the same watermark to estimate and remove it. Contrary to these methods, our method does not require any user intervention and the watermark can be removed using a single image only.

Image inpainting. Image inpainting is the process of restoring or completing portions of an image [1, 2, 3, 4, 22]. Most image inpainting approaches are *non-blind*, since they assume that the binary mask indicating the pixels to be restored is given [10, 12, 28, 32, 33].

Some works consider the case of *blind* inpainting, where the mask of the pixels to be restored should be detected as well. Zoran et al. [36] present a GMM based model for estimating the binary mask and restoring the latent image. Other approaches use deep neural networks with both blind and non-blind settings that can be used recover the latent image [18, 24]. Clearly, in all cases the performance degrades when the inpainting is performed blindly, without the mask of the missing pixels.

It is possible to use blind inpainting for watermark removal, but this would not take into account the latent image information provided by the watermark semi-transparency. In this sense, it would be inferior to methods which use this available information (*e.g.,* [5]).

Image restoration. Other previous works related to the removal of visual motifs include the removal of raindrops [6, 14, 23, 34], distracting objects [9], and reflections [8, 31, 35] from images. Our work takes inspiration from a reflection separation work [15], which separates an image with a reflection into two clear images. Contrary to this method, we use the separation to remove visual motifs from an image. The elements removed by our method can be of different sizes. They can be very small compared to the entire image and therefore hard to detect.

3. Method

Removing a visual motif from a single image is a difficult task to achieve without prior knowledge about the motif or the image. Our proposed approach tackles this problem using a convolutional neural network (CNN) trained to remove visual motifs embedded in an image. We train the network using a synthesized dataset of images with visual motifs (see Figure 4).

Our network learns to separate the visual motif from the image, by estimating the visual motif matte and reconstructing the latent image. During training, the loss computation uses the input image and the visual motif as ground-truth to train an encoder and decoder networks. Our network encodes the corrupted image into a latent representation, which is decoded by three parallel decoder branches: one for estimating the latent image, motif matte and motif image. The final image is generated by using the estimated motif matte to select pixels from either the input image or the reconstructed image. See Figure 2 for an overview.
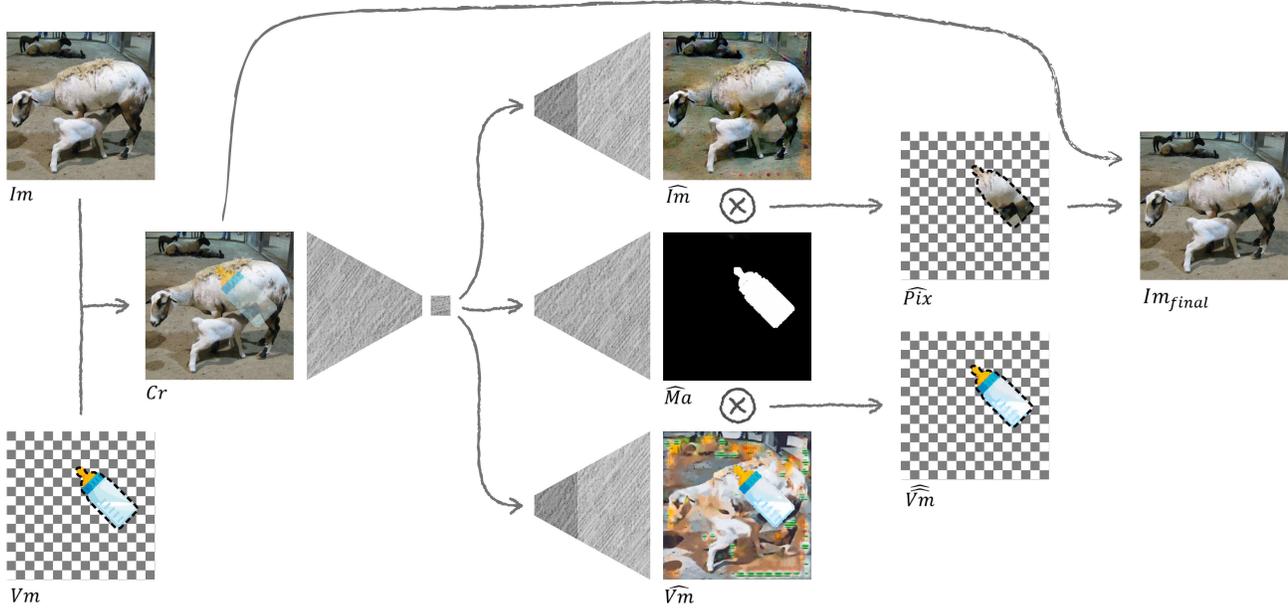
Figure 2: *Method overview.* The network consists of one encoder and three decoders. The top and bottom decoder branches reconstruct the background image and the overlaid visual motif, respectively. These decoders share weights in the first few layers (marked by darker gray color). The middle branch estimates the mask of the visual motif. The final output is generated by using the mask to select pixels from either the input image or the reconstructed image.

