# Light Field Messaging with Deep Photographic Steganography

Eric Wengrowski
Rutgers University
ericwengrowski@gmail.com

Kristin Dana
Rutgers University
kristin.dana@rutgers.edu

Figure 1. Goal of LFM (Light Field Messaging): embed a message within an image or video, display the image/video on-screen, photograph it with a handheld camera, and recover the hidden message. LFM significantly outperforms other synchronization-free steganography techniques for camera-display messaging in message bit recovery error (BER). Our code and dataset are available here [1].

## Abstract

*We develop Light Field Messaging (LFM), a process of embedding, transmitting, and receiving hidden information in video that is displayed on a screen and captured by a handheld camera. The goal of the system is to minimize perceived visual artifacts of the message embedding, while simultaneously maximizing the accuracy of message recovery on the camera side. LFM requires photographic steganography for embedding messages that can be displayed and camera-captured. Unlike digital steganography, the embedding requirements are significantly more challenging due to the combined effect of the screen's radiometric emittance function, the camera's sensitivity function, and the camera-display relative geometry. We devise and train a network to jointly learn a deep embedding and recovery algorithm that requires no multi-frame synchronization. A key novel component is the camera display transfer function (CDTF) to model the camera-display pipeline. To learn this CDTF we introduce a dataset (Camera-Display 1M) of 1,000,000 camera-captured images collected from 25 camera-display pairs. The result of this work is a high-performance real-time LFM system using consumer-grade displays and smartphone cameras.*

## 1. Introduction

In Light Field Messaging (LFM), cameras receive hidden messages from electronic displays concealed within ordinary images and videos. There are many applications for visually concealed information including interactive visual media, augmented reality, road signage for self-driving cars, hidden tags for robotics, privacy-preserving communication, and tagged digital artwork. When the hidden message is recovered from on-screen images, the task has significant challenges and is fundamentally different from the traditional task of steganography. The conversion of a digital image into a light field depends on the characteristics of the electronic display such as the spectral emittance function and spatial emitter pattern. Similarly, the transformation of light field to image depends on the camera pose, sensitivity curves, spatial sampling, and radiometric response. Our unique approach is to learn the entire pathway as a single camera-display transfer function (CDTF) modeled by a supervised deep network. This CDTF component is then used in a larger network that maximizes the accuracy of the camera-recovered message, while minimizing the perceived artifacts in the observed display image.

Steganography in prior years referred almost exclusively to the digital domain where images are processed and transferred as digital signals [3]. The classic methods for *digital steganography* range from simple alteration of least significant intensity bits to more sophisticated fixed-filter transform domain techniques [4]. Recent work has moved the prior fixed filter approaches to incorporate modern deep learning [2]; but these methods are designed for digital steganography and fail completely for the task of light field messaging as illustrated in Figure 2.

In this paper, we propose a single-shot end-to-end *photographic steganography* algorithm for light field messaging. Our method is comprised of: a CDTF network to model

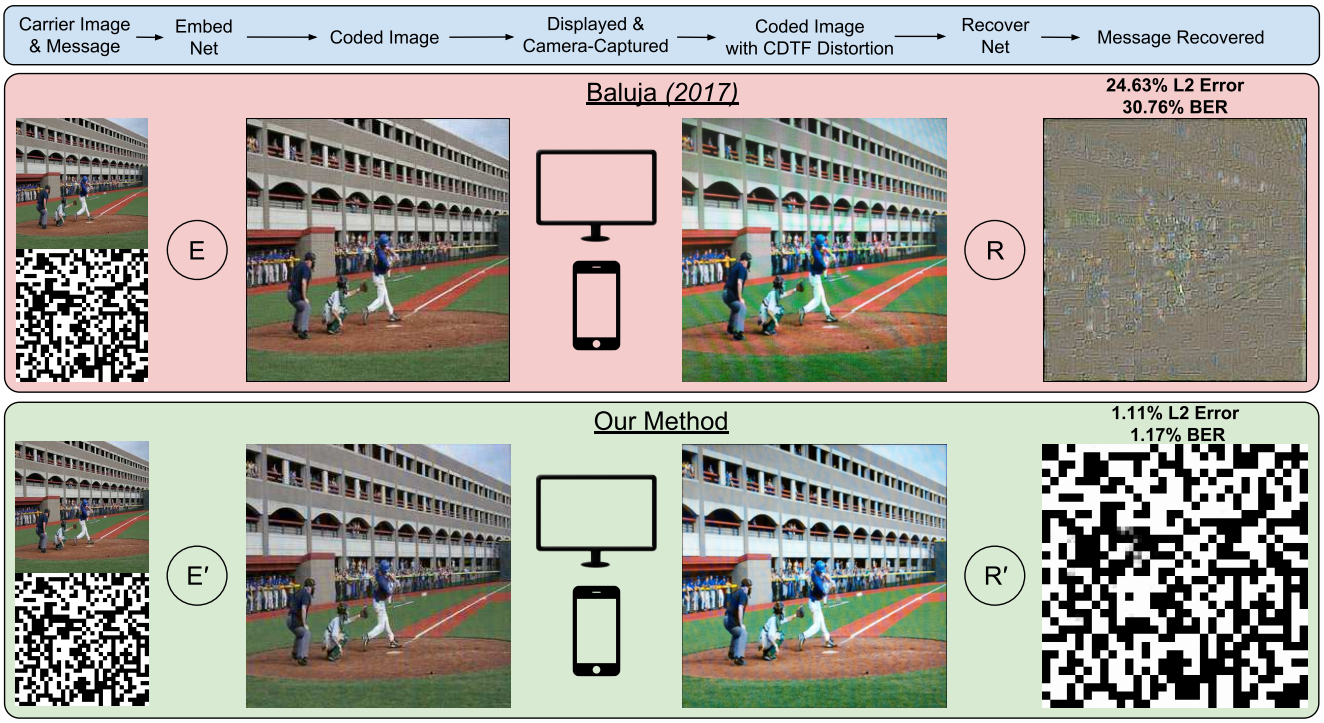


Figure 2. Digital steganography methods such as Baluja [2] are not suitable for photographic steganography. The distorting effect of the light field transfer, as characterized by the camera-display transfer function (CDTF), destroys the information steganographically encoded in carrier image pixels. We compare the digital steganography methods introduced by Baluja (top) with our proposed photographic steganography method (bottom). Unlike previous methods, the proposed method includes a model of the CDTF within the training pipeline so that a learned steganographic function for embedding and recovery is robust to CDTF distortion.

