RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] Reproducibility study of "Explaining Deep Convolutional Neural Networks via Latent Visual-Semantic Filter Attention"

Erik Buis[1], Sebastiaan Dijkstra[1], and Bram Heijermans[1]
[1]University of Amsterdam, Amsterdam, The Netherlands

Edited by
Koustuv Sinha,
Maurits Bleeker,
Samarth Bhargav

Received
04 February 2023

Published
20 July 2023

DOI
10.5281/zenodo.8173713

Reproducibility Summary

Scope of Reproducibility — In this work, we aim to reproduce the findings of the paper *Explaining Deep Convolutional Neural Networks via Latent Visual-Semantic Filter Attention* (LaViSE). This paper presents a global post-hoc explanation framework for deep learning models that generates semantic explanations for CNN filters. To assess the reproducibility of this work, we verify the main claims made in the paper. More specifically, we evaluate whether the framework creates an accurate mapping to the semantic space, generates words which were not seen in the training data, and is able to generalize to any pre-trained CNN.

Methodology — To reproduce the experiments detailed in the original paper, we first obtained the author's code. However, we had to modify the code for the experiments to be executable, adding missing code, debugging, and making the code more maintainable. Additionally, we evaluated the model's generalizability to other CNNs. The project required a total of 62 GPU hours.

Results — Our recall scores and qualitative experiments validate all claims of the authors: the framework creates an accurate mapping between the visual and semantic space, can analyze any trained CNN regardless of original training data availability, and is able to generate novel out-of-dataset descriptions for filters.

What was easy — The paper was well-written and easy to understand, with helpful figures illustrating the LaViSE framework that aided in the implementation process.

What was difficult — The implementation of the methodology outlined in the paper was particularly challenging due to limited documentation and insufficient details about parts that were not implemented in the existing codebase. Additionally, some experiments could not be recreated because they would require a significant amount of resources to verify.

Correspondence should be addressed to Erik Buis (ebbuis@gmail.com)
The authors have declared that no competing interests exist.
Code is available at https://github.com/ErikBuis/FACT2023 – DOI https://zenodo.org/badge/latestdoi/585223762. – SWH swh:1:dir:3b9cb41cd6d8680801f3b80b29410b641408348e.
Open peer review is available at https://openreview.net/forum?id=nsrHznwHhl.

Communication with original authors — We contacted the authors to clarify missing information and aspects that were not functioning as expected. However, we did not receive a response to our questions.

1 Introduction

Recent research has revealed that visual models like convolutional neural networks (CNNs) can exhibit societal biases based on protected characteristics such as race, gender, and age [1, 2, 3, 4, 5]. Transparency and interpretability of these models are crucial because it helps us make sense of their decision-making processes while identifying implicit biases that may negatively affect the fairness of their predictions.

In order to improve this aspect of deep neural networks, Yang et al. introduce the framework *Latent Visual-Semantic Filter Attention* (LaViSE) [6], which aims to generate textual explanations about the decision-making process of any CNN.

This report aims to replicate the authors' findings, verify their results and perform additional experiments to provide insights into the generalizability of their approach. Our main contributions comprise the following:

1. Enhancing the original code's completeness, reproducibility, maintainability, and efficiency (see section 4.5).
2. Replicating the authors' key experiments to assess their results' reproducibility and evaluate the resources required for replication, including computational cost, development effort, and communication with the authors (see sections 4.2.1, 4.4, 4.6, 5.1 and 6).
3. Extending and evaluating the experiments to verify the authors' claim of their method being generalizable across different CNNs (see section 5.2).

For further reproducibility, we have made all the code that produced the results in this report publicly available[1].

2 Scope of reproducibility

Existing global post-hoc approaches vary in their model explanation method (whether CNNs are explained semantically or visually) and their focus (whether individual filters or the model as a whole is interpreted). However, they tend to rely upon needing access to the original training dataset and may have limited generalizability to other models [7]. The original paper tries to overcome these limitations by proposing a framework that uses a separate dataset to train a separate model, which in turn generates semantic explanations of the filters of the model we want to interpret. This approach is post-hoc, because we can see existing models that we want to interpret as "black boxes" that do not have to be retrained.

This reproducibility analysis will verify the main claims made by the authors, which are:

1. The proposed framework creates an accurate mapping between the visual and semantic space by using generic object detection datasets.
2. The proposed framework can generate novel descriptions for learned filters beyond the categories defined in the reference dataset.

