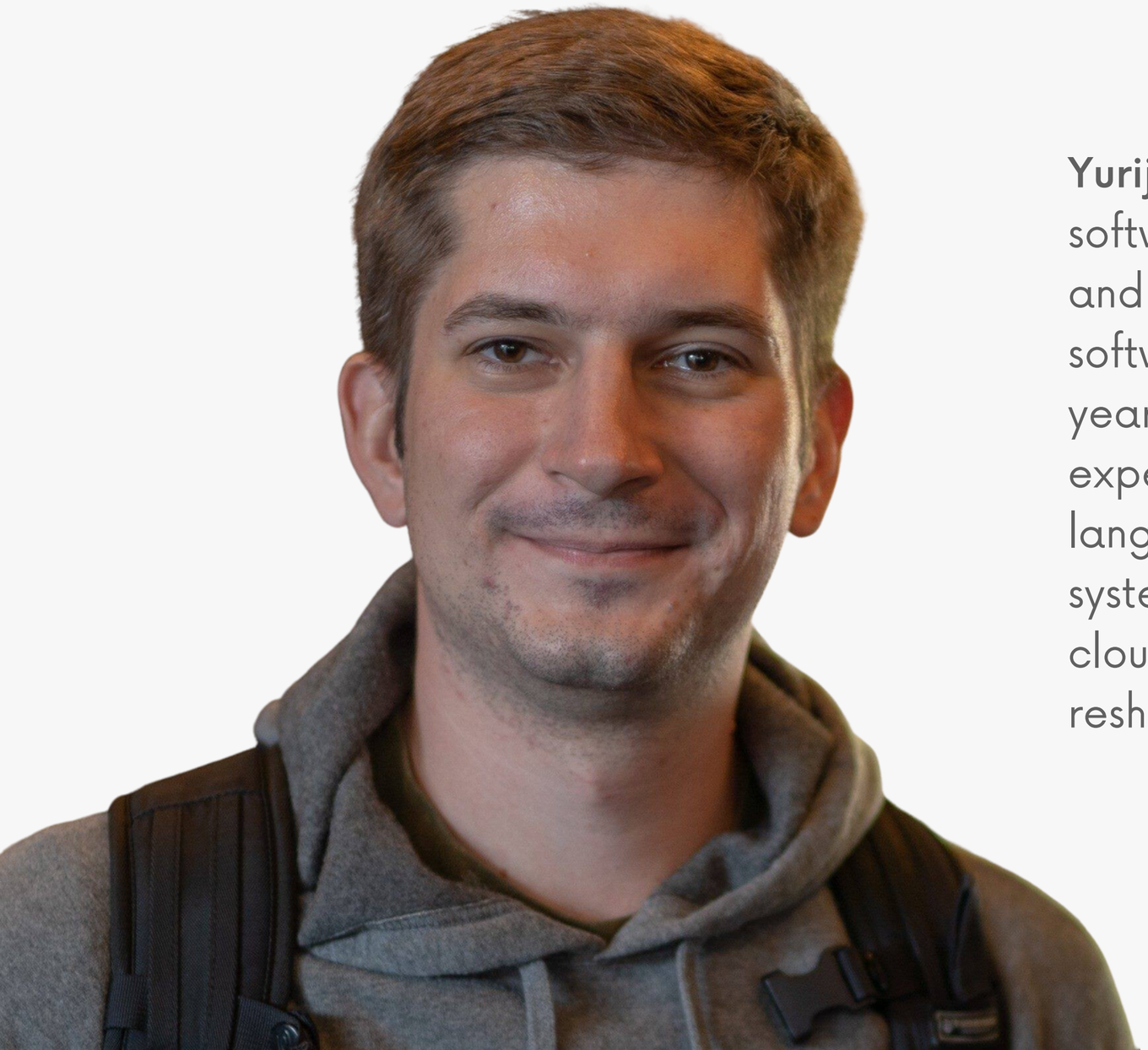
USING TEXT QUERIES TO LOOK UP UNLABELED IMAGES: A COMMAND-LINE SEARCH TOOL BASED ON CLIP

