# Visualizing Invariant Features in Vision Models

Fawaz Sammani, Boris Joukovsky, Nikos Deligiannis
ETRO Department, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium
imec, Kapeldreef 75, B-3001 Leuven, Belgium
fawaz.sammani@vub.be, bjoukovs@etrovub.be, ndeligia@etrovub.be

***Abstract*—Explainable AI is important for improving transparency, accountability, trust, and ethical considerations in AI systems, and for enabling users to make informed decisions based on the outputs of these systems. It provides insights into the factors that drove a particular machine learning model prediction. In the context of deep learning models, invariance refers to the property whereby diverse input transformations, such as data augmentations, result in similar feature spaces and predictions. The aim of this work is to unveil what invariant features the model has learned. We propose a method coined as *Pixel Invariance*, which measures the invariance of each pixel of the input. Our investigation involves an analysis of four self-supervised models, as these models are pre-trained to learn invariance to input transformations. We additionally perform quantitative evaluation measures to assess the faithfulness, reliability and confidence of the explanation map, and analyze the four self-supervised models both qualitatively and quantitatively.**

## I. Introduction

Explainable AI (XAI) refers to the development of algorithms that can provide a clear and understandable explanation for predictions of machine learning models. This ensures that the decision-making process is transparent and enables users to understand how and why their machine learning models make certain decisions, which also increases accountability. Many of today's AI systems are highly complex and learn using large amounts of data. This makes it difficult for users to trust and rely on these systems, particularly in critical applications such as healthcare, finance, and national security. Moreover, XAI can also allow users to identify and address any data biases or unintended consequences that may arise in AI systems. This can help ensure that AI systems are fair and equitable, and that they operate in a manner that aligns with ethical and societal values. The EU's General Data Protection Regulation (GDPR) has also added the right to explanation [1].

Most local post-hoc attribution methods [2], [3], [4], [5], [6], [7] are aimed towards attributing what features at an input are responsible for a specific prediction. The result is usually a heatmap that highlights which input patterns are relevant for the prediction of a single, specific instance. However, these types of explanations only reveal information about the input features that are relevant for a prediction, but not about the invariant features that the model has learned. Understanding the invariant features the model has learned is just as essential as identifying the crucial features for a prediction. There are several reasons for this: First, the explanation can provide insights into which examples or transformations are more effective in enabling the model to learn invariance. Consequently, this knowledge may guide us in determining which transformations to use. Secondly, invariant features are those that remain constant across different instances of the same problem. By identifying these features, we can ensure that our model is learning generalizable patterns that can be applied to new data. Thirdly, invariant features can help us make models more robust to changes in the input data. For example, if a model is trained to recognize faces, it should be able to do so regardless of factors such as lighting, pose, or facial expressions. Identifying the invariant features that are important for this task can help to ensure that the model is robust to these types of changes.

979-8-3503-3959-8/23/$31.00 ©2023 European Union

In this work, we present *Pixel Invariance*, an explainability technique which visualizes the invariant features learned by a model. The approach quantifies the contribution of each pixel in the input image towards achieving invariance, thereby measuring the extent to which individual pixels convey invariance. Pixel Invariance is model-agnostic, meaning it does not require access to the model structure or weights and instead relies solely on the output prediction of the model. However, it can also be partially model-agnostic, where access to the features of the last layer of the model can enhance its performance. In this work, we study the invariance of a model to different data transformations of the input (*e.g.,* color transformations). Our approach is based on training an approximate interpretable model, such as a linear model, to estimate the similarity between an instance and a vast range of possible data transformations. This asymptotically estimates the invariance of each pixel to all possible transformations of it.

The remainder of this paper is organized as follows: In Section II, we provide a background on the LIME [8] interpretability framework. In Section III, we present our proposed method, coined *Pixel Invariance*. In Section IV, we present our evaluation protocol and conduct experiments including quantitative and qualitative analysis. In Section V, we conclude this work.

