https://doi.org/10.31891/2219-9365-2025-81-51
UDC 004.85:004.93

DASHENKOV Dmytro
Kharkiv National University of Radio Electronics
https://orcid.org/0000-0001-9797-1863
dmytro.dashenkov@nure.ua
SMELYAKOV Kyrylo
Kharkiv National University of Radio Electronics
https://orcid.org/0000-0001-9938-5489
kyrylo.smelyakov@nure.ua

# METHOD OF EXPANDING IMAGE CLASSIFICATION MODEL WITH TEXT SUPERVISION

*This paper describes a method that addresses the problem of zero-shot image classification within an unbound domain. The goal of the research is to create a new method for expanding the set of classes supported by an image classifier model pre-trained on a large amount of data, without additional training of the model. An additional condition is the possibility to apply the proposed method to any conventional image classifier model, regardless of its architecture. The described method uses the additional information about objects in images obtained from text captions of images and descriptions of the classes. Text data is collected from open sources. An experiment is conducted to demonstrate the ability to generate part of the weights of the image classifier model by retraining a separate natural language processing model in order to add support for new image classes. For this, the classifier model is considered as a combination of an image encoder and a classifier layer that converts the vector representation of the image into class probabilities. When considering the mathematical model of the classifier layer, the task of creating a model without further training is reduced to the task of generating a vector of a predefined size. This transition allows to train a language model to generate weights for the image classifier model and add new classes. The resulting model demonstrates an acceptable level of accuracy on new classes with an average F-score of 0.731 for new classes, with an F-score of 0.844 for classes trained by the conventional method. Additionally, it is found that generating multiple weight vectors and using their average for classification allows to improve the quality of classification compared to using individual generated vectors.*

*Keywords: machine learning, artificial intelligence, natural language processing, image classification, neural networks, computer vision, information technology.*

ДАШЕНКОВ Дмитро, СМЕЛЯКОВ Кирило
Харківський національний університет радіоелектроніки

# МЕТОД РОЗШИРЕННЯ МОДЕЛІ-КЛАСИФІКАТОРА ЗОБРАЖЕНЬ З ТЕКСТОВОЮ СУПЕРВІЗІЄЮ

*Дана робота описує метод розглядає проблему класифікації зображень без донавчання в рамках необмеженої предметної області. Ціллю дослідження є створення нового методу розширення множини класів які підтримує модель-класифікатор зображень, попередньо натренована на великому обсязі даних, без додаткового тренування моделі. Додатковою умовою роботи даного методу є можливість застосовувати його до будь-якої класичної моделі-класифікатора зображень, незалежно від архітектури моделі. Описаний метод працює завдяки додатковій інформації про об'єкти на зображеннях, отриману з текстових описів зображень та самих класів. Текстові дані зібрані з відкритих джерел. Продемонстрована можливість, завдяки донавчанню окремої моделі з обробки природної мови, генерувати частину вагів моделі класифікатора зображень, таким чином аби додавати підтримку для нових класів зображень. Для цього, модель-класифікатор розглядається як комбінація енкодера зображення та шару-класифікатора, що перетворює векторне представлення зображення на вірогідності класів. При розгляді математичної моделі шару-класифікатора, задача створення моделі без донавчання зводиться до задачі генерації вектора визначеного розміру. Цей перехід дозволяє натренувати мовну модель для генерації вагів моделі-класифікатора зображень і додавати нові класи. Отримана модель демонструє прийнятний рівень точності на нових класах із середньою мірою F-score рівною 0,731 для нових класів, при рівні F-score 0,844 для класів натренованих конвенційним методом. Додатково, встановлено що генерування декількох вагових векторів і використання їх середнього для класифікації дозволяє покращити якість класифікації порівняно з використанням окремих згенерованих векторів.*

*Ключові слова: машинне навчання, штучний інтелект, обробка природної мови, класифікація зображень, нейронні мережі, комп'ютерний зір, інформаційні технології.*

## INTRODUCTION

Computer vision (CV) is a field of study that has gained a measurable boost from the development of ML algorithms in the recent years. Most CV tasks are being solved with a great efficacy using neural networks. At the same time, the level of accuracy expected from a machine learning (ML) model regarding a CV task is usually quite high since it is often trivial for a human to assess the performance of the model with their naked eyes.

