ECE 471: Exploiting Watermark Consistency

Joshua Hartmann
Department of Software Engineering
University of Victoria
Victoria, Canada
jchart@uvic.ca

Sukhdip Sandhu
Department of Software Engineering
University of Victoria
Victoria, Canada
sukhdips@uvic.ca

***Abstract*—“On the Effectiveness of Visible Watermarks” by Tali Dekal, Michael Rubinstein, Ce Liu, and William T. Freeman [1] is a paper that outlines an exploitation in current visible watermarking schemes in which the original image can be recovered with high accuracy. This approach capitalizes on the consistent nature in which watermarks are applied and automatically estimates the foreground (watermark), its alpha matte, and the background (original image). Dekal et al. further demonstrate techniques to strengthen watermarks, raising awareness to content providers to design watermarks not only robust to single image alterations, but also to mass scale removal. Our paper aims to replicate the algorithm proposed and evaluate it both quantitatively, from a collection of downsampled ImageNet [2] images in which we will watermark so that the ground truths are available, and qualitatively on the papers provided supplementary fotolia [3] dataset.**

I. Introduction

Watermarking is a technique that has been around since the 1200s [4], [5]. In its early years, it proved useful for identification and tracking. As time progressed, and especially when banknotes became more common, watermarks became synonymous with security to detect counterfeits. In the current digital age, watermarks are used extensively by content providers to protect their digital content online.

When designing a watermark, the creator has to find a compromise between how obstructive the watermark is and how secure they want their content. Trivially, a full opaque object in front of an image will provide the greatest security, but comes at the cost of obstructing the whole image. On the other hand, a simple transparent object makes the underlying image more visible but is easier to edit out using image editing software. [1] Typically, a good watermark will involve overlaying a semi-transparent image over the original image and will contain complex patterns such as thin lines and shadows to increase the time and complexity of manual or automated removal.

The authors [1] showed that the watermark removal process can be automated by leveraging the fact that stock image companies will use the same watermark across a large sets of their images. The consistent manner in which watermarks are applied provides ample information to estimate the watermark and reverse the process to output the original image. Alongside their findings, the researchers explored and released countermeasures to their attack. These suggested countermeasures include slight modifications to per-image watermarks such as warping and distorting. The importance of this research is to ultimately help keep copyrighted contented protected online, with the researchers hoping their findings will inspire development of more advanced watermarking techniques.

II. Literary Review

There have been many attempts to remove watermarks from images due to the high volume of stock content online and users wanting to avoid paying for content. Due to this, one can find an abundance of tutorials online demonstrating watermark removal using commercial software and techniques such as inpainting. However, these methods require extensive manual work, and are limited both in terms of effectiveness, and scalability.

As visible watermarking plays an important role in protecting image copyrights, researchers have both explored effective watermark design, and attack methods with the aim of creating more vigorous watermarks.

Braudaway et al. [6] were some of the early pioneers in the introduction of visible watermarks to digital images in 1996, (*for reference, the first image published on the internet was in 1992*) where they designed an adaptive non-linear pixel-domain technique to add a watermark as a method of identifying ownership. Since then, researchers have designed increasingly more complicated watermarking techniques including Meng and Cheng [7] who brought about watermarking to the discrete cosine transform (DCT) domain, and Hu and Kwong [8] who implemented adaptive watermarking in the wavelet domain to combat any artifacts brought about in the DCT domain. Despite the research backed methods, most content providers continue to use a simple additive model to apply watermarks. This additive model is the focus of this report.

Past attack methods include Pei and Zeng [9] who proposed to use Independent Component Analysis (ICA) to separate the source image from the watermark. The results of this study demonstrate that the proposed algorithm can blindly and successfully remove the visible watermarks without knowing the watermarking methods in advance. However, this approach suffers from time-consuming human intervention, requiring the user to manually mark the watermark area on individual

[1] *Photoshop*. Adobe. Available:
https://www.adobe.com/ca/products/photoshop.html

images. Huang and Wu [10] explored the use of classic image inpainting methods to fill in the watermarked regions, but again, suffer the same drawbacks. In addition, both methods perform poorly for large watermarked images.