3.1. Motif embedding

Matting a motif (Vm) onto an image (Im) can be obtained by:

$$Cr = \alpha \circ Vm + (1 - \alpha) \circ Im, \qquad (1)$$

where Cr is the synthesized corrupted image, and α is the spatially varying transparency, or *alpha matte*. The alpha matte contains a scalar α_i at pixel location i, such that $\alpha_i \in (0, 1)$ inside the motif region and 0 elsewhere. In the case where the corrupted image contains a completely opaque motif (*i.e.,* when $\alpha_i = 1$ for all motif pixels), recovering the latent image is equivalent to image inpainting. Following the work of [5], we also experiment with more challenging motifs by applying a spatial perturbation to motif pixels before matting. For recovering the latent image im, both α and the visual motif Vm must be known. Then it is simply given by

$$Im = \frac{Cr - \alpha \circ Vm}{1 - \alpha}. \qquad (2)$$

Similarly, the recovery of the visual motif requires the latent image Im in addition to the α matte information.

3.2. Blind visual motif removal

Our baseline network is composed of one encoder and two decoder branches: one for estimating the original background image $\widehat{Im}$, and another that reconstructs the visual motif mask $\widehat{Ma}$.

The final image is reconstructed by replacing all the pixels in the estimated visual motif region in the original corrupted image Cr with the corresponding pixels from the reconstructed image $\widehat{Im}$. The final resulting image is given by

$$Im_{final} = (1 - \widehat{Ma}) \circ Cr + \widehat{Ma} \circ \widehat{Im}, \qquad (3)$$

where $\circ$ denotes element-wise multiplication.

The loss term of the baseline network is composed of two reconstruction losses: one for the estimated visual motif mask (denoted L_{mask}), and another for the background image in the corrupted area (denoted L_{im}). The total loss for the baseline network is

$$L = L_{mask} + \lambda L_{im}. \qquad (4)$$

For the mask loss L_{mask}, we use binary cross entropy between the estimated $\widehat{Ma}$ and ground-truth Ma motif masks

$$L_{mask} = \qquad (5)$$
$$-\frac{1}{W \cdot H} \sum_{\substack{0<i<H \\ 0<j<W}} Ma_{ij} \log \widehat{Ma_{ij}} + (1 - Ma_{ij}) \log \left(1 - \widehat{Ma_{ij}}\right),$$

where W is the width and H is the height of the image.

For the image restoration loss L_{im}, we use the ℓ_1 distance between the recovered image $\widehat{Im}$ and the original image Im in the foreground mask pixels:

$$L_{im} = \frac{1}{|Ma|} \cdot \ell_1 \left(Pix, \widehat{Pix}\right), \qquad (6)$$

(a) Encoder-decoder main levels.