Image classification is one of the most basic CV tasks. The ImageNet benchmark evaluates the most accurate models in the image classification task (Ebrahim, Al-Ayyoub, Alsmirat, 2019). A typical solution is a model that is trained on a large dataset of images and performs well on the classes it knows. However, typically, models cannot be easily updated to work with new classes, especially with no examples of images of such classes. The task of classifying images into classes that may have not been available during the model training is called zero-shot image classification.

This paper describes a method of solving this task with the help of extra information about the objects that are represented by the image classes. This information comes from text captions of the images and descriptions of the classes. The text data is processed by a separate natural language processing (NLP) model and the result is used to modify a pre-trained image classifier model.

This work aims to establish a novel method of constructing image classification models on top of existing trained models, such that the new model supports previously unseen image classes, based solely on the parameters of the existing model with no extra training. The goal is to construct a step-by-step process of, given trained image classificator model and a text description of an image of a new class, obtain an image classification model with would support the new image class. Furthermore, it is expected that the initial classification model does not know interact with text descriptions of images and is not constructed specifically to be expanded in this way. At last, the received model and the whole process of modifying the classificator model must be measurably cheaper computation resource-wise than simply retraining the model with an updated dataset that would include the new image class.

Machine learning models are a universal tool for solving a number of computer vision problems, including image segmentation, image processing and modification to achieve the desired aesthetic effect, two-class and multi-class image classification, object tracking in video, generation of images in video or their fragments, and others. Given that the need for new data for training machine learning models in general and computer vision tasks in particular is constantly growing, the question arises of how to collect the necessary data. In general, the need for ever-increasing amounts of data is present in all areas of machine learning, including human language processing. A specific difficulty when working with images is that creating images is a relatively labor-intensive task or a relatively expensive process.

For the reasons stated above, the training a new model for image classification from scratch is often not possible due to lack of data or computational recourses or both. To overcome this issue, researchers use a number of techniques for extracting more useful trainable information from the same data and modifying already trained models in such a way that they fit the specific problems better. These techniques include:

- fine tuning;
- data augmentation;
- artificial data generation (Benbrahim, Behloul, 2021, Bartos, Ünaldı, & Yalçin, 2024).

These methods partially address the issue of data shortage, but require more computational power.

The OpenAI's CLIP model uses another approach to filling the gaps in training data. By training the base model on a combination of images and text descriptions, in a process called text-supervised learning, they created a model which can determine if a given image fits a number of text descriptions. By running this model with a single input image in combination with the necessary texts matching certain image classes, both seen and unseen by the model, it is possible to classify the image into an indefinite number of classes. This, however, requires much more computational resources than a regular image classification model (Radford et al., 2021).

There are several open datasets that combine text data and images which may assist in training models with the help of information gathered from text data, including MS COCO published by Lin et al. (2014), Conceptual Captions published by Sharma, Ding, Goodman, and Radu (2018), TextOCR published by Singh et al. (2021), and CUB-200-2011 published by He and Peng (2019).

A typical architecture of an image classifier ML model consists of two separate yet strongly connected segments. The first segment is the neural network (NN) which takes in image data and processes it in order to extract information from the given image. This NN is computationally heavy, consists of multiple layers, and may use different algorithms, some of which are the convolutional neural networks, residual NNs and, attention-based NNs (also known as transformers) (Sakthi, Lekha, Manesha, 2023). This segment is called the encoder. An encoder produces a vector representation of a given image as its output, known as an embedding of the image.

