Dataset for NLP-enhanced image classification

Dmytro Dashenkov[1], Kirill Smelyakov[1] and Nataliia Sharonova[2]

[1] *Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine*
[2] *National Technical University "KhPI", Kyrpychova str. 2, Kharkiv, 61002, Ukraine*

Abstract

In this paper we present a multi-modal image and text dataset. The dataset is based on images from the Open Images dataset and text descriptions of the class names obtained from Wikipedia. We provide an exemplary model for labeling images trained on top of the dataset. Lastly, we explore the applicability of this or similarly compiled datasets for various computer vision tasks, in particular for image classification with aid of a natural language processing model. With the help of the compiled dataset, we construct an image tagging model. The model represents a typical example of multi-class multi-label classification task. Using a pretrained model, we fine-tune a neural network classifier for adding one-word tags to the images based on the objects depicted in the images. We explore the performance of the classifier and argue for the benefits of the multi-modal datasets for this task as well as other vision tasks.

Keywords

Computer vision, image classification, natural language processing, multimodal learning

1. Introduction

Image classification is one of the typical tasks for computer vision algorithms. As such, many different approaches to the task have formed. In recent years, neural network-based approaches dominate the area. In particular, models using convolutions and attention mechanisms are popular and show great results.

In this work, we suggest a novel approach to solving the image classification task by using the latest findings in the field of natural language processing and combining them with the conventional models for image classification. For this, we have assembled a multi-modal dataset. The structure and more details on the dataset are presented further in this paper. The dataset is available publicly on GitHub. We also theorize as to what kinds of models might be built on top of this dataset.

Many multimodal datasets with text and images are built for the task of image description or for generating images from text. Such tasks require distinct and precise annotations for each image. Unlike those tasks, image classification works with a predefined set of ground truth labels. This gives us the ability to consider using general class descriptions as labels rather than individual image descriptions. Because of this simplification, we're able to assemble the dataset with less effort. Practitioners who apply the results of this work will be able to amend the dataset just as easily without spending resources on human annotators.

The end goal of this research is to come up with an approach to image classification that would be scalable, i.e., given a certain pretrained state, have an ability to receive new image classes with smaller amounts of extra training. By the virtue of being able to extract additional data from the class labels, rather than merely treating them as non-informative flags, we hope to achieve a better performance for the classes added after the main training stage (meta-training).

Also, the smaller training stages for the added classes (fine-tuning) should benefit from the knowledge extracted from the class descriptions. Such a technique may improve the rate at which new

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: dmytro.dashenkov@nure.ua (D. Dashenkov); kyrylo.smelyakov@nure.ua (K. Smelyakov); nvsharonova@ukr.net (N. Sharonova)
ORCID: 0000-0001-9797-1863 (D. Dashenkov); 0000-0001-9938-5489 (K. Smelyakov); 0000-0002-8161-552X (N. Sharonova)

classes may be added to a trained model. Such an achievement will benefit many practitioners, who take advantage of pretrained models to solve narrower cases.

This paper presents the preparatory yet important stage of the research, collecting and cleaning data for the models to learn upon. Our goal with this paper is to demonstrate the approach of data collection, present a complete usable and useful dataset, and illustrate the use of said dataset on a concrete problem. The demonstrative problem we have chosen is the image tagging task—generating multiple one-word tags that describe an image. Such an algorithm may prove useful for many practical scenarios, such as image search.

2. Related works

Currently, one can see significant progress in image classification using neural networks [1-3]. Including analysis of images with high spatial resolution [1], application of trending technologies of transfer learning and machine learning [2], use of multispectral data analysis [3]. In addition to these important areas of development, a number of private results have been obtained, which can significantly improve the efficiency of classification.

Thus, in [4], a new combined transfer learning technique for image segmentation based on the combination of image weighting and kernel training is proposed to improve performance on heterogeneous data. In [5], an effective model of voice labeling of images using neural networks is proposed, the application of which can significantly improve the accuracy of neural network training when considering non-trivial data. In [6], an aggregate network with context-sensitive learning for hyperspectral image classification is proposed, which can effectively reduce the influence of initial graph error on the classification result. In [7], a model for semi-supervised classification of hyperspectral images using spatial-spectral information is proposed to improve classification efficiency under conditions of limited data sampling. In [8], features, models and algorithms for volumetric image classification using multisample learning and extreme value theorem are described. In [9], a deep learning platform for converting image reconstruction into pixel classification for efficient local processing of a digital image is presented. In [10], models and algorithms for kernel-based constrained energy minimization for hyperspectral classification of mixed pixels are described. In [11], an algorithm for iteratively increasing the training sample to improve the accuracy of image classification is proposed.

At the same time, solutions for the classification task in NLP are being actively developed. Thus, the work [12] presents a unified understanding of deep NLP models for text classification at different levels of perception and detail. Work [13] proposes solutions in the sense of using deep learning architectures based on transducers for specific conditions of their application. In [14] a practically oriented model of automatic classification of sexism in social networks (Twitter network) is proposed. In [15], a mechanism for embedding user ID into pre-trained language models for document-level tone classification is proposed. In [16], a new method of MBTI classification based on the influence of class components is proposed. The method is used for subsequent prediction of personality type. In [17], the authors proposed a multitasking learning model based on multiscale CNN and LSTM for sentiment classification. In [18], the authors try to combine supervised machine learning and NLP algorithms into one method, which is called SECRET (Semantically Enhanced Classification of REal-world Tasks). This method does the classification by combining the semantic information of the labels with the available data. In [19] the state of the art of models and algorithms for classifying user-generated content from social networks in real time is described. Article [20] presents the results of an analysis of the application of text expansion methods combined with the latest data classification algorithms. Article [21] proposes an innovative method for the operation of a recommendation system for breast cancer diagnosis using patients' medical histories. The mechanism of machine learning and word embedding in the classification of the disease diagnosis is applied. In [22] the limitations of transducers for classification of clinical documents are presented and analyzed.

State of the art image classification analysis suggests a potential for improvement in classification performance with the methodology suggested in this work. Combining the convolutional neural networks and attention-based neural networks with the NLP models allows to compensate for the drawbacks of the generally accepted approach of image enhancement [23, 24] with further application of a neural network classifier [25, 26].

Simplistic approaches to natural language processing, such as statistical models, for example, Markov chains, can lead to significant results when applied to texts of limited scope, as shown in [27]. For more complex texts, the text styles can provide a useful heuristic for selecting the appropriate light-weight algorithm [28]. This ability to choose simpler NLP models allows us more flexibility to provide greater performance for many specific cases. Additionally, methods presented in [29, 30] for processing images of various sources provide a basis for a framework for heuristic-based and neural network-based approaches to the visual component of classification.

