**Analysis of Effective Methods for Watermark Detection and Removal**

Insert Subtitle Here

FirstName Surname†  
 Department Name  
 Institution/University Name  
 City State Country  
 email@email.com

FirstName Surname  
 Department Name  
 Institution/University Name  
 City State Country  
 [email@email.com](mailto:email@email.com)

FirstName Surname  
 Department Name  
 Institution/University Name  
 City State Country  
 email@email.com

**ABSTRACT**

Watermarks are a standard way of protecting a person’s rights to the photos and images they have created. However, with advancements in Deep Learning and other Machine Learning methods, it is now becoming possible to attack these watermarks, by detecting them and even removing them almost seamlessly from images. This project explores the effectiveness of Deep Learning methods against watermarks, by trying to do just that. We work with different styles of watermarks, and try to find the ones that are least susceptible to detection and removal.

**KEYWORDS**

Watermark, Embedment, Detection, Removal, Convolutional Neural Networks.

1**INTRODUCTION**

According to studies by IMGembed and Copytrack [9], around 85% of the three billion images shared online daily are unlicensed. Around 64% of professional photographers reported instances of image thefts over 200 times. Every time a professional image is stolen, photographers and agencies lose an average of $446. Many of these photographers do not take legal action or cannot afford to take legal action. The rate of image theft continues to rise due to the increase of media content on the internet. With people’s high usage of social media, it is easier than ever to distribute unlicensed images across the internet. A popular method of mitigating image theft is the embedment of watermarks on images. However, even with the addition of watermarks, around 68% of images containing watermarks have their watermarks removed since the watermarks are often not strong enough. Thus, this project analyzes the effectiveness of different styles of watermarks and finds which ones are most susceptible to detection and removal.

This project contains three main steps. First, we embedded watermarks of various styles on our dataset of 22762 scenery images. Then, we used Convolutional Neural Networks to train an object detection model to detect the region of interest of watermarks in these images. Lastly, we used Adversarial Learning-based approaches for pixel value prediction to reconstruct the original image after the watermark is removed. After we completed these steps, we compared the strengths of the different styles of watermarks using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and structural similarity index measure (SSIM). Using these metrics, we find out which watermark styles are most resistant to detection and removal.

2**RELATED WORK**

2.1**Model Free Methods**

The most primitive methods of embedding, detecting, and removing watermarks from images use various model-free methods without any deep learning techniques. There has been a plethora of research done over the last few decades to make watermarks harder to detect and remove in order to protect the images’ copyrights. One method to make watermarks harder to detect is by embedding watermarks in a way that adapts to the texture, edge, and luminance of the pixels in the underlying image [5]. This causes variations between different watermarks, making it difficult for detection algorithms to learn the patterns. Another method for embedding watermarks is an algorithm based on a user-key structure that adapts to the host image’s features [4]. With the correct key, the watermark can be removed without affecting the original image. If the watermark is removed illegally, then the image will have a lower quality.

For automatic detection of watermarks, one algorithm uses total variation based on L1 norm [8]. The watermarked image is decomposed based on its structure and texture, and the watermark’s edges are distinguished from the host image’s edges. Another model free method [3] uses a simple approach for the detection and removal of watermarks simply using mathematical equations and image processing techniques. This method is based on the assumption of consistency and uniformity of the locations, size, and types of watermarks.

These methods can be effectively used once we have a localized region for the watermark, which can be achieved by Convolutional Neural Networks.

2.2**Convolutional Neural Networks**

Convolutional Neural Networks (CNNs) are the traditional approach for any object detection tasks. This is due to the fact that they are much better suited to detect features, and simplify the processing of features in high dimensional data such as images. There is only one paper in this given grouping as this is the only approach we found that was using object detection using CNNs as an initial step. The method proposed in [2] is a two-step method, consisting of two specialized CNN models for two different tasks. The first task is to locate the watermark in the given image, and thus the first CNN model is an object detection model. The second model is trained as an Image Translation model, which takes in the region of pixels affected by the watermark and attempts to convert them back to their original form. We believe that by taking the learnings of this paper from task one, we can build a strong watermark detection model, and then try different approaches to convert the pixels in the watermarked region, using both Model Free and Model Based approaches. This is how we arrived at our initial technical approach in the proposal.

The CNN-based model’s method is adept at detecting/identifying the watermarks in the images, but the step for the reconstruction of the images leads to the creation of some artifacts. Adversarial methods are able to deal with this problem by using a generator-discriminator model.

