Supplementary Materials for

OSCAR-Net: Object-centric Scene Graph Attention for Image Attribution

1. Datasets

1.1. PSBattles24K construction

The original PSBattles dataset [4] contains 102,028 images grouped into 11,142 subsets where each subset has 1 original and several manipulated images. In total, 31,272 online amateur and professional artists contributed to make the manipulated set, averaging 7.9 manipulated variants per original image. In order to increase the challenge in this dataset, we consider manipulated images that had the most subtle changes or least visual differences from the originals. We did this by computing the distance of all original and manipulated image pairs using a pre-trained ImageNet ResNet50 model, and removed the pairs whose distance was larger than a threshold τ ($\tau = 150$, and selected the lower quartile of image pairs, determined by manually inspecting several value ranges). This left 25,046 original-manipulated image pairs. There were several manipulated images that were altered by insertion of imperceptible changes (e.g. introduction of invisible watermarking), which we treated as duplicates, and further removed using crowd-sourced image comparison (via Amazon Mechanical Turk). The final PSBattles24K dataset has 24,157 image pairs, which is split into a training and a test set of 21,197 and 2,960 pairs, respectively. We also ensured the original images do not overlap between the train and test sets. Fig. 1 depicts the histogram of object occurrence in the PSBattles24K test set using an off-the-shelf object detection MaskRCNN model [3]. Several examples are shown in Fig. 2. For the PSBattles24K benchmark and train/test split, please see our project page at https://exnx.github.io/oscar.

1.2. Benign transformations

Details of the benign transformations used in this work can be found in Tab. 1. The list of transformations used during the training and test of PSBattles24K are: 2 primary transformations (jpeg compression, resize) and 6 secondary transformations (flip, rotation, padding, sharpness enhancement, Gaussian noise and color saturation). The PSBattles24K benign-transformation test set is formed from random combinations of these transformations (c.f. subsec.3.5). We aim to learn a fingerprinting model robust to benign transformations at pre-defined parameter ranges (a transformation, although benign, can become destructive if the parameter that controls its severity is set beyond a certain level). It can be seen in Tab. 1 that the test parameters are near-identical to the training parameters except its range is shifted a bit so that the uniformly sampled values are slightly different.

Six other transformations (shot noise, impulse noise, speckle noise, Gaussian blur, defocus blur, pixelate) unseen by the model during training are used for evaluating the effects of individual transformations on performance (c.f. subsec.4.4). The seen and unseen individual transformations constitute the PSBattles360K-S dataset. Fig. 3 illustrate the effect of each individual transformation on an example image.

2. Losses

2.1. Hash loss analysis

As briefly described in subsec.3.4 in our main submission, we deliver true binary embedding during the training via:

$$u = \text{sign}(z) \in \{-1, 1\}^D, \tag{1}$$

where z is our continuous embedding. Our end-loss, SimCLR+ $\mathcal{L}_C$, operates directly on u. Since $\text{sign}()$ has an ill-posed gradient at zero, it is challenging to back propagate the loss gradient through this layer. Mathematically, we wish to compute $\partial\mathcal{L}_C/\partial z$ given $\partial\mathcal{L}_C/\partial u$. The problem is addressed using the Discrete Proximal Linearized Minimization (DPLM) method which was proposed in [11] and used first time in neural networks in [12]. Under DPLM, hashing, in general, can be viewed as an optimization problem:

$$\min_u \mathcal{L}_C(u), \quad s.t.\ u \in \{-1, 1\}^D \tag{2}$$

Here, our loss is optimized w.r.t. u (putting aside the model parameters for now). Then, using gradient descent:

$$u_{t+1} = u_t - \lambda \frac{\partial \mathcal{L}_C}{\partial u_t} \quad s.t.\ u \in \{-1, 1\}^D, \tag{3}$$

where λ is the learning rate. This is a NP-hard problem considering the binary constraints on u. DPLM [11] suggests

