Detecting Photoshopped Faces by Scripting Photoshop Supplemental Material

Sheng-Yu Wang[1] Oliver Wang[2] Andrew Owens[1] Richard Zhang[2] Alexei A. Efros[1]

UC Berkeley[1] Adobe Research[2]

1. Supplemental Video

We have included a supplementary video in the following link: https://youtu.be/TUootD36Xm0. We invite readers to view this video for better visualizations of our qualitative results.

2. Qualitative results

Local predictions Figure 5 shows a random selection of results from our validation dataset of automatically-generated manipulations. We conducted an experiment where the PSNR change with respect to scaled versions of the predicted flow field are shown over the validation set (Figure 1). We can see that the highest PSNR gain is where the scale factor is 1.0, which implies that our predicted flow fields do not contain a multiplicative bias, that might result from the regression loss.

Network visualization We visualize our global classifier using the class activation map method of Zhou *et al*. [9]. Figures 6, 7 show a random selection of class activation maps of our global classifier. Note that our global classifier model is able to achieve high accuracy (93.7%) despite the mismatch between class activation maps and ground truth flow. This suggests that the model may be able to pick up other cues to differentiate between original and manipulated images.

3. Robustness to corruptions

We tested the robustness of our model by perturbing the low-level statistics of our validation set through common corruptions such as lossy JPEG compression, blurring, and printing and scanning physical prints. This offers three interesting test cases, as we *did* train on JPEG compressed images, did *not* train on blurring, and *cannot* train on rescanned images due to the cost of dataset acquisition.

As shown in Fig. 2, the method with augmentation is fairly robust to JPEG compression. Though we did not train with blurring augmentations (as images are unlikely to be intentionally blurred), training with other augmentations helps increase resilience. However, with significant blur

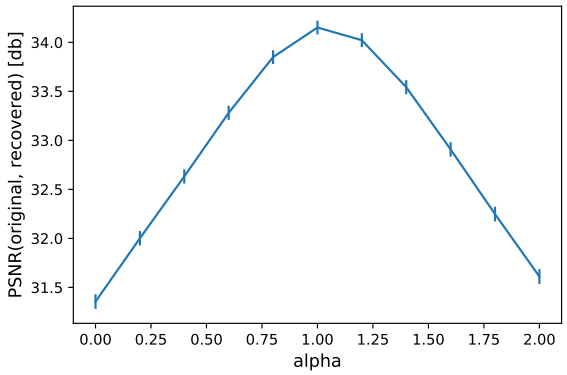

Figure 1: PSNR plots from our held-out validation subset. We plot the average PSNRs of the unwarped image to the original (y-axis), with respect to a multiplicative factor on the predicted flow field. The error bars are the standard errors. In the ideal case, this PSNR should peak at 1.0, the predicted flow.

($\sigma > 4$), performance degrades to chance levels. This indicates that the classifier is relying on some high frequency information, which is the main component attenuated by the Gaussian filter.

Lastly, we also test the robustness of our classifier to print rebroadcasting [1], testing on images that are printed, and then re-digitized by a scanner (*e.g.*, simulating the task of identifying manipulations in magazine covers). We used a Canon imageRunner Advance C3530i Multifunctional copier and standard 8.5×11 inch paper. We randomly selected 30 images each from the Flickr and OpenImages sets. Classification performance drops from 94.2% to 69.2% (standard error of 6.0%). While rebroadcasting hurts performance, our model still detects manipulated images significantly more accurately than chance.

4. Generalization

We are interested in what cues in the images the model learns to focus on, in order to detect warping. For example, is the model looking at low-level image statistics (*e.g.* resampling artifacts) or high-level cues (*e.g.* facial geometric inconsistencies)? This has larger implications for ex-