To work around the impracticalities of manual watermark detection, Santoyo-Garcia et al. [11] proposed an automatic watermark detection algorithm where the watermarked image is decomposed into a structure image and texture images using Total Variation L1 (TV-L1) method, in which the watermarked area can be automatically detected. This method can be used in conjunction with the conventional attacks [9], [10] to make them more efficient. The results illustrated by Santoyo-Garcia et al. are only for an embedded binary pattern, as they operated under the assumption that "watermark pattern[s are] generally flat and devoid of texture" so this approach will not work well, except in the most trivial of cases.

The rest of the report will contain details of our implementation, as well as a thorough evaluation of the algorithm both quantitatively and qualitatively.

III. Simplifications and Assumptions

There are a few simplifications and assumptions we made to make the work more feasible. For one, we opted to work with grayscale images as they are inherently less complex than color images. Moreover, we designed and chose our dataset so that it contains watermarks in the exact same location across every image. The paper operates on color images, and has an additional step of locating the watermark anywhere in the image. In addition, we opted to focus on just the watermark removal algorithm, as opposed to tackling datasets where the watermark has been inconsistently applied, which the paper discusses.

IV. Proposed Approach

A watermark W is typically applied to an image I, at some spatially varying opacity (alpha-matte) α, to produce a watermarked image J in the following way:

$$J = W\alpha + (1-\alpha)I. \tag{1}$$

The model of Eq. 1 for our watermarked image provides us with a trivial solution for reversing the process to recover the original image.

$$I = \frac{J - W\alpha}{(1-\alpha)} \tag{2}$$

Therefore, to extract the original image, we need to estimate an alpha-matte $\hat{\alpha}$ and watermark $\hat{W}$ (Eq. 2). This leads us to the proposed algorithm which is split into two processes: an initial watermark detection and estimation step, and an optimization.

Watermark Detection and Initial Estimation

In this step, we provide initial estimations of the watermark $\hat{W}_{init}$ and opacity $\hat{\alpha}_{init}$ to be used in the optimization. First, the alpha-matte and watermark are estimated by detecting the outline and the location of the watermark followed by a reconstruction of the watermark, using Poisson reconstruction, and a tuning of $\hat{\alpha}_{init}$.

Our first step is to find the outline and location of the watermark that has been consistently applied to our set of images J_k. To do this, we compute the magnitude of the median of gradients in the x and y directions across all N images (Eq. 3).

$$\nabla\hat{W}_{init} = \sqrt{\underset{\forall k \in N}{median}(\nabla_x J_k)^2 + \underset{\forall k \in N}{median}(\nabla_y J_k)^2} \tag{3}$$

Because we see the same gradients bordering the watermark on each image, $\nabla\hat{W}_{init}$ represents an estimate of the watermark gradients. All other gradients tend towards 0 because, across all images, they are distributed with a median of 0. Using a binary threshold on these gradients gives us an approximate outline of the watermark (Fig. 1 (a)).

The median of the gradients in the x and y direction are then used as input to Poisson reconstruction. The Authors [1] provided no details regarding implementation, therefore, we consulted open source code, implementing a modified solution from R. Jena, a senior undergrad at the Indian Institute of Technology in Bombay who has published his code freely for academic/research purposes [14]. The Poisson reconstruction process gives us our initial estimated watermark $\hat{W}_{init}$ (see Fig. 1(b)) and is used to estimate the alpha-matte which is refined in the following steps.

The alpha-matte can be broken down into two components, $\alpha = c\alpha_c$. The matte α_c is the mask where the blend-factor c is applied. The matte is estimated by taking the median over all the single image mattes which is computed by using a binary threshold on $\hat{W}_{init}$ (Fig. 1(c)) and running the single image matting algorithm adapted from Marco Forte [12] following A. Levin et al. [13].

The blend-factor is estimated by taking advantage of dark regions in the images. In these regions there is no image component to our model, therefore, $J = \alpha W = \alpha_c c W$. Using the $\hat{W}_{init}$ and α_c found previously, and the dark regions, the blend-factor c is estimated. Our estimated alpha-matte is then $\hat{\alpha}_{init} = \alpha_c c$, as seen in Fig. 1(d). The algorithm for this estimation was adapted from R. Jena [14].