$128^2 \times 3$ $64^2 \times 32$ $32^2 \times 64$ $16^2 \times 128$ $8^2 \times 256$ $4^2 \times 512$ $8^2 \times 256$ $16^2 \times 128$ $32^2 \times 64$ $64^2 \times 32$ $128^2 \times 3$

(b) Detailed view of an encode segment.

repeat x 3

$64^2 \times 32$ Conv 3 × 3   Relu   Conv 3 × 3   Batch Norm   Relu   Max Pool $32^2 \times 64$

(c) Detailed view of a decode segment.

concat from encoder

repeat x 3

$16^2 \times 128$ Transpose Conv 3 X 3 $32^2 \times 128$ Conv 3 × 3   Relu   Conv 3 X 3   Batch Norm   Relu $32^2 \times 64$
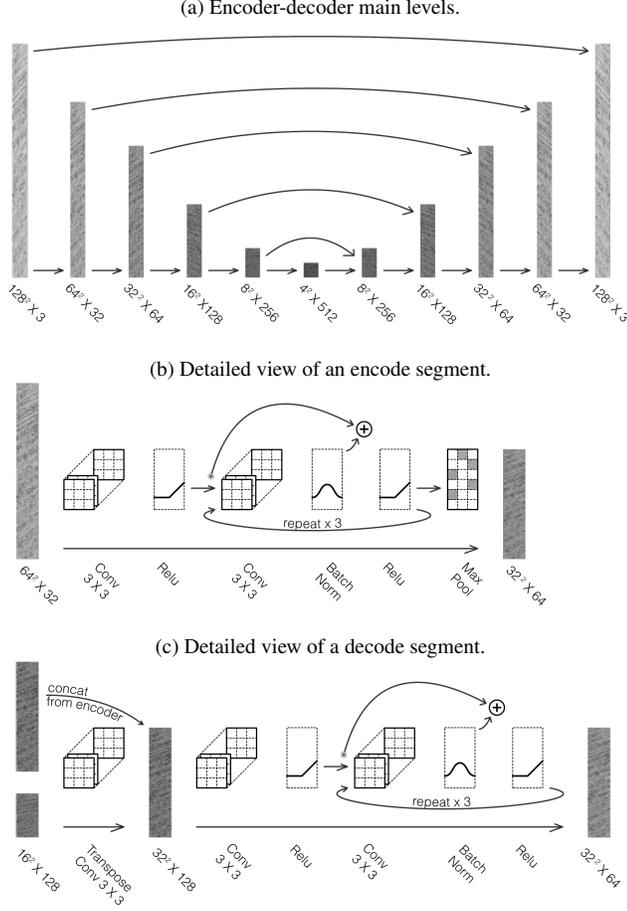
Figure 3: Detailed network architecture.

where $|Ma|$ is the size of the mask (in pixels); Pix and $\widehat{Pix}$ represent the values in the region of the visual motif, of the original background image Im and the estimated image $\widehat{Im}$ respectively. These are given by:

$$Pix = Im \circ Ma \qquad \widehat{Pix} = \widehat{Im} \circ Ma. \qquad (7)$$

Notice that we use the original mask Ma (and not $\widehat{Ma}$) in $\widehat{Pix}$ for calculating the L_{im} loss during training. We do this in order to separate between the loss terms of the two decoders (that generate $\widehat{Im}$ and $\widehat{Ma}$). Otherwise, L_{im} affects the mask reconstruction decoder such that it tends to converge to local minima (an empty motif mask).

Naturally, in some cases, the goal would be to reconstruct the visual motif. For such cases, we add another decoder branch for inferring the original visual motif. This network outputs the image $\widehat{Vm}$ and has shared weights with the image decoder in the first few layers (see Figure 2). Later, in the ablation study in Section 4.3, we show that the insertion of the additional branch does not only enable the recovery of the visual motif, but also leads to a small im-

provement in the quality of the reconstruction of the background image.

The addition of the visual motif branch adds the loss term

$$L_{Vm} = \frac{1}{|Ma|} \ell_1 \left(Vm, \widehat{\widehat{Vm}} \right), \qquad (8)$$

where Vm and $\widehat{\widehat{Vm}}$ represent the ground truth visual motif before blending and the generated visual motif given by:

$$\widehat{\widehat{Vm}} = \widehat{Vm} \circ Ma, \qquad (9)$$

where $\widehat{Vm}$ is the output of the bottom branch in Figure 2. All the above lead to the following final loss term:

$$L = L_{mask} + \lambda \left(L_{im} + L_{Vm} \right). \qquad (10)$$

The shared weights at the lower levels of the image and visual motif decoders emerges from the simple observation in equation 2, where recovering one helps in recovering the other.

