Facial Photoshop Detection

Ryan Jung, Jonathan Nushi, Justin Park

A. Photoshopped Image *B. Warp Detection* *C. Copy-Paste Detection*

Figure 1. Photo A is a photoshopped image where the woman's face has been warped and a fake pair of glasses has been added. Photo B is the result of the FALdetector identifying where her face has been warped, and photo C is a mask of MVSS_Net's foreign element prediction.

A. Introduction

Photoshop is a powerful and easily accessible tool, making it very easy to tamper with photos and the narratives that they create. In order to combat the capabilities of Photoshop and similar programs, several models have been developed to detect photos that have been edited in various ways. During our research, we discovered two models (detailed below in 'Related Work' section), one that detects facial warping [1], and another that detects when an image has a mask copy-pasted on top of it [2]. Our objective for this project is to combine the features of both models to create a more collectively exhaustive model that can identify facial photoshopped instances by its individual tools as well as fine-tune these pre-existing models to improve their accuracy. We also aim for this project to help us understand how to better design machine learning models in Computer Vision.

B. Related Work

We have researched two models that detect the effects of two different Photoshop tools. The first of which, the MVSS_Net model [1], detects parts of one image that were copied and pasted onto another. The model outputs a mask of the copy and pasted image. This model was compared against an FCN model, and was trained on CASIAv2 and DEFACTO-84k training data. We believe this model will be useful for detecting foreign elements that are masked onto a face to alter the viewers' perception of the portrait (eg. moles, glasses, etc.). Second, the FAL detector [2], is a CNN model trained on images of people's faces that were modified by Photoshop's Face-Aware Liquify tool. This model is designed to detect warping of facial features and produces a heat map of these edited features. Its local prediction method uses a combination of loss functions including flow warping prediction, relative warp preservation, and a pixel-wise reconstruction loss. We believe this model will be useful for detecting warping of cheekbones, chin, etc., which ultimately changes the integrity of the original persons' facial structure.

C. Method

Our greatest hurdle to overcome was the issue that although the FALdetector model was trained on faces, the MVSS_Net model was not. As a result, MVSS_Net was highly inaccurate when tasked with detecting copy-pasted elements on portraits.

A. Input to MVSS_Net *B. Mask Output*

Figure 2. MVSS_Net mistakenly identifies not just copy-pasted glasses as a mask but part of face as well, most likely due to differing focus of foreground/background.

A. Unmodified *B. Warped Only* *C. Unwarped + Mask* *D. Warped + Mask*

Figure 3. Photo A is an unmodified photo, used to train the model on photos with no editing done. Photo B is a photo that has only been warped and photo C is a photo that has only an added mask. These are used to see how the model analyzes photos with only one added feature. Photo D is a photo that has been both warped and masked to train the model to detect both.

Several features in the face dataset led to these inaccuracies, with the most prevalent being the contrasting focus between the face and blurred background. In order to resolve this issue, we decided to create a new dataset that would combine features from both models' datasets. This new dataset consisted of four different batches of photos; unmodified, warped only, masked only, and both warped and masked. The unmodified and warped images all originated from FALdetector's original dataset, and the unwarped + masked and warped + masked images are photos from FALdetector's dataset that we edited using Photoshop. These edits included adding glasses, face masks, tattoos, hats, extra facial features, cigarettes, etc. Initially, the plan was to include 250 photos in each, however, to more efficiently allocate our efforts, we decided to instead focus our time towards fine tuning the models. With these four batches of images, we trained the models on images that have been edited using different combinations of editing tools.

Batch	# of Images
Unmodified	250
Warped Only	250
Unwarped + Mask	184
Warped + Mask	61

Table 1. Distribution of images for the 4 different batches.

After creating this new dataset, we proceeded to fine-tune both models using the same images, but different labels. Our overall model inputs a single image and splits the detection into two processes/threads, simultaneously classifying the image using our fine-tuned FALdetector and MVSS-Net.

C.0.1 FALdetector fine-tuning

For the FALdetector model, we fine-tuned our model using the newly created data images, along with a label that denoted whether or not the image was warped. A label of 1 was given to warped images, and a label of 0 for those that were not. This model was fine-tuned for 3 epochs with a batch size of 32.

