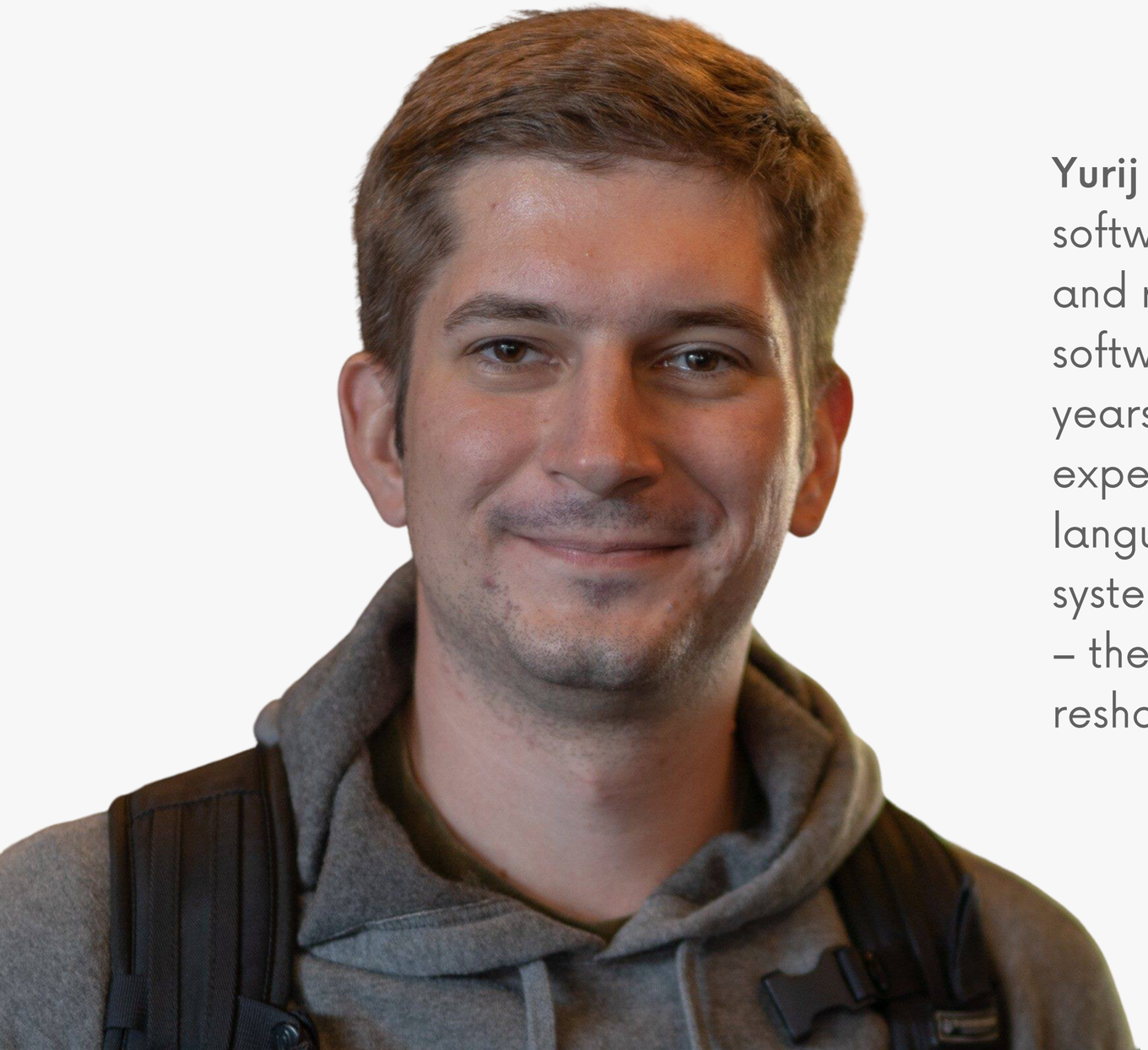
AUTOMATIC SEMANTIC IMAGE TAGGING AT SCALE: AI-POWERED COMMAND-LINE TOOL BASED ON CLIP

