Finding Optimal Models for Image Retrieval Tasks: A Comparative Approach

Ettore Miglioranza 257311
Luisa Porzio 255069
Anthony Tricarico 254957

Abstract

This work addresses the problem of robust image retrieval across domains with a focus on identity-level matching. The task involved retrieving the top-10 most similar images from a gallery, given a query set composed of celebrity faces displayed in visually distinct styles. The study systematically compared pre-trained convolutional and transformer-based models, including ResNet18, EfficientNet-B0, and multiple versions of CLIP. Initial experiments showed suboptimal performance from convolutional models, prompting a shift to CLIP-based architectures. Among these, CLIP ViT-L/14@336px demonstrated superior baseline retrieval accuracy, which was further improved via supervised contrastive fine-tuning. This study highlights the strength of CLIP models for cross-domain image retrieval and the benefits of contrastive fine-tuning in embedding space alignment.

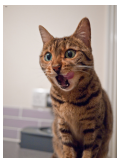
1. Introduction

In recent years, the field of image retrieval has undergone significant transformation, driven by advances in deep learning and the emergence of vision-language models. While traditional approaches have relied heavily on convolutional neural networks (CNNs) trained on large-scale datasets [4, 12], these models often struggle in scenarios involving domain shift, limited per-class examples, or stylistic variation. In contrast, transformer-based architectures, particularly those trained with multimodal supervision, such as CLIP, have shown growing promise in capturing semantically aligned visual representations [11]. This paper investigates these contrasting approaches in a competitive image retrieval setting, aiming to identify architectures that generalize effectively across styles and domains. The challenge posed by identity-level matching of stylized celebrity faces provided a rigorous testbed to evaluate not only retrieval accuracy but also training stability, architectural robustness, and practical deployability. Through a series of experiments, we explore the interplay between model choice, fine-tuning strategies, and system-level optimizations to derive insights into what constitutes an optimal solution for cross-domain image retrieval.

1.1. Task Description

The following study focused on comparing different models for an image retrieval task. The final goal was to retrieve the top ten most similar images. The dataset included a query of celebrity images exposed with a different visual style compared to the gallery set. The resulting model was expected to learn to map visually similar images by showing the same faces across query and gallery sets, identifying the closest embedded representations, in order to retrieve similar images even when they were visually represented with different styles.

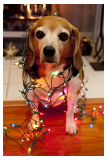
However, this experiment was carried out in two separate moments: first, an initial preparation of different models was made independently before the competition. In the competition setting, the official dataset was disclosed, containing the aforementioned celebrities' images, which contained several classes but few samples for each class, and the prepared models were tested. Finally, upon reviewing the accuracies achieved, these models were perfected, reaching in this way an optimal final model.


(a) Bengal Example 1


(b) Bengal Example 2


(c) Bengal Example 3

Figure 1. Sample images from the "Bengal" class in the Oxford Pet dataset.


(a) Beagle Example 1


(b) Beagle Example 2

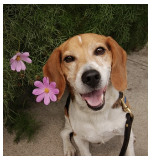

(c) Beagle Example 3

Figure 2. Sample images from the "Beagle" class in the Oxford Pet dataset.

1.2. Overview of Approaches

As a first initial approach, we based our project mainly on two main Deep Convolutional Neural Network models, ResNet18 and EfficientNet. We decided to further investigate CNN models as, historically, they proved to be efficient and widely used for image classification tasks, as they "naturally integrate low/mid/high-level features and classifiers in an end-to-end multilayer fashion" [3]. Indeed, given that the dataset provided prior to the competition included images of animals, this led us to select an additional dataset to understand how well the models considered would perform on such images. Specifically, the Oxford Pets dataset [9] was selected, considering its good intraclass variability of images, similar to the animals dataset provided. We started from convolutional baseline models, specifically, ResNet18 and EfficientNet-B0. Previous research has shown how the ResNet network reached very high accuracy in image classification tasks with large datasets and poorly annotated settings [5]. Also, the EfficientNet model was chosen due to its efficiency and being faster than other baseline models, despite having a more complex architecture [2].