the camera and display without radiometric calibration; an embedding network to optimally embed the message within an image; and a message recovery network to retrieve the message on the camera side. An important attribute of our approach is single-frame operation so that no temporal synchronization between camera and display is needed, greatly increasing the practical utility of the method. We assume that properties of the camera hardware, display hardware, and radiometry are not known beforehand. Instead, we develop a training dataset *Camera-Display 1M* with over one million images and 25 camera-display pairs, to train a neural network to learn the representative CDTF. This approach allows us to train the embedding network independently from the representative CDTF. The proposed photographic steganography algorithm learns which features are invariant to CDTF distortion, while simultaneously preserving perceptual quality of the carrier image.

The main contributions in this paper are: 1) a photographic steganography algorithm based on deep learning architectures; 2) development of a new paradigm for camera-display imaging systems, CDTF-network; 3) Camera-Display 1M: a dataset of 1,000,000 camera-captured images from 25 camera-display pairs.

## 2. Related Work

**Single vs. Dual Channel** Light field messaging, also known as camera-display or screen-camera communication, has been addressed by both the computer vision and the communications literature. Early systems in the communications area concentrate on the screen-camera transfer and do not seek to hide the signal in a display image [5, 6, 7, 8]. In computational photography, single channel systems have been developed for structured light [9] that develop optimal patterns for projector-camera systems. In the computer vision community, the theme of communicating hidden information in displayed images started with Visual MIMO [10, 11] and continued in other recent work such as InFrame[12, 13, 14, 15] and DisCo [16]. In these dual-channel methods, consistent with our approach, the display conveys information via human observation and the hidden channel transmits independent information via camera-captured video. Prior dual channel methods use fixed filter message embedding using either multiresolution spatial embedding or temporal embedding that requires high frequency displays and high-speed cameras to take advantage of human limitations in perceiving high frequency changes [13, 16, 17].

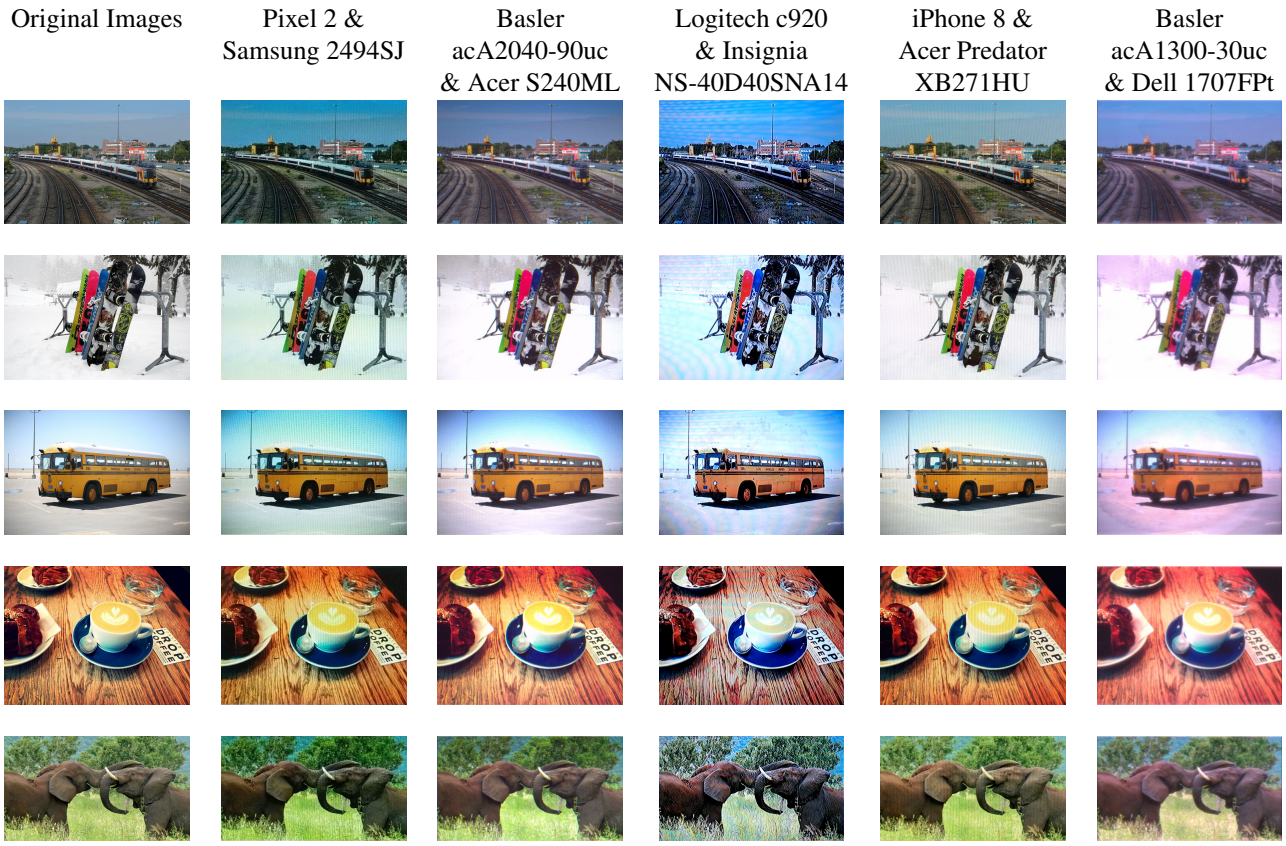


Figure 3. *Camera-Display 1M* examples: Our dataset contains over 1 million images collected from 25 camera-display pairs. Each column corresponds to a different camera-display pair (5 of 25 are shown). Camera properties (spectral sensitivity, radiometric function, spatial sensor pattern) and display properties (spatial emitter pattern, spectral emittance function) cause the same image to appear significantly different when displayed and captured using different camera-display hardware. (Best viewed as zoomed-in PDF.)

