

DASH: Visual Analytics for Debiasing Image Classification via User-Driven Synthetic Data Augmentation

Bum Chul Kwon¹ , Jungsoo Lee², Chaeyeon Chung², Nyoungwoo Lee² , Ho-Jin Choi² , Jaegul Choo²

¹IBM Research, Cambridge, Massachusetts, United States

² KAIST, Daejeon, Republic of Korea

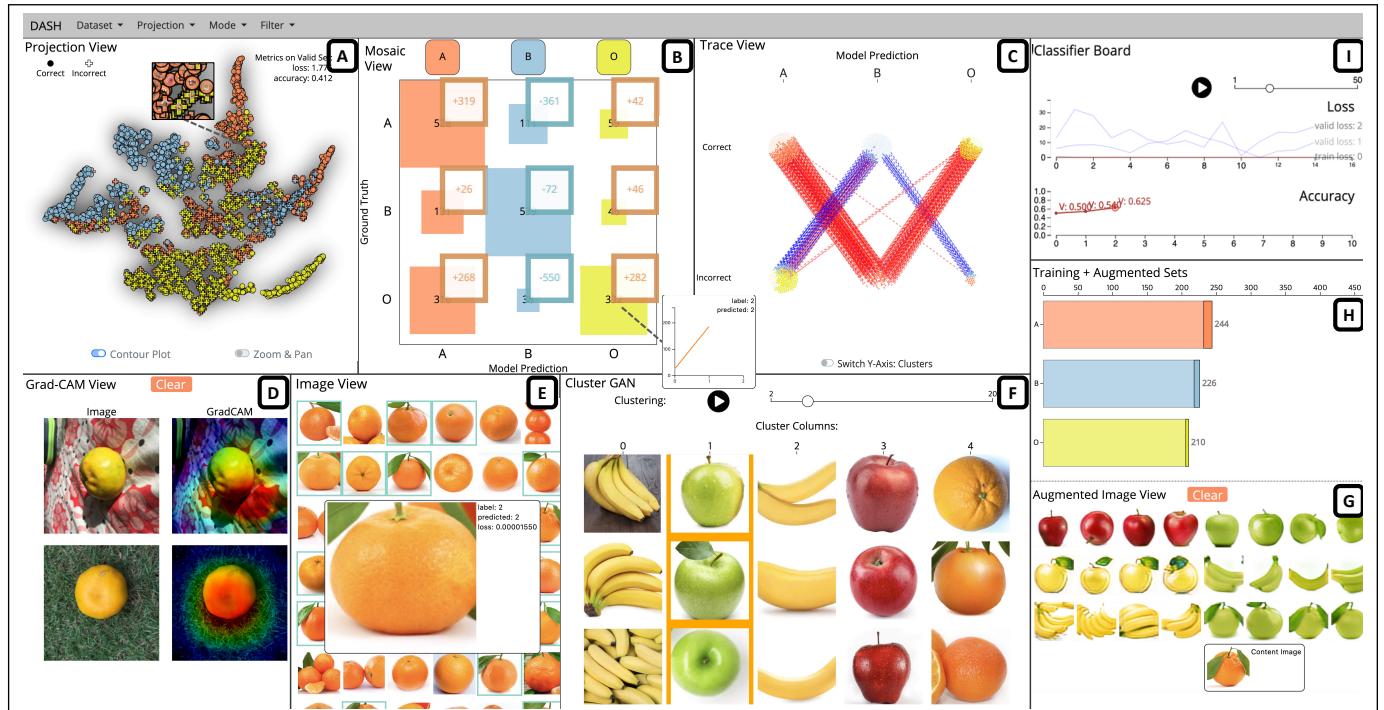


Figure 1: An overview of DASH: (A) Projection View shows the latent space representation of images using t-SNE; (B) Mosaic View summarizes the performance differences between two previously trained classifiers; (C) Trace View shows how the two classifiers predict individual images differently (red: incorrect to correct, blue: correct to incorrect); (D) Grad-CAM View shows the feature importance of images as heatmaps; (E) Image View shows a list of selected images; (F) Cluster GAN View shows clustering results which can be used to translate visual features of images to other images using XploreGAN; (G) Augmented Image View shows newly created images for retraining and (H) shows the summary of the new images; (I) Classifier Board shows the performance of different classifiers.