C.0.2 MVSS-Net fine-tuning

For the MVSS-Net model, we also fine-tuned using the newly created data images; however, we labeled our data with hand-made ground truth masks as our model did not have a built-in classifier. As a result, we instead trained our model on predicting a mask output and classified that if the maximum pixel in the predicted output (normalized from 0-255 to 0-1) was greater than or equal to 0.5, it predicted true, and if the maximum pixel in the predicted output was less than 0.5, it predicted false. This model was fine-tuned for 3 epochs with a batch size of 32.

D. Experiments

Model	Accuracy Pre Fine-Tuning	Accuracy Post Fine-Tuning
FALdetector	0.78	0.88
MVSS-Net	0.67	0.83

Table 2. Accuracies for both models pre/post fine-tuning.

C.1.1 FALdetector results

Prior to fine-tuning the FALdetector model, the reported baseline accuracy was around .78 on the dataset with the four separate batches. A possible explanation for this accuracy could be due to the familiarity of the model with the first two batches (unmodified and warped only), coupled with the unfamiliarity of the newly-introduced masks (which the model has not been trained on). Additionally,

many of these masks partially/fully obstructed the warping on the faces for the images that were in the warped + mask batch. Therefore, there could have potentially been large inaccuracies in the model after the introduction of these masks to the dataset.

After fine-tuning the FALdetector model, the accuracy rose to .88. This is most likely attributed to the model now being trained on copy/pasted masks on top of the faces. After being trained on this new dataset, the model was better at identifying these copy/pasted masks which explains the increased accuracies on the dataset post fine-tuning.

C.1.2 MVSS-Net results

The baseline accuracy of MVSS-Net prior to fine-tuning is 0.67, which is lower than FALdetectors' baseline accuracy of 0.78. This is reasonable as FALdetector has prior familiarity with a portion of the dataset whereas MVSS-Net does not. Furthermore, MVSS-Nets' baseline accuracy is also lower than three of the four other datasets that it has been tested on (CASIAv1plus, Columbia, COVER, DEFACTO-12k). This can be attributed to MVSS-Nets' struggle with portraits. Specifically, because the model has not been previously trained on portraits, it is unfamiliar to the focusing effect of cameras where the background is blurred. As a result, most of its incorrect predictions before fine-tuning were masks of the entire head.

The post-finetune accuracy of our MVSS-Net is 0.83, which shows to be a significant improvement. From this result, we can identify that fine-tuning our model on a dataset of portraits has had a positive impact on its accuracy towards identifying photoshop masking over faces.

E. Conclusion

We can conclude from our results that our model can not only detect different combinations of warping and masking over images, but that it also has a greater accuracy than if we were to simply run its pre-trained models.

Reproducibility: A limitation that restricts reproducibility is that our model is unique to Photoshop's Face-Aware Liquify tool. Due to this limitation, it is difficult to apply these models on other warping effect tools due to the uniqueness of this tool.

Strengths/Weaknesses: A strength of our model is that we were able to improve upon both baseline accuracies of MVSS-Net and FALdetector. However, the accuracy of our models can still be further improved.

Future work: In the future, we could use a larger dataset, as due to time constraints we only used 645 data points total. We could also try expanding our model to be trained on identifying other Photoshop tools.

F. Contribution

- Introduction - Jonathan, Ryan, Justin
- Related Work - Jonathan
- Method - Ryan, Jonathan, and Justin
- Experiments - Ryan and Justin
- Conclusion - Ryan, Jonathan, and Justin

https://github.com/ryanjung1211/FacialPhotoshopDetection

References

[1] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, Alexei A. Efros. Detecting Photoshopped Faces by Scripting Photoshop. In Proceedings of the The IEEE International Conference on Computer Vision (ICCV), Virtual, 71-80 June 2019

[2]Chen, Xinru and Dong, Chengbo and Ji, Jiaqi and Cao, Juan and Li, Xirong. Image Manipulation Detection by Multi-View Multi-Scale Supervision.In Proceedings of the The IEEE International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021