Overall, by using heuristic approach to both NLP and vision, many specific examples can be solved without using the more performant yet more resource intensive neural network-based approaches. However, in the general case, as well as for situations where determining the correct heuristics-based solution is impossible, neural network-based solutions are prevalent.

Existing multimodal datasets involving text and image data typically consist of images annotated with text description of what is happening on the image. Such datasets include:

- COCO (Microsoft Common Objects in Context) [31]
- Flickr30k [32]
- Conceptual Captions [33].

COCO (Microsoft Common Objects in Context) is a large-scale object detection, segmentation, and captioning dataset [31]. The dataset consists of images, polygon annotations for select objects, and a few statements about objects on the images in text form.

Flickr30k is a dataset of over 30 thousand images from Flickr. Each image is annotated with five sentences written by human annotators. The images are limited to educational use only [32].

Unlike in the COCO dataset, the five sentences are alternative descriptions of the image, not just class names. As seen from the example, some images have details that can be seen differently by different people. This feature of the dataset may infuse data with more variability.

Conceptual Captions is, similarly to Flickr30k, a dataset with annotated images. However, this dataset provides captions generated automatically by correlating images with text at the data source [33]. The dataset includes over 3 million captioned images.

However, when considering the image classification task, a typical dataset consists of images annotated with either one or several labels. Such datasets include ImageNet [34], MNIST [35], and CIFAR-10 and CIFAR-100 [36]. Each of the labels represent an object present on the image, or, sometimes, an action performed on the image by humans. At the core of these datasets are images. Even barring the text labels, classes, or action descriptions, the datasets may provide great value to researchers, e.g. in an unsupervised learning setting.

With these and other datasets, the list of models built for image classification is vast. At the time of writing, some of the more efficient models include transformer-based models, such as CoCa [37] and ViT-G/14 [38], residual neural network-based models, such as FixResNet-101 [39], and EffNet-L2 [40], which is based on the EfficientNet [41] scaling mechanism and the approach of minimizing training loss sharpness along with loss itself.

CoCa and ViT models use approaches derived from the initial Transformer model [42]. The attention mechanism is applied to convolutions derived from the input image. Both models perform well in few-shot scenarios and are suitable for fine-tuning.

FixResNeXt-101 model derives from the ResNet model [43]. The model is capable of high results on the classification task, having a lower number of parameters than the transformer-based models.

EffNet-L2 is a modification of other EfficientNet [41] models, that utilizes sharpness-aware minimization. Like all EfficientNet models, it is capable of scaling, thus may use less parameters than other model types.

Some of the most used benchmarks for image classification are based on datasets ImageNet [34] and CIFAR-100 [36]. It is impractical to compare specific models to one another if they are fine-tuned for different benchmarks. Thus, we choose the top performers in the three categories of models, transformers, ResNets and EfficientNets. See table 1 for benchmark values the ImageNet and CIFAR-100 values aggregates per model type with mentions of specific model names [42, 43]. For transformer-based models, consider CoCa [37] and ViT-B-16 [38]. For ResNet-based models, consider

FixResNeXt-101 [39] and BiT-L ResNet [44]. Finally, for EfficientNet-based models, consider EfficientNet-L2 [45] and EffNet-L2 SAM [40].

Table 1
Performance of some classification models

Model type	ImageNet	CIFAR-100
Transformer	91.0 (CoCa)	94.2 (ViT-B-16)
ResNet	86.4 (FixResNeXt-101)	93.51 (BiT-L ResNet)
EfficientNet	90.2 (EfficientNet-L2)	96.08 (EffNet-L2 SAM)

3. Methods and Materials

In order to assemble such a dataset, we compiled several data sources. The choice of data sources is based on several factors, such as:

- Images have to be of a relatively high resolution. Many datasets use low-resolution images. Such techniques work great for purposes of education, assembling proof-of-concept algorithms, basic demonstrations, etc. For purposes of solving real-world problems, we require high-resolution images. In the end, we settled on having images at least 360 by 480 pixels. This allows for fine details to be present on the images, as compared to low-resolution datasets, such as ImageNet [34], MNIST [35], and CIFAR-100 [36].
- Images have to be clearly labeled. Labels designate classes of objects on the image. There may be multiple object classes per image. There should not be any action labels, i.e. descriptions of actions, situations, etc. that are performed on the image.
- Text descriptions of the image classes have to be as informative as possible.
- Text descriptions are tokenized in order to simplify the preparation for NLP algorithms.
- Text descriptions must contain from 400 to 512 tokens. The upper limit comes from the common size limit for many NLP models, such as BERT [47].
- All data collected for the dataset has to be distributed under permissive open-source licenses.

Accounting for the listed requirements, we turned to the Open Images [48] dataset. The dataset provides 1.9 million images labeled with over 600 “boxable” classes, i.e. classes of objects present on an image that could be shown with a bounding box. However, the data we’re interested in for the purposes of our dataset is not the bounding boxes but the presence of a given class on an image. Class distribution is, while not uniform, is even enough to be sure that, given some thoughtful data sampling, the vision models will be able to learn all classes equally well. Figure 1 shows the histogram of occurrences of classes in the dataset.

As seen on the graph, most classes tend to have between one hundred and ten thousand images. There are a few classes that have less than ten images. In the training process, those classes could be excluded to later serve as the few-shot examples.

The boxable classes can also be used for bounding box labeling task.

With that in mind, we borrow the labeled images from the Open Images dataset.

For the purpose of obtaining class descriptions in text form, we fetch Wikipedia articles by the name of the class. If a total match exists, we use the article. If there is a redirected article, we use that article. In case of ambiguity, we manually select the article that fits the context of images the most. For example, the label “Stool” has more than one article matching the name. We manually select the one that describes a piece of furniture and proceed with it.

Once the article is obtained, we fetch the first few paragraphs. The goal here is to at least have the definition of the word. The table of contents, citations, links, and other markup elements are ignored.

Lastly, the definitions are tokenized, so that instead of working with whole texts, we are able to work with sequences of words that represent said text.

The resulting definition for “Stool” reads: “A stool is a raised seat commonly supported by three or four legs but with neither armrests nor a backrest in early stools and typically built to accommodate one occupant As some of the earliest forms of seat stools are sometimes called backless chairs despite how some modern stools have backrests Folding stools can be collapsed into a flat compact form typically by rotating the seat in parallel with fold-up legs".