## II. Background to LIME

LIME [8] is a popular model-agnostic interpretability technique that can be used to explain the predictions of any machine learning model, regardless of its complexity or architecture. The basic idea behind LIME is to train a simpler, interpretable model to approximate the predictions of the original model within a small, local neighborhood around a specific instance of interest. The interpretable model can then be used to provide insights into why the original model made its prediction for that instance. More formally, let $X$ be the input space, $Y$ be the output space, and $f$ be the original model we want to explain that maps an input $x \in X$ with $p$ features (*e.g.,* super-pixels) to output $y \in Y$. Given an instance of interest $x$, the goal of LIME is to find an interpretable model $g$ with parameters $\boldsymbol{w} \in \mathbb{R}^p$ that approximates $f$ in a

small neighborhood $N(x)$ around $x$. To do this, LIME uses a two-step process. In the first step, it generates a set of $K$ perturbed instances $\bar{X} = \{\bar{x}^i \dots \bar{x}^K\}$ by randomly sampling from the neighborhood $N(x)$ around $x$. Each perturbed instance is then mapped to a prediction $f(\bar{x}^i)$ by the original model $f$. In practice, the perturbed instances $\bar{X}$ are created by randomly masking out super-pixels of the instance. In the second step, LIME trains a local interpretable model $g$ using the perturbed instances $\bar{X}$ and their corresponding predictions as input-output pairs. The goal of the local model is to approximate $f(x)$ in the neighborhood $N(x)$ as closely as possible while being as interpretable as possible. The specific choice of the local model depends on the application and the type of input data, but some common choices include linear regression, decision trees, or sparse linear models. To ensure interpretability, LIME often imposes a sparsity constraint on the local model, which encourages it to use only a small number of input features to make its prediction. This can be accomplished through various regularization techniques such as Lasso regularization. The objective of LIME is given in Eq.1:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{K} \left(f(\bar{x}^i) - g(\bar{x}^i)\right)^2 + \lambda \|\boldsymbol{w}\|_1. \tag{1}$$

Once the local interpretable model $g$ has been trained, it can be used to explain the prediction of the original model $f(x)$ for the instance $x$. This can be done by examining the coefficients of the model, which indicate which features were most important in making the prediction. Alternatively, LIME can generate a visual explanation, such as a heatmap or saliency map, that highlights the regions of the input image that were most important in making the prediction.

## III. Proposed Method

We propose *Pixel Invariance*, a technique to explain the amount of invariance each pixel carries in a given image. Our method is inspired by LIME [8] and is based on training an approximate linear model which can then be easily interpreted. The output of the method is a heatmap highlighting the level of invariance exhibited by each pixel. Unlike LIME, our method requires two inputs, and approximates *invariance* as the similarity between an instance and a wide range of applied transformations to that instance. We note that this technique is model-independent, that is, it can be applied to both CNNs [9] or Transformers [10].

Given an input image $x_1$, we generate an auxiliary dataset $X$ with $B$ location-preserving augments of the image. Thus, $X = \{(x_1, x_2^i, s^i)\}_{i=1,\dots,B}$, where $x_2^i$ corresponds to a transformation of $x_1$, and $s^i$ is the cosine similarity score between $(x_1, x_2^i)$, obtained by feeding the pair $(x_1, x_2^i)$ to the original model we wish to explain. The surrogate model is defined using a *single* linear layer $W \in \mathbb{R}^{3uv}$, where $u, v$ are the spatial width and height of the image and 3 is the number of RGB channels[1]. In what follows, we define $y^i$ as the sigmoid-activated mean of the two outputs of the surrogate model applied separately on $x_1$ and $x_2^i$. By training $W$ to fit $y^i$ with $s^i$, each pixel will be assigned a weight reflecting its importance to the similarity between between $(x_1, x_2^i)$.

[1]images can optionally be resized before flattening them as input to the linear model

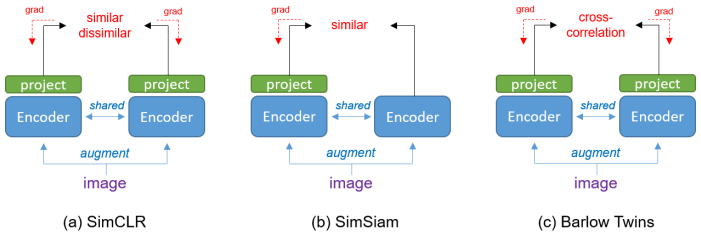


Fig. 1. The different self-supervised models we evaluate. The typical architecture consists of a visual encoder (*e.g.,* ResNet-50), and a projection layer (*e.g.,* MLP).