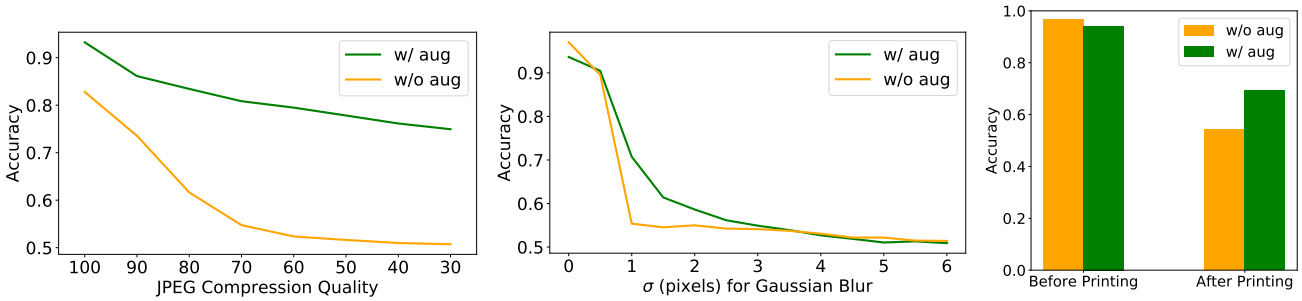

Figure 2: **Robustness to corruptions.** Accuracy of global classification with JPEG, Gaussian blur, and after rescanning, with and without data augmentation. (left) JPEG compression: a significant increase in robustness. Though unsurprising, as it is in our augmentation set, it is important, as compression and recompression is commonly applied to images. (middle) Blur: although this is not in our augmentation set, we observe a small increase in robustness. (right) Rescanning: a small increase in robustness. We corrupt by printing and rescanning a subset of photos, a perturbation that cannot be reproduced during training

	Global		Local	
	Acc.	**AP**	**ΔPSNR**	**IOU-3**
Face / FAL	93.7	98.9	+2.69	0.43
Face / X2face	64.7	74.0	+0.13	0.05
Noise / FAL	44.5	92.9	–	0.43
Noise / X2face	36.5	82.0	–	0.03
Natural / FAL	67.7	77.3	+0.12	0.05

Table 1: **Generalization results**. We tested the generalization of our global and local models on four out-of-distribution dataset. The top row (Face/FAL) contains the results of our original validation set for comparison.

ample in whether the model can detect warps only realizable by FAL, or can it detect more general warping scenarios? To investigate, we evaluate our global and local models in four different scenarios: (1) images composed of noise, warped with FAL warps, (2) images composed of noise warped with out-of-domain warps, (3) out-of-domain natural images warped with FAL warps, and (4) portrait images warped with out-of-domain warps.

To generate out-of-domain warps, we randomly sampled the latent space of the optical flow generator in the X2face model [6] to generate warps. We note that although the X2face model is trained to generate face-specific warps, the warping field will not necessarily align with the portrait; moreover, since a VAE loss is not included during X2face training, sampling the bottleneck does not guarantee to have realistic warping fields. However, empirically we observed our sampling method generates smooth warping fields that modifies the face in a "stochastic" fashion. That is, the X2face warping field will not specifically change a face in a meaningful way such as making someone's smile bigger or face smaller. On the other hand, for out-of-domain images we collected natural images from random samples in Open Images [4], which are not portrait images. Table 1 shows

Figure 3: **Noise experiment setup**. The Gaussian noise image (left) and the face image (middle) are deformed with the same warping field (right). Our model trained on faces can detect the warped noise (if well-calibrated), but a model trained on noise cannot detect the warped face.

the results.

Note that when there is a domain shift in warping field (face/X2face) or image space (natural/FAL), the performances of both models drop significantly although still perform above chance (50% Accuracy and 0 ΔPSNR). More interestingly, note that our global model is able to generalize to warped noise with FAL and X2face flows at a certain degree if well-calibrated (92.9, 82.0 AP), and our local model generalizes specifically to FAL-warped noise. This indicates they have learned low-level warping cues, while the local model is more specific to FAL warping field statistics. However, we trained global and local models solely on noise warped with FAL flows and tested on our validation set, and the models are only able to achieve 49.6% accuracy and 28.28 EPE respectively. This suggests that our model *has* learned low-level cues, but that low-level cues are not sufficient: *the face warping problem is much more difficult*.

5. Additional data collection details

Figure 4 shows a sample of the manipulations in our automatically-generated dataset. For each example photo, we show all 6 random manipulations that were applied to it.

Collecting real face images To obtain a diverse dataset of faces, we aggregate images from a variety of sources.