Note the absence of any punctuation in the example above. Some language models work with simple punctuation, such as commas, periods, etc. [39], while others don't. For our dataset, we're going with the simpler option of removing the punctuation. Partly, because we target the dataset to more simple language models, that may not need punctuation as they operate on words and word combinations, rather than whole sentences and text in general.

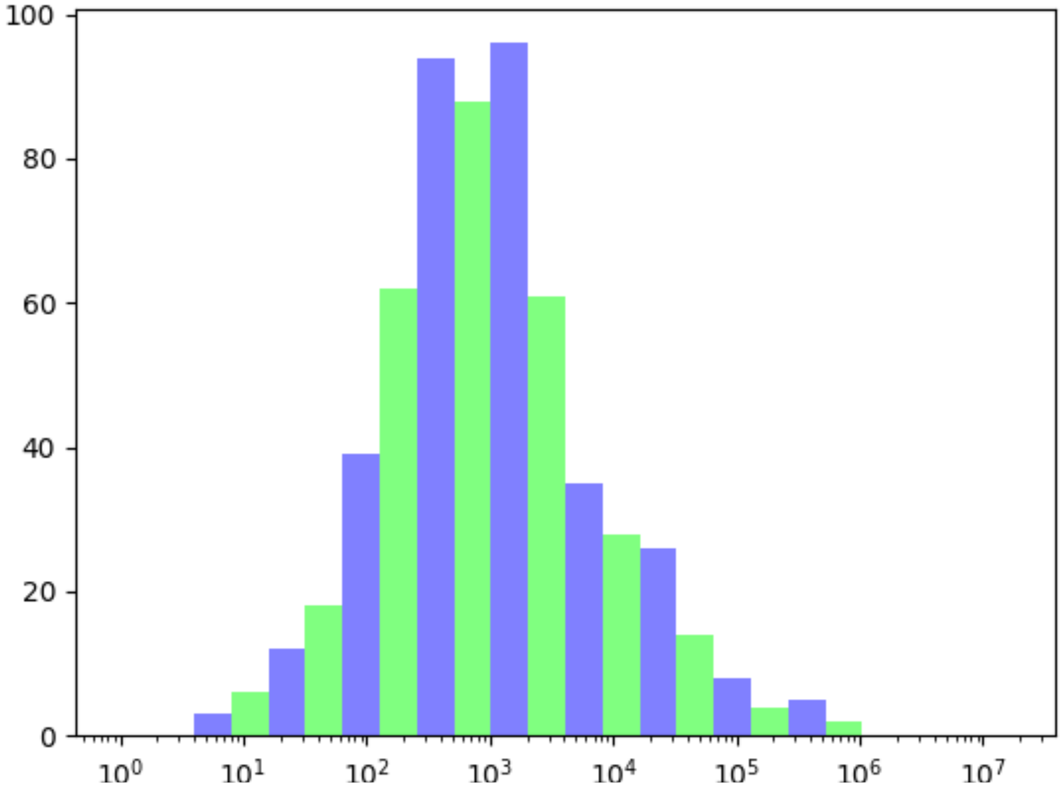


Figure 1: Histogram of the numbers of images per class in the dataset. The x-axis represents the count of images. The y-axis represents the number of classes with roughly this number of images

The resulting dataset is published on GitHub publicly [49]. Instructions on accessing the dataset are available on GitHub under the name "ImageD Dataset". The repository contains all the text data mentioned in this paper. The image data can be accessed by downloading the images from the Open Images dataset. For convenience, the repository also contains scripts for downloading images that can be copied or used as Python libraries. We do not redistribute the images from the Open Images dataset, but merely access them.

Models trained for vision tasks can be graded via several different metrics. In our research, we have developed a model for labeling images with several tags. This demonstration model is evaluated via the typical metrics, precision, recall, and the F-score.

The metrics are calculated with the following formulas:

$$precision = \frac{tp}{p}, \tag{1}$$

$$recall = \frac{tp}{ap}, \tag{2}$$

$$F = 2\frac{precision \cdot recall}{precision + recall}, \tag{3}$$

where *tp*—true positive answers, *p*—all positive answers, *ap*—actual positive examples.

The metrics do not mean anything without the appropriate context.. Only attached to a certain dataset, in our case, ImageD, do the metrics receive any meaning. But, as a rule of thumb, greater is better.

In binary classification problems, precision and recall must be adequately balanced, since random guessing would produce a high recall, nearly 1, and precision of nearly 0.5, resulting in F-score nearly 0.75. In our case of multi-class classification, this issue does not manifest.

For the tagging model, used to demonstrate the capabilities of the datasheet, we use a pre-trained ResNet 101 32x4d model [50]. The model is capped with a head which accepts the model's output embedding as its input and generates the vector of probabilities. Each probability value corresponds to a single tag. Figure 2 illustrates the general architecture of the model.