**Early Steganography** The early work of classic image-processing steganography can be divided into spatial and transform domain techniques. A simple and common form of spatial domain image steganography involves altering the least significant bits (LSBs) of carrier image pixels to encode a message [18]. Small variations in pixel values are difficult to detect visually and can be used to store relatively large amounts of information [19]. In practice, simple LSB steganography is not commonly used because it is easy to detect and requires lossless image compression techniques [20]. More sophisticated LSB methods can be used in conjunction with various image compression techniques such as graphics interchange format (GIF) and JPEG for more complex and difficult to detect steganography [18]. Transform domain techniques of traditional steganography embed using fourier, wavelet, and discrete cosine tranforms [21, 20, 22, 23]. While there is a large body of work in the steganography literature, the methods use fixed filters and these digital methods are not robust to the light transmission in LFM.

**From Fixed Filter to Deep Learning** In recent years, a new class of image steganography algorithms has emerged that utilize deep convolutional neural networks. Pibre *et al*. [27, 28] and Qian *et al*. [29] demonstrate that deep learning using jointly learned features and classifiers often outperform more established methods of steganalysis that use hand selected image features. Structured neural learning approaches have been explored that integrate classic image and transform domain steganography techniques, such as LSB selection in a carrier image for a text-based message [30, 31].

For deep steganography, Baluja [2] uses deep feed-forward convolutional neural networks that can directly learn feature representations to embed a message image into a carrier image. Rather than constraining the network to select pixels in a carrier image suitable for embedding, the end-to-end steganography networks are trained with constraints that preserve carrier and message image quality. Hayes *et al*. devised a similar steganography algorithm based on deep neural networks that utilizes adversarial learning to preserve the quality of the carrier image

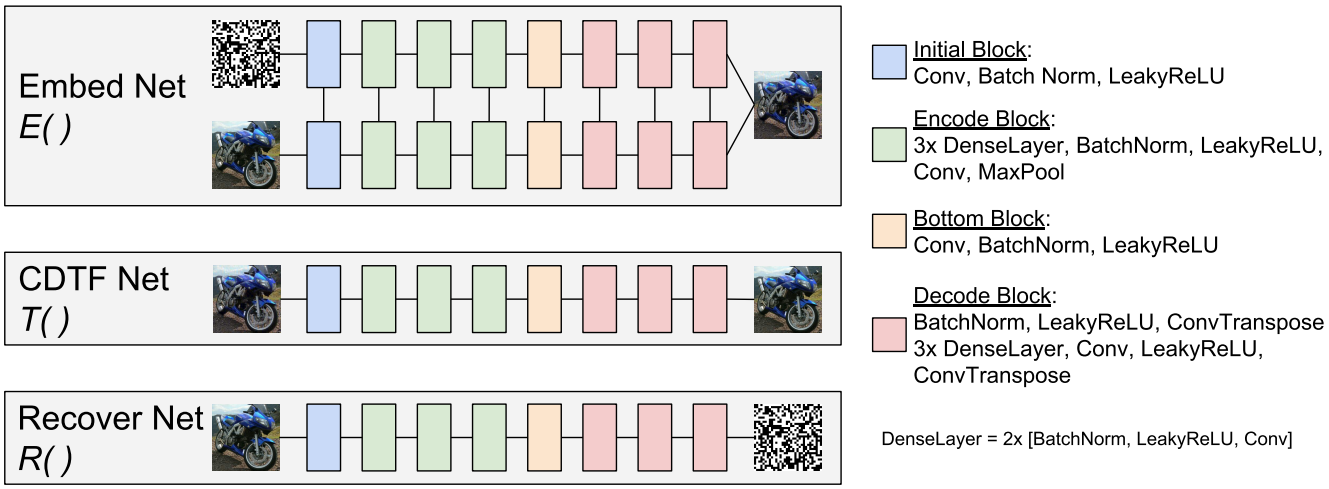


Figure 4. Our steganography model's deep convolutional network architecture. $R()$ and $T()$ are both constructed with an identical architecture inspired by U-net for multiscale analysis [24] and Dense blocks for feature reuse [25]. The embedding function $E()$ combines two images (carrier image and message) into one coded image. $E()$ has a siamese architecture [26] with separate network halves for carrier image and message. The features for carrier image and message are shared at different scales to ultimately produce a single coded image output. Each half of the siamese architecture of $E()$ is identical to $R()$.

and limit steganalysis detection [32]. Deep learning approaches such as these have been extended to include video steganography [33], high bits per pixel (BPP) embedding rates [34], resistance to JPEG compression [35], and new deep learning architectures [36, 37]. While our algorithmic approach also uses deep steganography, there is a significant key difference with prior work: we assume our covert message will be electronically displayed, transmitted as light in free space, and then camera-captured. That is, we address the problem of *photographic steganopraphy* for LFM that distinguishes our work from the prior methods (both classic and deep learning) that address *digital steganography*. Figure 2 demonstrates the clear problem in using digital steganography for LFM: the message cannot be retrieved accurately from the camera-captured image.

**Uniqueness of our Approach** Our work is distinct from prior work in that it simultaneously enables: 1) free space light communication, i.e. light field messaging, 2) dual channel communication where the machine-readable message is hidden from the human, 3) deeply learned embedding/recovery, 4) single-frame synchronization-free methodology, and 5) ordinary display hardware with no high frequency requirements. We are the first to explicitly model and measure the display-camera connection as well as build a first-of-its-kind network and database for learning the coefficients of the camera-display transfer function for use in experiments.