Abstract

Image classification models often learn to predict a class based on irrelevant co-occurrences between input features and an output class in training data. We call the unwanted correlations “data biases,” and the visual features causing data biases “bias factors.” It is challenging to identify and mitigate biases automatically without human intervention. Therefore, we conducted a design study to find a human-in-the-loop solution. First, we identified user tasks that capture the bias mitigation process for image classification models with three experts. Then, to support the tasks, we developed a visual analytics system called DASH that allows users to visually identify bias factors, to iteratively generate synthetic images using a state-of-the-art image-to-image translation model, and to supervise the model training process for improving the classification accuracy. Our quantitative evaluation and qualitative study with ten participants demonstrate the usefulness of DASH and provide lessons for future work.

CCS Concepts

- Human-centered computing → Visual analytics;

1. Introduction

Image classification models often learn to predict an output class based on irrelevant features co-occurring with the class within images in training data [SMG*20]. We call the undesirable correlation between some visual features and class labels in training data as “data biases,” and refer to such visual features causing the biases as “bias factors.” For example, as illustrated in the previous literature [BCY*20], many images of ‘frogs’ in training data are taken with ‘swamps’ in the background. Image classification models often make mistakes by predicting the class of frogs based on swamps in the background. In this example, swamp is a bias factor that causes the image classification model to be biased for the class label frogs. Though biased models may provide high accuracy in training data, they can result in fatal errors on unseen data beyond training data. Therefore, it is important for data scientists to identify and mitigate biases in models before deploying them.

To resolve data biases, image classification models need to unlearn irrelevant features and learn more important features that are related to the output class. In this context, data augmentation can be a viable solution that can generate synthetic images by artificially combining existing ones into new images. However, it is difficult to automatically identify bias factors of given models and generate images that can effectively target and remove the unwanted correlations. The process is often iterative and labor-intensive because data scientists need to inspect the models to discover bias factors among many potential features, generate images by augmenting existing images, and re-train and evaluate the model so that it can achieve a higher performance in testing data.

In this paper, we conducted a design study with thirteen experts in image classification to develop a visual analytics system for the model debiasing problem. First, we analyzed the user tasks with three experts to understand the model debiasing process. Second, based on the user tasks, we developed a visual analytics system called DASH (**D**ata **A**ugmentation **S**ystem for **H**uman-in-the-loop). DASH allows data scientists to visually identify bias factors among non-trivial visual attributes of images (e.g., colors and object presence). Using DASH, they can also synthesize new images by translating target attributes using a state-of-the-art image-to-image translation technique called XploreGAN [BCYC20] and evaluate the quality of the generated images. Finally, DASH allows them to retrain the model with the newly created images and evaluate the performance of the revised model against previous models. To evaluate DASH, we conducted a user study with ten machine learning experts on two real-world datasets. The results demonstrate that DASH helps data scientists discover and mitigate biases of image classification models.

Our main contributions include:

- We identify user tasks with three experts that capture the user-driven debiasing process for image classification models.
- We present a visual analytics system called DASH that allows users to identify the bias factors, to synthesize new images using image-to-image translation, and to visually supervise the model retraining process.
- We conduct a qualitative evaluation with ten machine experts to show the effectiveness of using DASH for debiasing image classification models.

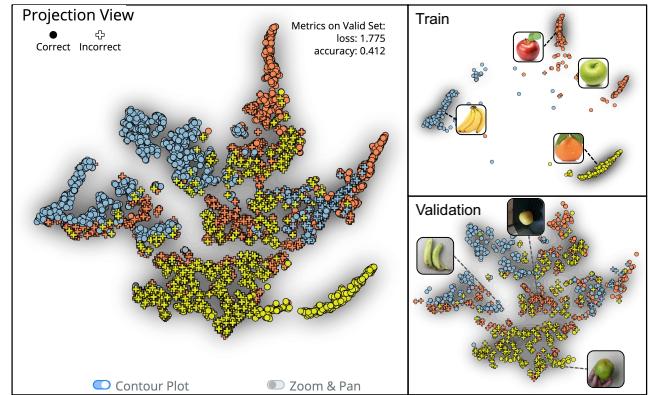


Figure 2: In Projection View, users find clusters of images. Circles and crosses represent correctly classified and misclassified images, respectively. The color inside each point shows its true class label, and the color of the stroke (border) shows its predicted class. Users can zoom in to view actual images.

2. User Tasks: Model Debiasing Process

We derived the tasks based on discussions among the co-authors of this work, who are experts in the fields of computer vision. The following tasks represent high-level objects that users need to perform in order to mitigate biases in image classification models. We assume that users already have trained an image classification model with a training dataset.