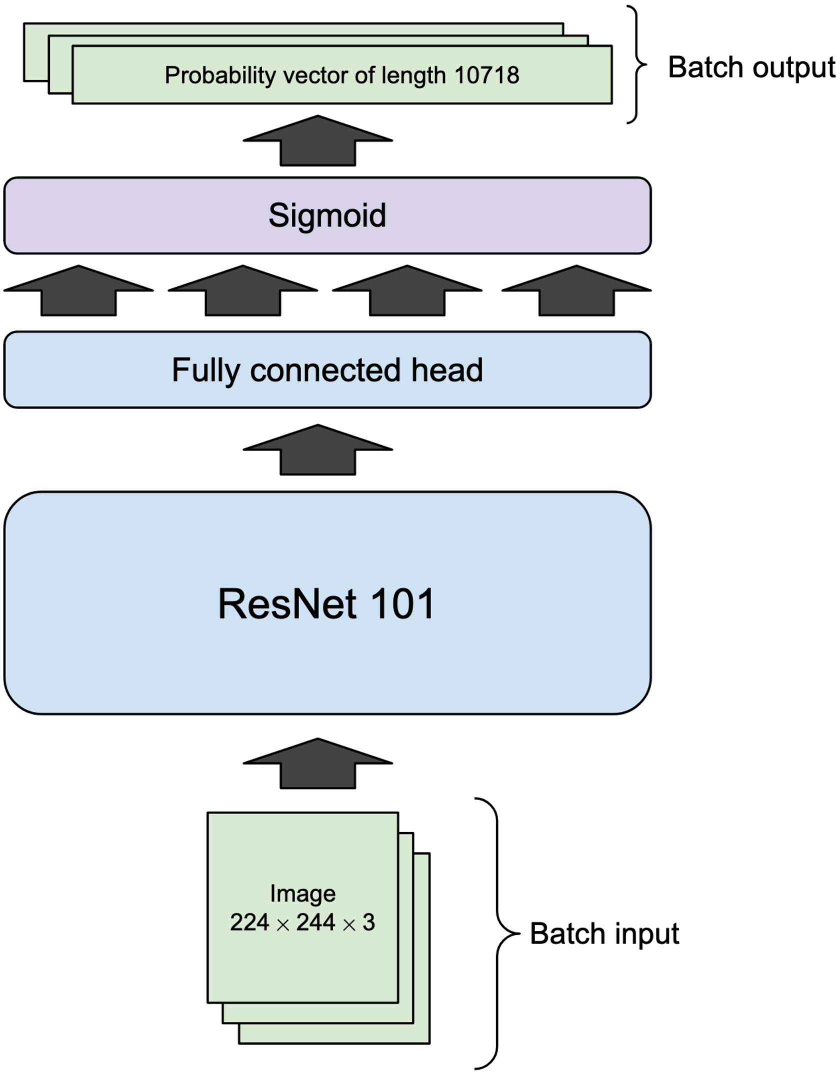


Figure 2: Structure of the tagging model.

As seen on the diagram, the head consists of a single fully-connected layer with a sigmoid activation. The resulting vector is then treated as a probability vector for the tags.

4. Experiment

Example learning experiment: convert text annotations into tags and train a model to tag images.

Given an integrated dataset of labeled images with text descriptions, we may build a variety of machine learning models. For the purposes of demonstrating the applicability of this dataset, we have developed a model that, given an image, tags it with several one-word tags.

The process of experiment preparation goes as follows:

1. Prepare one-word tags by tokenizing the class descriptions and selecting the rarely-occurring words.
2. Digitize tags by transforming them into binary unit vectors.
3. Assemble tag vectors for each image in the dataset.

The preparation effectively converts sets of labels for each image into sets of tags encoded in such a way that a neural network can be trained on them. Such a model, when used, would take an image and mark it with a set of tags. This can be useful for, e.g. building a primitive search engine that looks up images by code words.

In order to tokenize the class descriptions, we use the widely used “nltk” Python library. For each description, we receive the tokenized version that consists of word stems, endings, other grammatical parts, and punctuation. First, we remove the punctuation, as well as any non-word tokens, such as numbers. Then, we filter the tokens to remove those that occur too often. This is done by gathering the statistics of word occurrences in different class descriptions and only retaining a top P percentile. P is a hyperparameter of the model. For the purposes of demonstration, P was selected to be 0.2. This means that, if the word occurs in over 0.2% of the class definitions, we remove it from the list. All other words become the tags for the model to train upon. This gives us 10718 tags to train on, i.e. the output layer will have 10718 neurons.

Next, the tags are digitized, i.e. turned from words into a digital representation. For this, we sort the tags alphabetically and build a vector for each of the tags. The vector has the length of N, where N is the total number of tags. The vector is filled with zeros, except for one value, which is set to one. The position of the unit value corresponds to the order of the tag and uniquely represents the tag.

With the tags digitized, we now may label the images with them. To do so, we lookup all the classes associated with a given image and, given their descriptions as tags, find the set of digital representations of the tags. Then, we combine the tag vectors via the bitwise OR operation. This yields us vectors filled with ones and zeros, representing the tags associated with a given image.

Now, we proceed to training the model. For this, we selected a pre-trained core model that will provide us the needed level of accuracy with a needed level of performance. The choice of such a model has a significant impact on the final accuracy of our model. For the needs of this research, we have selected a small yet quite powerful model ResNet 101 32x4d model [50]. The model accepts images of resolution 224 per 224 pixels as input.

The process of enabling the model to solve our tagging task involves fine-tuning. In order to shorten training time and avoid pre-trained weights losing their efficiency, we substitute the final fully-connected layer of the model and freeze the rest of the layers. This means that the error propagation process will not alter any parameters except for the last layer.

The fresh fully-connected layer is shaped in such a way that it receives the output of the other layers of the ResNet model as the input and produces a vector of size N as the output, where N is the total number of tags. As calculated previously, N is equal to 10718.

The outputs of the final layer are passed through a sigmoid function to determine the probability of a given class being selected. If the output is high enough, the image is labeled with the tag associated with the given output.

To determine if the model considers the image to belong to a given class, we choose a threshold value for the output layer. All the outputs that are less than the threshold value are discarded and all the values that are equal to or greater than the threshold are considered to be confidence levels for the given class. Since the negative outcome (a zero value) is much more likely than a positive outcome (a one value) for any given output neuron, we take the threshold value quite low to be 0.3. This allows us to pick up on weak signals from the model when no confidence level is high enough. However, to avoid overselection, i.e. selecting too many classes per example, we also ignore all the values that are not in the top four classes per example, regardless of their confidence level. This is a tradeoff between the more extensive search for objects of the image and the accuracy. The more possible tags the model can find, the more false-positive errors it can make, and the more in-depth search for objects on the image can be performed.

Then, the model was trained to output the tag vectors on the data we assembled. The loss change during the training process is depicted in the graph in figure 3.