## 3. Methods

We define the terms *message* to refer to the covertly communicated payload, *carrier* to refer to the image used to hide the message, and *coded images* to refer to the combined carrier image and hidden message. Our approach has 3 main components:

- $E()$: a network that hides a message in a carrier image;
- $R()$: a network that recovers the message from the coded image;
- $T()$: a network that simulates the distorting effects of camera-display transfer (CDTF).

We denote the unaltered carrier image $i_c$, the unaltered message $i_m$, the coded image (carrier image containing the hidden message) $i'_c$, and our recovered message $i'_m$. $L_c$ and $L_m$ represent generic norm functions used for image and message loss, respectively. We wish to learn the functions $E()$ and $R()$ such that:

$$\begin{aligned} \text{minimize} \quad & L_c(i'_c - i_c) + L_m(i'_m - i_m) \\ \text{subject to} \quad & E(i_c, i_m) = i'_c \\ & R(i'_c) = i'_m \end{aligned} \tag{1}$$

In other words, our objective is to simultaneously minimize the distortions to the carrier image and minimize message recovery error. However, this simple formulation will not yield a solution to our problem. A naively trained steganography network will likely learn an embedding function $E()$ that encodes a message in carrier image LSBs [2]. LSB encoding will be overly distorted by the CDTF, yielding large message recovery errors [38]. Instead, we introduce

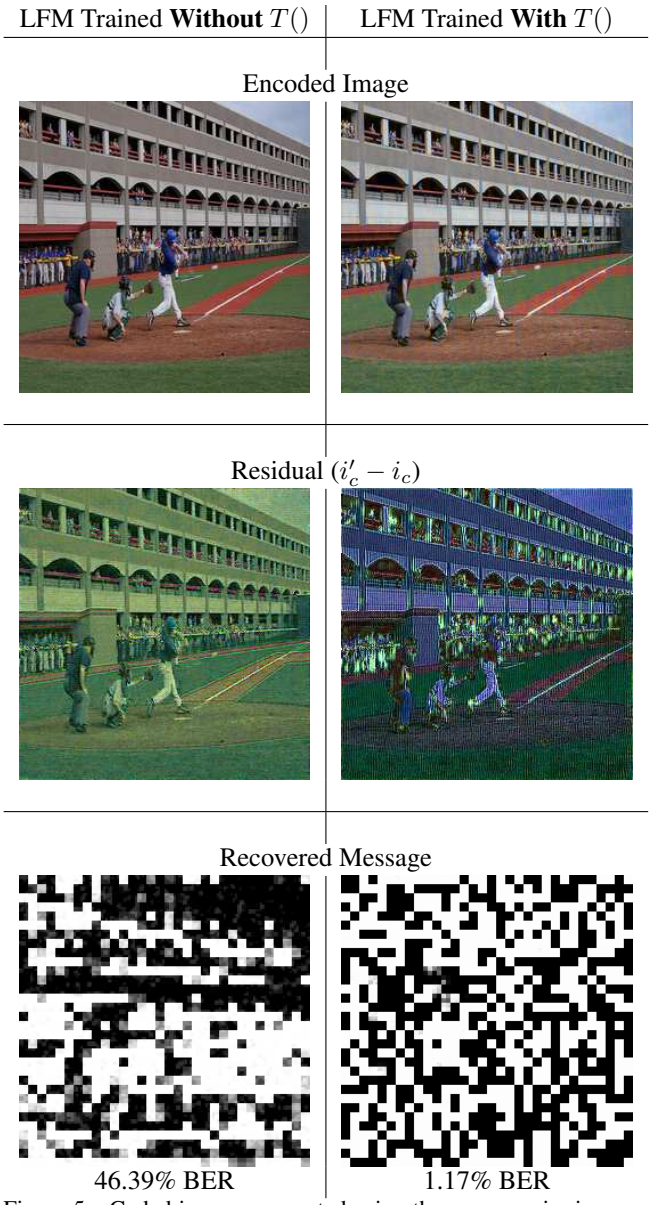


Figure 5. Coded images generated using the same carrier image and message, produced with two otherwise identical steganography architectures: **Left:** trained without the CDTF; **Right:** trained with $T()$ to model CDTF. The per-pixel changes ($i_c - i_c\prime$) in the two middle images are multiplied $\times 50$ for visibility. Notice the significant changes to coded image appearance that our photographic steganography model learns that anticipate the CDTF (right). This experiment was performed using the Pixel 2 camera and Acer Predator XB271HU display.

a third function $T()$ that simulates CDTF distortion. If $i_c$ represents an unaltered carrier image, and $i'_c$ represents a coded image, let $i''_c$ represent a coded image that has passed through the CDTF approximated by $T()$, such that

$\lambda_T = 0$ $\lambda_T = 0.001$ $\lambda_T = 0.01$

Figure 6. Examples of coded images generated by our photographic steganography model with various perceptual loss weights in training. As the perceptual quality metric $\lambda_T$ is increased, the image becomes sharper and has fewer color shift errors. If $\lambda_T$ is too large, BER increases, as is the case when $\lambda_T = 0.01$. (Best viewed as zoomed-in PDF)

$T(i'_c) = i''_c$. Now we denote a new objective:

$$\begin{aligned} \text{minimize} \quad & L_c(i'_c - i_c) + L_m(i'_m - i_m) \\ \text{subject to} \quad & E(i_c, i_m) = i'_c \\ & T(i'_c) = i''_c \\ & R(i''_c) = i'_m \end{aligned} \tag{2}$$

The CDTF function $T()$ must represent both the photometric and radiometric effects of camera-display transfer [38]. This is accomplished by training $T()$ using a large dataset of images electronically-displayed and then camera-captured using several combinations of cameras and displays. This training procedure is detailed in Section 4. After $T()$ is trained, the steganography networks $E()$ and $R()$ are trained, using $T()$ as a fixed constraint.