2.3**Adversarial Learning**

Adversarial Learning-based approaches have become more prevalent for watermark removal techniques. An approach using a Conditional GAN model that reconstructs images with a patch-based discriminator trained on reconstructed and original images [6] is able to deal with the problem of reconstruction of artifacts produced. This model incorporates a patch-based discriminator which uses the reconstructed image and original image to improve the reconstruction by reducing traces of watermarks (artifacts) that were left by the original model. This patch-based discriminator can thus be used on top of the CNN-based model previously mentioned to improve the results by removing the artifacts. However, it is difficult to train the discriminator to differentiate between the reconstructed and original images, as it could lead to overfitting.

Other recent approaches using Adversarial Learning are able to perform significantly well on the watermark removal task, but these methods utilize substantially greater computational power. One such approach is a general-purpose watermark removal framework called REFIT [1]. This technique uses fine-tuning and two different methods, EWC algorithm and unlabeled data augmentation, to remove watermarks without affecting the model's functionality under a weak threat model. It can remove watermarks against a wide range of watermarking schemes. Although it only evaluated image classification models under a weak threat model, this technique shows promising results in removing watermarks without affecting the model's performance or accuracy. An alternative approach used in [7] involves a deep learning framework based on an encoder-decoder structure for the task and is able to perform well even in the case when the images used have lower quality due to image compression or image resizing.

3**DATA**

We used the Scenery Watermark Detection dataset from Kaggle for this project. It contains 22762 scenery images. Of these images, 9983 images (43.9%) contained watermarks and 12779 images (56.1%) did not contain watermarks. Each image is 512 x 512 pixels and RGB-colored, thus yielding 786432 features. Having a large dataset helps reduce potential biases in the watermark detection and removal models since they will be trained with a wide variety of features. We splitted the dataset into 60% Train, 20% Validation, 20% Test classes to train, validate, and test our watermark embedment, detection, and removal models.

4**METHODS**

Our given problem is split into three major steps. First, we need to embed different styles of watermarks on our scenery images. Then, we need to detect the region of interest of a watermark in a given image. Once we are able to accurately locate the watermark, we need to fill in the watermark pixels with values that will blend in with the surrounding pixels.

4.1**Watermark Embedment**

For watermark embedment, we used Image Processing techniques with OpenCV and Python, and came up with a randomized text generator. This text generator chooses a small set of random english words, a random color, a random thickness and font, a random opacity level, and a random location on the original image. Combining all these parameters, we create our watermark and add it onto the image. We also save the bounding box of the watermark, and convert it into PASCAL VOC annotation format for object detection. We are currently working on randomizing the parameters in even more ways, to get as much variety in our watermarks as possible.



Figure 1: **Image before watermark embedment**



Figure 2: **Image after watermark embedment**

4.2**Watermark Detection**

2 different methods have been used for the task of watermark detection. First is Unet and second YOLO.

The second method which has been used is You Only Look Once Object Detection, also known as YOLO. YOLO is a very strong Object Detection algorithm, based on a Convolutional Neural Network and a unique approach to searching for objects in a given image. While other algorithms like R-CNN follow a two step approach to object detection, which involves initially detecting possible areas of interest, and then passing over those areas again to confirm whether there is an object present. With YOLO, we only pass through the image once, dividing it into blocks which the algorithm goes over once and gives a prediction score based on how likely there is an object in that block. It is much faster, and very accurate as well. Therefore, we have gone with the open source implementation of YOLO V8 for the Watermark Detection step.

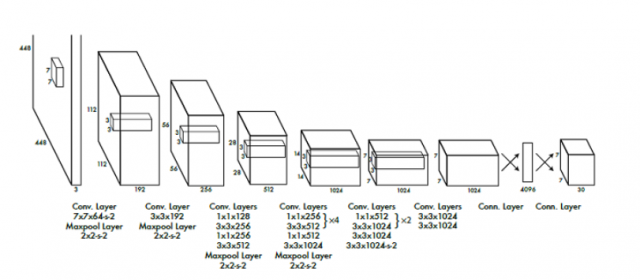


Figure 3: **CNN Architecture for YOLO**

We train the YOLO v8 Model on our generated dataset.



Figure 4: **Sample Image with Bounding Box**

4.3**Watermark Removal**

For the step of watermark removal, 2 different strategies have been implemented.