T1: Discover data biases in training data. With the initial model, users generate the accuracy of the model on test (unseen) data. The model generates lower accuracy, so the users investigate the source of errors. The errors usually originate from homogeneous distributions of training data, which include unintended correlation between visual attributes of images (swamp) and classes (frog) in training data. Users filter data by the specific class labels that cause errors in test data and then derive irregularities. They often use activation maps like GradCAM [SCD*17] to highlight important regions of images that the model uses for the classification task. After iterative exploration, they hypothesize the unwanted correlation between some visual features and class labels.

T2: Mitigate bias through data augmentation. Once users identified the sources of errors (target visual features to unlearn), users need to generate new images. Users need to generate images of frogs with diverse backgrounds like street, house, and tree. Users can use an image translation technique to change an attribute (swamp) of an image to another attribute (street) from other images without altering other attributes (frog). Image translation models require a “source” image including the class label and a “style” image containing diverse attributes (e.g., street, house, tree) to be fused into the source image. After training the image translation models, users evaluate how realistic the resulting image is. Once satisfied with the quality, users generate new images with diverse backgrounds and label them with the target class for retraining.

T3: Retrain, evaluate, and steer the classifier. With the newly generated images, users retrain the image classification model. In this process, users adjust model parameters and hyperparameters (e.g., epoch number, batch size, learning rate) to maximize the

learning outcome. After retraining the classifier, users evaluate the performance of classification model. In addition to accuracy, users need to assess whether the revised model correctly classify images of the target label. In particular, it is important to test whether the fused visual feature was helpful to resolve the target biases in the model. This process cannot be done at once. Some images that users generated may not help the model to improve their accuracy. In the worst case, some images may decrease the accuracy of the model. Then, users can discard the model instance and retrain the previous version of the model with newly generated images.

3. DASH: Visual Analytics for Data Augmentation

Based on the user tasks, we developed DASH which includes multiple, coordinated views. The following sections describe how the system supports the bias mitigation tasks.

3.1. Projection View

Projection View presents the overall distribution of the latent space in the training and validation sets with a two-dimensional scatter plot where each data point indicates each image. Following the recent studies [WGYS18, CPY*19, STN*16, CRHC18, KEV*18], we created a two-dimensional scatter plot of images using t-SNE [vdMH08]. We used each image's latent space representation extracted from the last convolutional layer of the image classification model. By doing so, the representation of each image captures its high-level semantics (e.g., background colors, objects, texture) [BZK*17]. Figure 2 (a) shows that Projection View separates images by the fruit types in the training set based on colors.

By exploring Projection View, users can discover a noticeable difference in the data distributions between the training and validation sets (**T1**). A contour plot of Projection View indicates the estimated density of image clouds. In addition, users can check whether each data point is predicted correctly or incorrectly with the marker shape, a circle (correct) or a cross (incorrect) respectively, which are colored differently according to its ground truth class (**T1**). We allow users to zoom into a cluster, where each point turns into the actual image at the maximum zoom level. They can also gain the additional information of an image, such as its class label, predicted label, and prediction loss value, in a popup. They can also select a group of data points by using lasso-selection to load the actual images on Image View right below Projection View. The validation set of Figure 2 (a) shows misclassified items, which include images of fruits in their unusual colors (e.g., green bananas).

3.2. Grad-CAM View

Grad-CAM View helps users to understand to which areas of an image the model attributes more importance while making decisions (**T1**). One can also consider using other existing explanation methods, such as saliency maps, Guided BackProp [SDBR15], and Guided Grad-CAM View [RDV*16]. As Figure 1 (D) shows, Grad-CAM View shows the heatmap. By interpreting Grad-CAM heatmaps over the images, users can estimate the regions that are correlated with its predicted class label. For example, as the first column of Figure 3 (a) shows, the model accurately classifies an