YURIJ MIKHALEVICH

DUBAI, UNITED ARAB EMIRATES

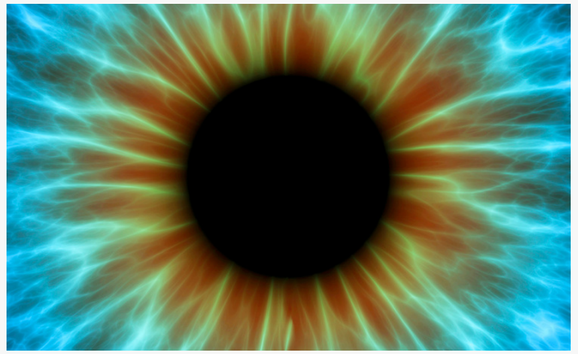
EMAIL: YURIJ@MIKHALEVI.CH



Yurij Mikhalevich, MSc Computer Science, is a software engineer, machine learning engineer, and researcher with over ten years of industrial software engineering experience and over eight years of industrial machine learning engineering experience focusing on computer vision, natural language processing, and recommendation systems. Presently, he is building the next-level cloud AI platform at Lightning AI, which is reshaping how AI products are built.

RESEARCH INTERESTS

On the right are Yurij's current primary research interests.

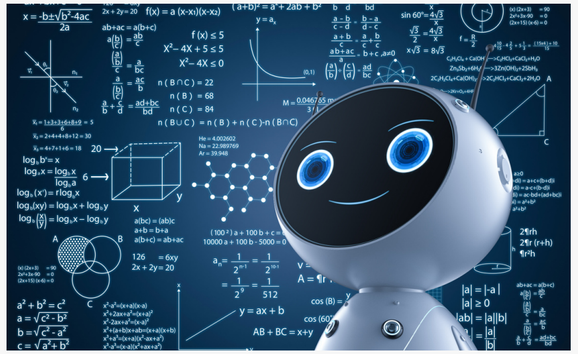


Computer Vision

Both image and video processing, with the current focus on diffusion models and vision transformers.

Natural Language Processing

With a focus on recommendation systems and generative language models.



Reinforcement Learning

Systems that learn from the environment are fascinating.

INTRODUCTION

The release of the Contrastive Language-Image Pre-Training (CLIP) model by OpenAI in 2021 has generated significant attention in the field of Natural Language Processing (NLP) and Computer Vision (CV). This model has demonstrated the ability to learn state-of-the-art image representations from a massive dataset of 400 million (image, text) pairs that were collected from the Internet. CLIP can predict the most relevant text snippet, given an image, using natural language instructions without explicitly optimizing for the task.
In addition to these capabilities, CLIP allows researchers to perform image searches using natural language queries. This article explores this particular application of CLIP.

RELATED WORKS

Traditional image search methods rely on bag-of-features, which are sets of features that describe the contents of an image. The features can be:

- created manually
- retrieved from the metadata injected into photos by the camera or mobile phone software
- added automatically by the image recognition algorithms, such as Convolutional Neural Networks (CNNs)
- obtained by using a combination of the methods listed above

These labels can then be searched using:

- (for GPS coordinates in the photo metadata) using distance-based search algorithms
- text-based search algorithms, ranging from substring matching to lemmatization-based search or even using word embeddings
- or using a combination of multiple approaches

METHOD: SIMILARITY

With CLIP's text transformer, it is possible to convert a text query to a **n**-dimensional vector. With CLIP's image transformer, it is possible to convert an image to a **n**-dimensional vector. Then, we can calculate the dot product of the normalized query vector and each of the normalized image vectors. After this, we sort the images by the decreasing dot product and take the first **k** images; this gets us the **k** images that match the query the most.

$$\boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

METHOD: CACHING

To ensure that the method scales well, the solution proposed in this paper suggests caching the image vectors. By caching the image vectors, repeated queries can be executed quickly, without having to wait for the image to be processed each time that a query is made.

Caching involves storing the computed image vectors on disk after the initial processing of images so that they can be accessed later. When a new query is received, the system retrieves the cached image vectors and computes only the query vector. This approach reduces the computational overhead associated with image querying and improves the response time for users.

METHOD: ADVANTAGE

The main advantage of the proposed method over the methods described earlier is that method does not require any metadata, labels, or annotations, which enables using it with any image catalog. Moreover, if used on labeled images, the proposed method allows searching for concepts not included in the image labels. The presence of these mechanisms enables users to utilize CLIP effectively in real-world scenarios involving image search.