The other segment is called the classifier. A classifier is a single fully-connected (dense) layer, which sometimes uses a dropout or other auxiliary techniques. The classifier takes the image embedding as its input and produces a vector of probabilities as the output. In most cases, the values of the vector would be normalized via a sigmoid, a ReLU, or a similar function (Kim, Mathai, Helm, Pinto, Summers, 2024). The probability at index $n$ in the vector represents the probability of the image belonging to the $n$-th class.

In other words, the encoder is the part if the model which processes the image data and presents it to the classifier, which makes the decision. This segmentation comes in handy to researchers and ML engineers who may:

- train a single encoder and use it for different domain tasks in a technique called transfer learning;
- take an already trained encoder and train a classifier which would be more efficient at a certain domain task, in a technique called fine-tuning (Kaur, Sharma, Chattopadhyay, Verma, 2024).

For the purposes of this paper, it's also important to consider the internal structure of a dense NN layer, such as the one used in the classifier. A dense NN layer is a set of neurons, each of which takes all the inputs of the whole layer and produces a single output scalar value as the result. Mathematically, each neuron is independent, as none of them effect the output of the forward pass of one another. It is technically true that the outputs of all neurons may affect the outcome of the backward pass for all neurons in case if some loss functions such as the cross-entropy function are used. However, this limited interference only happens of the training stage, thus allowing us to state that the neurons are independent of one

another on the evaluation stage. When we consider a static NN, i.e. the state of the network when not in training, the mathematical representation of each of the neurons is as shown in the formula (1):

$$y' = activation(WX), \tag{1}$$

where: $y'$ is the output of the layer; $W$ is the weight matrix of the layer; $X$ is the output of the image encoder; *activation* is an activation function of the layer, typically, the sigmoid function.

Note that this approach works for both single-class and multi-class classification, i.e. for tasks when the result must be a single image class of multiple image classes.

## MATERIALS AND METHODS

Let's consider the trainable state of the classifier dense layer. The state is represented by a single matrix $W$ of size $N_{emb} * N_{cls}$. This matrix $W$ can be further decomposed into $N_{cls}$ vectors, each of size $N_{emb}$. Each of the vectors is responsible for predicting the probability of one particular class. Therefore, the task of adding a new class to the image classification model can be simplified to obtaining such a vector of size $N_{emb}$, that the result of a vector multiplication operation of this vector with an embedding vector of an image of this new class is a a high enough scalar value that the class is considered detected by the whole model, and, the result of a vector multiplication operation of this vector with an embedding vector of an image that does not belong to the new class is a low-enough scalar value that the class is considered not detected, so that to avoid false-positive responses (Xu, 2023, Song, Lee, Bae, & Park, 2024).

Generating such a vector can be performed in multiple ways. The straightforward way of doing so is through the process of training the ML model. However, this is not always practical due to the need in a high number of data samples. Instead, this paper introduces another approach, which, through use of NLP ML models and text metadata for the images, allows to generate the target vector with no image data present. This, in turn, allows for a zero-short image classification for the new image class to be achieved.

In order to do this, two key components are required. First is the text data itself. For this experiment, the MS COCO dataset was used. The dataset contains images and text descriptions for each image. The descriptions are written by human operators, which makes them highly reliable.

In order to generalize image descriptions, text data from another source is added. For each image of a given class, a short description of the class is obtained and prepended to the text description of the image. Since images in MS COCO may belong to several classes at the same time, the resulting dataset contains duplicate images, each image is accompanied by a caption and a class description for a single class. Here is an example of a text metadata of an image with the "bicycle" class: "Bicycle also called a pedal cycle, bike, push-bike or cycle, is a human-powered or motor-assisted, pedal-driven, single-track vehicle, with two wheels attached to a frame, one behind the other. A bicycle rider is called a cyclist, or bicyclist. a bike sitting parked on the ground next to a building. a bike is parked on the side of a building. a bicycle leaned against the wall on the side of the street a bicycle stands against the walls of a train station. a bicycle leaning on a building near a train track.".