However, both models showed unsatisfactory levels of performance, so we shifted our focus to the CLIP model in Figure 3, which contributed to the final submission made during the challenge. The `CLIP-ViT-B/32` is widely recognized in the field of computer vision as a groundbreaking model, very strong for image classification tasks, specifically for tasks involving face recognition. This stems from the ability of the model to align text and images in the same space embedding, consequently giving high generalization to the model, as it is highly flexible for different tasks without the need for pretraining [8]. As mentioned above, this version of the CLIP model is what led to our final submission during the competition. Following that, after the challenge, we focused on improving even more the performance of this model, perfecting other versions of CLIP, based on the newly acquired dataset of faces.

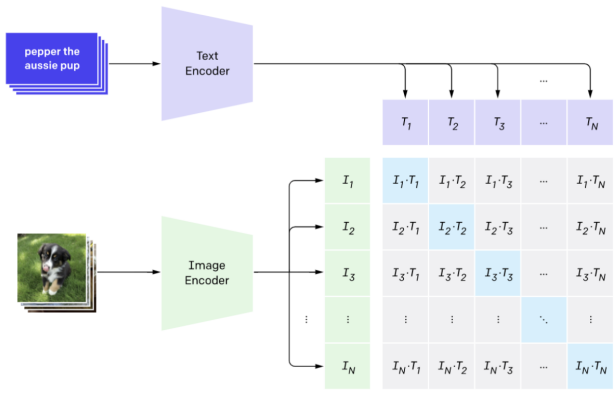


Figure 3. Original ViT-B/32 architecture

1.3. Summary Of Results

After the initial exploratory analysis of models, we focused on the fine-tuning selection. `CLIP-ViT-B/32` using a forward-pass/zero-shot classification procedure resulted in an initial accuracy of 351 (with 1000 representing the best possible score). With the same procedure, the `CLIP-ViT-L/14` architecture was used as shown in Figure 4, resulting in an accuracy score of 651. Therefore, the latter model was then selected for fine-tuning, which was carried out in three training loops. The first training loop resulted in an accuracy score of 362, which improved in the second training loop, reaching a score of 906. The third training loop was instead carried out using 4 different epochs, confirming the final score of 906.

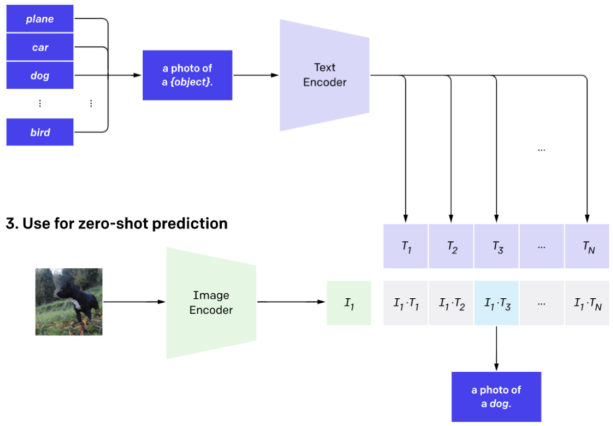


Figure 4. CLIP-ViT-L/14 architecture

2. Models Considered

Before deciding which specific models to pick for the purpose of this project, we first looked at some models that obtained a good response from the academic community and that were previously employed for image retrieval tasks. In our first evaluations we considered three models: EfficientNet [13], ResNet [3], and CLIP [1, 10]. We then decided to evaluate these models on the *Oxford-IIIT Pet Dataset* [9] and check how these models behaved out of the box, without performing any fine-tuning on this specific dataset. To this end, we organized the data (N = 7,350) according to the usual structure employed in similar retrieval tasks (i..e, query and gallery directories). To guide our decisions and to select the models to further analyze, we employed two different metrics: *Accuracy@1* (Equation 1) and *Precision@k* (Equation 2). We define *Accuracy@1* as the number of times a model picks the correct class as its top-ranking output over the total number of queries given as input. *Precision@k*, instead, refers to the number of correct classifications among the top-k ranking elements chosen by the model.