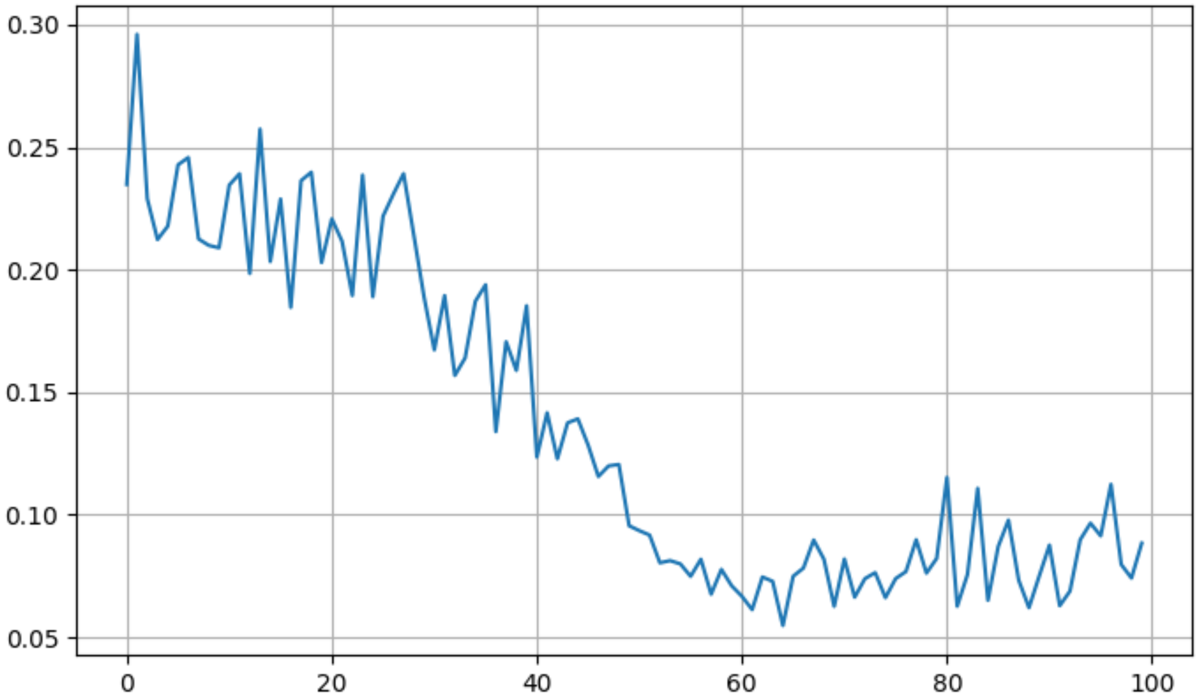


Figure 3: Change of the loss function value during the model training. The X axis represents the epoch number. The Y axis represents the MSE loss. For this graph, loss was sampled once per epoch every last batch of an epoch

For training we used the SGD optimizer with the mean squared error (MSE) loss function. SDG is the optimizer used during the training of the initial ResNet model. MSE loss fits best because of its simplicity when working with multi-class multi-label classification.

The fine-tuned model performs well enough on the test dataset, with the precision value of 71.11, the recall value of 74.2 and the F-score of 72.62. Note that, for calculating precision and recall, we used the following algorithm:

- an example is considered true positive if the four or less classes produced by the model are present among the ground truth labels;
- an example is considered false positive if the model produced at least one label that is not found among the ground truth labels;
- an example is considered false negative if the model did not produce a class that was present among the ground truth; for calculating this value, we omit the four classes per example rule.

This approach is known as micro-averaged precision and recall calculation, as opposed to macro-averaged and example-based calculation, both of which consider precision and recall for each class individually.

Note that in the process of training, there was a noticeable growth of loss after a certain number of iterations around epoch 64. This can be explained by the weight decay after a certain number of repetitions. For better performance, data augmentation could have been used.

5. Results

The trained model is able to tag images with some degree of accuracy. The performance of the model is defined by the several factors, such as:

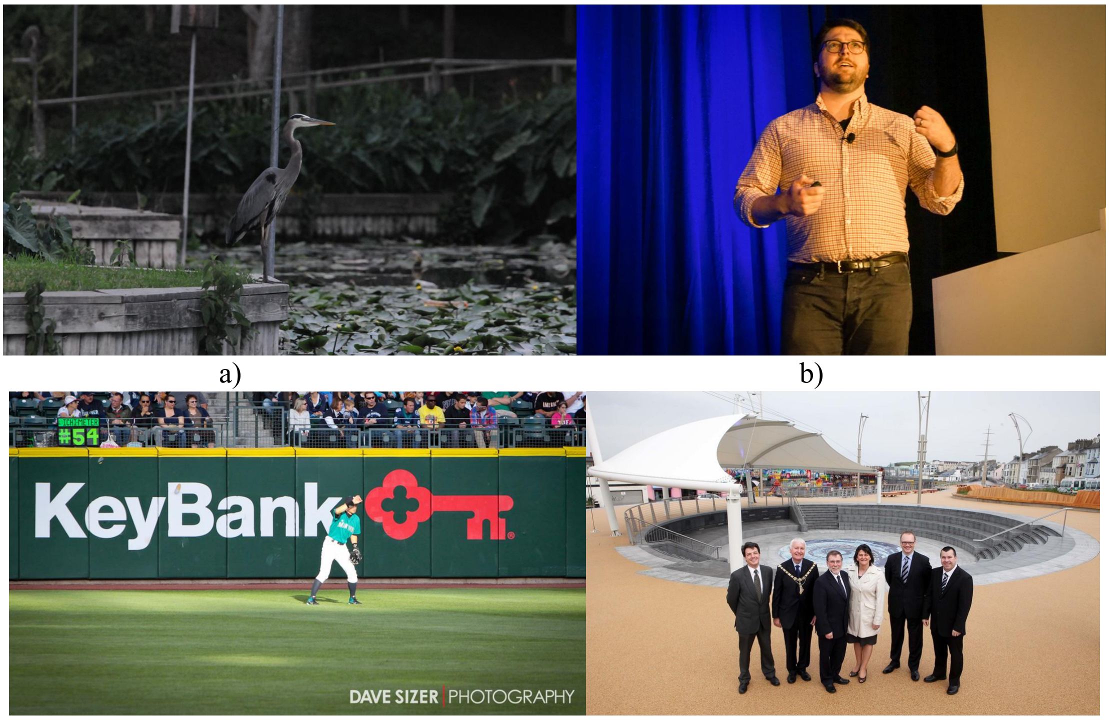
- learning hyperparameters;
- data augmentation;
- data shuffling;
- number of tags to be learnt.

For this demonstrative experiment, most learning hyperparameters were chosen empirically, with no prior cross validation. Exploring parameter hyperspace via simple validation or cross validation could enhance the training speed and resulting accuracy.

The lack of data augmentation, as mentioned previously, limits the effective number of training epochs that could be run without unintentionally messing up parameters. Data augmentation is the next best thing after sourcing more data, which is the whole point of this research. Same could be said about data shuffling.

The number of tags to be learnt in an important hyperparameter. As we took the P value to be 0.2, the number of tags became 10718. This is quite a large number of output values for a neural network. This means that there are much more negative cases for each output than positive ones. Thus, learning is complicated by the imbalance in the dataset.