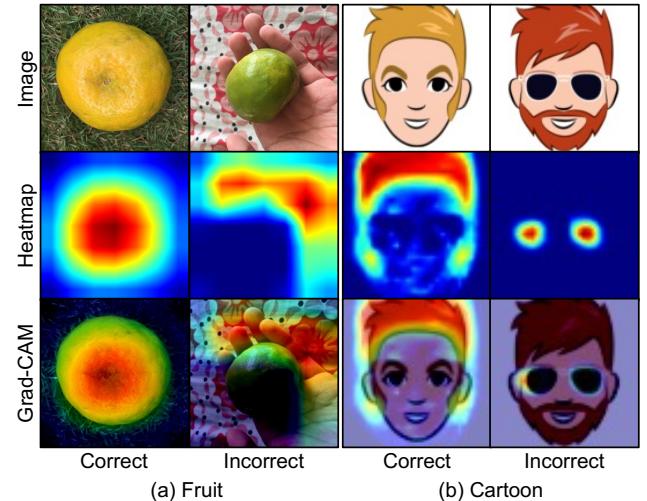


Figure 3: *Grad-CAM View shows the areas that the model gives the most attention as heatmaps (Red: High; Blue: Low). The three rows indicate i) original image; ii) heatmap; iii) original + heatmap. Users can infer where the model should focus on by comparing the correctly classified and misclassified images in Grad-CAM View.*

image of an orange with the focus on the orange. On the other hand, the model misclassifies another image of an orange as an apple in the second column of Figure 3 (a). Grad-CAM View shows that the model focuses on the peripheral region of the image instead of the green orange in the center.

3.3. Cluster GAN View

Cluster GAN View presents groups of images by discovering clusters of the latent space vectors (**T2**). We chose XploreGAN [BCYC20] over other image translation models (e.g., [CCK*18, HLBK18, LTH*18, IZZE17, ZPIE17, LDX*19]) because the model can generate new images without predefined labels for style features (e.g., swamp). XploreGAN clusters images into K groups using K-Means clustering [HW79], then uses the cluster indices as pseudo labels for style features shared by the images within each cluster. Then, XploreGAN can transfer visual features present in the images of a cluster to other images.

As Figure 1 (F) presents, users can run clustering by adjusting the target number of clusters ranging from 2 to 20. Once the clustering completes, it shows a table with columns (clusters) of N representative images that are the closest to the centroid of their respective clusters. Users choose a column (highlighted in orange in Figure 1 (F)) to use the images in the cluster as a target style images and choose an image from Image View as a source image in Figure 1 (E). After generating new images, users can validate the quality of the generated images on Augmented Image View as Figure 1 (G) shows (**T2**).

3.4. Classifier Board, Mosaic View, and Trace View

Classifier Board allows users to visually supervise the retraining process and to navigate the results (**T3**). While retraining, Classifier Board shows loss values of training and validation sets for

every epoch in a line chart (red: training, blue: validation). Moreover, Classifier Board enables users to switch back and forth among the previously trained models (**T3**). By doing so, users can discard unsuccessful retraining attempts if necessary.

Mosaic View shows the differences in classification results between two different models (**T3**). Inspired by prior studies on confusion matrix [KLTH10, Tor13, AHH*14, RAL*17], Mosaic View highlights the cell-level differences using a confusion matrix. Each cell in Mosaic View changes its size in proportion to the number of images in the corresponding cell. This allows users to understand the overall model performance at a glance (**T1**). Users can click on cells of interest by using CTRL + Click, and it filters other views by the images in the cells. For instance, users can select the two cells in (2,1) and (2,3) of [Figure 1](#) (B) to inspect the images of ‘banana’, which are misclassified as ‘apple’ and ‘orange’, respectively.

Trace View summarizes how individual images are predicted differently from different model instances. In Trace View, points in the upper row are correctly classified images while those in the lower row are incorrect ones, as [Figure 1](#) (C) presents. Red lines indicate that the items were previously predicted incorrectly; blue lines show that the items were previously predicted correctly. By browsing across multiple models from different iterations of training, users can gain insights about 1) the changes between different iterations and 2) edge cases, where models constantly make mistakes (**T3**). The insights can lead users to select the image group of interest and to analyze them in more detail to understand why they are predicted differently during different iterations of retraining.

4. User Study

We conducted a user study, where participants perform bias mitigation tasks on two datasets, 1) fruit dataset [Kal18, mes20] (450 images) and 2) cartoon dataset [RBG*20] (680 images), and provide their insights about DASH through interviews at the end. We initially trained models with low accuracy less than 55% with biases on colors of fruits and sunglasses of cartoon characters, respectively. Participants were asked to identify the source of biases and mitigate them using DASH. We recruited ten participants (9 graduate students and a recent graduate; 6 males and 4 females; mean age of 24.5), who are studying/working in computer vision for at least 6 months (10.7 months of working experiences in average). They were instructed with a video and provided with a tool on a toy dataset so that they can learn how to use the tool. Each participant took two hours to complete the study and received \$12.5 per hour for the reward. All users performed their tasks with a Macbook Pro (16-inch, 2019) monitor with a screen resolution of 2560×1600 . We also used one GPU (NVIDIA TITAN Xp 12GB VRAM) for the computation and Chrome (v. 85.0.4183.102) for the browser.