Using the initial estimates for $\hat{W}_{init}$, α_c and c, for a baseline, we considered the image reconstructions obtained using a direct per-pixel subtraction from the initial watermark and initial alpha-matte estimations to obtain initial results for the reconstructed image (Eq. 4).

$$\hat{I}_k = \hat{J}_k - (\hat{W}_{init} * \hat{\alpha}_{init}). \tag{4}$$

Optimization

Because minute deviations from the actual watermark cause noticeable image artefacts, our estimates for the alpha-matte and watermark need to be optimized. The optimization problem alternates between optimizing with respect to the watermark and the alpha-matte to find estimates of the alpha-matte $\hat{\alpha}$ and watermark $\hat{W}$. The optimization problem (Eq. 5) is defined as:

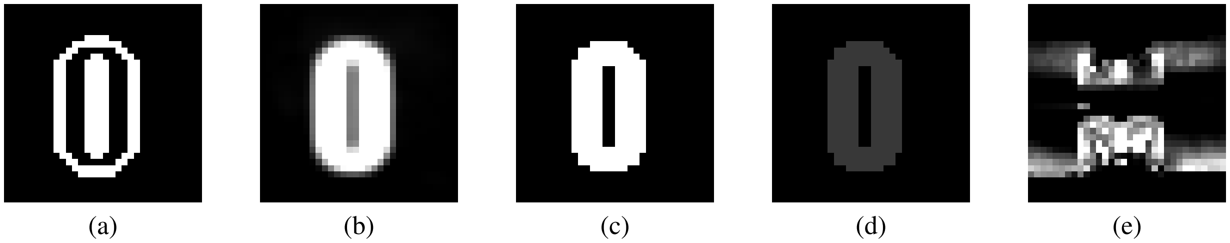


Fig. 1. The watermark outline estimate (a), Poisson reconstruction (b), alpha-matte (c), alpha-matte with blend factor (d), and optimized watermark*alpha*blend-factor (intensity increased for visualization) (e)

$$argmin \sum_k \Big(E_{data}(\hat{W}_k, \hat{\alpha}, \hat{I}_k) + \lambda_{\hat{I}} E_{reg}(\nabla \hat{I}_k) + \lambda_w E_{reg}(\nabla \hat{W}_k) + \lambda_{\hat{\alpha}} E_{reg}(\nabla \hat{\alpha}) + \beta E_f(\nabla(\hat{\alpha}\hat{W}_k))\Big) + \gamma \sum_k \Big(E_{aux}(\hat{W}, \hat{W}_k)\Big). \quad (5)$$

where $E_{data}(\hat{W}_k, \hat{\alpha}, \hat{I}_k)$ is the distance between the model $\hat{J} = \hat{\alpha_k}\hat{W}_k - (1 - \hat{\alpha_k})\hat{I}_k$ and the actual watermarked image J_k. The E_{reg} terms are regularization terms. $E_f(\nabla(\hat{\alpha}\hat{W}_k))$ is the squared error between the gradients of the estimated watermark and the watermark found in the watermark detection step $||\hat{W} - \hat{W}||^2$, this term ensures a single global minimum. The final term, $\sum_k(E_{aux}(\hat{W}, \hat{W}_k))$, is the absolute difference between the current best global watermark estimate and the optimized per-image estimate $|\hat{W} - \hat{W}_k|$.

The first step in the optimization is to optimize with respect to $\hat{W}_k$ and $\hat{I}_k$ on a per-image basis keeping $\hat{\alpha}$ constant. Next, a new estimated global watermark is taken to be the median of the per-image watermarks $\hat{W} = \underset{\forall k \in N}{median}(\hat{W}_k)$. Finally, the optimization problem is optimized with respect to $\hat{\alpha}$ (Fig. 1(d)).

The linear systems for optimization are derived in the papers Supplementary Material (SM).

Watermark Removal

The watermarks are then removed from the image by inverting our watermark model (Eq. 5), with the refined $\hat{W}$ and $\hat{\alpha}$ calculated during optimization.

$$\hat{I}_k = \frac{J_k - \hat{W}\hat{\alpha}}{(1 - \hat{\alpha})} \quad (6)$$

V. Evaluation & Datasets

Dataset

For evaluation, two datasets were used, a collection of downsampled ImageNet photos [2] (no apparent class) and a fotolia dataset in which the the papers SM provided [3].