YURIJ MIKHALEVICH

DUBAI, UNITED ARAB EMIRATES

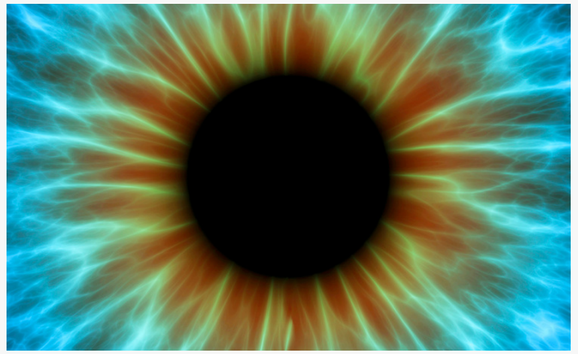
EMAIL: YURIJ@MIKHALEVI.CH



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

RESEARCH INTERESTS

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

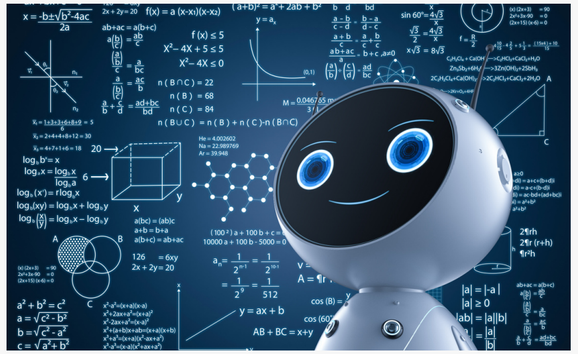


Computer Vision

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 unveiling of the CLIP model by OpenAI in 2021 has captured widespread interest across the domains of Natural Language Processing (NLP) and Computer Vision (CV). This innovative model is capable of understanding advanced image representations by analyzing a vast collection of 400 million image and text pairs sourced from the internet. CLIP uniquely identifies the most applicable text snippet for an image through natural language guidance without being directly trained for such tasks. Its zero-shot learning capabilities mirror those found in GPT-2 and GPT-3. The authors of CLIP have showcased its ability to match the performance of the original ResNet50 on the ImageNet dataset in a zero-shot setting without using any of the 1.28 million labeled examples. This breakthrough addresses some of the most daunting challenges in the field of computer vision.

Given its capabilities, CLIP emerges as an ideal foundation for developing a semantic image tagging tool capable of operating in a zero-shot manner with any user-provided tags or images. This article explores this particular application of CLIP.

RELATED WORKS

- Foundation: Early works combined neural networks with visual and textual data, pioneering multimodal learning.
- Image Classification Milestones:
  - AlexNet showcased deep learning's potential.
  - ResNet introduced residual learning for deeper networks.
- NLP Breakthroughs: BERT revolutionized language understanding, influencing multimodal models.
- Multimodal Learning Evolution:
  - ViLBERT enhanced tasks requiring both visual and textual comprehension.
  - CLIP, a significant leap, learns visual concepts from text, enabling zero-shot image classification.

METHOD: SIMILARITY

With CLIP's text transformer, it is possible to convert a text label (tag) into 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 image vector and each of the normalized tag vectors. After this, we pick tags with the dot product higher than the configured threshold; this gets us the tags that match the image 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. The solution reuses the cache implementation provided by the tool **rclip**. This approach reduces the computational overhead associated with image tagging and improves the response time for users. This is particularly useful when the user needs to retag the images with a different set of tags or when the user already has their images indexed by **rclip**.

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

IMPLEMENTATION

The method described above is implemented in the Python utility called **rtag**.
The solution uses the **rclip** library to compute the feature vectors. **rclip** uses ViT-B/32 version of the OpenAI's CLIP model. The tool writes the computed tags to the image IPTC metadata using the IPTCInfo3 Python library.

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

PERFORMANCE

rtag was benchmarked on the ObjectNet 50,273 images dataset, on the Apple M1 Max CPU. The table below shows how **rtag** performs when tagging previously indexed and unindexed images. As you can see from the table, running **rtag** on 50,273 unindexed images took 6m21.250s, while tagging 965 unindexed images took 0m12.040s. A similar linear scaling relationship is preserved when running **rtag** on indexed images. Tagging 50,273 indexed images took 4m44.790s while tagging 965 indexed images took 0m09.140s.