First, we take all images from the Open Images dataset [4] with the "human face" label. This dataset consists of humans in-the-wild, scraped from Flickr. We also scrape Flickr specifically for portrait photography images. To isolate the faces, we use an out-of-the-box CNN-based face detector from dlib [2] and crop the face region only. All together, our face dataset contains 69k and 116k faces from OpenImages and Flickr portrait photos, respectively, of which approximately 65k are high-resolution (at least 700 pixels on the shortest side). We note that the our dataset is biased toward Flickr users, who, on average, post higher-quality photographs than users of other Internet platforms. More problematically, the Flickr user base is predominantly Western. However, as our method is entirely self-supervised, it is easy to collect and train with new data to match the test distribution for a target application.

6. Implementation and training details

Flow consistency mask Given the original image X_{orig} and manipulated image X_{mod}, we compute the flow from original to manipulated and from manipulated to original using PWC-Net [5], which we denote U_{om} and U_{mo}, respectively.

To compute the flow consistency mask, we transform U_{mo} from the manipulated image space into the original image space, which is $U'_{mo} = \mathcal{T}(U_{mo};\ U_{om})$. We consider the flow to be consistent at a pixel if the magnitude of $U'_{mo} + U_{om}$ is less than a threshold. After this test, pixels corresponding to occlusions and ambiguities (*e.g.*, in low-texture regions) will be marked as inconsistent, and therefore do not contribute to the loss.

We take relative error of the flow consistency as the criterion. For a pixel p,

$$M_{inconsistent}(p) = \mathbb{1}\{\frac{||U'_{mo}(p) + U_{om}(p)||_2}{||U_{om}(p)||_2 + \epsilon} > \tau\}. \quad (1)$$

We take $\epsilon = 0.1$ and $\tau = 0.85$, then apply a Gaussian blur with $\sigma = 7$, denoted by G, and take the complement to get the flow consistency mask M:

$$M = 1 - G(M_{inconsistent}) \quad (2)$$

Training details for local prediction networks We use a two-stage training curriculum, where we first train a per-pixel 121-class classifier to predict the discretized warping field. We round the flow values into the closest integer, and assign class to each integer (u, v) value with a cutoff at 5 pixels. Therefore, we have $u, v \in \{-5, -4, \ldots, 4, 5\}$, *i.e.* 121 classes in total. We pretrained the model for 100k iterations with batch size 16. Our strategy is consistent to Zhang et al. [8], which found that (in the context of colorization) pretraining with multinomial classification and then fine-tuning for regression gave better performance than just training for regression directly.

The base-network of the regression model is initialized with the pretrained model weights, and the other weights are initialized with normal distribution with gain 0.02. We train the models for 250k iterations with batch size 32.

Both models are trained with Adam optimizer [3] with learning rate 10^{-4}, $\beta_1 = 0.9, \beta_2 = 0.999$.

Training details for global classification networks We initialized the base-network of the DRN-C-26 [7] network with the weights pretrained on the local detection task, and fine-tuned it for the global classification task. We use the Adam optimizer [3] with $\beta_1 = 0.9, \beta_2 = 0.999$, minibatch size 32 and 16 for the low and high-res models, respectively, and initial learning rate 10^{-4}, reduced by $10\times$ when loss plateaus. The models are trained for 300k iterations on 135.4k original images and 812.4k modified images, where the original images are sampled $6\times$ more frequently to balance the class distribution.

References

[1] Wei Fan, Shruti Agarwal, and Hany Farid. Rebroadcast attacks: Defenses, reattacks, and redefenses. In *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018. 1

[2] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009. 3

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[4] Ivan Krasin, Tom Duerig, Neil Alldrin, Andreas Veit, Sami Abu-El-Haija, Serge Belongie, David Cai, Zheyun Feng, Vittorio Ferrari, Victor Gomes, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2(6):7, 2016. 2, 3

[5] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 3

[6] Olivia Wiles, A Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–686, 2018. 2

[7] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[8] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 3

[9] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative

Figure 4: A random sample of manipulations from our dataset. For each photo, we show all 6 random edits that we made. We note that many of these modifications are subtle.

localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.
1

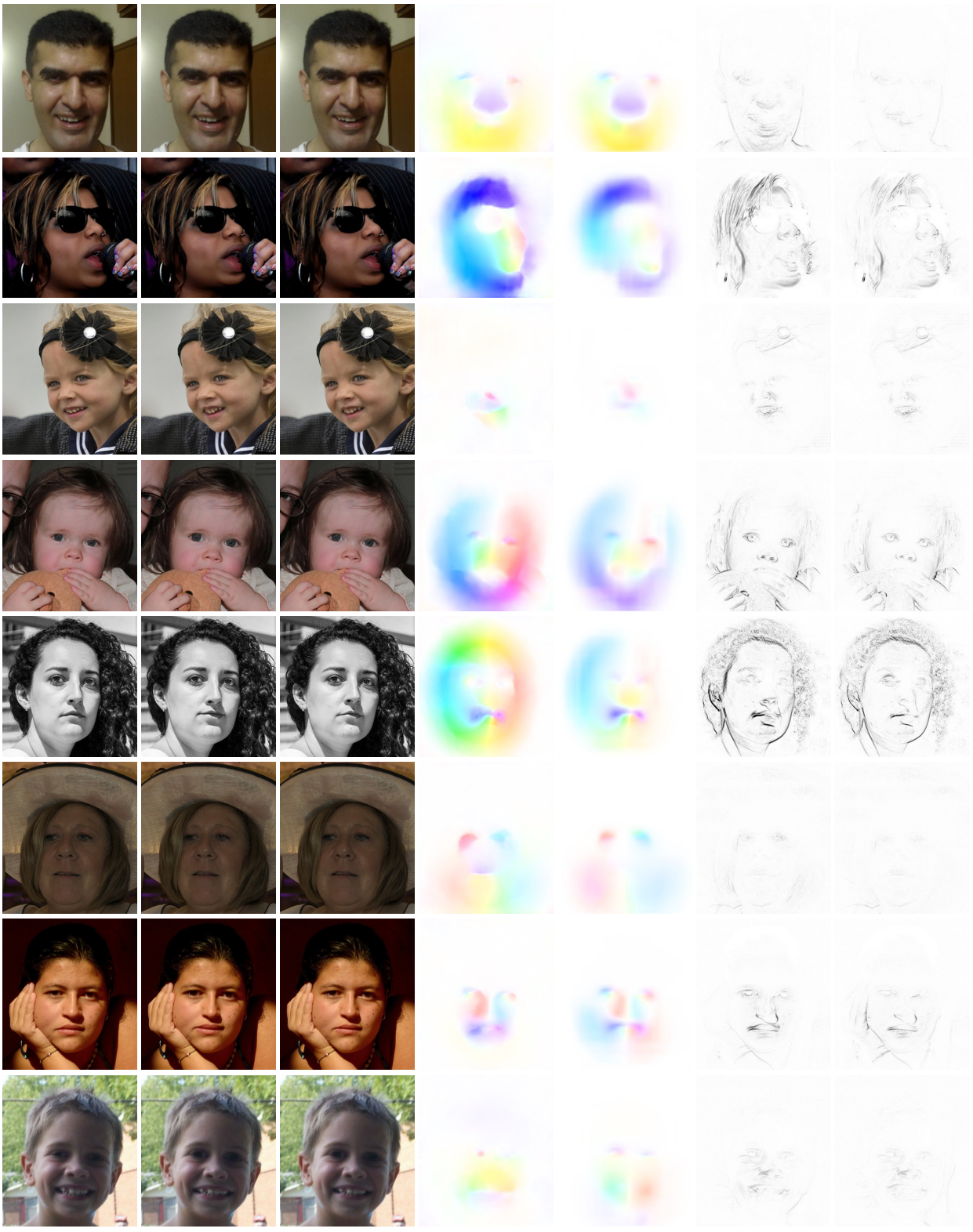

Figure 5: Randomly selected results from our held-out validation dataset, showing the original, warped, and unwarped images. The ground-truth and predicted flow fields, and the difference images between the manipulated and original image, and the unwarped and original images (enhanced for visibility).