With this in mind, here are some of the examples of the model output. The examples are split into positive and negative not by comparing model output to the recorded tags, but by human evaluation of the tags selected by the model. Figure 4 demonstrates some positive examples where the model tagged the images in the expected way.



a) b)

c) d)

Figure 4: Examples of well tagged images [48]

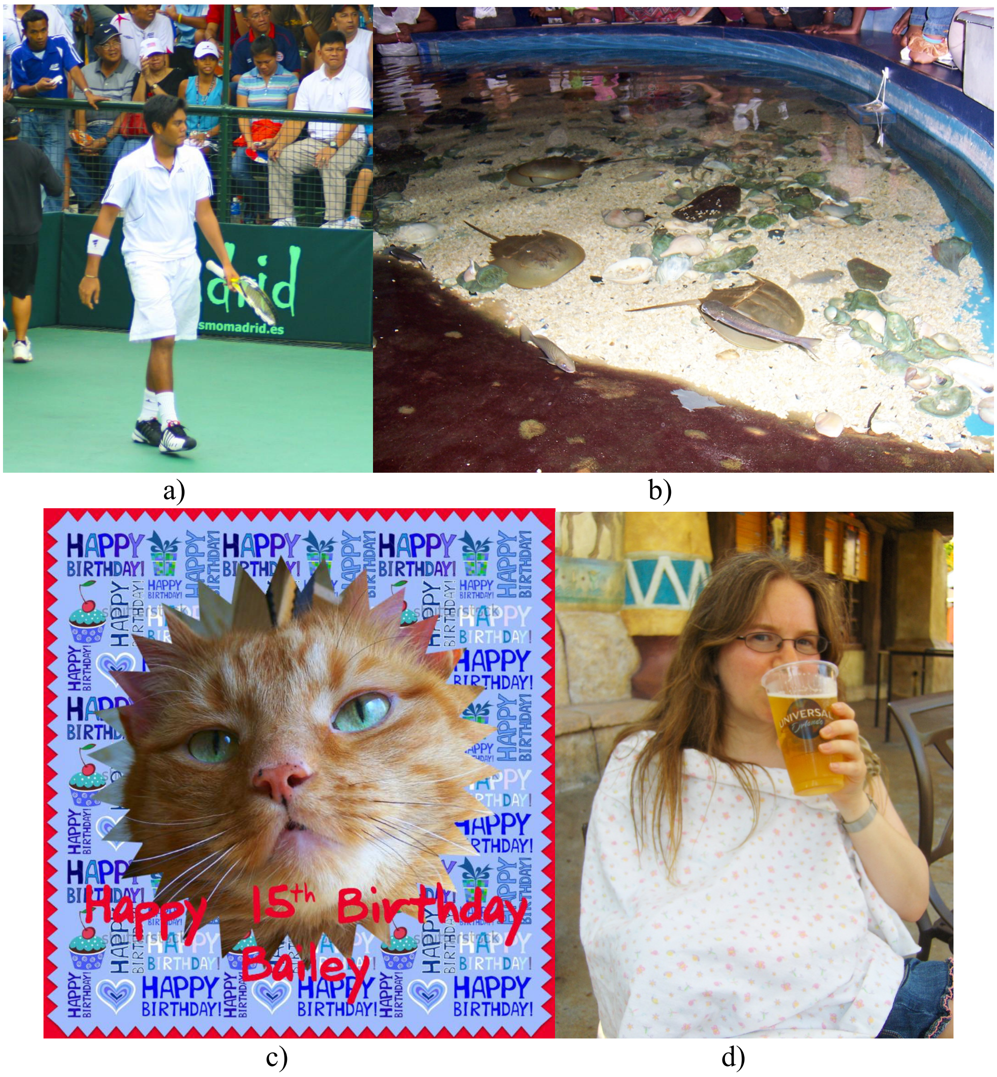
Image a) in fig. 4. shows a bird standing in an artificial enclosure by a pond. The model produced the following tags: “animal”, “bird”, “water”, “street”. Here and later, tags are provided in the order of descending confidence. However, the confidence levels themselves do not have any special meaning beyond the model, thus we omit them.

Image a) in fig. 4. shows a man making a speech in what appears to be a conference hall. The model produced the following tags: “person”, “gesture”, “hand”, “curtain”.

Image c) shows a sportsman in a competition. The model produced the following tags: “key”, “sport”, “cowd”, “street”.

Finally, image d) shows a group of people standing in an open area. The model produced the following tags: “person”, “theater”, “town”, “street”.

Figure 5 demonstrates the negative examples of tagging.



a) b)

c) d)

Figure 5: Examples of poorly tagged images [48]

Image a) in fig. 5. shows a tennis player. The model tagged it with “soccer”, “ball”, “cup”, “open”. Note that “open” is probably related to tennis, but the other tags are not correct.

Image b) in fig. 5. shows an aquarium. The model tagged it with “sea”, “ocean”, “fish”, “mammal”. Here, the model ignored more subtle clues like fragments of hands of people standing around the aquarium.

Image c) shows a cat birthday card, which is an example of bad data. The model tagged it with “cat”, “animal”, “box”, “fur”.

Image d) shows a woman drinking beer. The model tagged it with “person”, “glasses”, “barrel”, “street”. This is probably because of bad data from the definition of “beer”.

After analyzing the results, we conclude that the dataset is generally acceptable, though some records still contain unwanted noise. To improve the quality of the data, some human supervision would be beneficial.

6. Discussions

The obtained results show the clear efficiency of the approach of enhancing image data with text information. For the simple, tagging task, we managed to obtain adequate performance by simply fine-tuning an existing model on data generated by scraping Wikipedia.

For the presented task, as well as for any other task, it would be beneficial to use human-generated annotations for class names instead of snippets of Wikipedia. However, it can be enough for many tasks.

However, the presented experiment in training a labeling model only begins to explore the capabilities of the approach. Multi-modal data has many applications which, unlike the tagging task, cannot be solved by other means. However, multi-modal data is not always readily available, specially in more specialized domains. The approach of enhancing an image dataset with text data sourced from simple description of classes, rather than descriptions of each individual image, can bridge the gap between the required and the present datasets.

We presume some advances can be made in the task of image classification with the use of multi-modal datasets, such as the one presented in this paper. The approaches may include few-shot learning improvement based on similarities of text descriptions of the new and the existing classes, text generation from the images, which also can help with classification, using image labeling for the purposes of text classification, etc.

The idea behind this approach is to gain as much usable information from given data as possible with little effort. The approach targets the situations where it is not feasible to obtain more image data for one reason or another. Along with few-shot learning situations, where data for some classes is not as abundant as for others, this is useful for domain-specific tasks, where any data may be scarce.

Finally, the approach may be useful for achieving or beating state-of-the-art performance with fewer trainable parameters.

The need for fewer parameters is dictated by the growing computational complexity of modern machine learning models. This, in turn, leads to longer training times, slower inference, and more expensive hardware requirements. Thus achieving, or even approaching, state-of-the-art performance on vision tasks with significantly less parameters is a necessity for all the researchers and commercial users who do not have prolonged access to high-cost computational facilities.