3.3. Network architecture

Each sub-network is a fully convolutional deep neural network, based on the U-Net architecture [25]: the feature maps from decoder layer are combined with the corresponding encoder layer features maps at the same resolution.

Figure 3 shows the architecture details. The encoder and decoders contain five main segments. In the encoder, each segment increases the number of channels by a factor of two. It includes 3×3 convolutional layers, batch normalization [13] and the ReLU activation unit [19]. The spatial size of the input is reduced using max pooling with stride-2.

The decoders perform an *inverse operation*. Each decode segment expands the spatial size of the input and reduces the number of channels by a factor of two. It starts with a 3×3 transposed convolutional layer with stride-2. Afterwards, an additional convolutional layer is applied on both the previous output and the corresponding encoded data. Batch normalization and ReLU are also used. After the last segment of the decoder, the final output is generated by applying a 1×1 convolutional layer followed by a sigmoid activation for the mask decoder and a tanh activation for the image and visual motif decoders.

Notice that in both the encoder and decoders, three residual blocks are used in each segment. This expands the receptive field and improves the quality of the recovered image (see Section 4.3).

4. Experiments

We show the results of our method on various types of visual motifs: color and gray scale text, emojis [7] and geometric shapes. We have trained a different network for each type of visual motif. In all of our experiments, our background images consist of 2700 training

Figure 4: *Training patches*. First row contains 128×128 synthesized corrupted patches. The middle row shows the output of the mask decoder (estimated masks) and the bottom row presents the final reconstructed patches.

	Translation PSNR / SSIM	Pert. + Opc. PSNR / SSIM	Scale + Rot. PSNR / SSIM
CFM[16]	24.16 / 0.976	N/A	N/A
MMR[5]	37.41 / 0.977	33.07 / 0.966	N/A
SDB[31]	34.64 / 0.973	34.82 / 0.972	34.69 / 0.972
SIRF[35]	31.18 / 0.970	32.87 / 0.970	32.90 / 0.969
Ours	**38.46 / 0.986**	**38.08 / 0.986**	**37.63 / 0.983**

Table 1: Watermark removal comparison to existing approaches when applied on watermarks with perturbations to translation, scale and rotation.

	Light Font Test PSNR / SSIM	Bold Font Test PSNR / SSIM
Sh-CNN[24]	31.879 / 0.9522	28.436 / 0.9118
FoE[26]	38.155 / 0.9886	33.360 / 0.9675
EPLL[36]	38.672 / 0.9884	33.377 / 0.9675
Ours	**39.079 / 0.9890**	**34.676 / 0.9710**

Table 2: Quantitative inpainting results for images corrupted by lighter and bolder font.

and 300 separate testing images, randomly chosen from the Microsoft COCO val2014 dataset [17]. Our code is available at `https://github.com/amirhertz/visual_motif_removal`.

4.1. Transparent visual motifs

Training. To train the network, we synthesize several sets of images embedded with a certain visual motif type (see Figure 4). For each type, we have synthesized 20,000 512×512 corrupted images as follows:
(i) Colored texts with Latin characters that have random font and color. These are blended into the background images using a constant opacity $\alpha \in (0.3, 0.7)$ within each image.
(ii) Gray scale texts are characterized by random light color and dark borders. They are embedded with a uniformly dis-

tributed opacity blending field with 10% variance around the constant α and a random spatial perturbations of up to one pixel shifts.
(iii) The emojis dataset has 800 randomly chosen objects. They are blended with a constant opacity in the range $\alpha \in (0.4, 0.6)$ and are applied with the same perturbations and 10% opacity variance.
(iv) Geometric grayscale shapes such as lines, rectangles and ellipses; they are embedded with a constant opacity in the range $\alpha \in (0.2, 0.9)$.

For each training image, we embed up to 10 visual motifs using random positions, scales, crops and rotations for data augmentation. To speed up training, we use 128×128 sized patches randomly selected from the corrupted image as input to the network (see Figure 4), but test on full resolution images.