Transform	Train param.	Test param.	Method
Compress*	[0.50 – 0.90] @10	[0.52 – 0.92] @10	OpenCV imencode()
Resize*	[0.6 – 1.2] @10	[0.65 – 1.25] @10	OpenCV resize() Bilinear
Flip	{0, 1}	{0,1}	Numpy
Rotate	[-25 – 25] @15	[-26 – 26] @10	OpenCV warpAffine()
Padding	[0.01 – 0.1] @10	[0.015 – 0.105] @10	Numpy
Sharpness	[0.25 – 4.0] @15	[0.20 – 4.2] @10	PIL ImageEnhance
Gaussian noise	[0.1 – 1.1] @10	[0.15 – 1.15] @10	Numpy random Gaussian
Color enhancement	[0.5 – 2.0] @10	[0.45 – 2.05] @10	PIL ImageEnhance
Shot noise	N/A	[12 – 80] @10	Numpy random Poisson
Impulse noise	N/A	[0.01 – 0.1] @10	SkLearn random salt&pepper
Speckle noise	N/A	[0.15 – 0.35] @10	Numpy random speckle
Gaussian blur	N/A	[1.0 – 3.0] @10	ScikitImage Gaussian filter
Defocus blur	N/A	Radius [2 – 5]@10, std [0.1 – 0.5]@10	OpenCV disk filter
Pixelate	N/A	[0.25 – 0.6]@10	PIL box filter

Table 1. Transformation methods and its train/test parameter ranges used in PSBattles24K and PSBattles360K-S. Notation $[x_1 - x_2]@k$ indicates that k values are uniformly sampled from $[x_1, x_2]$ range where x_1 and x_2 are the lower and upper bound values. * indicates primary methods. The PSBattles24K benign test set contains images transformed using a combination of 2 primary and 1-3 secondary methods selected at random (Flip, Rotate, Padding, Sharpness, Gaussian noise, Color Enhancement). The PSBattles360K-S dataset has 360K images generated using individual transformations, of which a half is created via 6 seen methods (Compression, Rotate, Padding, Sharpness, Gaussian noise, Color enhancement) and the another half via 6 unseen methods (Shot noise, Impulse noise, Speckle noise, Gaussian blur, Defocus blur and Pixelate).

Method	Benchmark I		Benchmark II					
	mAP	**mmAP**	mAP	$\overline{mAP}$	$\boldsymbol{F_{mAP}}$	R@1	$\overline{R@1}$	$\boldsymbol{F_{R1}}$
OSCAR-Net	**0.8898**	**0.7411**	0.7866	0.7283	**0.3782**	0.6635	0.8105	**0.3648**
GNN	0.8807	0.7111	0.7682	0.6929	0.3643	0.6544	0.7720	0.3542
CNN	0.7980	0.6086	0.6924	0.7142	0.3516	0.5639	0.8003	0.3308
GreedyHash [12]	0.6635	0.3456	0.5893	0.3957	0.2367	0.4932	0.4784	*0.2428*
HashNet [2]	0.8093	*0.4031*	0.7354	0.2837	0.2047	0.6291	0.3736	0.2344
CSQ [13]	0.5785	0.2838	0.5104	0.4545	0.2404	0.4291	0.5226	0.2356
DFH [8]	0.3207	0.1595	0.3107	0.6657	0.2118	0.2470	0.7247	0.1842
DBDH [14]	0.6908	0.3339	0.5889	0.3508	0.2199	0.4818	0.4287	0.2268
DSDH [6]	0.6958	0.3280	0.5878	0.3214	0.2078	0.4693	0.4091	0.2186
ADSH [5]	0.3339	0.1887	0.2458	0.6112	0.1753	0.1578	0.7041	0.1289
DPSH [7]	0.8202	0.3917	0.8003	0.2197	0.1724	0.7159	0.2936	0.2082
DSH [9]	0.2416	0.1358	0.1962	0.7523	0.1556	0.1318	0.8274	0.1137
DHN [15]	0.1803	0.0898	0.1737	0.7396	0.1407	0.1291	0.7851	0.1108
wHash [1]	0.5338	0.2274	0.4981	0.1132	0.0922	0.4652	0.1372	0.1059
aHash [1]	0.5764	0.2668	0.5231	0.1114	0.0919	0.4892	0.1382	0.1077
pHash [1]	0.6008	*0.3260*	0.5515	0.0918	0.0787	0.5203	0.1196	0.0972
ISCC [10]	0.6003	0.3252	0.5506	0.0918	0.0787	0.5186	0.1189	0.0967
dHash [1]	0.6164	0.2890	0.5363	0.0681	0.0604	0.4993	0.0818	0.0703
cHash [1]	0.2509	0.1018	0.2866	0.5601	0.1896	0.2284	0.6264	*0.1674*