The first method uses a simple autoencoder structure. we compared a variety of methods for Pixel Value Prediction, ranging from simple sequential averaging of surrounding values to using Deep Learning for finding edges and filling pixels appropriately. We used… As for the tools and libraries, we used OpenCV for the Image Processing part, and for the Deep Learning models, we used Tensorflow or Pytorch.

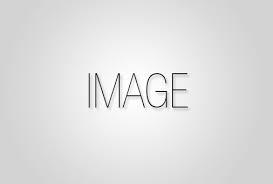


Figure 5: **Original image with watermark**

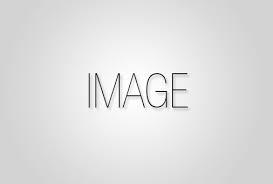


Figure 6: **Image after watermark removal**

5**RESULTS**

The YOLO V8 algorithm performed very strongly on our generated watermark dataset, and gave very high accuracy in predicting the bounding boxes for the watermarks.

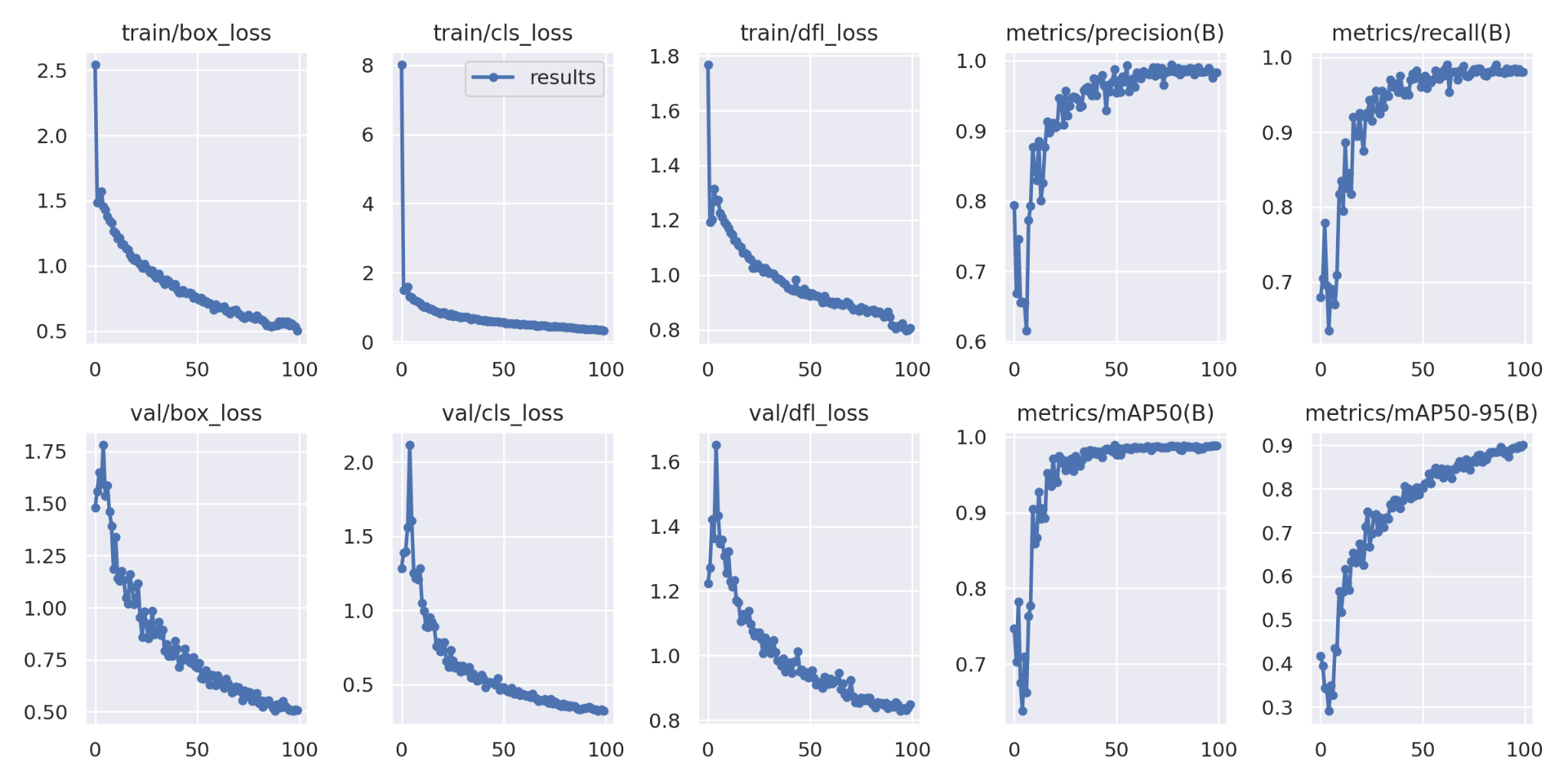


Figure 7: **YOLO V8 Training Metrics Graph**

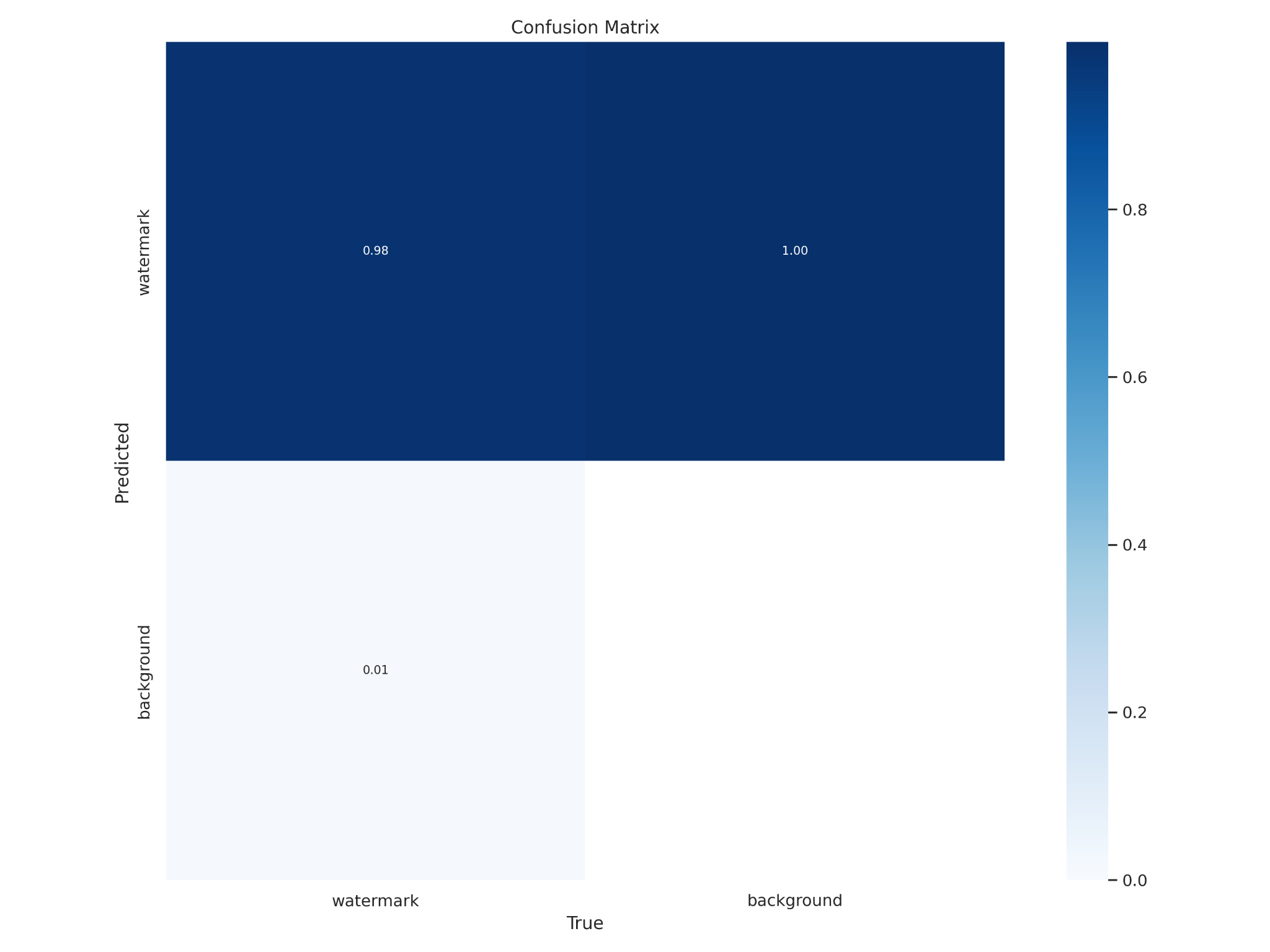


Figure 8: **Validation Data Confusion Matrix**

From the metrics, we see that the model is not overfitting on the training data, and having similar good performance on validation set. Also, in the confusion matrix, we see just how well the model is doing, with a 99% prediction accuracy for our watermark objects.