Testing. The test images consist of full resolution images, which are unseen during training. For the text experiments, we trained on Latin characters, but the visual motifs are unseen characters from other languages, *e.g.,* Chinese, Japanese, Hindi, or Georgian. The emojis test set consists of emojis that have not been used during training. Various reconstruction results are shown in Figures 1 and 5, as well as in the supplementary material. To highlight the difference between the test and train sets, we show several test emojis (*e.g.,* the ones that appear in figures 1 and 5) with their 5 Euclidean nearest neighbors in the supplementary material.

Single visual motif removal. We compare the performance of our method to four algorithms: closed-form solution to natural image matting (CFM) [16], multi-image reconstruction (MMR) [5] and two deep image reflection separation approaches (SIRF, BDN) [31, 35], which we adapt to the motif removal problem. To facilitate the task of MMR, we use only a single unseen emoji as the overlaid visual motif on the entire test set (1000 images in total). To facilitate the task for CFM, we provide the ground-truth motif blending image.

Our test images are divided into three groups with increasing levels of matting deformation. In the first group, the visual motif has a fixed size of 160×160 px. It appears at a random position and blended with a constant opacity within each image $\alpha \in (0.4, 0.6)$. For the second group, we apply the blending with added perturbations to the edges of the visual motif and a varying opacity. For the third group, variations in size and rotation were added. The visual motif's size was $120 \times 120 - 200 \times 200$ px and randomly rotated between $\theta \in (-20°, 20°)$.

The results are measured by comparing the reconstructed images to the ground truth one under PSNR and the structural similarity index (SSIM) [30]. As can be seen in both Table 1 and the sample images from the second test in figure 6, our method performs better both in the easier and more
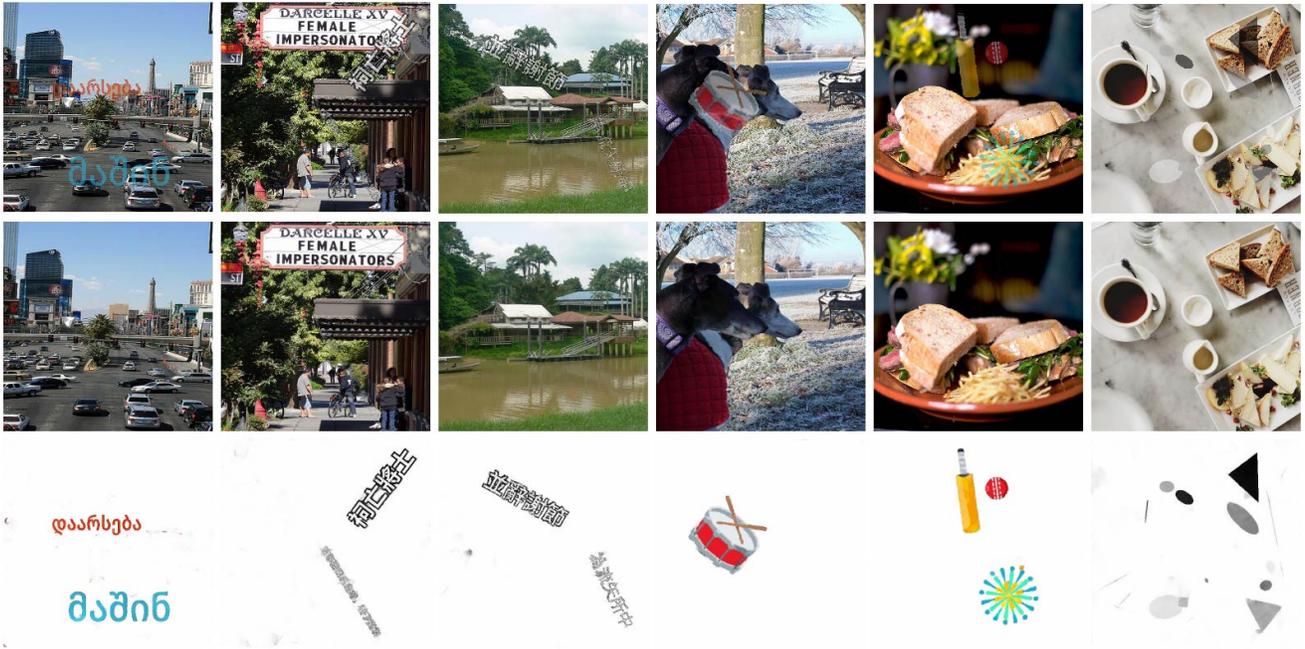
Figure 5: Semi-transparent visual motif removal. Top: test images embedded with unseen motifs. Middle: our reconstructed background images. Bottom: our reconstructed visual motifs. More results are presented in the supplementary material.