When using two models, one for vision and another for NLP, which have fewer parameters combined than typical modern vision models, applying the multi-modal approach can turn out beneficial for the combined system performance. The potential benefits of the approach with image class descriptions lie in several factors:

- Outputs of the NLP models can be cached and reused during training. If the NLP model is completely frozen and only performs inference with no error backpropagation, such outputs may be generated beforehand for all the possible classes and accessed as a simple read operation. This, in fact, transfers some of the load of learning to the preparation stage and thus speeds up both learning and inference.
- If the NLP model is being trained along with the vision model, its outputs can still be cached if the training is organized in such a manner that images of the same class are fed to the models in the same batch. In such a case, the batch size for the NLP model is effectively reduced to one. This speeds up the training process as well.
- The two models can be trained in parallel on different devices. The overhead of combining the two outputs of the models could be less than the overhead of training a single model in a distributed cluster of devices.

All the proposed approaches could be topics of further research in this sphere. For instance, providing a model ensemble that can run one of its key parts with a few times less resources, by the factor of the batch size, allows more flexibility for the researchers. Making the NLP model more lean can allow us to divert it to a CPU, while freeing costly GPU resources for the vision model. If the NLP model does require the GPU, we can easily split it onto a different machine and only combine the ensemble data after a pass is done.

7. Conclusions

In this paper we presented the ImageD dataset. The dataset combines the existing labeled images from the Open Images dataset with text descriptions for each of the classes of objects found in the images. The dataset can be used for a variety of research purposes, related to image classification, description, labeling, etc.

We also present a simple experiment in generating labels for images with data built on top of the said dataset. The trained model shows adequate results in labeling never before seen images. This leads us to believe that, given enough effort, such an approach could be scaled for greater efficiency and performance.

It is particularly interesting how the presented dataset may be used for image classification purposes to extract more data about the images at the training stage. Some techniques of combining text information with image data for the purpose of higher performance in classification already exist. One such technique, NLP supervised learning, seems to yield great results and deserves more attention.

As well as providing more information for the neural networks to train on, multimodal dataset also enable researchers to construct more complex and more scalable models. When combining the two factors of more data and more scalable models, this approach to gathering data has the power to optimize machine learning algorithms on several levels.

Multimodal datasets assembled from different existing sources of information are a viable first step towards harvesting the listed benefits. However, with application of human moderation and manual data cleaning, the tool becomes yet more efficient.

8. References

[1] J. Wang, Y. Zheng, M. Wang, Q. Shen and J. Huang, "Object-Scale Adaptive Convolutional Neural Networks for High-Spatial Resolution Remote Sensing Image Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 283-299, 2021, doi: 10.1109/JSTARS.2020.3041859.

[2] D. Xue et al., "An Application of Transfer Learning and Ensemble Learning Techniques for Cervical Histopathology Image Classification," in IEEE Access, vol. 8, pp. 104603-104618, 2020, doi: 10.1109/ACCESS.2020.2999816.

[3] T. Mao, H. Tang and W. Huang, "Unsupervised Classification of Multispectral Images Embedded With a Segmentation of Panchromatic Images Using Localized Clusters," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 11, pp. 8732-8744, Nov. 2019, doi: 10.1109/TGRS.2019.2922672.

[4] A. Van Opbroek, H. C. Achterberg, M. W. Vernooij and M. De Bruijne, "Transfer Learning for Image Segmentation by Combining Image Weighting and Kernel Learning," in IEEE Transactions on Medical Imaging, vol. 38, no. 1, pp. 213-224, Jan. 2019, doi: 10.1109/TMI.2018.2859478.

[5] E. Bonmati et al., "Voice-Assisted Image Labeling for Endoscopic Ultrasound Classification Using Neural Networks," in IEEE Transactions on Medical Imaging, vol. 41, no. 6, pp. 1311-1319, June 2022, doi: 10.1109/TMI.2021.3139023.

[6] Y. Ding, X. Zhao, Z. Zhang, W. Cai and N. Yang, "Multiscale Graph Sample and Aggregate Network With Context-Aware Learning for Hyperspectral Image Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 4561-4572, 2021, doi: 10.1109/JSTARS.2021.3074469.

[7] X. Ji, Y. Cui, H. Wang, L. Teng, L. Wang and L. Wang, "Semisupervised Hyperspectral Image Classification Using Spatial-Spectral Information and Landscape Features," in IEEE Access, vol. 7, pp. 146675-146692, 2019, doi: 10.1109/ACCESS.2019.2946220.

[8] R. Tennakoon et al., "Classification of Volumetric Images Using Multi-Instance Learning and Extreme Value Theorem," in IEEE Transactions on Medical Imaging, vol. 39, no. 4, pp. 854-865, April 2020, doi: 10.1109/TMI.2019.2936244.

[9] K. Pawar, Z. Chen, N. J. Shah and G. F. Egan, "A Deep Learning Framework for Transforming Image Reconstruction Into Pixel Classification," in IEEE Access, vol. 7, pp. 177690-177702, 2019, doi: 10.1109/ACCESS.2019.2959037.

[10] K. Y. Ma and C. -I. Chang, "Kernel-Based Constrained Energy Minimization for Hyperspectral Mixed Pixel Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-23, 2022, Art no. 5510723, doi: 10.1109/TGRS.2021.3085801.

[11] X. Shang, S. Han and M. Song, "Iterative Spatial-Spectral Training Sample Augmentation for Effective Hyperspectral Image Classification," in IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2022, Art no. 6005305, doi: 10.1109/LGRS.2021.3131373.

[12] Z. Li et al., "A Unified Understanding of Deep NLP Models for Text Classification," in IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 12, pp. 4980-4994, 1 Dec. 2022, doi: 10.1109/TVCG.2022.3184186.

[13] S. Singh and A. Mahmood, "The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures," in IEEE Access, vol. 9, pp. 68675-68702, 2021, doi: 10.1109/ACCESS.2021.3077350.

[14] F. Rodríguez-Sánchez, J. Carrillo-de-Albornoz and L. Plaza, "Automatic Classification of Sexism in Social Networks: An Empirical Study on Twitter Data," in IEEE Access, vol. 8, pp. 219563-219576, 2020, doi: 10.1109/ACCESS.2020.3042604.

[15] X. Cao, J. Yu and Y. Zhuang, "Injecting User Identity Into Pretrained Language Models for Document-Level Sentiment Classification," in IEEE Access, vol. 10, pp. 30157-30167, 2022, doi: 10.1109/ACCESS.2022.3158975.

[16] N. Cerkez, B. Vrdoljak and S. Skansi, "A Method for MBTI Classification Based on Impact of Class Components," in IEEE Access, vol. 9, pp. 146550-146567, 2021, doi: 10.1109/ACCESS.2021.3121137.