**Network Architecture** Recent trends in deep learning architectures have been to go deeper [39], with more connections between layers [25], and operate at multiple scales [24]. The proposed steganography networks draw heavily from the aforementioned architectures. The 3 networks $E()$, $R()$, and $T()$ all feature dense blocks with feature maps at different scales in the shape of U-Net. Only $E()$, the network used for embedding, features a siamese architecture [26]. One half of the network is directly linked to the carrier image $i_c$, while the other half is directly linked to the payload image $i_m$, and produces a single output $i'_c$. The outputs from each pair of blocks are concatenated and passed to subsequent blocks. The network architecture can be seen in Fig 4. See the supplementary material for further details of network architecture such as convolutional layer sizes.

**Perceptual Loss** Broadly, our photographic steganography method has 2 goals: 1) maximize message recovery; and 2) minimize carrier image distortion. For coded image fidelity, our objective function uses the $L_2$-norm to measure the difference between $i_c$ and $i'_c$. In prior work, photo-realistic image generation using deep neural networks was accomplished with perceptual loss metrics in

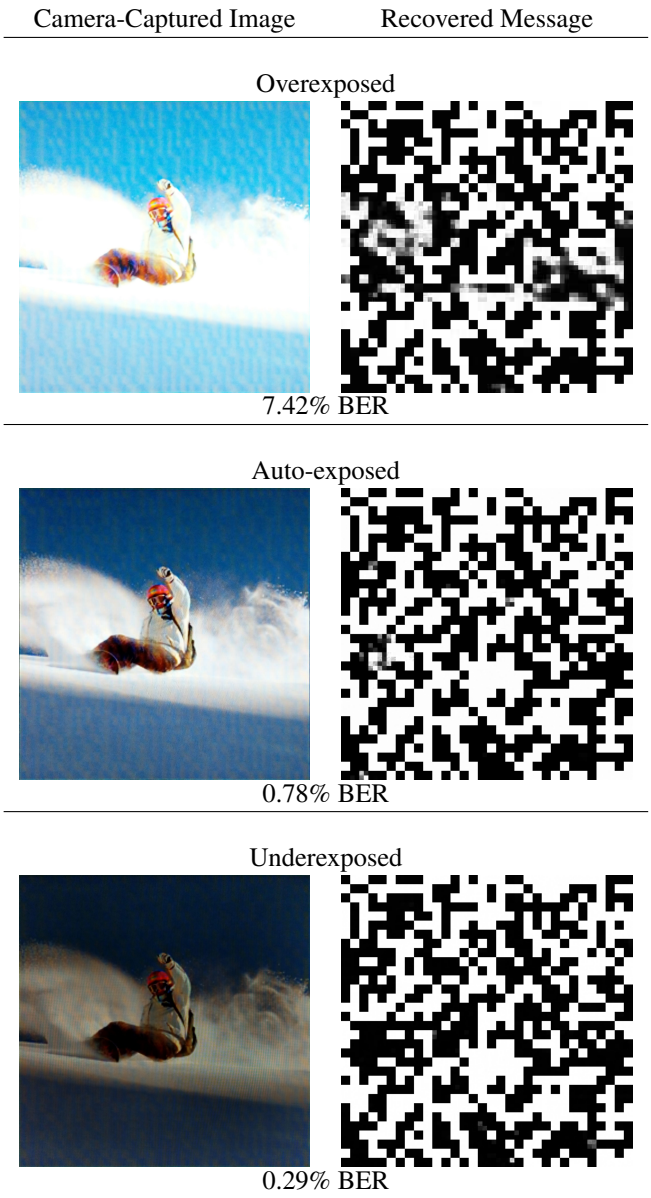


Figure 7. Our approach is robust to modifications of camera exposure, yielding low BER for multiple settings. Underexposure performs better than overexposure because the message cannot be recovered from the saturated snow pixels in the overexposed image. This experiment was performed using the Pixel 2 camera and Acer Predator XB271HU display.

training [40, 41, 42]. The validity of these perceptual loss metrics have been well established [43]. As is common when training neural networks that produce images as output [44], our perceptual loss metric also includes quality loss. Quality loss is calculated by passing $i_c$ and $i'_c$ through a trained neural network for object recognition, in this case VGG [45], and minimizing the difference of feature maps at several depths [46].

**Single Frame Advantage** Previous photographic steganography methods such as Visual MIMO [15, 38, 17] and DisCo [16] rely on temporal processing to isolate carrier image content (static) from message content (dynamic). Synchronization issues make this approach difficult in practice. Each display is operating at a frequency independent from each camera and there is no synchronization between camera and display. Even when a camera and display begin in-phase and at complementary frequencies, small changes in operating frequency, lag from computational load, screen-tearing, and rolling-shutter can all cause the system to quickly fall out of sync. The advantage of using a single frame for embedding is that the temporal synchronization problem is avoided.

### 3.1. Camera-Display 1M Dataset

We present *Camera-Display 1M*, a dataset containing over 1 million images collected using 25 camera-display pairs. Images from the MSCOCO 2014 training and validation dataset [47] were displayed on five electronic displays, and then photographed using five digital cameras. The five electronic displays used are the Samsung 2494SJ, Acer S240ML, Insignia NS-40D40SNA14, Acer Predator XB271HU, and Dell 1707FPt. The five cameras used are the Pixel 2 smartphone, Basler acA2040-90uc, Logitech c920 webcam, iPhone 8 smartphone, and Basler acA1300-30uc. The chosen hardware represents a spectrum of common cameras and displays. To achieve a set of 1M images, 120,000 images of MSCOCO were chosen at random. Each camera-captured image is cropped, warped to frontal view, and aligned with its original. The measurement process was semi-automated and required software control of all cameras and displays. The time-consuming acquisition process has produced a comprehensive dataset that will be made publicly available [1] along with the trained CDTF network parameters. See Figure 3 for examples of how different hardware in the imaging pipeline significantly alters the appearance of the same images.