Table 2. An expanded version of Tab.1 in the main submission, showing the components used to compute overall Benchmark II scores F_{mAP} and F_{R1} via eq. 14 of the main paper. For all metrics, higher is better.

that we can approximate u_{t+1} as the closest discrete point to the continuous $(u_t - \lambda \partial \mathcal{L}_C / \partial u_t)$, that is:

$$u_{t+1} = \text{sign}(u_t - \lambda \frac{\partial \mathcal{L}_C}{\partial u_t}), \tag{4}$$

which can be split into:

$$u_{t+1} = \text{sign}(z_{t+1}), \tag{5}$$

$$z_{t+1} = u_t - \lambda \frac{\partial \mathcal{L}_C}{\partial u_t}, \tag{6}$$

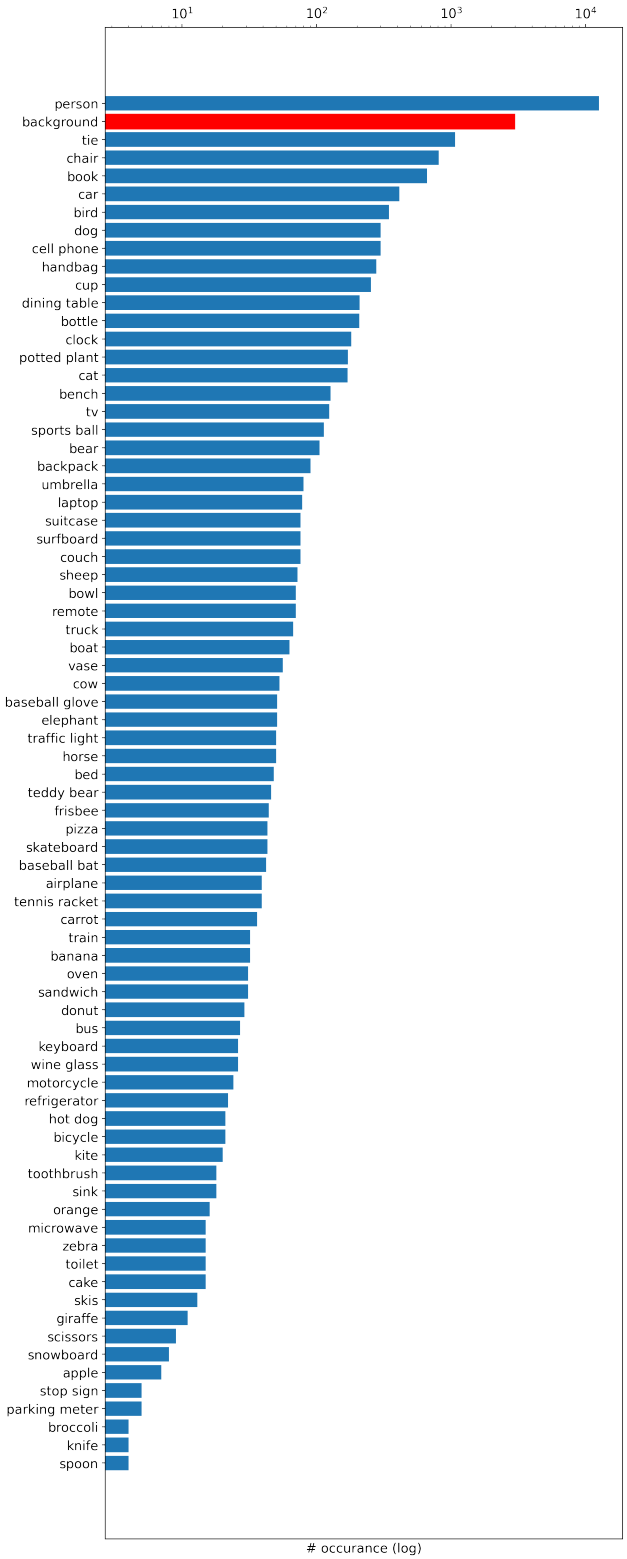

Figure 1. Object occurrence statistics on the PSBattles24K test set using the instance object detection model, Mask R-CNN [3], with objectness threshold 0.5. "Person" is the most popular object and the only class with average number of occurrence greater than 1 in this dataset.

Figure 2. Examples of PSBattles24K original-manipulated pairs.

but the gradient descent rules applies to z as well:

$$z_{t+1} = z_t - \lambda \frac{\partial \mathcal{L}_C}{\partial z_t} \tag{7}$$

$$= (z_t - u_t) + u_t - \lambda \frac{\partial \mathcal{L}_C}{\partial z_t}, \tag{8}$$

By comparing two equations 6 and 8, we can safely allow $\frac{\partial \mathcal{L}_C}{\partial z_t} = \frac{\partial \mathcal{L}_C}{\partial u_t}$ if we can regularize z_t close to u_t element-wise. This is implemented via the second loss term $\mathcal{L}_B = ||z - u||^p$, where the value of p defines the regularization's gradient surface ($p = 3$ in our work and [12], $p = 2$ in [11]). The total loss now becomes $\mathcal{L}(.) = \mathcal{L}_C(.) + \alpha \mathcal{L}_B(.)$, as shown in subsec.3.4. The gradient w.r.t. z can be back propagated as (assume $p = 3$):

$$\frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}_C}{\partial u} + 3\alpha \left|\left|z - u\right|\right|^2. \tag{9}$$