[17] N. Jin, J. Wu, X. Ma, K. Yan and Y. Mo, "Multi-Task Learning Model Based on Multi-Scale CNN and LSTM for Sentiment Classification," in IEEE Access, vol. 8, pp. 77060-77072, 2020, doi: 10.1109/ACCESS.2020.2989428.

[18] A. O. Akmandor, J. Ortiz, I. Manotas, B. Ko and N. K. Jha, "SECRET: Semantically Enhanced Classification of Real-World Tasks," in IEEE Transactions on Computers, vol. 70, no. 3, pp. 440-456, 1 March 2021, doi: 10.1109/TC.2020.2989642.

[19] D. Rogers, A. Preece, M. Innes and I. Spasić, "Real-Time Text Classification of User-Generated Content on Social Media: Systematic Review," in IEEE Transactions on Computational Social Systems, vol. 9, no. 4, pp. 1154-1166, Aug. 2022, doi: 10.1109/TCSS.2021.3120138.

[20] H. Q. Abonizio, E. C. Paraiso and S. Barbon, "Toward Text Data Augmentation for Sentiment Analysis," in IEEE Transactions on Artificial Intelligence, vol. 3, no. 5, pp. 657-668, Oct. 2022, doi: 10.1109/TAI.2021.3114390.

[21] A. A. R. Magna, H. Allende-Cid, C. Taramasco, C. Becerra and R. L. Figueroa, "Application of Machine Learning and Word Embeddings in the Classification of Cancer Diagnosis Using Patient Anamnesis," in IEEE Access, vol. 8, pp. 106198-106213, 2020, doi: 10.1109/ACCESS.2020.3000075.

[22] S. Gao et al., "Limitations of Transformers on Clinical Text Classification," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 9, pp. 3596-3607, Sept. 2021, doi: 10.1109/JBHI.2021.3062322.

[23] K. Smelyakov, M. Hvozdiev, A. Chupryna, D. Sandrkin and V. Martovytskyi, "Comparative Efficiency Analysis of Gradational Correction Models of Highly Lighted Image," 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 703-708, doi: 10.1109/PICST47496.2019.9061356.

[24] Y. Wang, W. Song, G. Fortino, L. -Z. Qi, W. Zhang and A. Liotta, "An Experimental-Based Review of Image Enhancement and Image Restoration Methods for Underwater Imaging," in IEEE Access, vol. 7, pp. 140233-140251, 2019, doi: 10.1109/ACCESS.2019.2932130.

[25] K. Smelyakov, A. Chupryna, O. Bohomolov and N. Hunko, "The Neural Network Models Effectiveness for Face Detection and Face Recognition," 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2021, pp. 1-7, doi: 10.1109/eStream53087.2021.9431476.

[26] K. Smelyakov, M. Shupyliuk, V. Martovytskyi, D. Tovchyrechko and O. Ponomarenko, "Efficiency of image convolution," 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), 2019, pp. 578-583, doi: 10.1109/CAOL46282.2019.9019450.

[27] G. Krivoulya, I. Ilina, V. Tokariev and V. Shcherbak, "Mathematical Model for Finding Probability of Detecting Victims of Man-Made Disasters Using Distributed Computer System with Reconfigurable Structure and Programmable Logic," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2020, pp. 573-576, doi: 10.1109/PICST51311.2020.9467976.

[28] Sharonova, N., Kyrychenko, I., Tereshchenko, G., “Application of big data methods in E-learning systems”, 2021 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), 2021. – CEUR-WS, 2021, ISSN 16130073. - Volume 2870, PP. 1302-1311.

[29] Gruzdo, I., Kyrychenko, I., Tereshchenko, G., Shanidze, N., "Metrics applicable for evaluating software at the design stage," 2021 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), 2021. – CEUR-WS, 2021, ISSN 16130073. - Volume 2870, PP. 916-936.

[30] K. T. Chitty-Venkata, M. Emani, V. Vishwanath and A. K. Somani, "Neural Architecture Search for Transformers: A Survey," in IEEE Access, vol. 10, pp. 108374-108412, 2022, doi: 10.1109/ACCESS.2022.3212767.

[31] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P., “Microsoft COCO: Common Objects in Context” arXiv, 2014

[32] P. Young, A. Lai, M. Hodosh, J. Hockenmaier, “From image descriptions to visual denotations”. URL: https://shannon.cs.illinois.edu/DenotationGraph/.

[33] Sharma, Piyush and Ding, Nan and Goodman, Sebastian and Soricut, Radu, “Conceptual Captions”, Proceedings of ACL, 2018.

[34] “ImageNet benchmark (Image Classification) | Papers with Code”. URL: https://paperswithcode.com/sota/image-classification-on-imagenet

[35] “MNIST Dataset | Papers with Code” URL: https://paperswithcode.com/dataset/mnist

[36] “CIFAR-100 (Image Classification) | Papers with Code”. URL: https://paperswithcode.com/sota/image-classification-on-cifar-100

[37] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, “CoCa: Contrastive Captioners are Image-Text Foundation Models.” arXiv, 2022

[38] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, “Scaling Vision Transformers.” arXiv, 2021

[39] H. Touvron, A. Vedaldi, M. Douze, H. Jegou, “Fixing the train-test resolution discrepancy”, Advances in Neural Information Processing Systems 32, Vancouver, Canada, 2019.

[40] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-Aware Minimization for Efficiently Improving Generalization.” arXiv, 2020. doi: 10.48550/ARXIV.2010.01412.

[41] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” arXiv, 2019, doi: 10.48550/ARXIV.1905.11946.

[42] A. Vaswani et al., “Attention Is All You Need.” arXiv, 2017. doi: 10.48550/ARXIV.1706.03762.

[43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, 2015. doi: 10.48550/ARXIV.1512.03385.

[44] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” arXiv, 2019, doi: 10.48550/ARXIV.1905.11946.

[45] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, “ImageNet-21K Pretraining for the Masses.” arXiv, 2021. doi: 10.48550/ARXIV.2104.10972.

[46] H. Pham, Z. Dai, Q. Xie and Q. V. Le, "Meta Pseudo Labels," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 11552-11563.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv, 2018. doi: 10.48550/ARXIV.1810.04805.

[48] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, “The Open Images Dataset V4”, IJCV, 2020.

[49] “GitHub: ddashenkov/ImageD” URL: https://github.com/ddashenkov/ImageD.

[50] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated Residual Transformations for Deep Neural Networks.” arXiv, 2016. doi: 10.48550/ARXIV.1611.05431.