Finally, each element of this dataset is enriched with a vector from the matrix $W$, which corresponds to the same image class as the text description. This will be the target variable for the training of the NLP model. The images themselves are not used in this training process. The goal is to train the model to generate a weight vector from the image descriptions.

The second key component is the model itself. For this experiment, the XLM-RoBERTa model was selected. The model was proposed by Conneau et al.(2019). The model is already pretrained on big text corpuses. Just like the image processing models described earlier in this paper, the XLM-RoBERTa generates an embedding of its input, allowing the end user to build upon it simpler NN for specific purposes. Unlike image processing models, XLM-RoBERTa generates a sequence of embeddings, one per input token. In order to bring them to a standardized format, a simple average pooling is used, resulting in a single embedding of the same dimension as the token embeddings, in a pooling technique described by Chen, Zhu, Yao, and Zhang (2023) and by Mayil and Jeyalakshmi (2023). In this case, the result vector has size 768. The goal of the training process is to train a simple dense layer that receives this embedding as the input and produces the target vector as the output.

Measuring the efficacy of the resulting NLP model is conducted in two steps. The first basic measurement is performed by comparing the target vectors that the model produced to the expected values. This comparison is used during training loss function calculation for the optimization algorithm and for supervising the training process via validation in between training epochs. Cosine distance loss function is used for training. The function is calculated according to the formula (2).

$$l^2 = \frac{\left(\sum_{i=1}^{768} y_i y_i'\right)^2}{\left(\sum_{i=1}^{768} y_i\right)\left(\sum_{i=1}^{768} y_i'\right)}, \tag{2}$$

where: $l$ is the loss value; $y_i$ is the $i$-th element of the ground truth vector; $y'_i$ is the $i$-th element of the predicted vector.

The second, more computationally heavy way of measuring the efficacy of the NLP model is by applying the resulting target vector to the image classifier model and assessing the efficacy of that model on the newly added class. This approach is used for comparing snapshots of the NLP model at different stages of training to one-another. The efficacy of the image classifier model is measured with the cross-entropy loss.

## EXPERIMENT

The MS COCO dataset was used for this experiment. The dataset contains over 200K captioned images with 3 to 5 captions for each image. The images are labeled with 80 classes (a.k.a. object categories). For each category, a small description of the object is sourced by taking the Wikipedia article which matches the category name best. Each description is limited to 256 words, so that the NLP model is able to process the class description and the image caption at the same time and not exceed the token count limits.

For NLP, the RoBERTa base model is used. The model contains roughly 125 million parameters.

For image classification, the EfficientNet v2 S model is used. It contains roughly 21 million parameters and produces embeddings of size 1280, which is the size of the target vector.

The MS COCO dataset was used to train the classifier model. The model was trained on a GPU with CUDA support. The model took 80 epochs to train in total. The peak performance of the model was achieved on the epoch 68 with an F-score of 0.844, after which, the model started overfitting on the data, as shown on figure 1. The training did not involve more sophisticated approaches, such as data augmentation, which are typically used to improve the classification efficacy, since this paper is mainly concerned with the later parts of the experiment, while training the EfficientNet model is rather a part of the setup stage.