The subset of ImageNet data contains a random selection of 1000 64x64px images. For each of these images, we further cropped them to be size 32x32px, due to the time and space complexity of the optimization stage (*not enough system memory to calculate*). Next, we overlaid a semi-transparent simple text-based object to simulate a consistent watermark across all images, with the same properties (size, location, opacity, rotation, etc). This watermark is overall flat, with some shadowing around the edges of the watermark. Since we have the ground truths, quantitative evaluation was then performed on our reconstructed watermark-removed images, and their true value.

Qualitative evaluation was performed on the 50 fotolia images provided directly from the papers SM [3]. Our motive behind choosing this dataset was, alongside the original watermarked images, the authors [2] provided their reconstructed results. This allows for direct visual comparison between our method, and what was expected. Note, to reduce complexity, we cropped each of the fotolia images centered around their watermark to dimensions 176x176px.

Quantitative Evaluation

To quantitatively evaluate our results, two well known metrics were determined which are shown to capture perceptible image degradations: Peak-Signal-to-Noise-Ratio (PSNR) and Structural dissimilarity Image Index (DSSIM). PSNR (Eq. 7) defined as:

$$PSNR = 10log_{10}(\frac{MAX^2}{MSE}) \quad (7)$$

where *MAX* is the maximum pixel value and *MSE* is the Mean Squared Error, which is the mean of the square of differences of pixel intensities between the ground truth and our reconstruction. A perfect reconstruction yields PSNR of infinity. DSSIM (Eq. 8) between the reconstructed image $\hat{I}_k$ and the ground truth image I_k is defined as:

$$DSSIM(\hat{I}_k, I_k) = \frac{1}{2}(1 - SSIM(\hat{I}_k, I_k)) \quad (8)$$

where $SSIM(x, y)$ is Structural Similarity [16]. A perfect reconstruction yields DSSIM of 0. The PSNR and DSSIM is calculated for each image and the mean is taken over the entire collection of images.

Table 1 contains Dekel et al. results for reference. It is worth noting, their evaluation was performed on a random subset of 1000 color images from the Microsoft COCO 'val2014 dataset [17]'. Table 2 contains the results of our implementation, for both direct subtraction and invertion of the watermark, both before and after optimization. Please note, the ground truths for the fotolia dataset are the reconstructions of the authors, therefore not *true* ground truths, albeit helpful in quantifying

TABLE I
Reconstruction quality and comparison of Authors results

Dekel et al.	Copyrights-fixed		CVPR2017-fixed	
	PSNR	DSSIM	PSNR	DSSIM
Attack	36.2	0.038	32.73	0.070
Direct Sub.	30.89	0.080	30.65	0.085

TABLE II
PSNR and DSSIM of our results for invertion(Inv) and subtraction (Sub) both before (B) and after (A) optimization on various datasets.

Ours	ImageNet(32x32)		ImageNet(64x64)		Fotolia(200x200)	
	PSNR	DSSIM	PSNR	DSSIM	PSNR	DSSIM
Inv. (B)	27.73	0.045	29.69	0.027	26.97	0.042
Inv. (A)	23.23	0.09	/	/	/	/
Sub. (B)	29.9	0.036	30.08	0.026	30.2	0.031
Sub. (A)	23.23	0.091	/	/	/	/

a comparison. Although we were unable to test our algorithm on the exact same dataset, we can compare how our results held up relative to the papers results on our dataset.

Our results for direct subtraction before optimization were in-line with what was expected, both visually, and quantitatively. Dekel et al. had a PSNR $\approx$ 30.77 and DSSIM $\approx$ 0.083 averaged across their datasets for direct subtraction, which is within a reasonable range of error from our direct subtraction results. This is about as close as our results got to replicating the papers. The inverting and optimization operators underperformed when compared to Dekel et al. results. Where they experienced a substantial improvement after optimization and inverting the image (PSNR 12% increase, DSSIM 65% decrease) our results got worse (PSNR 22% decrease, DSSIM 250% increase). This indicates errors in our implementation.

Qualitative Evaluation

Qualitative evaluation was performed on the fotolia dataset provided in the SM. The authors [1] provide both the watermarked image, and their reconstructed image, which allows for a direct comparison between our respective outputs. Due to the time and space complexity of optimization, only the subtraction and inversion results for pre-optimization were calculated for the fotolia dataset. Illustrated in Fig. 2 are good and poor results for the unoptimized direct subtraction. Figures 2(a,d) contain the original watermarked version, (b,e) is our reconstruction using unoptimized direct subtraction, and (c,f) contains Dekel et al. reconstruction. The top row in Fig. 2 highlight the reconstructed image that yielded the best PSNR value, and the bottom row yielded the worst. While analysing the dataset, it is clear that those images with backgrounds similar in intensity to the watermark give better results than those with higher contrast difference. Meaning, there is very little validity in direct subtraction as it can perform great for one image, but poor for another despite having the exact same watermark applied.