[1] https://github.com/ErikBuis/FACT2023

3. The proposed framework can analyze any trained CNN, regardless of whether or not the original training data is available.

The remainder of this study is organized as follows. Firstly, section 3 provides background information about the framework proposed. Next, we present our approach to reproduce this work in section 4. Section 5 summarizes our results and compares them to the original paper. Finally, section 6 evaluates the ease and difficulty of reproducing the results, discussing which aspects were more straightforward and which posed challenges.

3 Latent Visual-Semantic Filter Attention

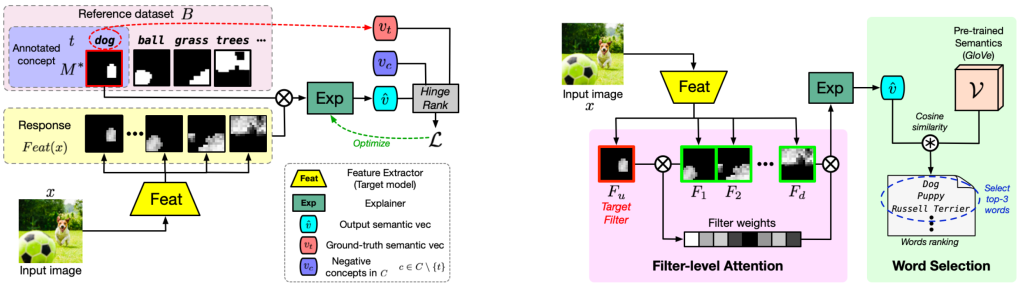
The LaViSE framework comprises two phases: a training phase and an inference phase, which can be seen in figure 1. During these phases, the framework uses two neural networks: a frozen *feature extractor* Feat, which consists of the first L layers of any CNN that was pre-trained on an unknown dataset D, and a *feature explainer* Exp, which is a trainable 2-layer fully connected network (see section 4.1). Finally, GloVe embeddings [8] are used to convert tokens to their semantic representations and vice versa.

3.1 Training phase

The main issue of earlier work was that it required access to the original training data. LaViSE alleviates this by using a *reference dataset* $B = \{(x_i, y_i)\}_{i=1}^{n}$ is used instead. Here, $x_i \in \mathbb{R}^{3\times h\times w}$ is an input image and $y_i = \{(t_j, M_j)\}_{j=1}^{k}$ is a set of k category labels t_j with their corresponding segmentation masks $M_j \in \mathbb{R}^{h\times w}$. The set of all category labels in B is given by C. During both inference and training, the first step is to use the extractor to transform an image and obtain its features with $F = \mathrm{Feat}(x) \in \mathbb{R}^{d\times h'\times w'}$. These features can be divided into d *filter activation maps* $[F_1, \ldots, F_d]$, which have dimensionality $\mathbb{R}^{h'\times w'}$.

Subsequently, the masked filter activation maps are multiplied elementwise with each mask. In turn, the result is passed through the explainer via $\hat{v} = \mathrm{Exp}(F \otimes M_j')$, which corresponds to the $\otimes$ symbol in figure 1a. If we define the semantic representation of the ground truth category to be v_{t_j} and the set of all other categories to be $\{v_c \mid \forall c \in C,\ c \neq t_j\}$, we can formulate a variant of contrastive loss to train the explainer, which is given by

$$\mathcal{L}(\theta) = \sum_{j=1}^{k} \sum_{c\neq t_j} \max(0, 1 - \hat{v}\cdot v_{t_j} + \hat{v}\cdot v_c).$$



(a) Training the explainer

(b) Inference with the filter attention

Figure 1. Training and inference phase of the LaViSE framework [6].

3.2 Inference phase

During inference, we try to explain one target filter F_u at a time. To do this, LaViSE uses a novel technique called *filter-level attention*, which is visualized in figure 1b. In summary, the input to the explainer is F^{att}, which is defined with $F_k^{\text{att}} = a(F_u, F_k) \cdot F_k$ where a represents the cosine similarity measure. The intuition behind this method is that important concepts are often implicitly represented and distributed over many filters. Lastly, after passing these features through the explainer, we search for the GloVe embedding closest to the model's prediction, establishing a mapping between the visual and semantic space.

4 Methodology

This section outlines which steps we undertook to replicate the experiments detailed in the original paper and how we resolved the ambiguity in the original paper. We focus on further reproducibility by describing our approach and design choices in detail.