IMPLEMENTATION

The method described above is implemented in the Python utility called **rclip**. The utility uses OpenAI's ViT-B/32 version of the CLIP model to compute the vectors and the SQLite 3 RDBMS to implement the vector cache. The cache table is defined as follows:

```
CREATE TABLE IF NOT EXISTS images (
  id INTEGER PRIMARY KEY,
  deleted BOOLEAN,
  filepath TEXT NOT NULL UNIQUE,
  modified_at DATETIME NOT NULL,
  size INTEGER NOT NULL,
  vector BLOB NOT NULL
)
```

The **rclip** source code is published on GitHub under the MIT license:
github.com/yurijmikhalevich/rclip

PERFORMANCE: MODEL COMPARISON

The table below shows how **rclip** performs when using ViT-B/32 and ViT-L/14@336px CLIP models. The tests were run on the **Intel(R) Celeron(R) CPU J3455 @ 1.50GHz**. Given the poor performance of the ViT-L/14@336px model, the decision was made to use the ViT-B/32 model in **rclip**.

Model	Indexing Time	Search Time
`ViT-B/32`	3m56.626s	0m18.064s
`ViT-L/14@336px`	125m0.507s	3m19.742s
Difference	x31.70	x11.06

PERFORMANCE: SCALABILITY

The table below shows how the **rclip** performance scales. The tests were performed on the **Apple M1 Max CPU**. The indexing time scales linearly with the number of images when the search time increases only slightly, even when going from searching through 50 thousand images to searching through 1.28 million images.

Dataset	# of images	Indexing Time	Search Time
`ImagNet-1k` validation set	50k	19m24.750s	0m4.04s
`ImagNet-1k` train set	1.28m	8h31m26.680s	0m11.49s
Difference	x25.62	x26.35	x2.84