Additionally, we wish to capture the relationship between the image and the invariant features learned by the model. However, these features are too complex to be modeled by a simple layer, hence we apply a dimension reduction scheme: we collect and stack the extracted features from the projection head output of the model for all $x_2^i$ in a matrix $F \in \mathbb{R}^{B \times D}$, where $D$ is the dimension of the projection head output. We feed these features through a ReLU function and use Non-Negative Matrix Factorization (NMF) [11] to decompose $F$ into the matrices $M \in \mathbb{R}^{D \times V}$ and $H \in \mathbb{R}^{V \times B}$, where $V < (D, B)$ is the number of factorized components. The sample-wise average of $H$ is then used as an extra label to train the surrogate model. We train $W$ using stochastic gradient descent to minimize the following loss function in Eq. (2). It is common to induce sparsity in the weights using $\ell_1$ regularization with a weight of $\lambda$, which we also include during training.

$$\begin{aligned}\mathcal{L}^i = &- \left(s^i \log y^i + \left(1 - s^i\right) \log\left(1 - y^i\right)\right) \\ &+ \alpha \left(y^i - \frac{1}{V} \sum_{v=1}^{V} H_{vi}\right)^2 + \lambda \|W\|_1\end{aligned} \tag{2}$$

Once trained, the weights associated to every pixel can be visualized as a heatmap, where each weight indicates how sensitive the pixel is to transformations and whether it contributes to the invariant features. Having larger values of $V$ harms the performance, as we slowly approach the original number of features $D$. For visually appealing results (not used during evaluation), we trim values smaller than 0.5 to 0 and blur the heatmap with a Gaussian kernel.

## IV. Experiments

In this section, we showcase qualitative examples of our methodology and elaborate on the quantitative evaluation protocol and provide scores for four self-supervised models. We chose to evaluate self-supervised models due to their pretraining objective of learning invariance, which is distinct from image classification models that are trained to perform the specific task of classification. Nevertheless, our methodology can be applied to image classification models to explain the invariance they possess. We evaluate four high-performant self-supervised models developed in the past two years and have gained increasing attention and popularity: SimCLRv2 [12], SimCLRv2 (2×) [12], Barlow Twins [13] and SimSiam [14]. Each of them consists of a vision backbone (*e.g.,* ResNet-50 [15]) and a projection head which learns invariance. Figure 1 shows a diagram describing these four models on a high level.