### 3.2. Training $T()$

The network $T()$ is trained using 1,000,000 image pairs, $i_{COCO}$ representing the original image and $i_{CDTF}$ representing the same image displayed and camera-captured. These images used for training are MS-COCO images [47] that are rendered on an electronic display and then camera-captured using 25 camera-display pairs. The objective of $T()$ is to simulate CDTF distortion by outputting $i_{CDTF}$ given $i_{COCO}$ as input. The objective function we wish to minimize is:

$$\begin{aligned} T_{loss} = & L_2(i_{COCO} - i_{CDTF}) + \\ & \lambda_T * L_1(VGG(i_{COCO}) - VGG(i_{CDTF})). \end{aligned} \tag{3}$$

We include a perceptual loss regularizer for $T()$ to preserve the visual quality of the network output $i''_c$. The percep-

| | Pixel 2 & Samsung 2494SJ | Basler acA2040-90uc & Acer S240ML | Logitech c920 & Insignia NS-40D40SNA14 | iPhone 8 & Acer Predator XB271HU | Basler acA1300-30uc & Dell 1707FPt |
|---|---|---|---|---|---|
| LFM without $T()$, frontal | 49.961% | 50.138% | 50.047% | 50.108% | 50.042% |
| LFM with $T()$, 45°(ours) | 29.807% | 15.229% | **10.217%** | 5.1415% | 10.01% |
| LFM with $T()$, frontal (ours) | **10.051%** | **6.5809%** | 10.333% | **5.0732%** | **4.8305%** |

Table 1. BER for various camera-display pairs (*lower is better*). One thousand randomly generated $32 \times 32$ (1024-bit) messages were embedded into one thousand previously unused MSCOCO images. Message recovery was evaluated using 5 cameras and 5 displays. The distances between camera and display range from 23cm to 4.3 meters. The table shows the mean BER for each camera-display pair. While 0% BER would be a perfectly recovered message, 50% BER corresponds to randomly classified bits. Each device was operated with its default manufacturer settings for normal use.

tual loss weight $\lambda_T$ is 0.001. $T()$ is trained for 2 epochs using the Adam optimizer with a learning rate of $0.001$, betas equal to $(0.9, 0.999)$, and no weight decay [48]. Total training time was 7 days.

### 3.3. Training $E()$ and $R()$

The networks $E()$ and $R()$ are trained simultaneously using 123,287 images from MS-COCO [47] for $i_c$, and 123,287 messages for $i_m$. The objective of $E()$ is to produce a coded image $i'_c$ that is visually similar to $i_c$, and encodes all the information from $i_m$ such that it is robust to CDTF distortion. The objective of $R()$ is to recover all information in $i_m$ despite CDTF distortion. The objective functions we wish to minimize are:

$$\begin{aligned} E_{loss} =& L_2(i_c - i'_c) + \\ & \lambda_E * L_1(VGG(i_c) - VGG(i'_c)). \\ R_{loss} =& \phi * L_1(i_m - i'_m) \end{aligned} \tag{4}$$

Again here, we include a perceptual loss regularizer for $E()$ to preserve the visual quality of the network output $i'_c$. The perceptual loss weight $\lambda_E$ is 0.001, and the message weight $\phi = 128$. $E()$ and $R()$ are trained for 3 epochs using the Adam optimizer with a learning rate of $0.001$, betas equal to $(0.9, 0.999)$, and no weight decay [48, 49]. Total training time was 18 hours. The networks $E()$, $R()$, and $T()$ were all trained using PyTorch 0.3.0 with an Nvidia Titan X (Maxwell) compute card [50].

## 4. Experiments and Results

To study the efficacy of our approach, we constructed a benchmark with 1000 images, 1000 messages, and 5 camera-display pairs. The images are from the MSCOCO 2014 test dataset, and each message contained 1024 bits. Two videos were generated, each containing 1000 coded images embedded using a trained LFM network, one trained with $T()$ and one without. As shown in Table 1, the proposed LFM algorithm trained with $T()$ achieved $7.3737\%$ BER, or $92.6263\%$ correctly recovered bits on average for frontally photographed displays. The same algorithm achieved $14.0809\%$ BER when camera and display were aligned at a $45\,\mathrm{deg}$ angle. The example in Figure 5 illustrates the differences between coded images $i'_c$ generated with and without the CDTF network $T()$ in the training pipeline. All BER results in this paper are generated without any error correcting codes or radiometric calibration between cameras and displays.

We wish to understand the effects of perceptual loss in our steganography model. In particular, we examine the effects of $\lambda_T$ by varying its weight in the loss function during training. Figure 6 features an ablation study of the effects of perceptual loss. Figure 8 features an example of the same image and message camera-captured at different angles. The LFM algorithm trained without $T()$ is analogous to digital steganography deep learning techniques, and was unable to successfully recover coded messages even when frontally viewed, the simplest case. Figure 5 illustrates the difference that the inclusion of $T()$ in LFM training makes. Without $T()$, the message is encoded as small per-pixel changes that are near-uniform across the image. With $T()$, the message is encoded as patches where the magnitude of pixel changes varies spatially. We show an empirical sensitivity analysis of camera exposure settings in Figure 7. Our LFM method is robust to overexposure and underexposure, provided pixels are not in saturation.

Finially, we motivate the need for photographic steganography with a comparison to existing methods. Are existing synchronization-free steganography algorithms such as Baluja [2] sufficient for photographic message transfer? As shown in Figure 2, even simple binary messages are not stably transmitted photographically using existing methods. Our CDTF simulation function $T()$ is trained with 25 camera-display pairs, but we want to know how well $T()$ generalizes to new camera-display pairs. Using the 1000-image, 1024-bit test dataset, we test two additional cameras and two additional displays. We create coded images using various embedding algorithms and measure message recovery accuracy for each of the four camera-display pairs. Table 2 shows that LFM trained with $T()$ significantly outperforms existing methods, even when camera and display are at a $45°$ angle.

| | Sony Cybershot DSC-RX100 & Lenovo Thinkpad X1 Carbon 3444-CUU | Sony Cybershot DSC-RX100 & Apple Macbook Pro 13-inch, Early 2011 | Nikon Coolpix S6000 & Lenovo Thinkpad X1 Carbon 3444-CUU | Nikon Coolpix S6000 & Apple Macbook Pro 13-inch, Early 2011 |
|---|---|---|---|---|
| DCT [51], frontal | 50.01% | 50.127% | 50.001% | 49.949% |
| Baluja [2], frontal | 40.372% | 37.152% | 48.497% | 48.827% |
| LFM without $T()$, frontal | 50.059% | 49.948% | 50.0005% | 49.997% |
| LFM with $T()$, 45°(ours) | 12.974% | 15.591% | 27.434% | 25.811% |
| LFM with $T()$, frontal (ours) | **9.1688%** | **7.313%** | **20.454%** | **17.555%** |

Table 2. Generalization to new camera-display pairs: Our LFM model generalizes to new camera and display hardware, outperforming traditional fixed-filter Discrete Cosine Transform (DCT) [51] and deep-learning-based [2] steganography approaches. Here, we show BER for 1000 1024-bit messages transmitted with 4 new camera-display pairs that were not in the training set.