Let N be the number of queries, and for each query $i \in \{1, \ldots, N\}$, let:

- y_i be the ground-truth label of the i-th query,
- $\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{ik}$ be the labels of the top-k retrieved images,
- $\mathbb{1}(\cdot)$ be the indicator function.

$$\text{Accuracy@1} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left(y_i = \hat{y}_{i1}\right) \tag{1}$$

$$\text{Precision@}k = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} \sum_{j=1}^{k} \mathbb{1}\left(y_i = \hat{y}_{ij}\right) \tag{2}$$

We chose to focus on these metrics since we wanted to see how well the models under consideration were able to react to a potentially large and diverse dataset containing images with varying levels of quality. Also, we decided to not perform any fine-tuning to assess the baseline quality of the models without providing them with any domain-specific knowledge relative to the field of application.

Model	Accuracy@1	Precision@k
CLIP	0.9730	0.9027
ResNet	0.9189	0.8568
EfficientNet	0.8919	0.8784

Table 1. Comparison of retrieval performance across different pre-trained models.

From Table 1 we can see that CLIP is the model obtaining the best performance on both metrics, exhibiting a particularly high *Accuracy@1* indicating its capability to consistently pick the correct class as its top-ranking choice, thus representing a more robust model. Moreover, its *Precision@k* indicates that among the top-k picks about 90% of the cases are correctly classified as belonging to the correct class.

On the other hand, ResNet and EfficientNet produced mixed results with ResNet producing the higher *Accuracy@1* score between the two. Yet, it did not prove to be consistent since EfficientNet reached a higher *Precision@k* score, thus showing that even though the top-ranking choice of ResNet consistently classifies the correct class, the remaining $k - 1$ picks are not as likely to be equally good.

3. Evaluation

In this section we systematically evaluate the performances of the chosen model (`ViT-L/14@336px`) in different settings. Additionally, we document how our approach evolved since the day of the competition.

3.1. Results from Models used in the Competition

During the competition, each team member brought one of the three models presented in Section 2 to produce submissions. As expected, the best performance among the three was reached by a `clip-vit-base-patch32` model (not fine-tuned) with a score of 351. Other scores are reported in Table 2.

Model	Score
ResNet	21.7869
EfficientNet	33.7457
CLIP	351.2714

Table 2. Model performance scores (not fine-tuned)

3.2. Fine-tuning with Supervised Contrastive Learning

Overview. We began with suboptimal model and pipeline configurations that resulted in poor performance and high training latency. Through careful selection of the model, we replaced the initial `clip-vit-base-patch32` with the more powerful `ViT-L/14@336px`, resulting in an 85% improvement in zero-shot accuracy, as reported in Table 3. We then introduced a structured fine-tuning approach involving the employment of single-epoch hyperparameters exploration followed by multi-epoch validation. In parallel, we speed up the training pipeline by adding mixed precision floating point, efficient data loading, and GPU memory management. These refinements led to better retrieval performance and a 70% reduction in training time per epoch.

3.2.1 CLIP-Based Fine-Tuning Strategy: Model Selection, Hyperparameter Search, and Training Dynamics

During the early stages of the competition, we adopted the `clip-vit-base-patch32` model and tested its performance in both zero-shot classification and fine-tuning. However, the results were unsatisfactory: the zero-shot baseline achieved only modest accuracy, and our early attempts at fine-tuning led to severe overfitting and poor generalization (see Table 3). These results were mainly attributed to:

- suboptimal choice of hyperparameters (e.g., overly large learning rate and weak regularization);
- I/O bottlenecks in the data loading and image preprocessing pipeline;
- insufficient training monitoring and model selection.

However, qualitative insights from competition revealed that CLIP-based models were effective when properly tuned. By performing more experiments we achieved a better accuracy by adopting more refined training strategies

and better architectural variants. In short, we redesigned our training pipeline and explored improved model configurations.

Model upgrade. We conducted a comparative evaluation of different CLIP architectures and found that `ViT-L/14@336px`, a larger variant of the CLIP Vision Transformer, achieved significantly higher zero-shot performance. As shown in Table 3, switching from `clip-vit-base-patch32` to `ViT-L/14@336px` resulted in a relative accuracy improvement of approximately 85% in the zero-shot setting, without any additional training.

Procedure	Model	Number of Queries	Accuracy
Forward Pass / Zero-shot classification	clip-vit-base-patch32	1455	351.27
Fine-tuning (initial attempt)	clip-vit-base-patch32	1455	17.06
Forward Pass / Zero-shot classification	ViT-L/14@336px	1455	**651.41**

Table 3. Comparison of CLIP model variants in zero-shot and fine-tuning configurations.