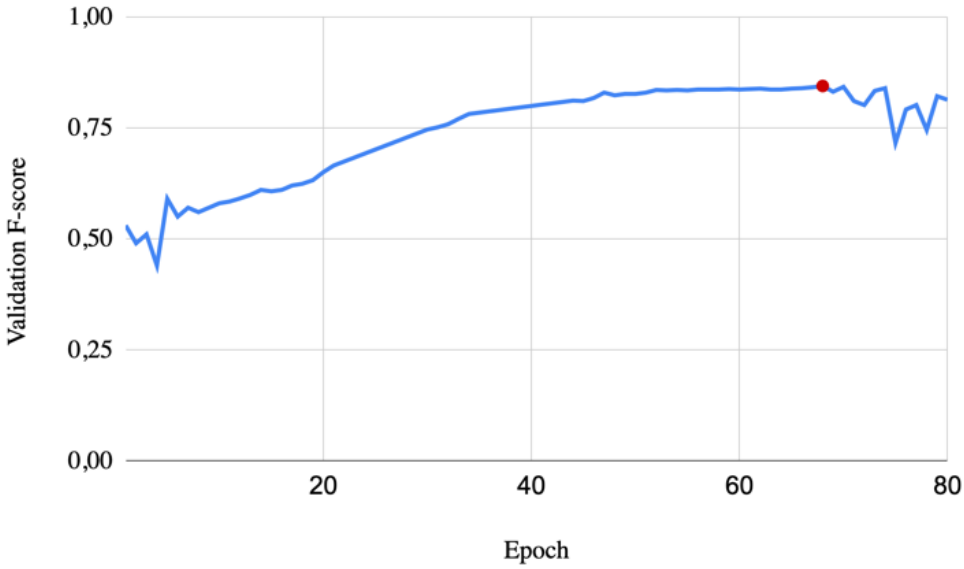



**Fig. 1. The validation F-score change with the training. The best performance point is highlighted in red.**

The RoBERTa model used in this experiment is a pretrained one, available publicly. The model was trained on large text corpora and can be customized for a specific task by adding a domain-specific head, in this case, a simple dense NN layer, which converts a pooled embedding of size 768 into a target vector of size 1280.

In order to improve the performance of the classifier with the target vector, a few vectors are generated for each class. The vectors are obtained by passing a few different examples of the text image descriptions to the NLP model. Then, the resulting target vectors are averaged in order to obtain a target vector with a more precise representation of the class.

## RESULTS AND DISCUSSION

The results of the experiment show that the efficacy of the new class with a generated weight vector is somewhat similar yet still lower than that of the normal classes. The efficacy was measured by testing the generated weights for six new classes. The classes are “warship”, “birdhouse”, “bee”, “snake”, “leopard”, and “canoe”. Images for the class “warship” are assembled by hand (25 images in total, gathered among the images available freely on the Internet). Images for the other classes are taken from the ImageNet dataset. While “birdhouse”, “bee”, “leopard”, and “canoe” are existing ImageNet classes, the “snake” class is sourced from three different ImageNet classes, namely “thunder snake”, “water snake”, and “green snake”. For each of the five ImageNet classes, 100 images are taken per

class. The text descriptions for all the five classes are taken from Wikipedia in the same way as were the descriptions for the known classes. Text captions for the images themselves are written by hand. In order to generate the target vectors, 10 images from each class are captioned. The other images are used as the test sample. Table 1 demonstrates the results of using the target vectors as the weights for the classifier on the five new image classes. The core measure to be considered regarding the model efficacy is the model's F-score. For the test classes, the binary classification F-score was measured for each class. Each measurement involved an equal number of images of the new class and other images which do not belong to the new class.

Table 1.