| Ground Truth | Corrupted | MMR | SIRF | SDB | Ours |

Figure 6: Visual comparisons for the second test in Table 1 for removing a single watermark from a multi-image collection. More results may be found at the supplementary material.

challenging scenarios. More visual results of all tests are displayed in the supplementary material.

Note that non-deep approaches do not handle the more difficult scenario (our method is not effected). In addition, the MMR algorithm has a missed detection rate of 4.5% (the location of the visual motif is not correctly identified). These cases are excluded from the overall results.

The reflection separation works follow an *encoder-decoder* framework, since they must generalize to reflections potentially present in the entire image. The major difference between reflection and visual motif removal in images is the distribution of transparent pixels. Motif embedded images will always contain content from the *background* image. Since our method specializes in motifs, we constrain the reconstruction to the (learned) binary motif

mask. In our ablation study (see section 4.3), we demonstrated that using a regular auto-encoder is inferior to a network with a motif mask branch.

Watermark removal. Naturally, our method can be *abused*, and be used for removing watermarks from protected images. See Figure 8 for examples of removing watermarks from various stock photography services. More examples are shown in the supplementary material.

For this task, we trained the network on a combination of the different training sets outlined above. Specifically, we used the synthetic light texts set (ii) and the gray scale shapes set (iv), in addition to a dataset of synthetic white texts. These texts were embedded using a constant opacity $\alpha \in (0.2, 0.7)$, followed by a 3×3 Gaussian blur filter,
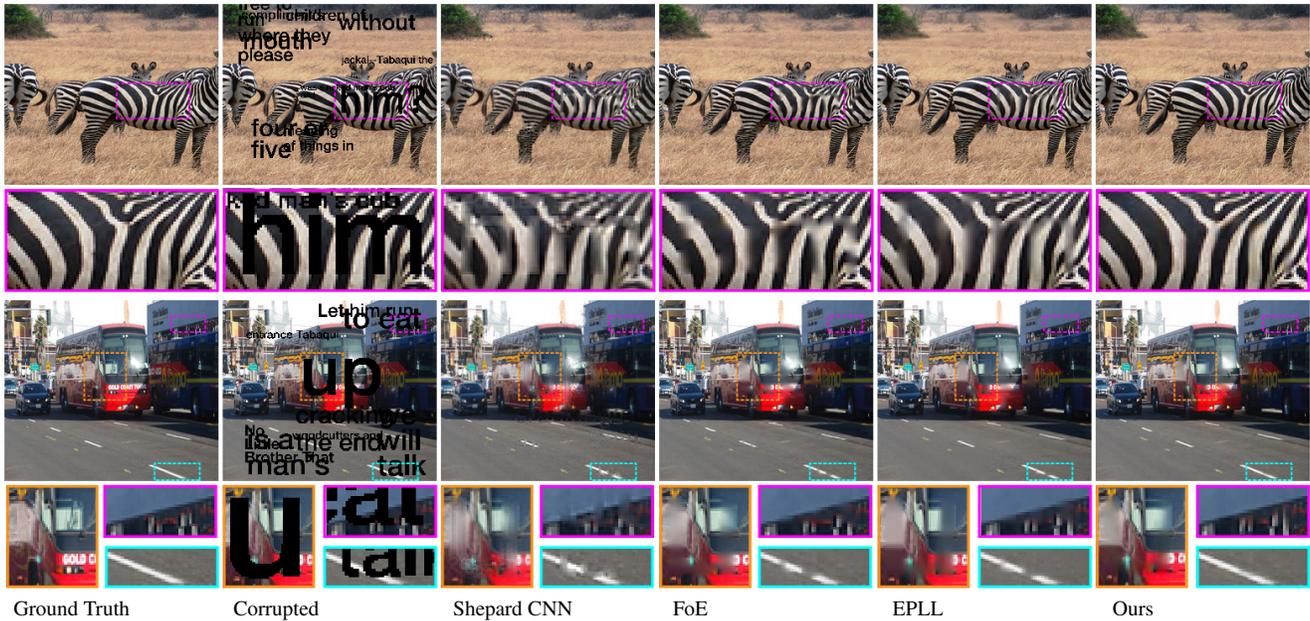
Figure 7: Visual comparisons with other inpainting methods. Our results are less blurred and the patterns distracted by the letters are inpainted in a more continuous way.