Figure 3 shows a side-by-side visual comparison of direct subtraction reconstructions vs. inverted reconstructions. When comparing the two methods to the desired reconstruction (Fig. 3 (b)), it is shown subtraction (Fig. 3(c)) outperforms invertion (Fig. 3 (d)). This is because the initial estimates for the watermark, alpha matte, and blend factor are not sufficient to reverse the process, hence, the need for optimization to combat visible artifacts.

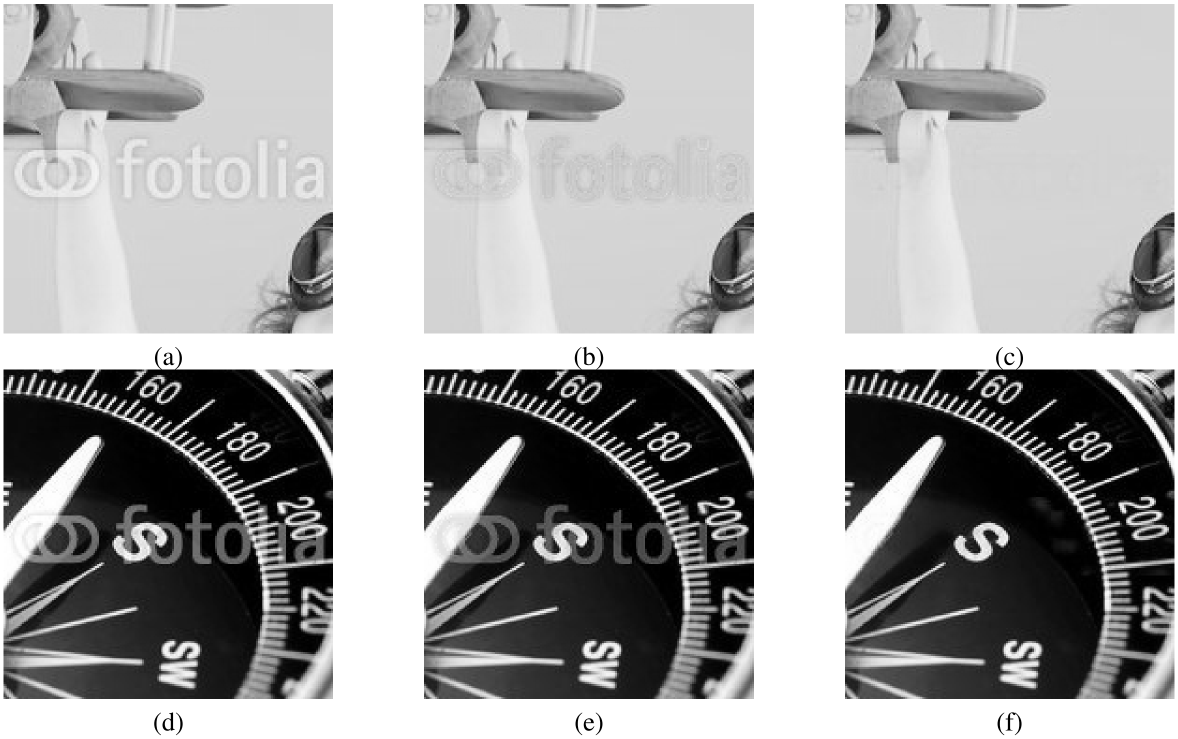
To explore the role optimization plays in the invertion stage, we generated images on the smaller 32x32 ImageNet dataset. Figure 4 shows the results of inverting the watermark before optimization (c), and after optimization (d). The before optimization results still perform better, further indicating error in the optimization part of our algorithm.

VI. Conclusion

Tali Dekal, Michael Rubinstein, Ce Liu, and William T. Freeman's paper "On the Effectiveness of Visible Watermarks" [1] revealed a loophole in the way visible watermarks are used online, which had thus far been overlooked. Their attack exploits the consistency in which watermarks are applied and effectively takes a large collection of watermarked images and automatically recovers the watermark, its alpha-matte, and the original image with high accuracy.