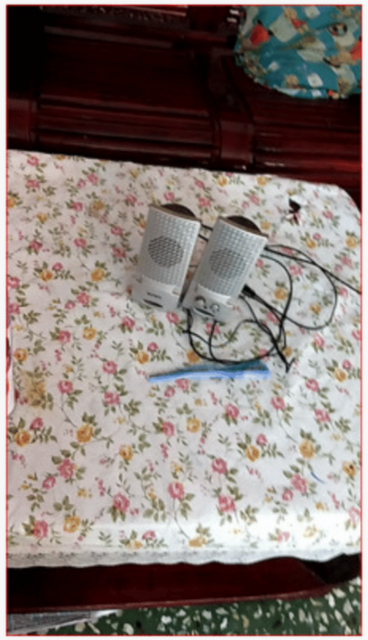
Dataset	# of images	Indexed	Processing time
`ObjectNet` dataset	50,273	No	6m21.250s
`ObjectNet` dataset	50,273	Yes	4m44.790s
`ObjectNet` subset	965	No	0m12.040s
`ObjectNet` subset	965	Yes	0m09.140s

SEARCH QUALITY: BENCHMARK

As the Table below shows, **rtag** achieves 27.22% top-1 accuracy and 50.42% top-5 accuracy rate on the ObjectNet 50,273 images dataset. The ObjectNet dataset was chosen for the quality measurement because it is a diverse and challenging dataset that contains images of objects in their natural environments.

Model	Top-1 accuracy	Top-5 accuracy
ObjectNet	25.28%	48.05%
ObjectNet photo of [tag]	27.22%	50.42%

SEARCH QUALITY: AN IMAGE OF SPEAKERS TAGGED BY RTAG



```
Profile-iptc: 334 bytes
  unknown[2,0]:
  Keyword[2,25]: cassette
  Keyword[2,25]: cassette player
  Keyword[2,25]: dial telephone, dial phone
  Keyword[2,25]: electric fan, blower
  Keyword[2,25]: handkerchief, hankie, hanky, hankey
  Keyword[2,25]: iPod
  Keyword[2,25]: loudspeaker, speaker, speaker unit, loudspeaker system, speaker system
  Keyword[2,25]: modem
  Keyword[2,25]: mosquito net
  Keyword[2,25]: notebook, notebook computer
  Keyword[2,25]: radio, wireless
  Keyword[2,25]: sewing machine
  Keyword[2,25]: tape player
```

SEARCH QUALITY: AN IMAGE OF A SOAP TAGGED BY RTAG

```
Profile-iptc: 280 bytes
  unknown[2,0]:
  Keyword[2,25]: bathtub, bathing tub, bath, tub
  Keyword[2,25]: dishwasher, dish washer, dishwashing machine
  Keyword[2,25]: maraca
  Keyword[2,25]: plunger, plumber's helper",
  Keyword[2,25]: screwdriver
  Keyword[2,25]: soap dispenser
  Keyword[2,25]: toilet seat
  Keyword[2,25]: tub, vat
  Keyword[2,25]: washbasin, handbasin, washbowl, lavabo, wash-hand basin
  Keyword[2,25]: butternut squash
```

SEARCH QUALITY: AN IMAGE OF A CAMERA TAGGED BY RTAG

```
  Profile-iptc: 270 bytes
    unknown[2,0]:
    Keyword[2,25]: binoculars, field glasses, opera glasses
    Keyword[2,25]: bolo tie, bolo, bola tie, bola
    Keyword[2,25]: espresso maker
    Keyword[2,25]: holster
    Keyword[2,25]: lens cap, lens cover
    Keyword[2,25]: loupe, jeweler's loupe",
    Keyword[2,25]: Polaroid camera, Polaroid Land camera
    Keyword[2,25]: reflex camera
    Keyword[2,25]: slot, one-armed bandit
    Keyword[2,25]: tripod
```

FUTURE PLANS

- to improve the tool's tagging performance by processing multiple images in parallel;
- to add support for writing tags into the XMP sidecar files and leaving the image files intact;
- to improve the tool's performance by preventing it from re-indexing files when they are renamed;
- to assess the tool's performance on other datasets;
- to enrich **rtag**'s tagging capabilities by utilizing metadata, which already exists within the images, e.g., existing GPS coordinates can be used to tag images with human-readable location-based tags;
- to do an in-depth comparison of rtag with other existing image tagging tools.

CONCLUSION

In summary, this paper introduces a practical and scalable method of tagging images of any kind with any set of labels. The method is based on the CLIP model. The approach has demonstrated impressive results on the ObjectNet dataset, indicating its potential applicability to a wide range of industries reliant on visual data and using image processing tools requiring images to be tagged.

AUTOMATIC SEMANTIC IMAGE TAGGING AT SCALE: AI-POWERED COMMAND-LINE TOOL BASED ON CLIP

YURIJ MIKHALEVICH

DUBAI, UNITED ARAB EMIRATES

EMAIL: YURIJ@MIKHALEVI.CH