All participants successfully achieved the test accuracy of 90% for the cartoon dataset, and seven out of ten participants (P2, P4, P6-10) reached the test accuracy of 65% for the fruit dataset. In the process, participants retrained 2.1 and 3.3 times on average for cartoon and fruit datasets, respectively. Wilcoxon Signed Rank Test indicates that there was a statistically significant difference in retraining iterations between the cartoon dataset and the fruit dataset ($p=0.048$). Understandably, participants perceived the fruit dataset

more difficult and did more poorly on it than the cartoon dataset because the fruit dataset includes real images of fruits.

Participants shared their experiences with DASH. Here we provide some areas for improvements based on their comments. Three participants (P2, P4, P5) pointed out that DASH requires domain expertise and prior experiences in deep learning and data augmentation. P2 added, “Novice users would take more time to learn how to use DASH. They may need hands-on tutorials for a longer period of time. The instructional video was useful to understand the tool.” Participants also described why some views were difficult to use. P6 reported “While exploring the images, I observed only subtle differences in visual characteristics between clustering results with different K s.” Additionally, we observed that prior knowledge about particular views in DASH may prevent participants from using the views. For example, P7 finished his tasks for both datasets without using Grad-CAM View. He said “I do not trust the robustness and usefulness of GradCAM [SCD*17] because I didn’t find the technique useful in the past. This prior knowledge prevented me from using Grad-CAM View and I trusted my own judgment when I inspected individual images.”

Participants also shared future ideas to improve the bias mitigation processes using DASH. First, three participants (P1, P3, P4) wanted to keep the data augmentation history. While repeatedly generating and discarding images, the participants easily forgot what they already did or what they should do. Thus, the participants wanted to keep track of their previous attempts in order to save time and efforts. Three participants (P3-5) also reported that they wanted to separately analyze images which were frequently misclassified over previously trained models. In that way, they can further investigate why the model keeps making mistakes and derive a strategy to mitigate the specific biases.

5. Conclusions

Our work studies a visual analytic approach to tackle the problem of debiasing image classification models through data augmentation. We designed DASH by conducting a design study with experts in deep learning. DASH integrates the state-of-the-art image translation technique with various views as a unified system which saves time and cognitive efforts of users. In particular, various views of DASH lead users to gain key insights that are required for debiasing. The user study and the quantitative evaluation demonstrate that DASH can provide users with capabilities to effectively solve real-world biases in image data. Future work can investigate ways to help novice users learn how to use the tool. Our experiment is limited because we used a small dataset with relatively simple biases due to time constraints. Future work can also investigate the use of bias mitigation tools like DASH on a large-scale dataset like ImageNet [KSH12] for a longer period of time through a long-term case study [SP06]. Future studies can embed such tools in notebook environments like Jupyter Notebook so that data scientists can develop their own models. Users aim to achieve high quality for translated images, so it will be useful to develop a user interface to retrain XploreGAN interactively. The study shows task analysis, tool design, and user experiments which can be useful to conduct future studies on developing visual analytics tools for bias mitigation in image classification models.