Despite the simplifications and assumptions we made for development, replicating the algorithm proved unsuccessful. The paper provided very little detail regarding implementation, and when it did, it was often convoluted or required a much deeper understanding of computer vision and linear algebra.

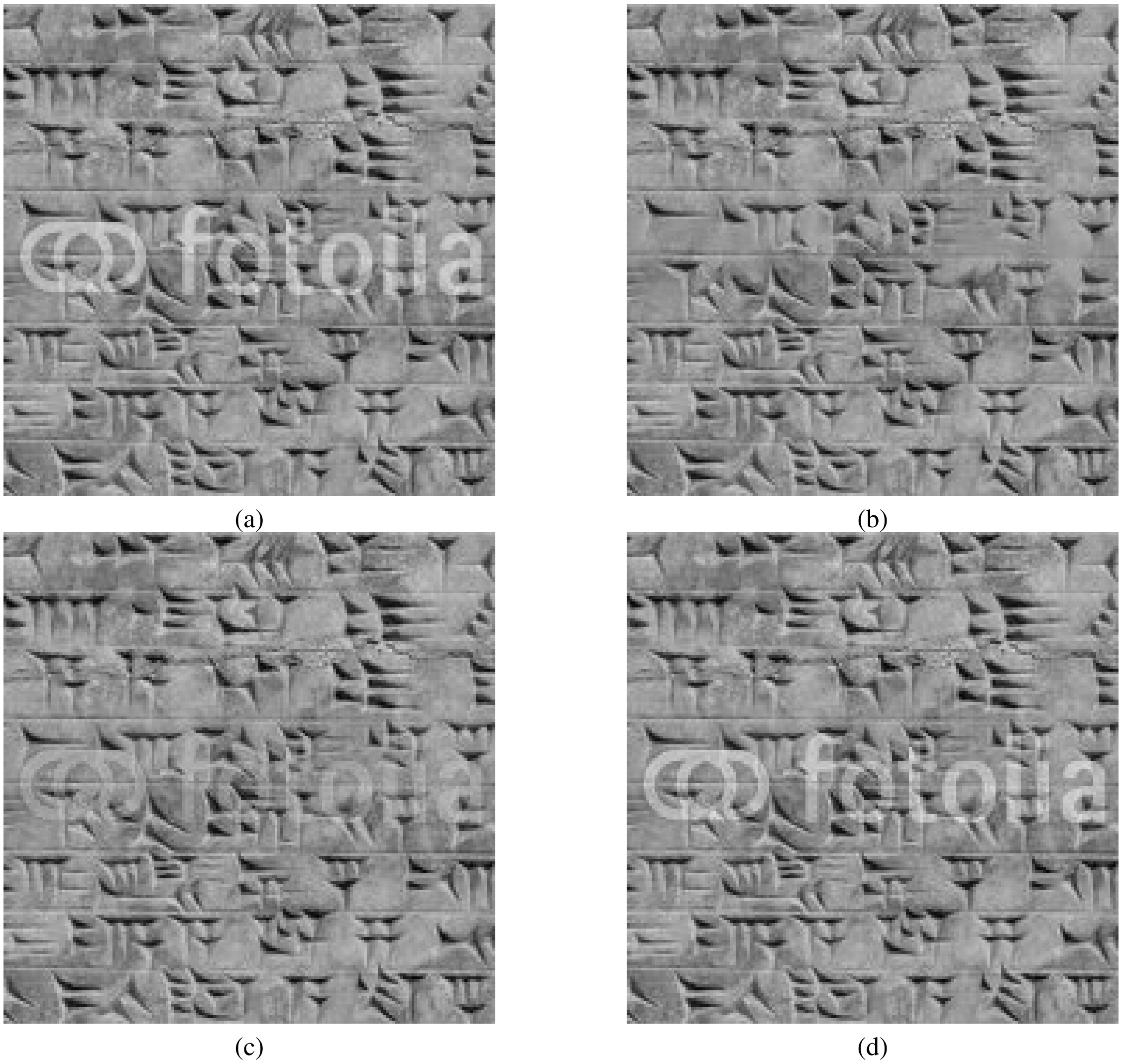
Although the overall algorithm did not perform as expected, this project was still very much a success. The intended purpose was to give real-world experience in solving a problem using computer vision techniques. We both applied knowledge learned in the classroom, and learned to decipher other researchers work. These new skills will prove to be beneficial moving forward in our software engineering careers.