2.2. Triplet+ details

Recall, the formula of Triplet+ loss used in our ablation study (subsec. 4.4 in the main paper submission) is:

$$L(z_i, z_{i+}, z_{i-}) = \max(0, m + d(z_i, z_{i-}) - d(z_i, z_{i+})) + \beta d(z_i, z_{i-}), \tag{10}$$

where $d(.)$ is cosine similarity, m is the margin that defines a distance threshold between the positive and negative pairs ($m = 0.2$), β is a constant weighting the second loss term ($\beta = 0.01$). Unlike standard Triplet, the anchor image i in Triplet+ can be either an original *or manipulated image*. The positive image i_+ is a random benign transformation of i (so it can be either original or manipulated derivatives). The negative image i_- is sampled stochastically from either two sets: same-instance-different-variant (SIDV) and different-instance (DI). For SIDV, the negative is the same image as the anchor, but if anchor is manipulated, then the negative is the original image (and vice-versa). For DI, a different image to the anchor is used. The sets are sampled with ratio 90:10 respectively. This allows the model to learn from negative samples with both small semantic changes (90%) as well as large global differences (10%). As the difference between the original and manipulated pairs is often subtle, the second term $d(z_i, z_{i-})$ was added to explicitly encourage the separation of negative and anchor images. Triplet+ achieves comparable performance with SimCLR+ however its training strategy is more complex (sampling of triplets) and requires more hyper-parameters (β, SIDV-DI ratio) therefore is not favored.

3. Experiments

3.1. Table 1 full results

Tab. 2 extends Tab. 1 in the main submission, including mAP, $\overline{mAP}$, $R@1$ and $\overline{R@1}$ scores for benchmark II. Within this benchmark, mAP and $R@1$ are performance metrics of the benign query set, while $\overline{mAP}$ and $\overline{R@1}$ are scores when querying the manipulated set.