Refined fine-tuning strategy. Building on the improved backbone and a newly optimized pipeline (described in the next paragraph), we designed a structured fine-tuning strategy based on two phases:

- **Hyperparameter search via single-epoch training**: we tested multiple configurations with different learning rates and weight decay values to locate a stable and effective training regime.
- **Multi-epoch fine-tuning**: using the best hyperparameters identified during the search, we performed a longer fine-tuning run to monitor convergence and stability across epochs.

This iterative design, combined with better utilization of the GPU, the adoption of mixed-precision floating point during fine-tuning, and a balanced sampling strategy, led to consistent improvements in loss reduction and retrieval accuracy. The detailed results of the fine-tuning experiments are presented in the following paragraphs.

Shared configuration. All training runs shared the following architectural choices and hyperparameters:

- **Encoder**: ViT-L/14@336px (CLIP)
- **Loss Function**: Supervised Contrastive Loss
- **Similarity Metric**: Cosine similarity
- **Optimizer**: Adam
- **Sampler**: BalancedBatchSampler with $k = 4$ classes and $n = 2$ samples.
- **Temperature (SupCon)**: 0.07

Training results. To identify the most promising hyperparameter region, we conducted a series of short training instances, each consisting of a single epoch, using the `ViT-L/14@336px` encoder with supervised contrastive loss. The aim was to assess the impact of varying learning rates and regularization strength (i.e., weight decay) on training stability and retrieval accuracy.

Instance	Learning Rate	Weight Decay	Loss	Accuracy
Instance 1	$3 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	0.6166	362.20
Instance 2	$1 \cdot 10^{-6}$	0.01	0.3731	**906.74**
Instance 3	$1 \cdot 10^{-6}$	0.1	**0.0871**	895.81
Instance 4	$1 \cdot 10^{-7}$	0.01	0.9370	865.84

Table 4. Single-epoch fine-tuning runs with different hyperparameters.

The first configuration (i.e., the first training instance), characterized by a relatively high learning rate and weak regularization, produced poor results with high loss and low accuracy. In contrast, the second training instance provided a substantial improvement in both metrics, with a loss of 0.3731 and an accuracy of 906.74 (as it is possible to see from Table 4). This indicated that reducing the learning rate and increasing the weight decay led to more stable and effective training. The third training instance pushed regularization further by increasing the weight decay to 0.1, achieving the lowest loss (0.0871), but at the cost of a slight drop in accuracy, likely due to over-regularization. To explore the behavior of more conservative learning dynamics, the fourth training instance decreased the learning rate to $1 \cdot 10^{-7}$. However, this configuration resulted in underfitting: the loss increased to 0.9370 and accuracy dropped to 865.84, suggesting that learning was too slow to make significant progress in a single epoch. Based on this exploration, the second training instance offered the best trade-off and was selected as the configuration to be further validated with multi-epoch training.

We then performed a 5-epoch fine-tuning using the selected configuration to evaluate the training dynamics across epochs.

Epoch	Loss	Accuracy	LR	Weight Decay
1	0.4098	908.59	$1 \cdot 10^{-6}$	0.01
2	0.2056	900.21	–	–
3	0.1324	903.85	–	–
4	0.1189	894.36	–	–
5	**0.0842**	905.64	–	–

Table 5. Full training loop with 5 epochs using the best hyperparameter configuration from Table 4.

During the 5-epoch training loop, the model progressively improved in terms of loss, while accuracy fluctuated within a narrow band. The best overall accuracy was reached in

the first epoch, while the lowest loss occurred in the fifth epoch, indicating improved embedding separation without necessarily improving retrieval ranking. This suggests that additional hyperparameter tuning or changing in the system architecture and the components may be required to further improve retrieval performance beyond the current accuracy ceiling.

3.2.2 Bottlenecks Encountered During Training and Optimization Steps

At the beginning of the project, the training loop was significantly slower than expected. This slowdown was due to several bottlenecks in data loading, memory management, and computation flow. Below, we describe the main causes and the corresponding optimizations that were implemented to improve the efficiency of the training process.