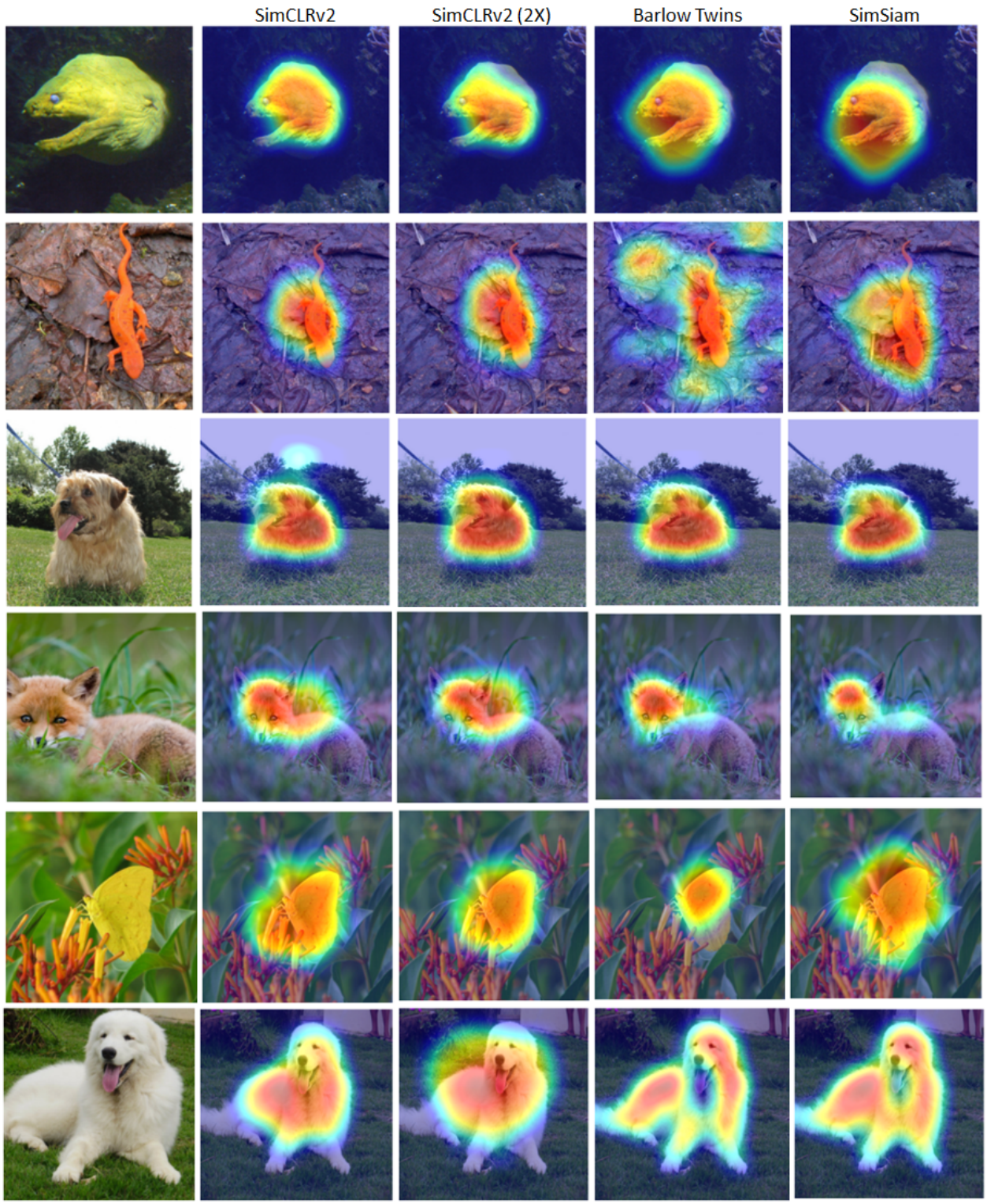


Fig. 2. Qualitative examples of our proposed Pixel Invariance method for 4 different self-supervised models: SimCLRv2 (1X) [12], SimCLRv2 (2×) [12], Barlow Twins [13] and SimSiam [14]. The heatmap displays the degree of invariance of each pixel towards data transformations, with the color red indicating strong invariance and blue indicating weak invariance.

The objective of SimCLRv2 is to maximize the agreement of learned representations between two augmented views of the same image (positive pairs) while minimizing the agreement between other images in the same batch (negative pairs) [16]; namely, augments of the same image should similar, while augments from different images should be dissimilar. SimCLRv2 (2×) doubles the dimension width of the model. SimSiam uses positive pairs only. We refer to these models as *negative pair-free* models. However, several tricks such as an exponentially moving average encoder, gradient stopping and a predictor network are required for these models to function properly and avoid collapsing into a trivial solution. Barlow Twins is another simple self-supervised model with a pretext task based on cross-correlation. As in SimCLR, data augmentation is used to obtain two different views of the same image. Given the outputs from the model, the cross-correlation matrix is measured and trained to be as close to the identity matrix as possible. The objective is similar to how Principal Component Analysis (PCA) operates. For fair comparison, all self-supervised models use the ResNet-50 model architecture [15] as the vision backbone.

### A. Implementation Details

We generate an auxiliary dataset of 1000 samples to train the surrogate model by using a combination of the color jitter transform with a probability of 0.8, grayscale transform with a probability of 0.2 and random erasing with a probability of 0.5 masking 2% - 33% of the image. In Equation 2, we set $\alpha = 2$, $\lambda = 0.2$ and $V = 1$. The linear model is trained for 100 epochs with a constant learning rate of 0.1 using gradient descent.