| Ground Truth | Corrupted | Shepard CNN | FoE | EPLL | Ours |

Figure 8: *Watermark removal examples*. Top: input images, middle: reconstruction results, and bottom: enlarged patches.

which causes a gradual matting between the texts and the background images.

4.2. Inpainting

Another type of visual motif we tested our method on is blind inpainting tasks: the network has no explicit priors on the background image or the mask of the corrupted regions. We evaluate our network on two levels of inpainting regions. The first is composed from 256×256 pixels images on which we tiled random black texts of the font Hel-

vetica light and with random range of rotations and sizes between 80 to 120 pixels per word. For the second experiment, we used a bolder Helvetica medium font with up to 150 pixels per word. For this task, we have trained our baseline configuration without the Vm branch since the retrieval of the pixel-mask is equivalent to the retrieval of the black regions. Our network was trained on the same 2700 images which were randomly cropped to 64×64 pixels patches. Each test dataset contains 100 unseen images and unseen words. We compared our method with state-of-

	Auto-encoder	Baseline	Baseline + VM	**Baseline + shared VM**
PSNR	27.729	37.359	37.564	**37.636**
SSIM	0.9443	0.9882	0.9883	**0.9884**

Table 3: Ablation study of the network components.

# res blocks	PSNR / SSIM	Shared depth	PSNR / SSIM
1 block	35.964 / 0.9848	1 level	37.414 / 0.9883
3 blocks	**37.636 / 0.9884**	**2 levels**	**37.636 / 0.9884**
5 blocks	37.411 / 0.9880	3 levels	37.4387 / 0.9882

Table 4: Ablation study of the network depth.

the-art methods [24, 26, 36], which unlike our method, were supplied with the ground truth masks of the opaque regions. Table 2 summarizes the quantitative results on the 100 test images of each experiment. Figure 7 shows visual comparisons to the other methods. The results clearly demonstrate that our method performs better in the task of completing real imagery patterns and achieves sharper results when filling large corrupted regions.

We also used our pre-trained network on the sample images from these methods (see supplementary material). Our method still outperforms even when using their sample images.

4.3. Ablation study

In our ablation study, we verify our network configurations by testing the quality of the results obtained by the different branches configurations and the inner architecture setup.

We generate the training dataset for the study as described previously, using emojis [7] as the visual motifs blended on images of size 256×256 pixels. The motif size is uniformly randomly sampled in the range $(80, 160)$ pixels. We blend the motif with a blending constant α which is randomly sampled $\in (0.3, 0.7)$. Each tested network is trained for 100 epochs on patches of size 128×128. We test the performance of the networks on 200 unseen images blended with unseen motifs.

The network configurations we compared include an auto-encoder U-Net model, our baseline two branches model, a three branches model with a separated branch for the motif (Baseline + Vm) and the chosen model (Baseline + shared Vm) with three branches as well as shared weight between the image and the motif decoders.

Table 3 summarizes the quantitative results on the test set (visual comparisons in the supplementary material). Both results show that the auto-encoder network, without the mask branch, cannot separate the motif from the background image. The chosen model with the shared motif de-

Figure 9: *Failure case*. Our method may struggle when the motif is embedded in flat uniform regions.

coder (Baseline + shared Vm) generates the best restoration within the corrupted regions.