Main bottlenecks identified. Initially, our training pipeline suffered from several performance issues. First, image pre-processing during training was inefficient. Each image was individually loaded using `PIL.Image.open()` and processed on the fly with the `CLIPProcessor`, resulting in high I/O latency and substantial CPU overhead. Second, all training operations were conducted in full 32-bit precision (`float32`), which significantly increased memory consumption and slowed down computations. Third, we had included unnecessary calls to `torch.cuda.empty_cache()` within the training loop, which not only introduced overhead but also contributed to memory fragmentation without tangible benefits. Lastly, the `DataLoader` was configured with `num_workers=1`, meaning all data loading and preprocessing occurred in the main process, severely underutilizing the available CPU resources.

Optimizations applied. To address these inefficiencies, we implemented several targeted optimizations. We enabled Automatic Mixed Precision (AMP) using `torch.cuda.amp`, which allowed us to reduce memory usage and accelerate training, particularly for the computationally intensive ViT-L/14 backbone. We reconfigured the `DataLoader` by setting `num_workers=4` and enabling both `pin_memory=True` and `persistent_workers=True`, thereby enabling parallel data loading and efficient non-blocking transfers to the GPU. In particular, enabling the pinned memory of the CPU allows overlapping data loading and calculation on the GPU, improving overall throughput. The redundant cache-clearing operation was removed from the batch loop to reduce unnecessary overhead. Furthermore, during the evaluation phase, we moved embeddings to the CPU only after model inference, helping to prevent GPU memory saturation while computing similarity scores between gallery and query sets. This move is necessary mainly because subsequent operations, such as concatenating the embeddings and calculating similarities, are performed by libraries operating on the CPU. Leaving the embeddings on the GPU would not only be inefficient, but would also lead to unnecessary utilization of the video memory, with the risk of saturating it unnecessarily.

Optimizations results. These optimizations resulted in a significant improvement to the training pipeline, reducing per-epoch training time by approximately 70%. They also stabilized GPU memory usage and boosted overall throughput. Together, these enhancements were essential for making the entire training and evaluation process both efficient and scalable.

Condition	Time per Epoch	Relative Improvement
Before Optimization	$\sim$10 minutes	—
After Optimization	$\sim$3 minutes	**70% faster**

Table 6. Training time per epoch before and after optimization.

4. Discussion

Our study set out to identify the most effective model for cross-domain image retrieval, with a specific focus on identity-level matching under domain shift conditions. The initial performance gap between convolutional models (ResNet18, EfficientNet-B0) and transformer-based models (CLIP variants) underscores a key finding: while CNNs remain strong contenders for general visual tasks, their capacity to generalize across stylized domains without explicit supervision is limited. This was particularly evident in our low retrieval scores during the competition phase, where CNNs failed to adapt effectively to the stylized celebrity face dataset.

Conversely, CLIP-based models demonstrated remarkable resilience to style variations, with even the baseline `ViT-B/32` achieving a far superior zero-shot retrieval performance. This confirmed CLIP's inherent advantage in aligning visual representations in a semantically meaningful embedding space, especially for tasks involving face recognition and style variance. Our experiments further validated the hypothesis that transformer-based encoders trained on multimodal data provide stronger generalization for retrieval tasks.

Moreover, our work exposed several practical bottlenecks: inefficient data loading, poor GPU utilization, and training instability due to hyperparameter misconfiguration. Once those were addressed, this led to a 70% reduction in training time and a more stable fine-tuning pipeline. This underscores the need for a systems-aware perspective in

deep learning research, where engineering decisions can be as impactful as modeling choices. In sum, our work illustrates both the power and the limits of pre-trained vision-language models for image retrieval. While CLIP offers strong out-of-the-box performance and fine-tuning potential, careful attention to training strategy, hyperparameters, and infrastructure remains critical to fully unlock its capabilities.

Code Availability. The full code to reproduce the analysis and review the development history for this project is available in a dedicated GitHub[1] repo.

5. Conclusion

In this section we outline the main pitfalls of our current implementation while providing some hints as to how those could be mitigated and potentially solved with additional development.