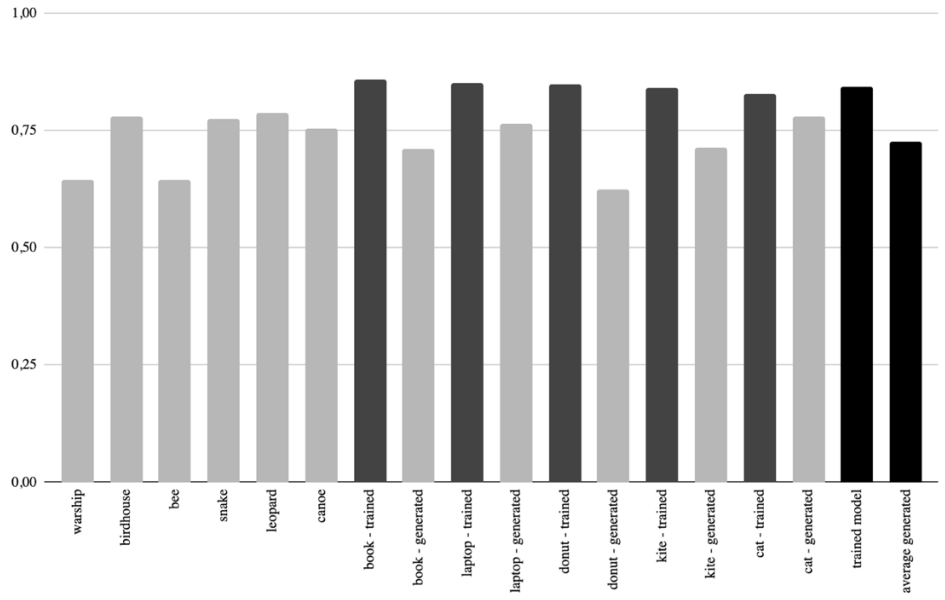
**Efficacy of the model with the generated weight vector on the test classes**

| Image class | Precision | Recall | F-score | Test sample image count |
|---|---|---|---|---|
| warship | 0.625 | 0.667 | 0.645 | 30 |
| birdhouse | 0.713 | 0.855 | 0.778 | 180 |
| bee | 0.75 | 0.567 | 0.645 | 180 |
| snake | 0.68 | 0.9 | 0.775 | 180 |
| leopard | 0.755 | 0.822 | 0.787 | 180 |
| canoe | 0.712 | 0.8 | 0.754 | 180 |
| average among all classes | 0.706 | 0.768 | 0.731 | |

As seen in the table, the performance of the model with generated weight vectors is relatively consistent among different test classes. However, the model obtained in such a way is less precise in the image classification task than a model trained for the task. Furthermore, the model performs better on the trained classes than on the classes for which the weight vectors are generated. This can be discovered when generating a new weight vector for a class of images that was present during model training. This test was conducted with five image classes from the MS COCO dataset. The F-score of binary classification was calculated using each of the generated weight vectors as well as the original weight vectors, obtained when training the model. For each image class, descriptions for 10 images from the validation sample, 30 descriptions in total, were presented to the NLP model and the generated target vectors were averaged. Then, all the other images in the validation sample were tested on the model with the resulting weight vector. This test showed the following results:

1) For the class "book", the trained F-score is 0.859 and the generated F-score is 0.711.
2) For the class "laptop", the trained F-score is 0.851 and the generated F-score is 0.763.
3) For the class "donut", the trained F-score is 0.847 and the generated F-score is 0.623.
4) For the class "kite", the trained F-score is 0.841 and the generated F-score is 0.713.
5) For the class "cat", the trained F-score is 0.828 and the generated F-score is 0.78.

The graph on figure 2 illustrates the performance of the model with the generated weight vectors.




**Fig, 2. Binary classification F-score values for the model with the generated and trained weight vectors. Values for the generated vectors are shown in light gray. Values for the trained vectors are shown in dark gray. Reference value of the multiclass classification F-score of the trained model and the average value of binary F-score classification are shown in black.**

As seen in the presented results, the suggested method demonstrates a relatively high efficacy rate, allowing to achieve zero-shot learning with relatively few resources. The experiment included training an image classificator NN from scratch, however, this is merely a setup step and not a part of the suggested approach to zero-shot learning.

Key benefits of the described method are:

1) Low requirements to the NN training setup. Since the only training required by the method is fine-tuning of the NLP model, which could be done on a modern consumer-grade GPU, in a few hours, rather than specialized NN servers and many processing hours it takes to train an image-processing model from scratch.

2) Low requirements to the image data. No examples of images of the new classes are required. The experiment only used images of the new classes for testing purposes. However, it is necessary to provide hand-written descriptions of what an example image might look like and what category does the image class represent in the real life.