Figure 9: **Test Image with Predicted Watermark**

Our goal here is not just to make a good object detection model for watermarks, it is to find the styles of watermarks that are strongest at being undetected by such a model. Therefore, having set our baseline detection model, for the final project, we will be generating datasets of different kinds of watermarks, and individually evaluating how well this model performs on them.

6**DISCUSSION**

In this sample-structured document, neither the cross-linking of float elements and bibliography nor metadata/copyright information is available. The sample document is provided in “Draft” mode and to view it in the final layout format, applying the required template is essential with some standard steps.

These steps, which should require generation of the final output from the styled paper, are mentioned here in this paragraph. First, users have to run “Reference Numbering” from the “Reference Elements” menu; this is the first step to start the bibliography marking (it should be clicked while keeping the cursor at the beginning of the reference list). After the marking is complete, the reference element runs all the options under the “Cross Linking” menu.

7**CONCLUSION**

In this sample-structured document, neither the cross-linking of float elements and bibliography nor metadata/copyright information is available. The sample document is provided in “Draft” mode and to view it in the final layout format, applying the required template is essential with some standard steps.

These steps, which should require generation of the final output from the styled paper, are mentioned here in this paragraph. First, users have to run “Reference Numbering” from the “Reference Elements” menu; this is the first step to start the bibliography marking (it should be clicked while keeping the cursor at the beginning of the reference list). After the marking is complete, the reference element runs all the options under the “Cross Linking” menu.

**REFERENCES**

[1] Xinyun Chen, Wenxiao Wang, Chris Bender, Yiming Ding, Ruoxi Jia, Bo Li, and Dawn Song. 2021. REFIT: A Unified Watermark Removal Framework For Deep Learning Systems With Limited Data. In Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (Virtual Event, Hong Kong) (ASIA CCS ’21). Association for Computing Machinery, New York, NY, USA, 321–335. DOI:https://doi.org/10.1145/3433210.3453079.

[2] Danni Cheng, Xiang Li, Wei-Hong Li, Chan Lu, Fake Li, Hua Zhao, and Wei-Shi Zheng. 2018. Large-Scale Visible Watermark Detection and Removal with Deep Convolutional Networks. In Pattern Recognition and Computer Vision, Jian-Huang Lai, Cheng-Lin Liu, Xilin Chen, Jie Zhou, Tieniu Tan, Nanning Zheng, and Hongbin Zha (Eds.). Springer International Publishing, Cham, 27–40. DOI:https://doi.org/10.1007/978-3-030-03338-5\_3

[3] Tali Dekel, Michael Rubinstein, Ce Liu, and William T. Freeman. 2017. On the Effectiveness of Visible Watermarks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6864–6872. DOI:https://doi.org/10.1109/CVPR.2017.726.

[4] Yongjian Hu, S. Kwong, and Jiwu Huang. 2005. An algorithm for removable visible watermarking. IEEE Transactions on Circuits and Systems for Video Technology 16, 1 (December 2005), 129–133. DOI:http://dx.doi.org/10.1109/tcsvt.2005.858742.

[5] M.S. Kankanhalli, Rajmohan, and K.R. Ramakrishnan. 1999. Adaptive visible watermarking of images. In Proceedings IEEE International Conference on Multimedia Computing and Systems, Vol. 1. 568–573 vol.1. DOI:https://doi.org/10.1109/MMCS.1999.779263.

[6] Xiang Li, Chan Lu, Danni Cheng, Wei-Hong Li, Mei Cao, Bo Liu, Jiechao Ma, and Wei-Shi Zheng. 2019. Towards Photo-Realistic Visible Watermark Removal with Conditional Generative Adversarial Networks. In Image and Graphics, Yao Zhao, Nick Barnes, Baoquan Chen, Rüdiger Westermann, Xiangwei Kong, and Chunyu Lin (Eds.). Springer International Publishing, Cham, 345–356. DOI:https://doi.org/10.48550/arXiv.1905.12845.