5.1. Limitations

We conclude the discussion of our experiment framework on CLIP's ViT encoder with a more in-depth analysis of the limitations of our current development, mainly due to time constraints. Firstly, the dynamic fine-tuning strategy to find the best setting of the hyper-parameter region has some important limitations due to the temperature hyper-parameter being held constant in the search for the optimal region. Indeed, the temperature value of 0.07 is kept constant and the contrastive loss is not optimized. Another fundamental limitation is the loading strategy of the training data. The balanced loading class implemented to load batches of images to perform anchor-positive-negative matching is rather rigid, leaving out classes with a small number of samples. Furthermore, there is no strategy for image augmentation or dynamic sample sizing in the batching strategy. These have implications on the scoring mechanism. Finally, a caching strategy could be useful to reduce the loading time of images on the device, thus reducing the overall training time. Concerning the loss strategy, due to time constraints, we only used one framework, the one we learned to be working best for CLIP fine-tuning according to the literature [6]. Other loss frameworks could be tested. As far as the chosen optimizer is concerned, Adam seems to fit the scope of the project well, allowing us to significantly improve our baseline accuracy. Having said this, we are using a regularization procedure with weight decay, which has proved less effective due to Adam's adaptive normalization. For this reason, we would like to point out this limitation of our project in its current state and will propose a solution in the section on future improvements.

[1] https://github.com/anthony-tricarico/I2ML_ImageClassifier

5.2. Future Work

Building on the current implementation, several improvements could enhance the robustness and performance of our CLIP-based retrieval system. A primary direction is to extend the current hyperparameter search strategy by including the temperature parameter, which was fixed at 0.07. Varying this value in a controlled range (e.g., [0.05, 0.20]) may help better shape the similarity distribution in the contrastive learning process. With regard to data loading, the current batch sampling strategy could be improved by introducing a more flexible sampling mechanism, including low-sampling classes and starting with easier training examples and gradually introducing more difficult ones, including visually similar images of different classes that are more difficult to distinguish. We also plan to enrich the training pipeline with data augmentation techniques, such as random cropping, flipping, and color jitter, to increase variability and improve generalization. On the optimization side, we aim to switch from Adam to AdamW, which more effectively handles weight decay by decoupling it from gradient normalization [7]. Furthermore, adding more regularization techniques could be a valid way to enhance our model performance (e.g., dropout). Finally, future iterations will explore combining the current loss function with additional objectives like triplet or center loss, using fixed or simple weighted combinations, to better capture both inter-class and intra-class relationships in the embedding space.

6. Workload Table

In this section we detail individual contributions for this submission highlighting the main tasks carried out by each team member.

- **Ettore**: Ettore took care of developing and systematically optimize the code implementation of the fine-tuning pipeline for the CLIP-based model. He also experimented with different hyperparameters and implemented a dynamic fine-tuning strategy to find the optimal combination of hyperparameters to use for the optimized model. Finally, he delved deeper into the limitations of the current implementation and thoroughly investigated potential future improvements.
- **Luisa**: Luisa focused on the selection of the preliminary models to be used by investigating the available literature, working on the EfficientNet model used in the competition. She also contributed to finding available and commonly used datasets for the ensuing model evaluation step.
- **Anthony**: Anthony worked mainly on the evaluation of out-of-the-box performances of the selected models on external datasets. He led the evaluation and design of the metrics to be used for performance assessment (as seen in Section 2). He also reviewed the content on GitHub

to make sure the results obtained during the competition were reproducible.

References

[1] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Effective conditioned and composed image retrieval combining CLIP-based features. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21434–21442, New Orleans, LA, USA, 2022. IEEE. 2

[2] Raafi Careem, Md Gapar Md Johar, and Ali Khatibi. Deep neural networks optimization for resource-constrained environments: techniques and models. *Indonesian Journal of Electrical Engineering and Computer Science*, 33:1843, 2024. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[5] Yujie Jing, Chen Li, Tianming Du, Tao Jiang, Hongzan Sun, Jinzhu Yang, Liyu Shi, Minghe Gao, Marcin Grzegorzek, and Xiaoyan Li. A comprehensive survey of intestine histopathological image analysis using machine vision approaches. *Computers in Biology and Medicine*, 2023. 2

[6] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 6

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 6

[8] Nhan Luu. Clip unreasonable potential in single-shot face recognition. 2024. 2

[9] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021. arXiv:2103.00020 [cs]. 2

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 1

[12] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 1

[13] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020. arXiv:1905.11946 [cs]. 2