## 5. Conclusion

In this paper, we extend deep learning methods for digital steganography into the *photographic* domain for LFM where coded images are transmitted through light, allowing users to scan televisions and electronic signage with their cameras without an internet connection. This process of *photographic steganography* is more difficult than digital steganography because radiometric effects from the camera-display transfer function (CDTF) drastically alter image appearance [38]. We jointly model these effects as a camera-display transfer function (CDTF) trained with over one million images. The resulting system provided embedded messages that are not detectable to the eye and recoverable with high accuracy.

Our LFM algorithm significantly outperforms existing deep-learning and fixed-filter steganography approaches, yielding the best BER scores for every camera-display combination tested. Our approach is robust to camera exposure settings and camera-display angle, with LFM at $45°$ outperforming all other methods at $0°$ camera-display viewing angles. Along with our LFM algorithm, we introduce Camera-Display 1M, a dataset of 1,000,000 image pairs generated with 25 camera-display pairs. Our contributions open up exciting avenues for new applications and learning-based approaches to photographic steganography.

## 6. Acknowledgements

The authors would like to thank Gradeigh Clark and Professor Thomas Papathomas for our insightful discussions on human perception, Jane Baldwin for generously lending several cameras. We would also like to thank Professor Athena Petropulu for her generous support through a Graduate Assistance in Areas of National Need (GAANN) fellowship. Finally we would like to thank Vishal Patel, Thomas Shyr, Matthew Purri, Jia Xue and Blerta Lindqvist for their time and thoughtful suggestions.

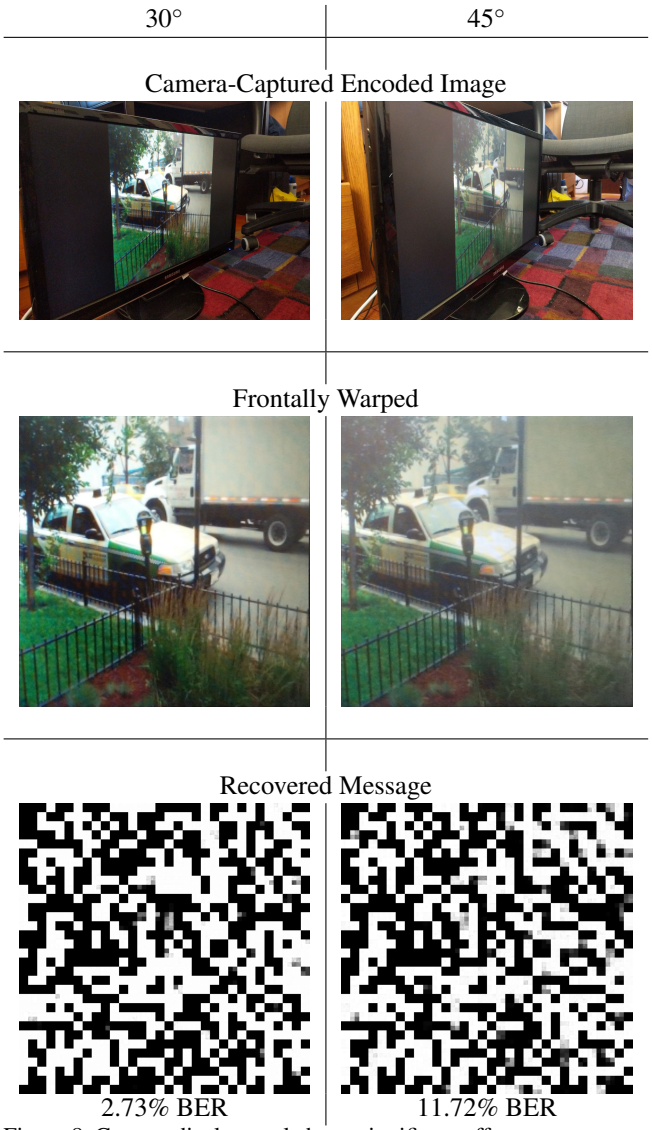


Figure 8. Camera display angle has a significant effect on message recovery. This experiment was performed using the Pixel 2 camera and Samsung 2494SJ display. Our LFM method performs well for oblique views, but experiences a steep dropoff in BER as the camera-display angle increases.

## References


[1] https://github.com/mathski/LFM 1, 6

[2] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Advances in Neural Information Processing Systems*, pp. 2066–2076, 2017. 1, 2, 3, 4, 7, 8

[3] R. Chandramouli and N. Memon, "Analysis of lsb based image steganography techniques," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3, pp. 1019–1022, IEEE, 2001. 1

[4] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal processing*, vol. 90, no. 3, pp. 727–752, 2010. 1

[5] W. Hu, H. Gu, and Q. Pu, "Lightsync: Unsynchronized visual communication over screen-camera links," in *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, MobiCom '13, (New York, NY, USA), pp. 15–26, ACM, 2013. 2

[6] A. Ashok, S. Jain, M. Gruteser, N. Mandayam, W. Yuan, and K. Dana, "Capacity of pervasive camera based communication under perspective distortions," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 112–120, March 2014. 2

[7] S. D. Perli, N. Ahmed, and D. Katabi, "Pixnet: Interference-free wireless links using lcd-camera pairs," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pp. 137–148, ACM, 2010. 2