4.1 Model descriptions

We followed the paper's approach by conducting experiments with the PyTorch [9] implementations of ResNet-18 (11.2M parameters) and ResNet-50 (25.6M parameters) [10] as feature extractors. To test the generalizability of the method, we also trained the explainer with PyTorch's implementation of AlexNet [11] (62.4M parameters) as the feature extractor. All these models were pre-trained on ImageNet [12].

The feature explainer first squishes each filter activation map down to a single value using pooling layers used in the target CNN. This is followed by two linear layers separated by a ReLU activation function. In addition, batch normalization and dropout ($p = 0.1$) are applied before each linear layer. Moreover, when there are d filters, the feature explainer will have $d^2 + 303d + 900$ parameters.

4.2 Reference datasets

The original paper used two reference datasets: Common Objects in Context (COCO) [13] and Visual Genome (VG) [14]. These datasets contain images with one or multiple *annotations*, where each annotation contains an object label and its position in the image, given by a binary mask. Furthermore, COCO contains segmentation masks, while VG contains bounding boxes around objects. General statistics about these datasets are presented in table 1. We adhered to the training/validation split used in the original paper to improve consistency in our experiments.

Pre-processing — As in the original paper, pre-processing is performed exclusively on images in the VG dataset. In their paper, the authors describe that images without box-able annotations are removed, object categories are defined based on WordNet [15] synsets,

Table 1. Reference datasets. The reference datasets used to train the feature explainer. The number of images and annotations are rounded to the nearest thousand.

Dataset	Images			Annotations			Categories			URL
	Train	Val	Total	Train	Val	Total	Train	Val	Total	
COCO	117k	5k	122k	860k	37k	897k	92	92	92	Link
VG	90%	10%	93k	90%	90%	1963k	70%	30%	1128	Link

and annotations of the same category in the same image are combined. Rare categories that appear fewer than 100 times are deleted. While we followed these pre-processing steps to the best of our ability, the results did not perfectly align. More specifically, the number of remaining images and categories we ended up with was around 12% and 7% lower than what was mentioned in the original paper. Further analysis revealed that our version was a subset of the original, which could be due to the order in which the pre-processing steps were applied; for example, our script removes images if they do not contain any annotations after the annotations with rare categories were removed. For exact replicability, we have made this pre-processing script available in our git repository[2].

4.3 Experimental setup

All experiments were designed to assess at least one of the author's claims (introduced in section 2).

Claim 1: The proposed framework accurately maps the visual and semantic space using generic object detection datasets. — To evaluate this claim, we replicated three original experiments as closely as possible. To be precise, we trained the explainer with ResNet-18 and ResNet-50 as the feature extractor, where VG was used as the reference dataset. However, for one of the experiments, we decided to use ImageNet for pre-training instead of COCO for three reasons. Firstly, training it ourselves was not feasible due to time and resource constraints. Secondly, using COCO as both a pre-training and reference dataset could lead to overfitting (as was also hypothesized by the original authors). Finally, pre-training all networks on ImageNet leads to a fairer comparison because only one variable (the reference dataset) is changed between experiments. For all these experiments, we followed the paper's approach in letting the explainer interpret each extractor's deepest filter layer, called `layer4` in both ResNet versions implemented in PyTorch. In contrast, the number of epochs used to train the model was not given in the original paper. Realistically, we had limited resources, so we trained all models for exactly 30 epochs, where the model with the best validation loss was used during inference. By training for this long, we could still recognize trends in the data to assess the original work's reproducibility.

Claim 2: The proposed framework can generate novel descriptions for learned filters beyond the categories defined in the reference dataset. — To address this claim, we performed a qualitative analysis of the models trained on COCO. More specifically, we analyzed whether the model generated predictions that are not present in COCO and, if so, whether they are accurate. To accomplish this, we overlaid the activation heatmap over the corresponding input images. Next, we manually examined whether the model's novel predictions accurately described what the filters were focusing on.

Claim 3: The proposed framework can analyze any trained CNN, regardless of whether or not the original training data is available. — The original authors have limited their presentation of results to the ResNet architecture. To evaluate whether the LaViSE framework can generalize to work with other CNNs, we utilize AlexNet [11] as a feature extractor, as it is widely recognized as a prominent network in the deep learning field [16]. We again trained a network to explain its deepest layer, which is called `features` in PyTorch.

Evaluation measure — The explainer models are evaluated using recall@s, which calculates the ratio of relevant results among a system's top s outputs $W_{u,i}$. More precisely, we compute the recall@s for each of the top p images that activate a certain target filter u

[2] https://github.com/ErikBuis/FACT2023

the most and then average over the obtained scores. Here, we define the top p images for filter u to be $\{x_1^u, \ldots, x_p^u\}$. Intuitively, we are looking for a fraction of ground-truth categories that the model predicted in a certain region of each image. To do this, we first find the region of activations $R_{u,i}$ that the model focuses on when looking at image x_i^u, which is a binary mask that indicates whether each activation is higher than the top 0.5% of activations[3]. Mathematically, $R_{u,i} = [\text{Feat}(x_i^u) > T_u]$, where $p(\text{Feat}(x_i) > T_u) > 0.005$. Furthermore, since we are using the validation set, we can access k annotations corresponding to the image. These can be used to calculate the ground-truth concepts to compare to, which are given by $G_{u,i} = \{t_j \mid \forall_{1 \leq j \leq k},\ \text{IoU}(M_j, R_{u,i}) > 0.04\}$, where IoU calculates the intersection-over-union score between its arguments. Finally, we can calculate the recall as follows:

$$\text{Recall@}s_{u,i} = \frac{|W_{u,i} \cap G_{u,i}|}{|G_{u,i}|}.$$

4.4 Hyperparameters

Due to the learning rate used in the original experiments being unknown, we employed the `DAdaptAdam` optimizer [17], which utilizes a dynamic adaptation technique for the learning rate, eliminating the need for manual tuning. Additionally, we selected the maximum batch size that could fit in the GPU (2048 samples per batch).

4.5 Improving the codebase

In order to reproduce the experiments described in the original paper, we obtained the authors' PyTorch implementation from their public repository on GitHub [18]. However, the code lacked completeness, efficiency, and documentation. Therefore, the first step was to ensure the code was functional by adding missing code and debugging where necessary.

Most prominently, we created functions for calculating the recall measure, plotting activation heatmaps, and pre-processing and loading the VG dataset. Next, we added docstrings and type hints to all functions to improve code readability and catch potential errors. We added the functionality for using a fixed seed, added scripts to download and pre-process both datasets automatically, and made the coding style consistent. In doing this, we aimed to facilitate future maintainability and reproducibility.

Lastly, we improved the code's efficiency to save on required computational resources. More specifically, this was accomplished by creating a faster algorithm for finding the p images that activate certain filters the most, memoizing the results of expensive function calls, decreasing the number of I/O calls to speed up data loading, moving operations from the CPU to the GPU, and parallelizing sequential operations. Especially the inference code underwent extensive re-engineering to achieve substantial performance improvements, resulting in a speedup of 12x.

4.6 Computational requirements

Information about neither the hardware used nor the GPU hours required to perform the experiments was given in the original work. Consequently, we chose the fastest option available to us to perform our experiments: an Nvidia A100 GPU with 40 GB of RAM running on a Google Cloud Platform (GCP) computing instance. Training each model cost 15 hours on average, and to compute the heatmaps and recall score, 30 extra minutes were required. In the end, the total hours spent to obtain all results presented in this report amounted to 62 GPU hours.

[3] This mask is overlaid with the original image to form the *heatmap* shown in figure 2.

Table 2. **Results reproducing original paper.** Recall (R@s) scores of the original paper versus ours.

Model & Layer	Implementation	Original dataset	Reference dataset	R@5	R@10	R@20
ResNet-18	Original	COCO	COCO	0.675	0.728	0.776
`layer4`	Ours	ImageNet	COCO	0.448	0.477	0.497
ResNet-18	Original	ImageNet	VG	0.273	0.353	0.429
`layer4`	Ours	ImageNet	VG	0.163	0.198	0.243
ResNet-50	Original	ImageNet	VG	0.226	0.302	0.373
`layer4`	Ours	ImageNet	VG	0.132	0.164	0.197

5 Results

Table 2 presents our quantitative results, comparing them to the original paper's results. The subsequent sections will assess the validity of the authors' claims when considering these results.

5.1 Results reproducing original paper

Claim 1: The proposed framework accurately maps the visual and semantic space using generic object detection datasets. — Table 2 presents our experiment outcomes and a comparison with the authors' results. While the original work recorded considerably better recall scores on all experiments, we can still see that the main trends are comparable. The rationale behind this is that the recall scores we get are still sufficient to support the claim: for example, the recall@5 score of ResNet-50, which is only 0.132, can intuitively be interpreted as the model predicting the correct category out of VG's 1128 categories around 13.2% of the time. Adding to this, most of our trained models reached their highest validation loss after the last training epoch, indicating that improvements were still being made. Moreover, figure 2, which displays some examples of the explainer's predictions, shows that the framework predicts primarily meaningful and relevant words for the masked objects in the images.

Claim 2: The proposed framework can generate novel descriptions for learned filters beyond the categories defined in the reference dataset. — This claim is validated by figure 2, which illustrates that the model accurately predicts words (shown in green) that are not present in the reference dataset but accurately describes the filters. Remarkably, the filters mostly predict out-of-dataset words, which suggests that the model's capabilities are very generalizable. However, this may be caused by the predictions being morphological variants of words contained in the dataset, which could be an interesting area for future research. Finally, LaViSE can accurately generate multiple semantic categories when explaining some filters that focus on more than one category.

5.2 Results beyond original paper

Claim 3: The proposed framework can analyze any trained CNN, regardless of whether or not the original training data is available. — To assess the validity of this claim, we have tested LaViSE's applicability to AlexNet. Table 3 presents these results quantitatively. Our implementation achieved recall scores comparable to ResNet-18 trained with COCO (shown in table 2), confirming the claim's validity.

Table 3. **Results beyond original paper.** Recall (R@s) score of an additional experiment to assess the generalizability of the method.

Model & Layer	Original dataset	Reference dataset	R@5	R@10	R@20
AlexNet `features`	ImageNet	COCO	0.551	0.571	0.600

6 Discussion

Our results show that the LaViSE framework effectively maps the visual and semantic space, as demonstrated by the recall scores recorded on both ResNet models and our additional test on AlexNet, where LaViSE achieved a comparable score. This supports the first and third claims of the authors (see section 5.1). Additionally, our results confirm that LaViSE can generate novel descriptions for latent representations of filters: the explainer accurately predicts words that are not present in the reference dataset (see section 5.2).

Despite this, the author's recall scores were consistently higher. The most important reason for this was that it was challenging to reproduce the authors' results accurately due to the absence of crucial information regarding the implementation, which was discussed more thoroughly in section 4.5. Moreover, as discussed in section 4.2.1, slightly different pre-processing methods may have affected the training procedure and, in turn, the final results. Furthermore, the number of epochs to train for may have been much higher, which would severely limit the model's ability to perform rigorously.

Even though our results aligned with the author's claims, there are still some limitations and potential areas for improvement. Further experiments could be conducted on a broader range of CNN architectures to evaluate the generalizability of LaViSE to a fuller extent, as our research was limited to AlexNet and ResNet.

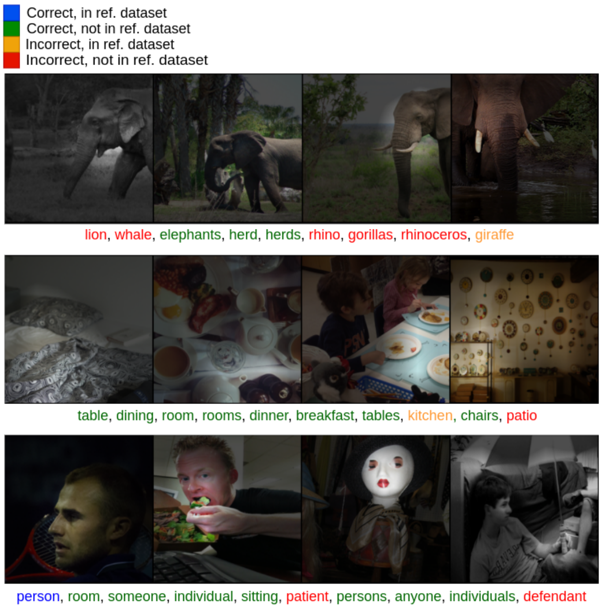

Correct, in ref. dataset
Correct, not in ref. dataset
Incorrect, in ref. dataset
Incorrect, not in ref. dataset
lion, whale, elephants, herd, herds, rhino, gorillas, rhinoceros, giraffe
table, dining, room, rooms, dinner, breakfast, tables, kitchen, chairs, patio
person, room, someone, individual, sitting, patient, persons, anyone, individuals, defendant

Figure 2. Visualization of explainer predictions when interpreting filters 150, 275 and 400 in `layer4` of ResNet-18, with COCO as the reference dataset.

In conclusion, our results support the claims made in the original paper and show that LaViSE is a reliable and effective tool for mapping the visual space to the semantic space. However, more research could be conducted to assess its capabilities more precisely and to find its strengths and weaknesses.

6.1 What was easy

We particularly appreciated the paper's easily comprehensible writing style, which made it easy to understand. The use of figures, particularly those illustrating the phases of the LaViSE framework, was helpful in providing a clear and concise overview of the training and inference processes, which helped us in our implementation.

6.2 What was difficult

The implementation of the methodology outlined in this paper presented challenges, primarily due to the limited documentation and insufficient details regarding parts that were not included in the existing codebase. This made it challenging to comprehend the existing code's functioning and pinpoint the bugs' origin. Furthermore, the bugs impeded the implementation process by causing unforeseen errors and crashes, complicating the determination of whether the issues arose from implementing the new method or were intrinsic to the existing codebase. Lastly, it was only feasible to recreate some experiments; in particular, the experiments that required an extractor to be pre-trained would take significant time and resources to verify.

6.3 Communication with original authors

We sought assistance from the original authors to clarify missing information and aspects that needed to be fixed as expected. Unfortunately, they informed us that they were occupied with other commitments. Nevertheless, we still sought further understanding of the repository and respectfully requested information regarding its completeness, the settings employed for training, and the pre-trained weights on the COCO dataset. Alas, we did not receive any further response.

References

1. J. Buolamwini and T. Gebru. "Gender shades: Intersectional accuracy disparities in commercial gender classification." In: **Conference on fairness, accountability and transparency**. PMLR. 2018, pp. 77–91.
2. A. Das, A. Dantcheva, and F. Bremond. "Mitigating bias in gender, age and ethnicity classification: a multi-task convolution neural network approach." In: **Proceedings of the european conference on computer vision (eccv) workshops**. 2018.
3. A. Kortylewski, B. Egger, A. Schneider, T. Gerig, A. Morel-Forster, and T. Vetter. "Empirically analyzing the effect of dataset biases on deep face recognition systems." In: **Proceedings of the IEEE conference on computer vision and pattern recognition workshops**. 2018, pp. 2093–2102.
4. M. Wang and W. Deng. "Mitigating bias in face recognition using skewness-aware reinforcement learning." In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. 2020, pp. 9322–9331.
5. L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. "Women also snowboard: Overcoming bias in captioning models." In: **Proceedings of the European conference on computer vision (ECCV)**. 2018, pp. 771–787.
6. Y. Yang, S. Kim, and J. Joo. "Explaining Deep Convolutional Neural Networks via Unsupervised Visual-Semantic Filter Attention." In: **2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. 2022, pp. 8323–8333. DOI: 10.1109/CVPR52688.2022.00815.

7. D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach. "Multimodal explanations: Justifying decisions and pointing to the evidence." In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2018, pp. 8779–8788.
8. J. Pennington, R. Socher, and C. D. Manning. "Glove: Global Vectors for Word Representation." In: **EMNLP**. Vol. 14. 2014, pp. 1532–1543.
9. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library." In: **Advances in neural information processing systems** 32 (2019).
10. K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016, pp. 770–778.
11. A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In: **Communications of the ACM** 60.6 (2017), pp. 84–90.
12. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database." In: **2009 IEEE conference on computer vision and pattern recognition**. Ieee. 2009, pp. 248–255.
13. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft COCO: Common Objects in Context." In: **Computer Vision – ECCV 2014**. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Cham: Springer International Publishing, 2014, pp. 740–755.
14. R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations." In: **International journal of computer vision** 123 (2017), pp. 32–73.
15. C. Fellbaum, ed. **WordNet: An Electronic Lexical Database**. Language, Speech, and Communication. Cambridge, MA: MIT Press, 1998.
16. M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari. "The history began from alexnet: A comprehensive survey on deep learning approaches." In: **arXiv preprint arXiv:1803.01164** (2018).
17. A. Defazio and K. Mishchenko. "Learning-Rate-Free Learning by D-Adaptation." In: **arXiv preprint arXiv:2301.07733** (2023).
18. Y. Yang, S. Kim, and J. Joo. **LaViSE**. https://github.com/YuYang0901/LaViSE. 2022.