3.2. Visual explanation

In addition to Figure 1 of the main paper, we present further visual explanation examples in Fig. 5, with failure cases. We extend GradCam to visualize the dis-similarity between images by feeding the gradient of an objective function (c.f. eq.14 subsec.4.6) back to the early layers, and observe which areas of the intermediate feature maps are activated most. Note that in eq.14, we freeze the gradient w.r.t. $f(x_+)$ and $f(x_-)$ and compute $\partial \mathcal{L}(x|x_+, x_-)/\partial x$ only. The objective function is illustrated in Fig. 4.

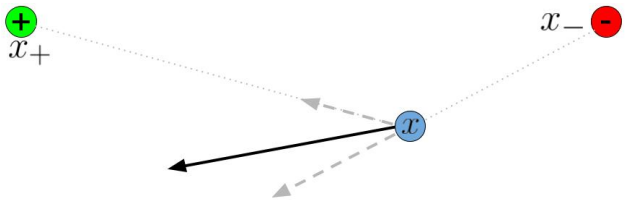

Figure 4. Illustration of the GradCam objective function in eq.14.

References

[1] J. Buchner. Imagehash. https://pypi.org/project/ImageHash/, 2020. 2

[2] Z. Cao, M. Long, J. Wang, and P. S. Yu. Hashnet: Deep learning to hash by continuation. In *Proc. CVPR*, pages 5608–5617, 2017. 2

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proc. ICCV*, pages 2961–2969, 2017. 1, 3

[4] S. Heller, L. Rossetto, and H. Schuldt. The PS-Battles Dataset – an Image Collection for Image Manipulation Detection. *CoRR*, abs/1804.04866, 2018. 1

[5] Q-Y. Jiang and W-J. Li. Asymmetric deep supervised hashing. In *AAAI*, 2018. 2

[6] Q. Li, Z. Sun, R. He, and T. Tan. Deep supervised discrete hashing. In *Proc. NeurIPS*, pages 2482–2491, 2017. 2

[7] W. Li, S. Wang, and W-C. Kang. Feature learning based deep supervised hashing with pairwise labels. In *Proc. IJCAI*, pages 1711–1717, 2016. 2

[8] Y. Li, W. Pei, and J. van Gemert. Push for quantization: Deep fisher hashing. *BMVC*, 2019. 2

[9] H. Liu, R. Wang, S. Shan, and X. Chen. Deep supervised hashing for fast image retrieval. In *Proc. CVPR*, pages 2064–2072, 2016. 2

[10] T. Pan. Digital-content-based identification: Similarity hashing for content identification in decentralized environments. In *Proc. Blockchain for Science*, 2019. 2

[11] F. Shen, X. Zhou, Y. Yang, J. Song, H. T. Shen, and D. Tao. A fast optimization method for general binary code learning. *IEEE TIP*, 25(12):5610–5621, 2016. 1, 4

Figure 3. Examples of benign transformations seen (a-i) and unseen (j-o) by the model during the training. (p) is a mix of several random transformations above (used in the PSBattles24K test set).

[12] S. Su, C. Zhang, K. Han, and Y. Tian. Greedy hash: Towards fast optimization for accurate hash coding in CNN. In *Proc. NeurIPS*, pages 798–807, 2018. 1, 2, 4

[13] L. Yuan, T. Wang, X. Zhang, F. Tay, Z. Jie, W. Liu, and J. Feng. Central similarity quantization for efficient image and video retrieval. In *Proc. CVPR*, pages 3083–3092, 2020. 2

[14] X. Zheng, Y. Zhang, and X. Lu. Deep balanced discrete hashing for image retrieval. *Neurocomputing*, 2020. 2

[15] H. Zhu, M. Long, J. Wang, and Y. Cao. Deep hashing network for efficient similarity retrieval. In *Proc. AAAI*, 2016. 2

