CS 536 Project:
Watermark Removal

Alexander Saff, Alon Flor, Kaushik Vakadkar

Team members: Alexander Saff (ars369), Alon Flor (af656), Kaushik Vakadkar (kav126)

What to solve. The goal of this project is to automatically remove watermarks from images. Watermarks are a common tool used to prevent unlicensed distribution or copying of intellectual property. In this instance, the watermarks are generally logos, text, or both placed over the image. Watermarks may be somewhat transparent or fully opaque, although if they are fully opaque they will usually be somewhat smaller than their translucent counterparts. Watermarks are usually either identifiable because the text and/or logo is made by darkening or lightening the parts of the image underneath the logo/text. Very rarely are translucent watermarks formed by tinting the underlying image to a color other than black or white. While there have been previous attempts to create automated watermark removal tools, our approach is unique because it is the first attempt that we are aware of to make use of transformer blocks. Previous attempts have either not used neural networks at all, utilized convolutional neural networks, or used generative adversarial networks (GANs). Recently transformers have been shown to match CNN performance on vision tasks with better efficiency. Most notable is Google's recent ViT[1].

How to solve. Our proposed solution is to create a neural network that takes in images and outputs the same image with the watermark removed using a transformer architecture. Images would be sourced from the CIFAR10 dataset, and watermarks would be applied to the images for training and testing purposes. The images with watermarks applied will serve as input data, while the original pictures sans watermark will serve as the ground-truth target output. Since the watermark size, location, rotation, and contents are randomly applied to each image, we are able to augment the CIFAR10 dataset to nearly limitless size, since the number of watermark-image combinations are very high. There has been some work around pretrained computer vision transformer networks that may allow for higher accuracy while keeping training time and computational load at a reasonable level.

How to evaluate. The model will be evaluated on how closely its output images match the ground truth on image-watermark combinations that it (the model) has not seen before. Ideally, images are scored only on the difference between pixel values in the region affected by the watermark. The images will be sourced from the CIFAR10 dataset, with rotations and watermarks applied to the images as stated above. Pixel values should be averaged across RGB channels and then the weighted average applied over all images according to image class. This will prevent the network from specialising on only one class of the CIFAR10 dataset. To prevent overfitting and improve generalization, we will use an 80-10-10 split for training, validation and final evaluation/testing respectively. In addition to data augmentation, we even propose to make use of k-fold cross-validation.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020.