Final Project Proposal

Oree Leibowitz
Roey Magen

1 Problem statement

Digital watermarking is the process of embedding information into an image (or other types of multimedia) such that it can survive under distortions, while requiring the encoded image to have little or no perceptual difference from the original image. We can model the problem in cryptographic terms as three parties: Alice wishes to encode a fingerprint in an image. Eve will then somehow distort the image (by cropping, blurring, etc), and Bob should be able to detect the fingerprint in the distorted image. Digital watermarking can be used to identify image ownership: It can be used to identify an image that posted online even if the posted version is modified. In the last two decades, watermarking technology has been applied to protect multimedia documents in order to prove image ownership as a form of copyright protection.

One motivation for considering neural networks for this task is the phenomena of adversarial examples (can be shown as a "bright side of adversarial examples"). While the existence of adversarial examples is usually seen as a disadvantage of neural networks, it can be desirable for watermarking: if a network can be fooled with small perturbations into making incorrect class predictions, it should be possible to extract meaningful information from similar perturbations. Furthermore, the adversarial nature of these generated examples is preserved under a variety of image transformations. [1].

2 Related work

A wide variety of watermarking settings and methods have been proposed in the literature. Some watermarking methods encode information in the least significant bits of image pixels [3]. For more robust encoding other methods proposed to encode information in the frequency domain [4].

Also neural networks have been used for watermarking. Old work used neural networks for one stage of a larger pipeline, such as determining watermarking strength per image region, or as part of the encoder or the decoder. Later work modeled the entire data hiding pipeline with neural networks and train them end-to-end. The networks are encoder-decoder networks, where given an input message and cover image, the encoder produces a visually indistinguishable encoded image. This image go over noise layers and the decoder have to decode the noisy image to recover the original message. Zhu at el. [2] showed encoder-decoder network based on convolutions layers. They trained the network to be robust to noise using different noise layers, such as Gaussian blurring, pixel-wise dropout, cropping, and JPEG. Later work influenced by [2] done by Xiyang at el. [5] used adversarial training to generate learnable noise layers in order to generate Watermarking that is more distortion agnostic.

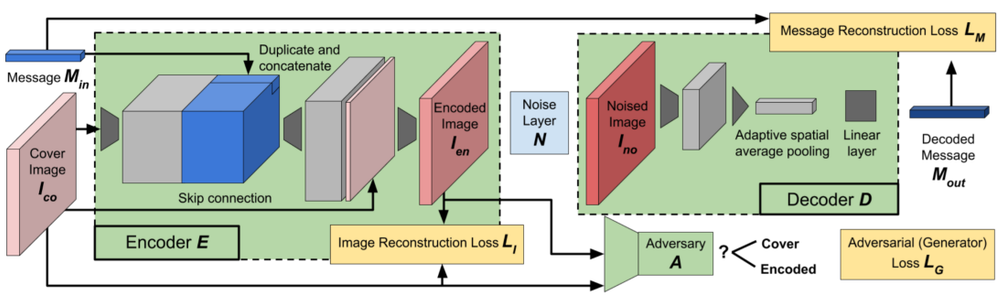

Figure 1: The network architecture proposed in [2]. Note that this architecture includes also a discriminator in order to tackle also the Steganography problem.

3 Proposed method

Our goal is to develop a learnable, end-to-end model for image watermarking that can be made robust for a large extent to arbitrary types of image distortions. Nowadays we know that self-attention mechanisms can draw global dependencies between pixels in the image and between inputs and outputs. We suggest that treating each image pixel differently can improve the robustness of the data to distortions.

The neglect of considering the sensitivity of each pixel will inevitably affect the model robustness for information hiding.

Our suggestion is to use an encoder-decoder architecture influenced by [2], but instead of convolutions layers we use self-attention blocks. Hopefully our model will be able to learn a pattern of pixels that supply robustness for the Watermarking.

If time allows, we might want to combine the construction from [5] and use adversarial training in order to get distortion agnostic robustness.

One more problem we might want to tackle is to train the decoder to decide whether an image encodes a watermark at all. The model from [2] as well as other models train the decoder to decode a watermark, assuming the image is watermarked. But the training didn't include any images without watermark. We believe that in real world scenarios it is important to distinguish between the cases.

4 Possible pitfalls

We consider the following possible pitfalls:

- **Resources.** The original implementation of [2] is trained on 10,000 COCO images and evaluated on 1,000. We believe that training on that amount of data will take too long. In addition, our second stage (involving the architecture from [5]) involves training a GAN which is very resource demanding.
- **Achieving the results of the paper.** We found a non-formal implementation of the architecture from [2], which we don't know if will work and achieve the results of the paper. We also found other implementations for later work which we can adapt to implement the architecture of [2].
- **Training on small amount of data.** In order to solve the resources problem, we might want to train the model on small amount of cover images. That might cause the model to learn the specific cover images we train on. Therefore we need to pay attention to over-fitting through the learning process.
- **Evaluation.** The original paper [2] compared itself to a closed-source tool for image watermarking named Digmarc. The tool provides no information about its bit error rate, which makes comparing with our model difficult.

It is important to mention that we will try to train the [2] architecture with similar amount of data in order to make a fair comparison.

5 Data

Microsoft COCO: Common objects in context. a large-scale object detection, segmentation, and captioning dataset. All of the papers that we saw in context of digital watermark used this data set. We will use a small subset of the dataset.

6 Planned evaluation

There are several parameters of comparison between methods:

- **Capacity** The size in bits of the watermark per image bit.
- **Robustness** The extent of robustness to image distortions. Can be evaluated using bit accuracy: The hamming distance between the encoded message and the decoded message.
- **Perceptibility** The distrotion between the cover image and the watermarked image. We consider the PSNR (peak signal-to-noise ration) distance to measure it.

We plan to fix the capacity between the compared methods, and compare the robustness and perceptibility gained by each of the methods. We plan to compare our method to [2] and to the (previously mentioned) tool Digimarc.

References

[1] Kurakin, A., Goodfellow, I., Bengio, S. *Adversarial examples in the physical world* In: ICLR Workshop. (2017)

[2] Jiren Zhu, Russell Kaplan, Justin Johnson and Li Fei-Fei. *HiDDeN: Hiding Data With Deep Networks.* ECVV (2018)

[3] Van Schyndel, R.G., Tirkel, A.Z., Osborne, C.F. *A digital watermark. In: IEEE Converence on Image Processing, 1994, IEEE (1994)*

[4] Bi, N., Sun, Q., Huang, D., Yang, Z., Huang, J. *Robust image watermarking based on multiband wavelets and empirical mode decomposition.* IEEE Transactions on Image Processing (2007)

[5] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. *Distortion Agnostic Deep Watermarking.* Google Research, Stanford University (2020)

[6] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P., Zitnick, C.L.: *Microsoft COCO: Common objects in context.* In: ECCV. (2014)