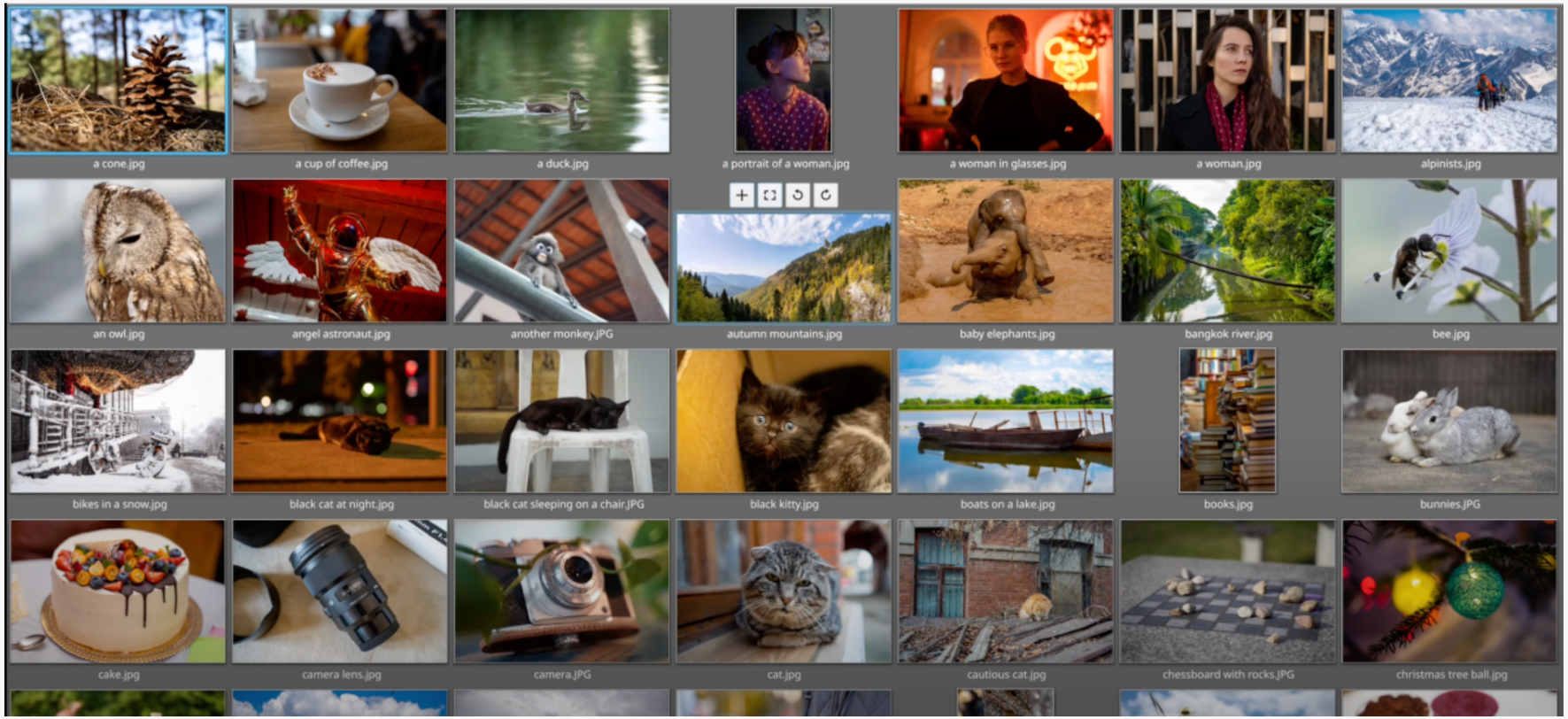
SEARCH QUALITY: BENCHMARK

rclip achieves 31.17% top-1 accuracy and 44.80% top-5 accuracy rate on the **ImageNet-1k** 1.28 million images train set and 55.15% top-1 and 81.34% top-5 accuracy on the **CIFAR-100** 10 thousand images test set.

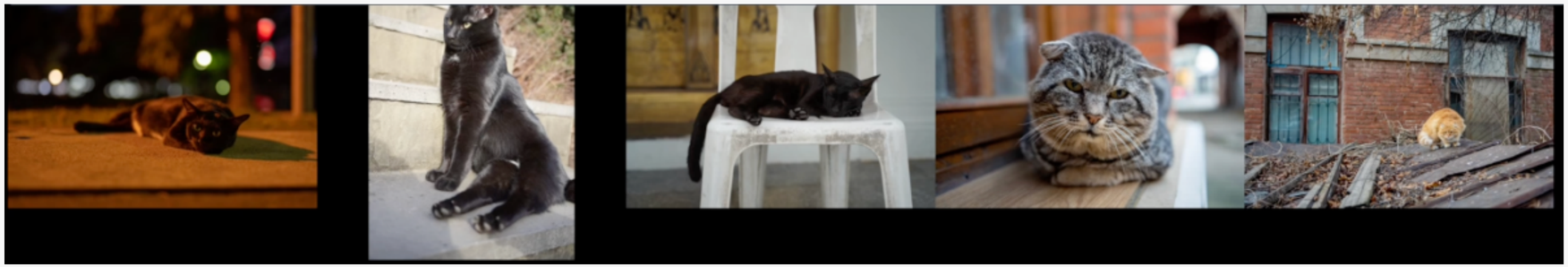
Model	Top-1 accuracy	Top-5 accuracy
`ImageNet-1k` 1.28m	31.17%	44.80%
`CIFAR-100` 10k	55.15%	81.34%

SEARCH QUALITY: DEMO SET SAMPLE

Search results presented on the following slides were obtained by querying this demo set.



SEARCH QUALITY: SEARCH RESULT FOR QUERY "CAT"



SEARCH QUALITY: SEARCH RESULT FOR QUERY "LEADING LINES"

SEARCH QUALITY: TOP RESULT FOR QUERY "A KITTEN PEEKING FROM BEHIND A CORNER"

FUTURE PLANS

- to create a separate model for the CLIP text transformer and to load only it when users initiate a search that does not require indexing, thereby avoiding the loading of the CLIP vision transformer and improving the querying performance;
- to push the tool's scaling limits even further by introducing the sharded cache;
- to improve the tool's performance by preventing it from re-indexing files when they are renamed;
- to enable **rclip** to write tags to the image file metadata for third-party software to utilize them;
- to enrich **rclip**'s search capabilities by utilizing metadata, which already exists within the images, like GPS coordinates, image capture date, camera model, tags, etc.;
- to do an in-depth comparison of **rclip** with other existing image search tools;
- to explore how well CLIP handles distorted and corrupted images.

CONCLUSION

The development of the command-line tool, **rclip**, which employs OpenAI's CLIP model, has resulted in an efficient and user-friendly utility for image search. This paper introduces a practical and scalable method of searching images using natural language queries based on the CLIP model. The approach has demonstrated impressive results on the ImageNet-1k and CIFAR-100 datasets, indicating its potential applicability to a wide range of industries reliant on visual data.