### B. Quantitative Evaluation

We randomly select 100 samples from the ImageNet [17] validation set for evaluation. We use four quantitative evaluation measures: Insertion, Deletion, Average Drop and Increase in Confidence. The insertion and deletion game [18] is used to evaluate image classification explanations. It is based on the motivation presented in [19]. In the insertion game, we start from a highly blurred image and gradually add image pixels starting from the pixels identified as most important by the explanation algorithm. This creates an output score curve that we use to calculate the Area Under the Curve (AUC). A high AUC generally means a good explanation, as adding relevant pixels forces the model to change its decision. The deletion game starts with the full image and gradually removes important pixels, with a lower AUC indicating a better explanation. The Average Drop and Increase in Confidence are used to evaluate image classification explanations as proposed in [20]. The Average Drop measures the change in confidence between the full image and the highlighted part identified by the explanation map. A lower Average Drop indicates a better explanation. The Increase in Confidence measures how much the model's confidence increases when only the explanation map regions are provided as input. A higher Increase in Confidence indicates a better explanation. In order to accurately assess the invariance of added or removed pixels, it is essential to utilize a similarity measure in conjunction with another image. At each step of the evaluation process, we calculate the similarity score between the added or removed pixels of the evaluated image and a random transformation of that same image. The intuition is that adding or removing invariant pixels will cause an increase or drop in the similarity score when measured with a transformation of that image. This procedure is repeated using 10 different random transformations of the evaluated image, and the resultant scores are subsequently averaged to derive a single value. In Table I, we present scores of the four self-supervised models. We observe that SimSiam and Barlow Twins are more invariant than SimCLRv2, despite SimCLRv2's superior performance on the downstream task of image classification after fine-tuning. This suggests that better invariant models does not always result in improved accuracy upon fine-tuning.

TABLE I. EVALUATION SCORES ON DIFFERENT MODELS

| | Insertion ↑ | Deletion ↓ | Avg. Drop ↓ | Inc. Conf. ↑ |
|---|---|---|---|---|
| SimCLRv2 (1x) [12] | 0.648 | 0.281 | 0.562 | 0.004 |
| SimCLRv2 (2x) [12] | 0.689 | 0.269 | 0.448 | 0.024 |
| Barlow Twins [13] | 0.619 | **0.210** | 0.623 | **0.084** |
| SimSiam [14] | **0.713** | 0.330 | **0.413** | 0.054 |

### C. Qualitative Evaluation

In Figure 2, we provide qualitative examples of the four self-supervised models. Upon observation, it is generally evident that all the self-supervised models studied exhibit a similar capacity to learn invariant features. This outcome implies that the adoption of diverse data augmentations to learning invariance can lead to learning meaningful concepts that are also representative of the object present in the image. It is also evident that the heatmaps generated by SimSiam and Barlow Twins exhibit less scattering and diffusion over the background as compared to those produced by the SimCLRv2 model. This finding aligns with the results presented in the preceding section, which indicated that SimSiam and Barlow Twins outperform the other models in terms of quantitative measures. Based on this correlation, it can be inferred that an explanation that effectively covers the object suggests that the model has achieved greater invariance and has learned more robust concepts.

## V. CONCLUSION

We proposed Pixel Invariance, an explainability technique for visualizing the invariant features learned by a model. By using this approach, we are able to assess the contribution of each pixel in an input image towards achieving invariance. This provides valuable insights into how invariant a given model is to different data transformations. We then presented qualitative examples and evaluated the method quantitatively.


## ACKNOWLEDGMENT

This research received funding from the FWO (Grants G014718N, G0A4720N and 1SB5721N) and from imec through AAA project Trustworthy AI Methods (TAIM).


## REFERENCES


[1] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation"," *AI Mag.*, vol. 38, pp. 50–57, 2017.

[2] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, pp. 1–46, 07 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0130140

[4] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2014.

[5] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, pp. 336–359, 2019.

[6] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *ArXiv*, vol. abs/1706.03825, 2017.

[7] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 111–119, 2020.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, pp. 2278–2324, 1998.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[12] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in neural information processing systems*, vol. 33, pp. 22 243–22 255, 2020.

[13] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *ICML*, 2021.

[14] X. Chen and K. He, "Exploring simple siamese representation learning," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15 745–15 753, 2021.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.

[18] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," in *BMVC*, 2018.

[19] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3449–3457, 2017.

[20] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847, 2018.