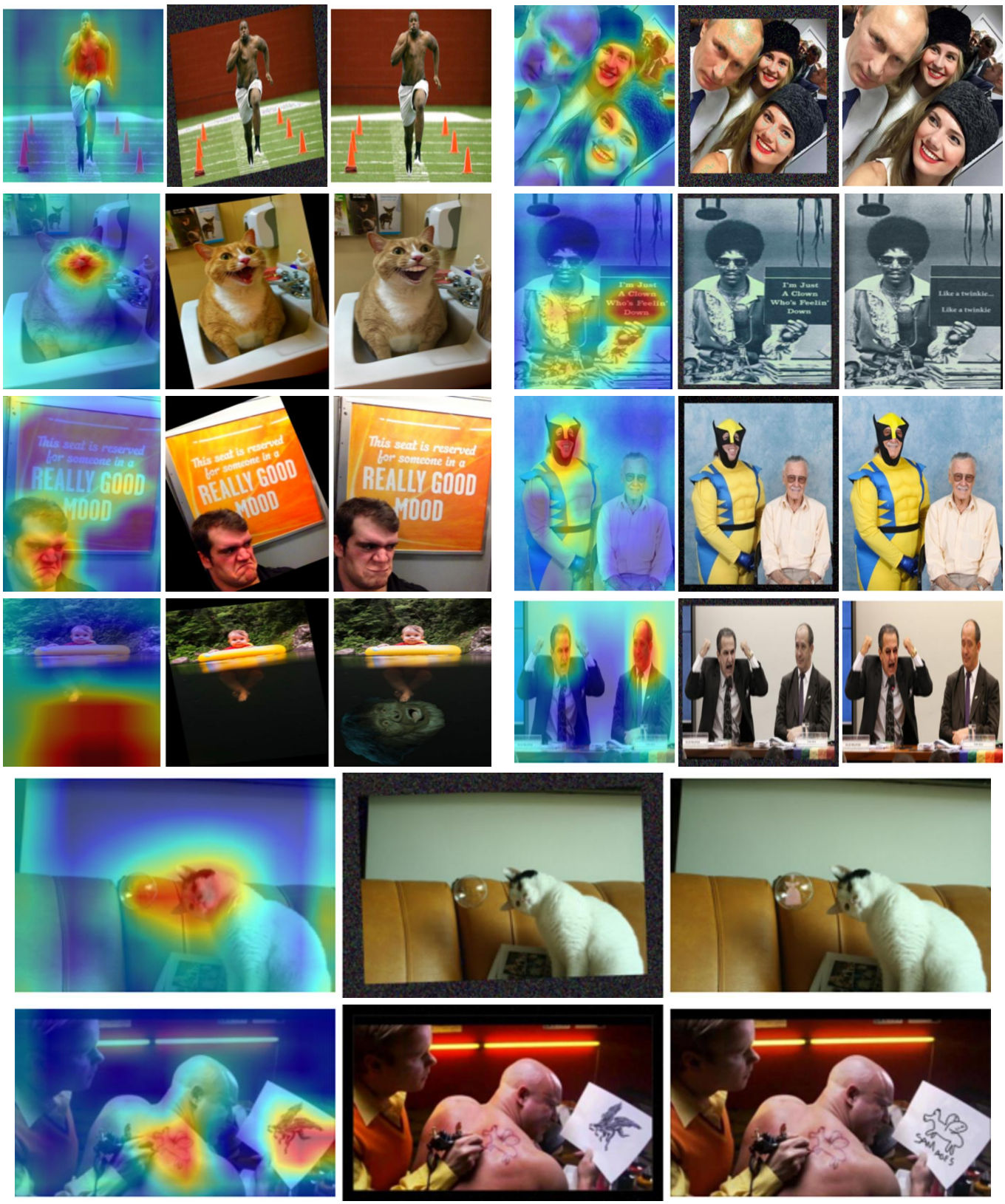

Figure 5. Further visual explanations of dis-similarity via our adapted GradCAM visualization technique. For each example, the visualization of dis-similarity is leftmost. An image with benign transformations is shown middle. The manipulated image is rightmost. The benign transformation is ignored and the manipulation highlighted. The last two rows are failure cases. In the former, the transparent bubble is manipulated but the heat map shifts towards the nearby cat's face. The latter sees the heat map focusing correctly at the paper but incorrectly at the tattoo. In both cases benign transformation is ignored correctly.