[8] T. Hao, R. Zhou, and G. Xing, "Cobra: color barcode streaming for smartphone systems," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 85–98, ACM, 2012. 2

[9] P. Mirdehghan, W. Chen, and K. N. Kutulakos, "Optimal structured light à la carte," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6248–6257, 2018. 2

[10] W. Yuan, K. J. Dana, M. Varga, A. Ashok, M. Gruteser, and N. B. Mandayam, "Computer vision methods for visual mimo optical system," *CVPR 2011 WORKSHOPS*, pp. 37–43, 2011. 2

[11] W. Yuan, K. Dana, A. Ashok, M. Gruteser, and N. Mandayam, "Dynamic and invisible messaging for visual mimo," in *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*, pp. 345–352, Jan 2012. 2

[12] A. Wang, C. Peng, O. Zhang, G. Shen, and B. Zeng, "Inframe: Multiflexing full-frame visible communication channel for humans and devices," in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, (New York, NY, USA), pp. 23:1–23:7, ACM, 2014. 2

[13] A. Wang, Z. Li, C. Peng, G. Shen, G. Fang, and B. Zeng, "Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, pp. 181–195, 2015. 2

[14] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou, "Real-time screen-camera communication behind any scene," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, (New York, NY, USA), pp. 197–211, ACM, 2015. 2

[15] E. Wengrowski, K. J. Dana, M. Gruteser, and N. Mandayam, "Reading between the pixels: Photographic steganography for camera display messaging," in *Computational Photography (ICCP), 2017 IEEE International Conference on*, pp. 1–11, IEEE, 2017. 2, 6

[16] K. Jo, M. Gupta, and S. K. Nayar, "Disco: Display-camera communication using rolling shutter sensors," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 5, p. 150, 2016. 2, 6

[17] V. Nguyen, Y. Tang, A. Ashok, M. Gruteser, K. Dana, W. Hu, E. Wengrowski, and N. Mandayam, "High-rate flicker-free screen-camera communication with spatially adaptive embedding," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pp. 1–9, IEEE, 2016. 2, 6

[18] T. Morkel, J. H. Eloff, and M. S. Olivier, "An overview of image steganography.," in *ISSA*, pp. 1–11, 2005. 3

[19] J. Fridrich and M. Goljan, "Practical steganalysis of digital images: state of the art," in *Security and Watermarking of Multimedia Contents IV*, vol. 4675, pp. 1–14, International Society for Optics and Photonics, 2002. 3

[20] H. Wang and S. Wang, "Cyber warfare: steganography vs. steganalysis," *Communications of the ACM*, vol. 47, no. 10, pp. 76–82, 2004. 3

[21] R. Chandramouli, M. Kharrazi, and N. Memon, "Image steganography and steganalysis: Concepts and practice," in *International Workshop on Digital Watermarking*, pp. 35–49, Springer, 2003. 3

[22] L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Spread spectrum image steganography," *IEEE Transactions on image processing*, vol. 8, no. 8, pp. 1075–1083, 1999. 3

[23] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, 1998. 3

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015. 4, 5

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks.," in *CVPR*, vol. 1, p. 3, 2017. 4, 5

[26] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, vol. 2, 2015. 4, 5

[27] L. Pibre, P. Jérôme, D. Ienco, and M. Chaumont, "Deep learning for steganalysis is better than a rich model with an ensemble classifier, and is natively robust to the cover source-mismatch. arxiv preprint," *arXiv preprint arXiv:1511.04855*, 2015. 3

[28] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont, "Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch," *Electronic Imaging*, vol. 2016, no. 8, pp. 1–11, 2016. 3

[29] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Media Watermarking, Security, and Forensics 2015*, vol. 9409, p. 94090J, International Society for Optics and Photonics, 2015. 3

[30] I. Khan, B. Verma, V. K. Chaudhari, and I. Khan, "Neural network based steganography algorithm for still images," in *Emerging Trends in Robotics and Communication Technologies (INTERACT), 2010 International Conference on*, pp. 46–51, IEEE, 2010. 3

[31] S. Husien and H. Badi, "Artificial neural network for steganography," *Neural Computing and Applications*, vol. 26, no. 1, pp. 111–116, 2015. 3

[32] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Advances in Neural Information Processing Systems*, pp. 1951–1960, 2017. 4

[33] X. Weng, Y. Li, L. Chi, and Y. Mu, "Convolutional video steganography with temporal residual modeling," *arXiv preprint arXiv:1806.02941*, 2018. 4

[34] P. Wu, Y. Yang, and X. Li, "Stegnet: Mega image steganography capacity with deep convolutional network," *arXiv preprint arXiv:1806.06357*, 2018. 4

[35] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," *arXiv preprint arXiv:1807.09937*, 2018. 4

[36] R. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster r-cnn," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 1–1, 2018. 4

[37] S. Dong, R. Zhang, and J. Liu, "Invisible steganography via generative adversarial network," *arXiv preprint arXiv:1807.08571*, 2018. 4

[38] E. Wengrowski, W. Yuan, K. J. Dana, A. Ashok, M. Gruteser, and N. Mandayam, "Optimal radiometric calibration for camera-display communication," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pp. 1–10, IEEE, 2016. 4, 5, 6, 8

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 5

[40] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, pp. 694–711, Springer, 2016. 6

[41] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network.," in *CVPR*, vol. 2, p. 4, 2017. 6

[42] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA*, pp. 6228–6237, 2018. 6

[43] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 6

[44] H. Talebi and P. Milanfar, "Learned perceptual image enhancement," in *Computational Photography (ICCP), 2018 IEEE International Conference on*, pp. 1–13, IEEE, 2018. 6

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 6

[46] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016. 6

[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014. 6, 7

[48] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, vol. 5, 2015. 7

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 7

[50] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017. 7

[51] S. Kamya, "Watermark dct." https://www.mathworks.com/matlabcentral/fileexchange/46866-watermark-dct, 2014. 8