[7] Jing Liang, Li Niu, Fengjun Guo, Teng Long, and Liqing Zhang. 2021. Visible Watermark Removal via Self-calibrated Localization and Background Refinement. CoRR abs/2108.03581 (2021). DOI:https://doi.org/10.48550/arXiv.2108.03581.

[8] Hector Santoyo-Garcia, Eduardo Fragoso-Navarro, Rogelio Reyes-Reyes, Gabriel Sanchez-Perez, Mariko Nakano-Miyatake, and Hector Perez-Meana. 2017. An automatic visible watermark detection method using total variation. 2017 5th International Workshop on Biometrics and Forensics (IWBF) (April 2017), 1–5. DOI:http://dx.doi.org/10.1109/iwbf.2017.7935109.

[9] Marcus Schmitt. 2019. Global infringement report 2019 - Copytrack. (2019). Retrieved April 13, 2023 from https://www.copytrack.com/wp-content/uploads/2019/03/Global\_Infringement\_Report\_2019\_EN.pdf

**END OF DOC. WILL REMOVE BELOW CONTENT BEFORE SUBMISSION**

**ACM Reference format:**

FirstName Surname, FirstName Surname and FirstName Surname. 2018. Insert Your Title Here: Insert Subtitle Here. In *Proceedings of ACM Woodstock conference (WOODSTOCK’18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1**Insert Heading Level 1**

The updated template, user manuals, samples, and required fonts, all are available at the URL <https://www.acm.org/publications/proceedings-template>. It contains said information for all three versions of MS Word (Windows and 2 versions of Mac). There are also separate links to the user guide, which can be referred to by the user. This URL also contains some useful video links, which describe how to add the template, structure the paper, and generate the layout, in different clips. **Display Formula with Number**

(1)



**Continuation part of Paragraph Text** The user must style this paragraph in **ParaContinue** style, which follows immediately after the **DisplayFormula** (numbered equation). The **DisplayFormula** style is applied only in case of a numbered equation. A numbered equation always has a number to its right. Insert paragraph text here. **Display Formula without Number**



The **DisplayFormulaUnnum** style is applied only in case of an unnumbered equation. An unnumbered display equation never contains an equation number to its right, and this unique property distinguishes it from a numbered equation.

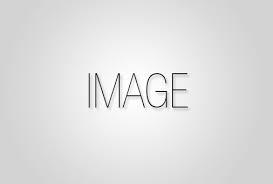


Figure 1: **Figure Caption and Image above the caption [In draft mode, Image will not appear on the screen]**

**Theorem/Proof/Lemma.** Insert text here for the enunciation or Math statement. Insert text here for the enunciation or Math statement. Insert text here for the enunciation or Math statement. Insert text here for the enunciation or Math statement. Insert text here for the enunciation or Math statement.

....Insert text here for the Quotation or Extract, Insert text here for the Quotation or Extract, Insert text here for the Quotation or Extract, Insert text here for the Quotation or Extract, Insert text here for the Quotation or Extract, Insert text here for the Quotation or Extract.

1.1**Heading Level 2**

In the below paragraph, it is explained how alt-txt value is placed in **MS Word 2010**. To add alternative text to a picture in Word 2010, follow these steps:

1. In a Word 2010 document, insert a picture.
2. Right click on the inserted picture and select the **Format Picture** option.
3. Select the **Alt Txt** option from the left-side panel options.
4. In the "Title:" and "Description:" text boxes, type the text you want to represent the picture, and then click "Close".

Below are steps to place alt-txt value in **MS Word 2013/2016**. To add alternative text to a picture in Word 2013/2016, follow these steps:

1. In a Word 2013/2016 document, insert a picture.
2. Right click on the inserted picture and select the **Format Picture** option.
3. In the settings at the right side of the window, click on the "Layout & Properties" icon (3rd option).
4. Expand **Alt Txt** option.
5. In the "Title:" and "Description:" text boxes, type the text you want to represent the picture, and then click "Close".

*1.1.1 Heading Level 3.* Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

*1.1.1.1 Heading Level 4.*Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

**ACKNOWLEDGMENTS**

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

**REFERENCES**

[1] Patricia S. Abril and Robert Plant, 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan, 2007), 36-44. DOI: <https://doi.org/>10.1145/1188913.1188915.

[2] Sten Andler. 1979. Predicate path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226-236. DOI:https://doi.org/10.1145/567752.567774

[3] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:https://doi.org/10.1007/3-540-09237-4.

[4] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY..