Incremental Automatic Image Annotation

Luigi Sabetta*[1], Francesco Pelosin*[1], Giulia Denevi[1] and Alessandro Nicolosi[1]

[1]Leonardo Labs, via Tiburtina, Roma, Italy

Abstract

The availability of labelled data is often limited, which hinders the potential of deep learning pipelines in industry. To address this issue, many industries resort to third-party solutions that involve human annotators manually labeling data. However, these solutions are costly, time-consuming, and their accuracy may be questionable. In this paper, we propose an alternative approach that utilizes a deep learning system capable of automatically labeling images with varying levels of supervision from human annotators. Our proposed Automatic Image Annotation system encodes a class using a prototype vector obtained by averaging the projections of images annotated as belonging to that class by a pre-trained backbone. The system efficiently annotates images in real-time without the need to memorize them. It can remember past annotations and also effectively identify new classes. We have developed a web application (link to code) to demonstrate the effectiveness of our approach.

Keywords

Automatic Image Annotation, Incremental Learning, Few-Shot

1. Introduction

In the last years, Deep Learning has achieved impressive results on a variety of tasks, from computer vision [1] to NLP [2] and also as a tool to help natural sciences model our world such as in biology [3] and in physics [4]. This powerful tool is becoming more and more pervasive. But, it comes with a drawback: in the supervised learning realm the training procedure *needs* labeled data. With the pace of our world, data generation does not constitute a problem; the bottleneck lies in the slow and painful annotation procedure.

The standard way to cope with this incomplete data is to rely on human annotators. Human annotation is typically performed by companies that, after a careful interaction with the costumer, agree on a labeling scheme. When such scheme has been defined, the data is forwarded to several humans that subjectively carry the job. This subjective step intrinsically carries low homogeneity on the final labeling. Another problem, is the quantity of data to be curated. Obviously, this reflects on the quantity of time required to accomplish the task which, in the end, results in more expensive services. Due to these reasons, the call to use automatic tools to annotate data is nowadays taking place [5]. Although an important topic, Automatic Image Annotation (AIA) has not received enough attention from the research community. In fact, according to latest reviews on the topic [5, 6] most of the published works are from 2003-2016.

We then tackle such problem by devising a pipeline to assist humans during the labeling process. By exploiting the generalization power of pretrained backbones and a minimal human feedback, we can cut down the time-onerous and error-prone process of image annotation.

Contribution The main contributions of this work are the following.

1. We develop **an Automatic Image Annotation (AIA) system to support humans in labelling a stream of images** by designing an appropriate variant of the method described in [7].
2. **The system is robust to domain-shifts**. Since the prototype vectors representing the different classes are computed by projecting into the embedding space of CLIP [8], the system is resilient to domain-shifts and is almost free of catastrophic-forgetting.
3. **The system is efficient and user friendly**. The disentangled representation provided by CLIP does not require additional expensive training procedures and it reveals to be very effective for this kind of application. The cost to store the protypes for each class is negligible and performs a on-line update which is computationally efficient. Moreover it allows human interaction at different levels. Such an interaction is also facilitated by the development of a web app implementing the system.

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy

luigi.sabetta.ext@leonardocompany.com (L. Sabetta*);
francesco.pelosin.ext@leonardocompany.com (F. Pelosin*);
giulia.denevi.ext@leonardocompany.com (G. Denevi);
alessandro.nicolosi@leonardocompany.com (A. Nicolosi)
0000-0002-0865-5891 (L. Sabetta*); 0000-0002-5548-2620 (F. Pelosin*); 0000-0001-7181-9660 (G. Denevi); 0009-0007-5071-5687 (A. Nicolosi)

* These authors contributed equally to this work

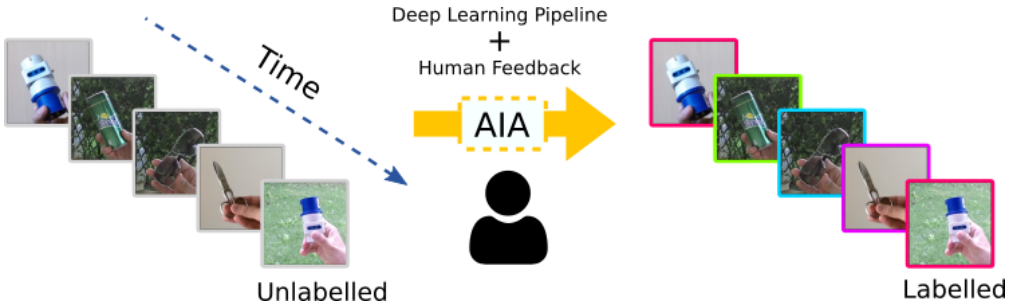

Figure 1: Complete visualization of our pipeline. Each image is processed incrementally, the system then through different level of human feedback, speedup the human on the task of dataset labeling through a pipeline that exploit the powerful feature representation of CLIP [8].

4. **The good performance of the system is confirmed through numerical experiments**. We analyze the proposed system under different datasets and assess the optimal perfomance.

Organization The work is organized as follows. In section 2, we present an overview of the most related work in literature. In section 3, we describe in details the Automatic Image Annotation method we propose. In section 4, we report the numerical experiments we used to test the performance of the method. Finally, in section 5, we draw conclusion and possible future directions.

2. Related Works

Our proposed method in this work combines Automatic Image Annotation (AIA) with Incremental Learning. In this section, we mention the most related literature of these two fields with our work.

Automatic Image Annotation AIA has been the subject of numerous studies in recent years, and the research community has developed a common taxonomy for its different categories [5, 6]. We briefly describe these categories below. One category is generative model-based AIA, which involves learning a joint probabilistic model of image features and words from training datasets. Another category is nearest neighbor model-based AIA, where the tag of the query data point is derived from the most similar data points. For example, in [9], low-level features are combined with distances to find the nearest neighbor. Discriminative model-based AIA methods, on the other hand, view image annotation as a multi-label classification problem [10]. The third category, tag completion models, works by assuming an optimal matrix dataset describing the correspondence between data and labels, and recovering such initial matrix. Lastly, deep learning-based solutions couple feature extractors with side information, such as semantic label relationships, to correctly predict tags [11].
Our approach combines the deep learning and nearest neighbor-based approaches, falling into a mixture of these two categories. By leveraging the strengths of both approaches, we achieve better performance in handling the challenges posed by incremental learning scenarios, as shown in our experimental results.

Incremental Learning Nowadays the need of using Incremental Learning (or Continual Learning) approaches to overcome data shortage is becoming more and more critical. These approaches aim at facilitating the learning process of new tasks, by exploiting the knowledge accumulated by solving previous tasks. However, these Incremental Learning systems have often revealed to be subject to an undesired negative effect: the so-called catastrophic forgetting. More specifically, during the incremental learning process, these models gradually forget the tasks they previously learnt in the past. In quite recent years, the usage of pre-trained backbones has revealed to be a possible and effective solution to overcome this issue, see e.g. [12, 13, 14, 15, 16]. The main idea supported in these works is that pre-training mitigates forgetting by exploiting the disentangling power of the pre-trained backbones.

In the next section we describe in detail the method we propose for Incremental Automatic Image Annotation.

3. Method

Let $\mathcal{X}$ be the images space. We propose a method to automatically label a sequence of images $(x_i)_{i=1}^n \in \mathcal{X}^n$. The proposed method is reported in algorithm 1. As explained in detail below, the algorithm allows the interaction with a human annotator, at different levels.

The algorithm takes in input a sequence of images, a pretrained backbone $\Phi : \mathcal{X} \to \mathbb{R}^d$ mapping the input images in embedded vectors in $\mathbb{R}^d$, a distance $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ among these vectors and a threshold parameter $\epsilon < 0$ used to identify new class labels. For each class label the algorithm memorizes a prototype vector that is computed by averaging the embedded images annotated as belonging to the same class. Such prototype vectors are memorized into a memory $\mathscr{M}$. In order to denote the different class labels, we use incremental integer numbers $\hat{c} = \{1, 2, 3, \dots\}$. The metrics reported in order to evaluate the performance of the algorithm are the classification accuracy A_{class} and the new class detection accuracy A_{det}, which are incrementally computed as new images arrive.

More specifically, at each iteration $i = 1, \dots, n$, the algorithm performs the steps below in order.

1. The algorithm receives the current image $x_i \in \mathcal{X}$ to be labelled.
2. The algorithm computes the corresponding embedded vector $z_i = \Phi(x_i) \in \mathbb{R}^d$ by the backbone Φ.
3. If there exist a prototype vector in the current memory with distance less than ϵ to the current embedded vector z_i, the algorithm associates to the current image the class index $\hat{k}_i$ associated to the closest prototype vector in the memory and increases the frequency of that class represents$\hat{n}_{\hat{k}_i}$ by one. The algorithm also returns the indicator $\hat{nc} = 0$, indicating that the returned class is among the classes already observed in the memory. On the contrary, if there no exist a prototype vector in the memory distant at most ϵ to the current embedded vector, the algorithm associates the current image to a new class label $\hat{k}_i = \hat{c}$, with frequency $\hat{n}_{\hat{k}_i} = 1$. In such a case, the algorithm also returns the indicator $\hat{nc} = 1$, indicating that the returned class is a new class not contained in the actual memory.
4. The human annotator tells to the algorithm if the current image belongs to a previously observed class ($nc = 0$) or a new one ($nc = 1$) and it provides to the algorithm the right class index k_i the current image belongs to.
5. The algorithm uses the feedback received by the human annotator in order to update its memory. Specifically, if the class has been already observed before, the algorithm updates the prototype vector associated to that class by computing an incremental average of the prototype vectors associated to that class. On the contrary, if the class is new, the algorithm adds the new prototype vector z_i to its memory.
6. The algorithm updates the computation of the classification accuracy and the new class detection confusion matrix until that time, by comparing the quantities estimated by the algorithm (denoted by the symbol $\hat{\cdot}$) with the corresponding exact counterparts returned by the human annotator (denoted by the same letters without the symbol $\hat{\cdot}$).

Algorithm 1 Incremental Automatic Image Annotation

Input A sequence of images $(x_i)_{i=1}^n \in \mathcal{X}^n$, an embedder (backbone) $\Phi : \mathcal{X} \to \mathbb{R}^d$, an embedding distance $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$, an out-of-class threshold $\epsilon > 0$

Initialization Memory $\mathscr{M}_1 = []$, class label $\hat{c} = 0 \in \mathbb{R}$, accuracy $A_{class} = 0 \in \mathbb{R}$, confusion matrix $C = 0 \in \mathbb{R}^{2\times 2}$

For $i = 1$ to n

- 1) Observe the image x_i
- 2) Compute the embedding $z_i = \Phi(x_i)$
- 3) **If** there exist $y \in \mathscr{M}_i$ s.t. $d(y, z_i) \leq \epsilon$:
  - Define $\hat{nc} = 0$ (old class)
  - Define $\hat{k}_i = \text{argmin}_{y_k \in \mathscr{M}_i} d(y_k, z_i)$
  - Update $\hat{n}_{\hat{k}_i} = \hat{n}_{\hat{k}_i} + 1$
- **Else**:
  - $\hat{c} = \hat{c} + 1$
  - Define $\hat{nc} = 1$ (new class)
  - Define $\hat{k}_i = \hat{c}$
  - Define $\hat{n}_{\hat{k}_i} = 1$
- 4) Receive user's check: $nc \in \{0, 1\}$, $k_i \in \mathbb{N}$
- 5) **If** $nc = 0$:
  - Update $n_{k_i} = n_{k_i} + 1$
  - Pick up $y_{k_i} \in \mathscr{M}_i$
  - Update $y_{k_i} = \frac{n_{k_i}-1}{n_{k_i}} y_{k_i} + \frac{1}{n_{k_i}} z_i$
- **Else**:
  - Update $n_{k_i} = 1$
  - Define $y_{k_i} = z_i$
  - Update $\mathscr{M}_{i+1} = \mathscr{M}_i \cup \{y_{k_i}\}$
- 6) Update the classification accuracy

$$A_{class} = \frac{i-1}{i} A_{class} + \frac{1}{i} \mathbb{1}_{\{\hat{k}_i = k_i\}} \tag{1}$$

- 7) Update the new class confusion matrix

$$C(nc, \hat{nc}) = C(nc, \hat{nc}) + 1 \tag{2}$$

Return A_{class}, C

Interaction with the human annotator In algorithm 1, the interaction with the human can be also queried less frequently, only at some iterations. In such a

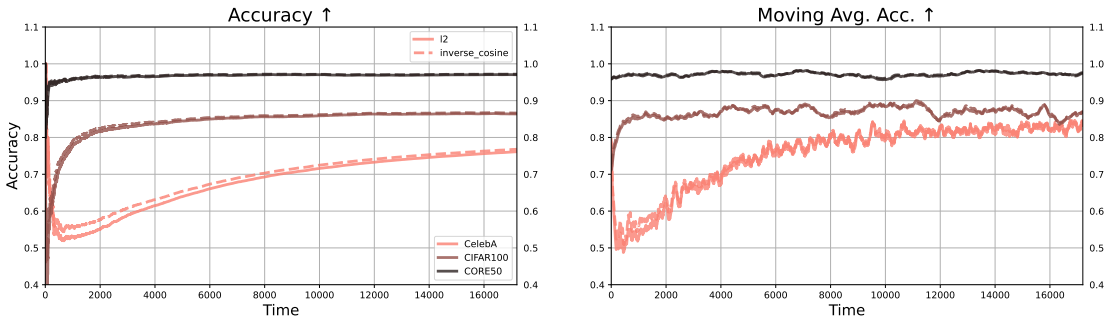

Figure 2: On left we report the accuracy per iteration over all the datasets to show the capabilities of the system on three different datasets. On the right, we report the moving average over variable time frame of the accuracy to show that the system improves over time. In both the plots we compare two distances to compute the closest prototype. As can be seen CosDist provide slightly better performance.

case, at the iterations with no human annotator feedback, the update of the memory can be done in a similar way as described in the step 5) in the algorithm, by replacing the true quantities with the corresponding estimates (denoted by the symbol $\hat{\cdot}$). In section 4 we will propose an analysis on the performance of the system under different quantities of human supervision.

4. Experiments

To assess the performance of our method we defined four different experimental settings and used the datasets below.

- CIFAR100 [17]: the dataset is composed by 50000 train, 32×32 RGB images subdivided in 100 classess with 600 images each. This dataset has been chosen to provide a *comparative benchmark* in line with the research community.
- CelebA [18]: the dataset is composed of 64×64 RGB images divided in 10177 classes; it is composed of 202599 images. This dataset represents a *fine grained* benchmark to assess our system.
- Core50 [19]: the dataset is composed by 164866, 128×128 RGB images of 50 domestic objects divided in 10 classes. Each object appears in 11 different scenarios. We opted for this dataset to provide a more *realistic dataset* benchmark and to test the system under the *domain shift*. In Figure 4 we show the structure of the data.

In all the experiments we implemented algorithm 1 with backbone Φ equal to CLIP [8].

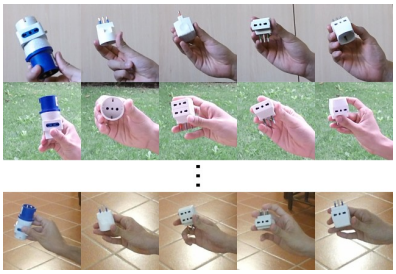

Figure 3: Visualization of one class (socket) of 10 of Core50 dataset. Each row represents a scenario where the same objects appear in different backgrounds. This realistic dataset allows us to stress the system on domain shift setting.

Accuracy and Distance Analysis The first experiment is aimed at measuring the performance in terms of the classification accuracy for the images in the dataset. We did not consider the first occurrence of each class when computing the accuracy. In order to assess the incremental improvement of the algorithm, we plot as well the moving average accuracy with a variable time frame. We implemented algorithm 1 by using two different distances $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$:

- the Euclidian distance ($l2$)

$$d(y_1, y_2) = \|y_1 - y_2\|_2$$

- the cosine distance (*CosDist*)

$$d(y_1, y_2) = 1 - \frac{\langle y_1, y_2 \rangle}{\|y_1\|_2 \, \|y_2\|_2}.$$

The comparison results in terms of accuracy across the three different datasets are presented in Figure 2. The system's performance shows a noticeable improvement over time, starting off with poor accuracy and gradually

	l2 (%)	CosDis (%)
Core50	97.1	97.2
CIFAR100	86.3	86.6
CelebA	77.3	78.0

Table 1
Accuracy for different datasets using *l*2 distance and *CosDist* distance.

increasing its accuracy across all datasets. This behavior is expected, as the centroids need to adjust to the data and "warm up" before delivering optimal performance. For a summary of the numerical accuracy values obtained, refer to Table 1. In the case of the Core50 dataset, the system is highly effective in separating all classes, achieving exceptional performance with just a few centroid updates. These results demonstrate the effectiveness of our approach in tackling real-world classification tasks. It is worth noting that while challenging, the CIFAR100 dataset may not be fully representative of real-world usage. Nevertheless, we report our system's performance on this dataset to facilitate future comparisons. It is worth mentioning that the system requires 2000 iterations before achieving stable labeling on this dataset. The CelebA dataset poses the greatest challenge among the three datasets, as it represents a fine-grained benchmark with a large number of classes and few examples per class. As a result, our system's performance on this dataset is relatively lower than that of the other two datasets. This observation highlights the importance of having a robust system that can align with the data, which requires a larger number of images (around 10k) for this particular dataset. Since the performance for *CosDist* are slightly better and stabler, we choose to use it for all the other experiments.

OOD (new class detection) Analysis In this experiment, we conducted out-of-distribution (OOD) analysis on the Core50 dataset by varying the classification threshold used to determine whether a new class instance is OOD or not. The results are presented as the precision vs. recall curve relative to the confusion matrix in Figure 4. While these results are empirical and may not generalize to different datasets, they provide a starting point for more thorough threshold estimation that could potentially be applicable to unseen datasets.

Domain Shift Analysis In the third experiment, we evaluate the robustness of our pipeline under incremental domain shift on the Core50 dataset. Specifically, we compare the system's performance over a *iid* set of images (similarly to the previous plots) against the same set of images featuring coherent-ordered backgrounds. In other words, all images from the same background

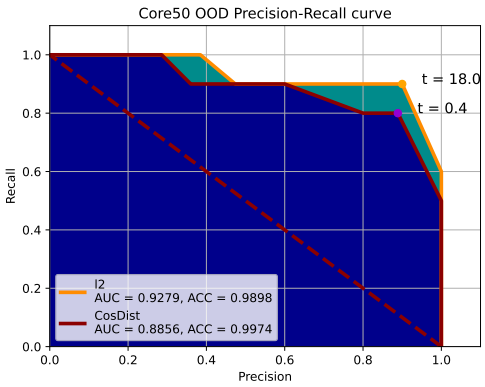

Figure 4: Precision vs recall curve for the two different types of distance, *l*2 (orange curve) and *CosDist* (red curve) over the Core50 dataset.

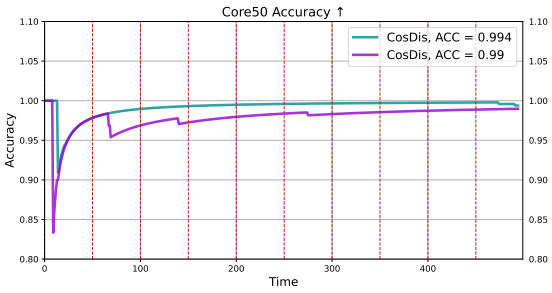

Figure 5: Accuracy as the system encounters new task and experience domain-shift. Each dotted vertical red line represents a change in the scenario therefore a change in the background of the images. The system show resilience on to the distributional shift

scenario are presented before moving to the next scenario. This experiment aims to demonstrate that the CLIP space is resilient enough to cope with distributional shift. As shown in Figure 4, there is only a slight drop in performance when the background scenario changes, which becomes increasingly irrelevant as the centroids fine-tune. These results demonstrate the effectiveness of our approach in handling domain shift, which is a critical aspect of real-world applications.

Self-Annotation In our final experiment, we evaluated the performance of our pipeline under minimal human feedback. We present the results on the challenging Core50 dataset under domain-shift in Figure 4. The findings reveal that even with minimal interaction, our system can achieve good results, indicating that it can autonomously propose correct labels for the input data. These results demonstrate the effectiveness and efficiency

of our approach in minimizing human intervention, making it suitable for real-world applications where manual labeling can be time-consuming and expensive.

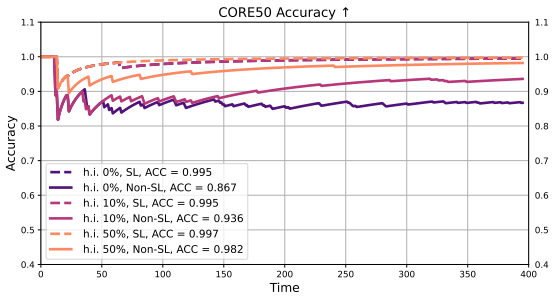

Figure 6: Comparison of different levels of human interaction (h.i.) in the self-labeling (SL) case. As can be seen with 10% of probability of interaction of human feedback the system is able to autonomously label images with small amount of data.

5. Conclusion

In this work, we proposed a deep learning system for automatically annotating a sequence of images with different levels of active human supervision. The system encodes a class by a prototype vector that is computed by averaging the projections of the images annotated as belonging to the same class by a pretrained backbone. The system is computationally efficient and does not require memorizing the images. Our pipeline efficiently keeps memory of the past, and, at the same time, identifies new classes. We also developed a web app for our method and carried extensive numerical analysis to assess the robustness of the system.

In the future, it would be interesting to further investigate the applicability of the proposed method to different scenarios and extend the pipeline with a learnable module. It would be also interesting to provide theoretical certification for its performance.

References

[1] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), NeurIPS, 2020.

[2] R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, Y. Qiao, Llama-adapter: Efficient fine-tuning of language models with zero-init attention, 2023. arXiv:2303.16199.

[3] J. M. J. et al, Highly accurate protein structure prediction with alphafold, Nature (2021).

[4] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, et al., Magnetic control of tokamak plasmas through deep reinforcement learning, Nature (2022).

[5] I. Namatevs, K. Sudars, I. Polaka, Automatic data labeling by neural networks for the counting of objects in videos, Procedia Computer Science (2018). ICTE in Transportation and Logistics.

[6] Q. Cheng, Q. Zhang, P. Fu, C. Tu, S. Li, A survey and analysis on automatic image annotation, Pattern Recognition (2018).

[7] F. Pelosin, Simpler is better: off-the-shelf continual learning through pretrained backbones, in: Transformers 4 Vision Workshop CVPR, 2022.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, in: M. Meila, T. Zhang (Eds.), ICML, 2021.

[9] A. Makadia, V. Pavlovic, S. Kumar, A new baseline for image annotation, in: ECCV, 2008.

[10] G. Carneiro, A. B. Chan, P. J. Moreno, N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, TPAMI (2007).

[11] Y. Niu, Z. Lu, J.-R. Wen, T. Xiang, S.-F. Chang, Multi-modal multi-scale deep learning for large-scale image annotation, 2018. arXiv:1709.01220.

[12] S. V. Mehta, D. Patil, S. Chandar, E. Strubell, An empirical investigation of the role of pre-training in lifelong learning (2021). arXiv:2112.09153.

[13] A. Cossu, T. Tuytelaars, A. Carta, L. C. Passaro, V. Lomonaco, D. Bacciu, Continual pre-training mitigates forgetting in language and vision (2022). arXiv:2205.09357.

[14] T. Wu, G. Swaminathan, Z. Li, A. Ravichandran, N. Vasconcelos, R. Bhotika, S. Soatto, Class-incremental learning with strong pre-trained models, in: CVPR, 2022.

[15] K. Lee, Y. Zhong, Y. Wang, Do pre-trained models benefit equally in continual learning?, in: WACV, 2023.

[16] T. Wu, M. Caccia, Z. Li, Y. Li, G. Qi, G. Haffari, Pretrained language model in continual learning: A comparative study, in: ICLR, 2022.

[17] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Technical Report, University of Toronto, 2009.

[18] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: ICCM, 2015.

[19] V. Lomonaco, D. Maltoni, Core50: a new dataset and benchmark for continuous object recognition, in: CoRL, 2017.