The suggested also has a few drawbacks, which can be improved with further research into the method. These drawbacks are:

1) When comparing to classes that were present during NN training, the classes added with the suggested method receive lower efficacy rates, meaning that the model cannot classify images containing the zero-shot classes as well as it can images containing regular classes.

2) The method includes a NN training step. This can by an issue to adopters who do not possess resources needed for training models. Since the method aims to cover practical use-cases with low-resource conditions, this is drawback to some of the potential adopters. However, the scale of training is quite limited, as described earlier, meaning that the cost of resources needed for it is not that taxing comparing to a typical image classifier training.

3) The need for high quality image captions. This is potentially the most difficult obstacle to overcome for an adopter of the suggested method. The lack of high-quality text captions for many open-source datasets is the reason why the set up for the experiment described in this paper included training an image classifier NN from scratch.

Results of this paper contribute to the study of zero-shot learning as a section of the machine learning research.

For instance, Ruffino et al. (2024) demonstrate the application of a convensional non-LLM model to the zero-shot approach. The model compares well to much larger conventional models while having much less trainable parameters and therefore being less resource hungry, having faster inference and shorter training time.

Also, in the works of Li, Tang, Tang, and Yang (2023) and Sivarajkumar and Wang (2023), the influence of zero-shot learning methods is demonstrated on practical problems in the fields of industrial system management and medicine respectively. Their findings indicate that the use of zero-shot learning methods is invaluable in domains with a few to none publicly available training data.

Finally, Patil and Ravindran (2024) demonstrate the application of the zero-shot learning approach in a field where data is not just unavailable, but also hard or impossible to gather, in particular, in searching for software defects. Their findings show that the suggested method is more capable of solving the task than the conventional supervised learning techniques.

## CONCLUSIONS

This paper presents a method for solving the zero-shot image classification task.

The scientific novelty of obtained results lies in the novel method for zero-shot image classification using image captions and class descriptions as a source of additional information for the images. The method decomposes the task of image classification into the task of encoding an image into a vector representation and converting the vector representation into a decision regarding the image class. It is noticed that a typical NN image classifier performs the latter step using a simple dense layer of neurons, which is essentially a matrix multiplication operation. With the help of a NLP model trained on the additional information regarding image classes extracted from text metadata, the weight matrix of the said dense layer can be manipulated to work on new classes which were not previously seen by the classifier model, thus achieving zero-shot classification.

The practical significance of obtained results is in giving adopters of the method a way to obtain image classifier models with support of some domain specific classes, based on an existing model that supports other image classes. It is also possible to build software that would generate such models with no input from the user except for the seed image captions. Such models work faster and require less computational resources than the alternative zero-shot models.

Prospects for further research are to improve the performance of the zero-shot classes to bridge the gap between them and the trained classes. It can also be speculated that with the zero-shot image classification solved, the task of zero-shot object detection on images and later — on video becomes more approachable. Hence this paper lays the groundwork for solving more complicated computer vision tasks.

## References

[1] Ebrahim, M., Al-Ayyoub, M., & Alsmirat, M. A. (2019). Will Transfer Learning Enhance ImageNet Classification Accuracy Using ImageNet-Pretrained Models? In *10th International Conference on Information and Communication Systems (ICICS)* (pp. 211–216). https://doi.org/10.1109/iacs.2019.8809114

[2] Benbrahim, H., & Behloul, A. (2021). Fine-tuned Xception for Image Classification on Tiny ImageNet. In *International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP)* (pp. 1–4). https://doi.org/10.1109/ai-csp52968.2021.9671150

[3] Bartos, G. E., Ünaldı, S., & Yalçin, N. (2024). Impact of image augmentation on Deep Learning-Based Classification of granite tiles. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 796–799. https://doi.org/10.1109/ubmk63289.2024.10773433

[4] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2103.00020

[5] Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Dollár, P. (2014). Microsoft COCO: Common Objects in context. In *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1405.0312

[6] Sharma, P., Ding, N., Goodman, S., & Radu, S. (2018). Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of ACL*.