References

- [AHH*14] ALSALLAKH B., HANBURY A., HAUSER H., MIKSCH S., RAUBER A.: Visual Methods for Analyzing Probabilistic Classification Data. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec. 2014), 1703–1712. 4
- [BCY*20] BAHNG H., CHUN S., YUN S., CHOO J., OH S. J.: Learning de-biased representations with biased representations, 2020. 2
- [BCYC20] BAHNG H., CHUNG S., YOO S., CHOO J.: Exploring unlabeled faces for novel attribute discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020). 2, 3
- [BZK*17] BAU D., ZHOU B., KHOSLA A., OLIVA A., TORRALBA A.: Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition* (2017). 3
- [CCK*18] CHOI Y., CHOI M., KIM M., HA J.-W., KIM S., CHOO J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 3
- [CPY*19] CHOI M., PARK C., YANG S., KIM Y., CHOO J., HONG S. R.: Aila: Attentive interactive labeling assistant for document classification through attention-based deep neural networks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), CHI ’19, Association for Computing Machinery. doi:10.1145/3290605.3300460. 3
- [CRHC18] CHAN D. M., RAO R., HUANG F., CANNY J. F.: T-sne-cuda: Gpu-accelerated t-sne and its applications to modern data. In *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)* (2018), pp. 330–338. 3
- [HLBK18] HUANG X., LIU M.-Y., BELONGIE S., KAUTZ J.: Multi-modal unsupervised image-to-image translation. In *ECCV* (2018). 3
- [HW79] HARTIGAN J. A., WONG M. A.: A k-means clustering algorithm. *JSTOR: Applied Statistics* 28, 1 (1979), 100–108. 3
- [IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. *CVPR* (2017). 3
- [Kal18] KALLURI S. R.: Fruits fresh and rotten for classification: Apples, oranges, bananas. Retrieved from <https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification>, 2018. 4
- [KEV*18] KWON B. C., EYSENBACH B., VERMA J., NG K., FILIPPI C. D., STEWART W. F., PERER A.: Clustervision: Visual supervision of unsupervised clustering. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan 2018), 142–151. 3
- [KLTH10] KAPOOR A., LEE B., TAN D., HORVITZ E.: Interactive optimization for steering machine classification. In *ACM SIGCHI Conference on Human Factors in Computing System* (2010), ACM Press, p. 1343. 4
- [KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, Pereira F., Burges C. J. C., Bottou L., Weinberger K. Q., (Eds.). Curran Associates, Inc., 2012, pp. 1097–1105. 4
- [LDX*19] LIU M., DING Y., XIA M., LIU X., DING E., ZUO W., WEN S.: Stgan: A unified selective transfer network for arbitrary image attribute editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 3
- [LTH*18] LEE H.-Y., TSENG H.-Y., HUANG J.-B., SINGH M. K., YANG M.-H.: Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision* (2018). 3
- [mes20] MESHRAM V.: Fruitsgb: Top indian fruits with quality. *IEEE Dataport* (2020). <https://ieee-dataport.org/open-access/fruitsgb-top-indian-fruits-quality>. 4
- [RAL*17] REN D., AMERSHI S., LEE B., SUH J., WILLIAMS J. D.: Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan. 2017), 61–70. 4
- [RBG*20] ROYER A., BOUSMALIS K., GOUWS S., BERTSCH F., MOSSERI I., COLE F., MURPHY K.: XGAN: Unsupervised Image-to-Image Translation for Many-to-Many Mappings. In *Domain Adaptation for Visual Understanding*, Singh R., Vatsa M., Patel V. M., Ratha N., (Eds.). Springer International Publishing, Cham, 2020, pp. 33–49. 4
- [RDV*16] RS R., DAS A., VEDANTAM R., COGSWELL M., PARikh D., BATRA D.: Grad-cam: Why did you say that? 3
- [SCD*17] SELVARAJU R. R., COGSWELL M., DAS A., VEDANTAM R., PARikh D., BATRA D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Oct 2017). 2, 4
- [SDBR15] SPRINGENBERG J., DOSOVITSKIY A., BROX T., RIEDMILLER M.: Striving for simplicity: The all convolutional net. In *ICLR (workshop track)* (2015). 3
- [SMG*20] SINGH K. K., MAHAJAN D., GRAUMAN K., LEE Y. J., FEISZLI M., GHADIYARAM D.: Don’t Judge an Object by Its Context: Learning to Overcome Contextual Bias. pp. 11070–11078. 2
- [SP06] SHNEIDERMAN B., PLAISANT C.: Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization* (New York, NY, USA, 2006), BELIV ’06, ACM, pp. 1–7. 4
- [STN*16] SMILKOV D., THORAT N., NICHOLSON C., REIF E., VIÉGAS F., WATTENBERG M.: Embedding projector: Interactive visualization and interpretation of embeddings. *NIPS 2016 Workshop* (11 2016). 3
- [Tor13] TORKILDSON M. K.: Visualizing the performance of classification algorithms with additional re-annotated data. ACM Press, p. 2767. 4
- [vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. 3
- [WGY18] WANG J., GOU L., YANG H., SHEN H.: Ganviz: A visual analytics approach to understand the adversarial game. *IEEE Transactions on Visualization and Computer Graphics* 24, 6 (2018), 1905–1917. 3
- [ZPIE17] ZHU J., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR abs/1703.10593* (2017). arXiv:1703.10593. 3