In addition, we explore the impact of the size of the shared weights between the image and the motif branches and try shallower and deeper setups by changing the number of residual blocks within each encoder-decoder segment. The quantitative results for these studies in Table 4 show that the addition of (up to three) res-blocks and the use of the shared weights (up to two decode levels) improve the reconstruction of the image.

5. Conclusion

We have presented a method for the identification and removal of visual motifs embedded in images. Our method is the first to remove visual motifs from images blindly (*i.e.,* without explicit prior information). It should be stressed that unlike previous methods, our method does not require any user input or making intricate assumptions about the visual motif.

Deep neural networks are capable of generalizing beyond the examples presented during training. This enables us to train on one set of visual motifs, and during test time remove a large variety of other visual motifs, which may differ significantly in size, location, colors and even shape from the training set. This generalization power is clearly demonstrated in the successful removal of Indian, Chinese and Japanese characters when training on Latin characters (*e.g.,* Figure 1 and 5).

A notable advantage of our approach is the addition of a decoder network for reconstructing the distracting visual motif. We have shown that this leads to a better latent image reconstruction and visual motif detection. This is particularly notable in the removal of semi-transparent visual motifs, which are embedded with varying transparency or small spatial perturbations.

When the visual motif is embedded over a uniform background, the motif traces are still somewhat noticeable (see example in Figure 9). We believe this limitation can be alleviated by training our network using hard example mining to make it more accurate in cases where the background of the visual motif is relatively uniform. Another interesting future direction would be to add a generative network for synthesizing complex visual motifs, such as raindrops or other visual distractors.

References

[1] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3):24, 2009.

[3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[4] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

[5] Tali Dekel, Michael Rubinstein, Ce Liu, and William T Freeman. On the effectiveness of visible watermarks.

[6] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE international conference on computer vision*, pages 633–640, 2013.

[7] Emojione. Signature emoji collection. `https://www.emojione.com`, 2018.

[8] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3238–3247, 2017.

[9] Ohad Fried, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. Finding distractors in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1703–1712, 2015.

[10] Andrew Gilbert, John Collomosse, Hailin Jin, and Brian Price. Disentangling structure and aesthetics for style-aware image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1848–1856, 2018.

[11] Chun-Hsiang Huang and Ja-Ling Wu. Attacking visible watermarking schemes. *IEEE transactions on multimedia*, 6(1):16–30, 2004.

[12] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[14] Idan Kligvasser, Tamar Rott Shaham, and Tomer Michaeli. xunit: Learning a spatial activation function for efficient image restoration. In *CVPR*, 2018.

[15] Donghoon Lee, Ming-Hsuan Yang, and Songhwai Oh. Generative single image reflection separation. *arXiv preprint arXiv:1801.04102*, 2018.

[16] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2008.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[18] Yang Liu, Jinshan Pan, and Zhixun Su. Deep blind image inpainting. *arXiv preprint arXiv:1712.09078*, 2017.

[19] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[20] Jaesik Park, Yu-Wing Tai, and In So Kweon. Identigram/watermark removal using cross-channel correlation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 446–453. IEEE, 2012.

[21] Soo-Chang Pei and Yi-Chong Zeng. A novel image recovery algorithm for visible watermarked images. *IEEE Transactions on Information Forensics and Security*, 1(4):543–550, 2006.

[22] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003.

[23] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2018.

[24] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2015.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] Stefan Roth and Michael J Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205, 2009.

[27] Hai Tao, Li Chongmin, Jasni Mohamad Zain, and Ahmed N Abdalla. Robust image watermarking theories and techniques: A review. *Journal of applied research and technology*, 12(1):122–138, 2014.

[28] T. Tirer and R. Giryes. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing*, 28(3):1220–1234, March 2019.

[29] Sviatolsav Voloshynovskiy, Shelby Pereira, Thierry Pun, Joachim J Eggers, and Jonathan K Su. Attacks on digital watermarks: classification, estimation based attacks, and benchmarks. *IEEE communications Magazine*, 39(8):118–126, 2001.

[30] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[31] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for

single image reflection removal. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[32] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *arXiv abs/1006.3056*, 2010.

[33] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention.

[34] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017.

[35] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. *CVPR*, 2018.

[36] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 479–486. IEEE, 2011.