(a) (b) (c)

(d) (e) (f)

Fig. 2. Good pre-optimized watermark subtraction results: Original watermarked image (a), our watermark subtraction reconstruction (b), Dekel et al. reconstruction(c). Poor pre-optimized watermark subtraction results: Original watermarked image (d), our watermark subtraction reconstruction (e), Dekel et al. reconstruction(f)



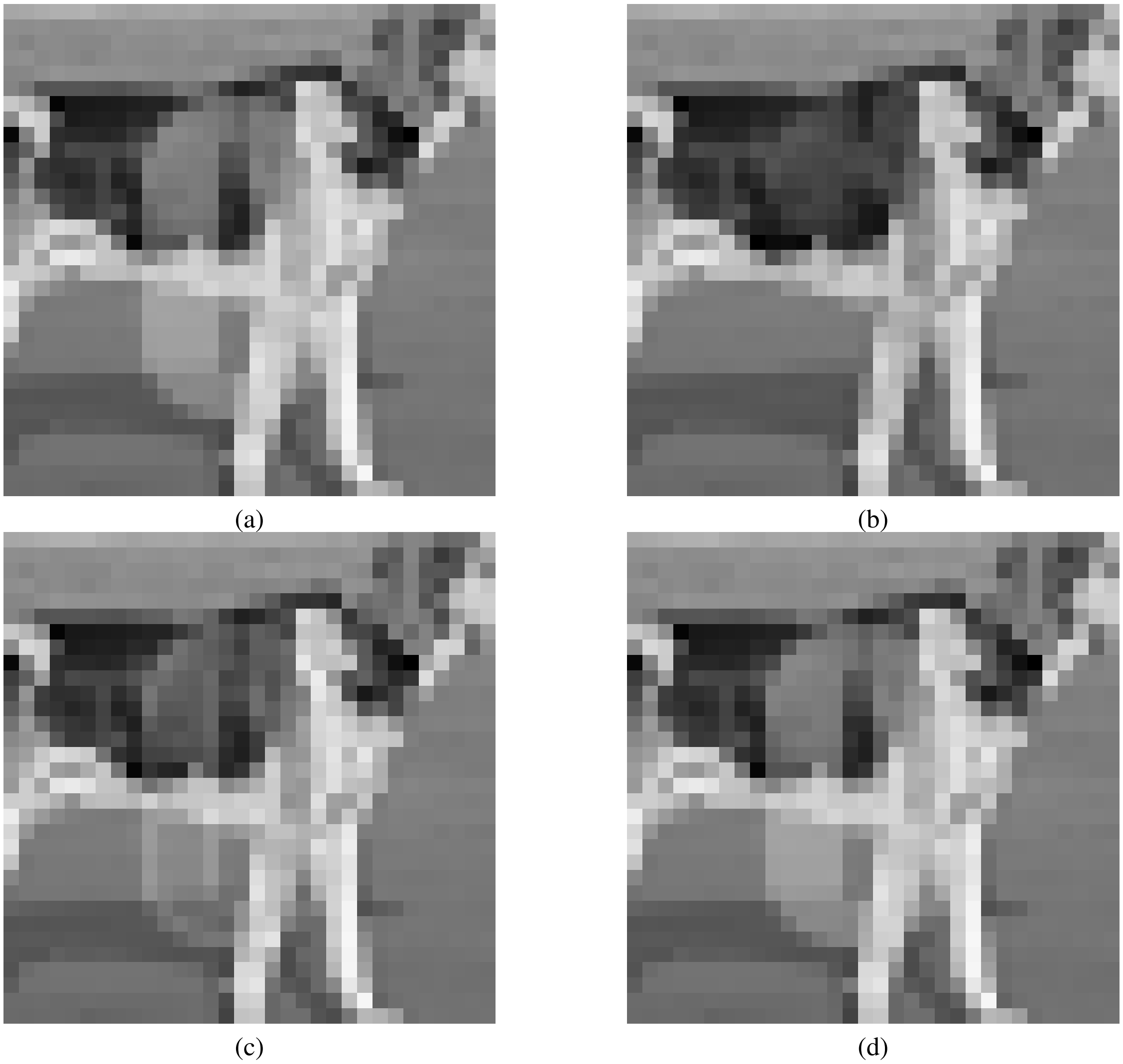
(a)