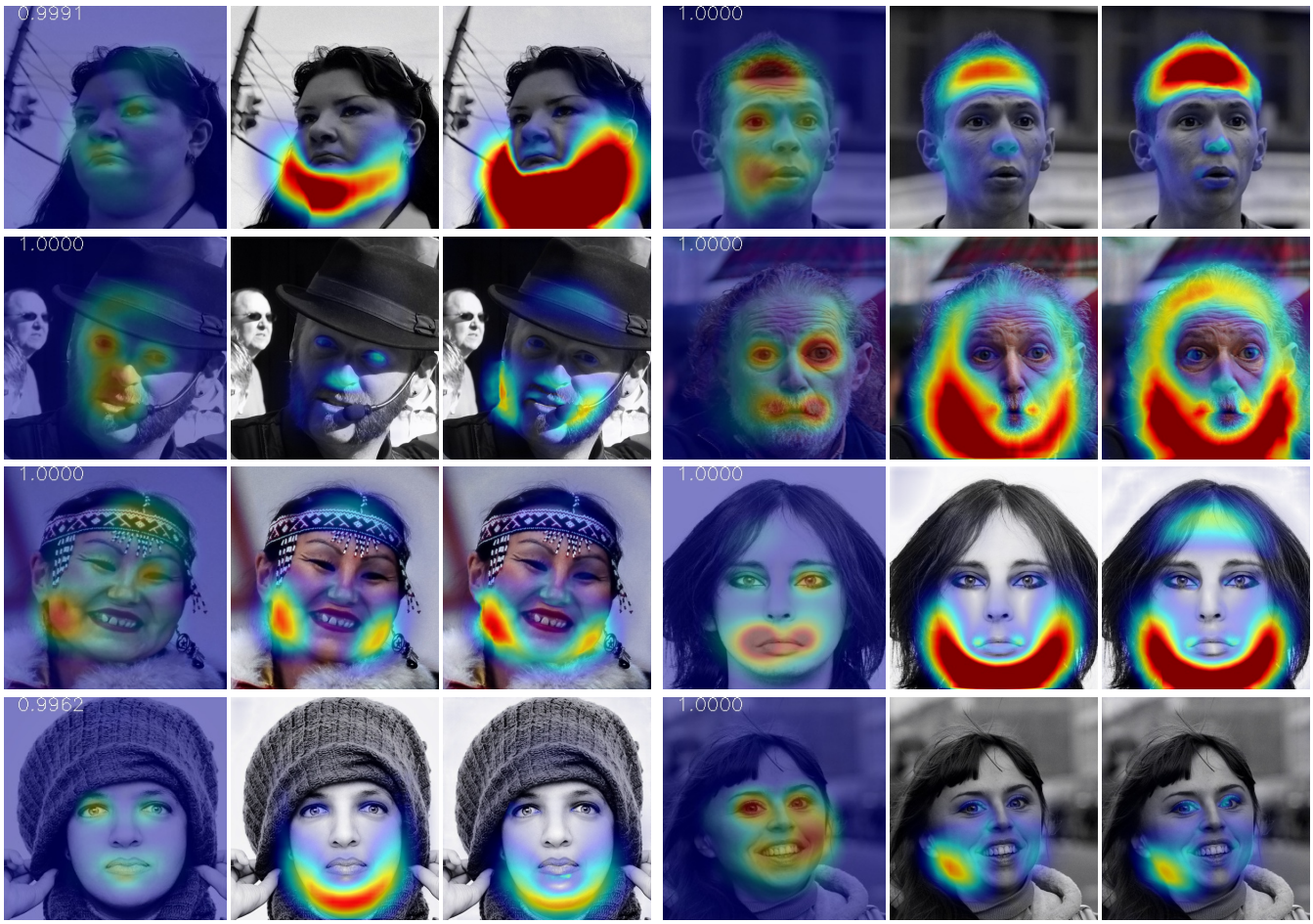

Figure 6: Class activation maps on modified images. Numbers on the upper-left corners of the class activation maps are the modification probability assigned by our model. For reference, we also include the ground truth flow, and our prediction of it.

Figure 7: Class activation maps on randomly sampled original (unmodified) images. Numbers on the upper-left corners of the class activation maps are the modification probability assigned by our model.