[7] Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., & Hassner, T. (2021). TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2105.05486

[8] He, X., & Peng, Y. (2019). Fine-Grained Visual-Textual Representation learning. In *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(2), 520–531. https://doi.org/10.1109/tcsvt.2019.2892802

[9] Sakthi, K., Lekha, M., & Manesha, P. A. (2023). Exploring active machine learning techniques to boost classification accuracy in image and text models. In *2022 8th International Conference on Smart Structures and Systems (ICSSS)*, 1–6. https://doi.org/10.1109/icsss58085.2023.10407201

[10] Kim, B., Mathai, T. S., Helm, K., Pinto, P. A., & Summers, R. M. (2024). Classification of Multi-Parametric Body MRI Series using Deep Learning. In *IEEE Journal of Biomedical and Health Informatics*, *28*(11), 6791–6802. https://doi.org/10.1109/jbhi.2024.3448373

[11] Kaur, A., Sharma, R., Chattopadhyay, S., & Verma, A. (2024). Automated Multiclass Classification of Groundnut Leaf Diseases Using Fine-Tuned InceptionV3 Model. In *2nd World Conference on Communication & Computing (WCONF)*, 1–4. https://doi.org/10.1109/wconf61366.2024.10692187

[12] Xu, X. (2023). Research on Multi-Labels Image Classification Based on Self-Supervised Model. In *2022 International Conference on Image Processing and Computer Vision (IPCV)*, 56–59. https://doi.org/10.1109/ipcv57033.2023.00017

[13] Song, S., Lee, D., Bae, H., & Park, C. (2024). Improving image classification accuracy through cluster optimization in latent space. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, 1342–1346. https://doi.org/10.1109/ictc62082.2024.10827620

[14] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F. (2019). Unsupervised cross-lingual representation learning at scale. In *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1911.02116

[15] Chen, J., Zhu, Y., Yao, H., & Zhang, W. (2023). Adaptive Visual Semantic Embedding with Adaptive Pooling and Instance-Level Interaction. In *2023 7th International Conference on Electrical, Mechanical and Computer Engineering (ICEMCE)*, 853–857. https://doi.org/10.1109/icemce60359.2023.10490683

[16] Mayil, V., & Jeyalakshmi, T. (2023). Pretrained Sentence Embedding and Semantic Sentence Similarity Language Model for Text Classification in NLP. In *2023 3rd International Conference on Artificial Intelligence and Signal Processing (AISP)*, 1–5. https://doi.org/10.1109/aisp57993.2023.10134937

[17] Ruffino, S., Karunaratne, G., Hersche, M., Benini, L., Sebastian, A., & Rahimi, A. (2024). Zero-Shot Classification Using Hyperdimensional Computing. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 1–2. https://doi.org/10.23919/date58400.2024.10546605

[18] Li, J., Tang, H., Tang, D., & Yang, Z. (2023). Multi-Label Zero-Shot Learning for Industrial Fault Diagnosis. In *2023 6th International Conference on Information Communication and Signal Processing (ICICSP)*, 1235–1240. https://doi.org/10.1109/icicsp59554.2023.10390617

[19] Sivarajkumar, S., & Wang, Y. (2023). Evaluation of HealthPrompt for zero-shot clinical text classification. In *2022 IEEE 10th International Conference on Healthcare Informatics (ICHI)*, 492–494. https://doi.org/10.1109/ichi57859.2023.00081

[20] Patil, S., & Ravindran, B. (2024). Zero-shot learning based Alternatives for class imbalanced learning problem in enterprise software defect analysis. In *2024 IEEE ACM 21st International Conference on Mining Software Repositories (MSR)*.