(b)

(c)

(d)

Fig. 3. Comparison of direct subtraction vs. image inverting: Original watermarked image (a), Dekel et al. reconstructed image (b), our reconstruction using direct subtraction (c), our reconstruction using invertion (d)



(a)

(b)

(c)

(d)

Fig. 4. Comparison of before and after optimization inverting: Original watermarked image (a), ground truth reconstructed image (b), our reconstruction invertion before optimization (c), our reconstruction invertion after optimization (d)

References

[1] T. Dekel and M. Rubinstein and C. Liu and W. T. Freeman, ”On the Effectiveness of Visible Watermarks”, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/papers/Dekel_On_the_Effectiveness_CVPR_2017_paper.pdf . [Accessed Feb 26, 2019]

[2] ”ILSVRC2015”, Image-net.org, 2019. [Online]. Available: http://image-net.org/small/download.php. [Accessed: 27- Feb- 2019]

[3] ”fotolia: Input Images”, Watermark-cvpr17.github.io, 2019. [Online]. Available: https://watermark-cvpr17.github.io/supplemental/fotolia/input.html. [Accessed: 14- Apr- 2019].

[4] E. Manus ”Watermarks: An Appreciation for a Timeless Feature of Currency” Sept 2013. [Online]. Available: https://www.americanbanker.com/news/watermarks-an-appreciation-for-a-timeless-feature-of-currency . [Accessed Feb 26, 2019]

[5] ”Watermark”, [Online]. Available: https://en.wikipedia.org/wiki/Watermark . [Accessed Feb 26, 2019]

[6] G. W. Braudaway, K. A. Magerlein, and F. C. Mintzer. Protecting publicly available images with a visible image watermark. In Electronic Imaging: Science and Technology, pages 126–133. International Society for Optics and Photonics, 1996

[7] J. Meng and S.-F. Chang. Embedding visible video watermarks in the compressed domain. In Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, volume 1, pages 474–477. IEEE, 1998.

[8] Y. Hu and S. Kwong. Wavelet domain adaptive visible watermarking. Electronics Letters, 37(20):1219–1220, 2001.

[9] S. C. Pei and Y. C. Zeng, ”A Novel Image Recovery Algorithm for 1Visible Watermarked Image,” IEEE Trans on Information Forensics and Security, vol. 1, pp. 543-550, Dec. 2006.

[10] C. H. Huang, and J. L. Wu, ”Attacking Visible Watermarking Schemes”, IEEE Trans. Multimedia, vol. 6, no. 1, pp. 16, Feb. 2004. [Accessed: 21- Mar- 2019]

[11] H. Santoyo-Garcia, E. Fragoso-Navarro, R. Reyes-Reyes, G. Sanchez-Perez, M. Nakano-Miyatake and H. Perez-Meana, ”An automatic visible watermark detection method using total variation,” 2017 5th International Workshop on Biometrics and Forensics (IWBF), Coventry, 2017, pp. 1-5. doi: 10.1109/IWBF.2017.7935109

[12] Marco Forte, closed-form-matting, GitHub repository: https://github.com/MarcoForte, [Accessed: 20- Mar- 2019].

[13] A. Levin, D. Lischinski, Y. Weiss, ”A Closed Form Solution to Natural Image Matting”, *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2006. [Online]. Available: http://webee.technion.ac.il/people/anat.levin/papers/Matting-Levin-Lischinski-Weiss-CVPR06.pdf . [Accessed: 20- Mar- 2019]

[14] Rohit Kumar Jena, automatic-watermark-detection, (2017), GitHub repository: https://github.com/rohitrango, [Accessed: 18- Mar- 2019]

[15] R. Raskar, ”Matlab Code for Poisson Image Reconstruction from Image Gradients”, Web.media.mit.edu, 2004. [Online]. Available: http://web.media.mit.edu/ raskar/photo/code.pdf. [Accessed: 18- Mar- 2019].

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on, 13(4):600–612, 2004.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014, pages 740–755. Springer, 2014.