Efficient Image Clustering Conditioned on Text Criteria

Sheikh Shafayat

Recap on my previous paper presentation…

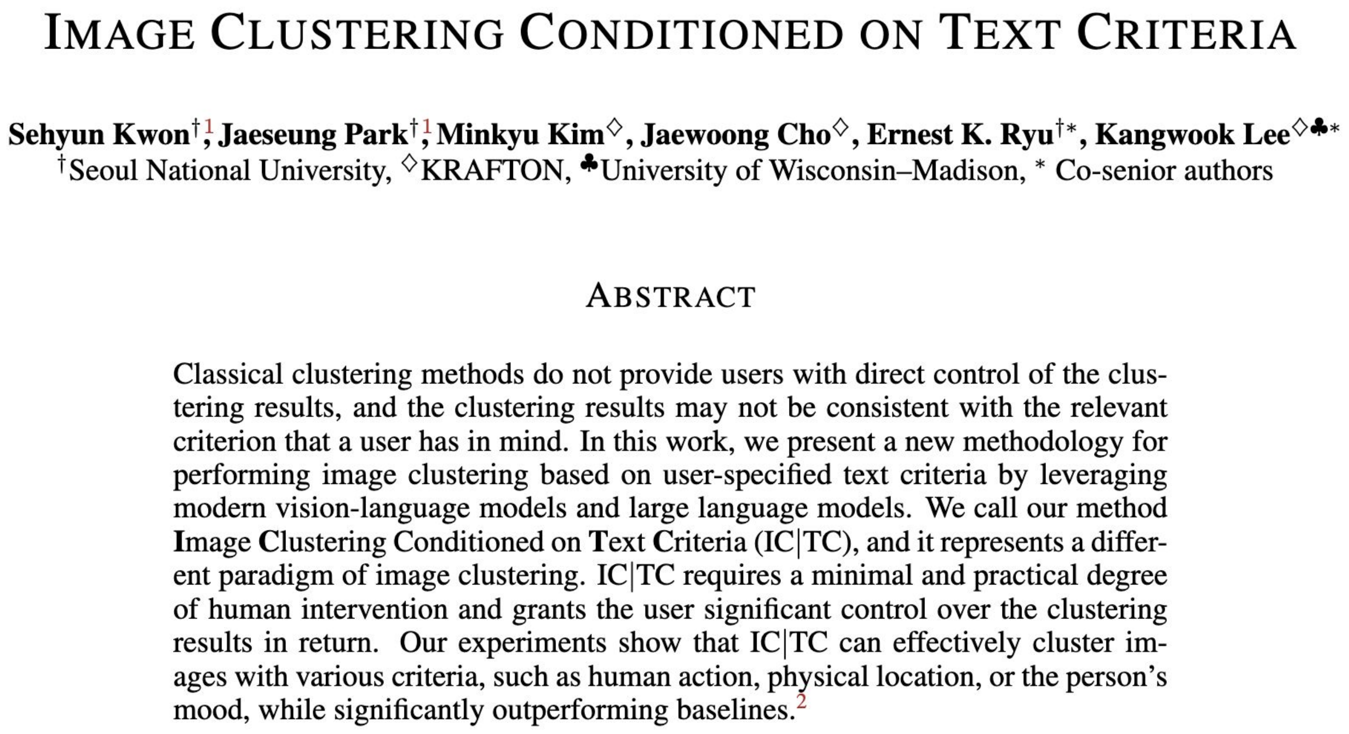


IMAGE CLUSTERING CONDITIONED ON TEXT CRITERIA

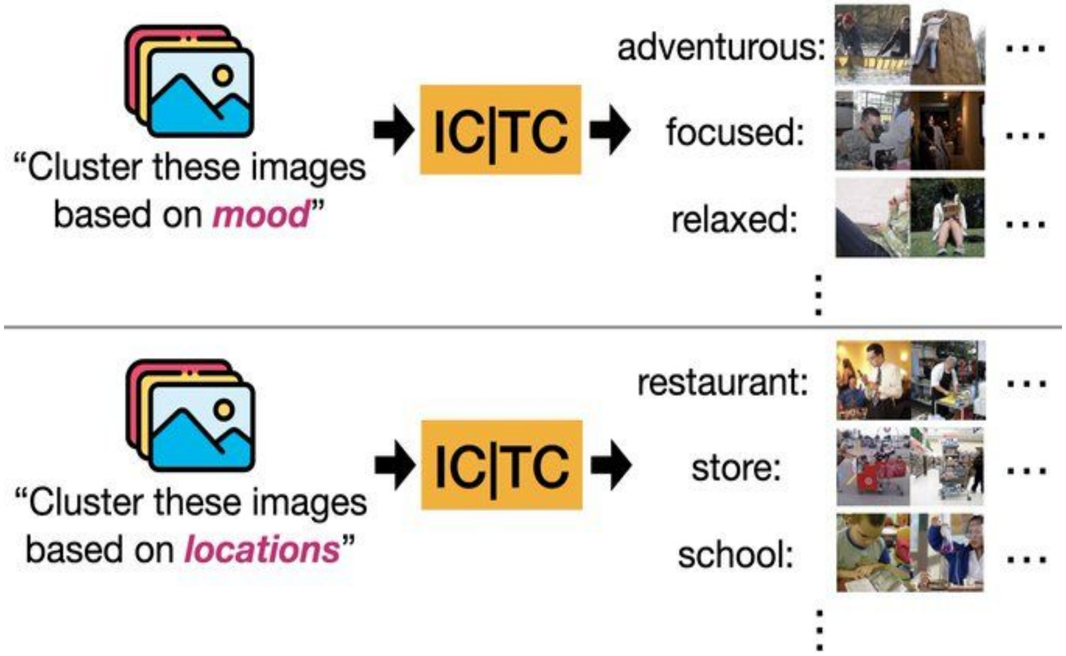
Sehyun Kwon[†],[1] **Jaeseung Park**[†],[1] **Minkyu Kim**[◇], **Jaewoong Cho**[◇], **Ernest K. Ryu**[†*], **Kangwook Lee**[◇♣*]
[†]Seoul National University, [◇]KRAFTON, [♣]University of Wisconsin–Madison, [*] Co-senior authors

ABSTRACT

Classical clustering methods do not provide users with direct control of the clustering results, and the clustering results may not be consistent with the relevant criterion that a user has in mind. In this work, we present a new methodology for performing image clustering based on user-specified text criteria by leveraging modern vision-language models and large language models. We call our method **I**mage **C**lustering Conditioned on **T**ext **C**riteria (IC|TC), and it represents a different paradigm of image clustering. IC|TC requires a minimal and practical degree of human intervention and grants the user significant control over the clustering results in return. Our experiments show that IC|TC can effectively cluster images with various criteria, such as human action, physical location, or the person's mood, while significantly outperforming baselines.[2]

What is the problem?

- They are doing image clustering
- Not just any kind of clustering
  - Clustering based on user query
  - Query is **word** based
- Use cases:
  - You can cluster **the same images in many different ways**
  - By mood, location, event


"Cluster these images based on *mood*"
IC|TC
adventurous:
focused:
relaxed:
"Cluster these images based on *locations*"
IC|TC
restaurant:
store:
school:

How does it work?

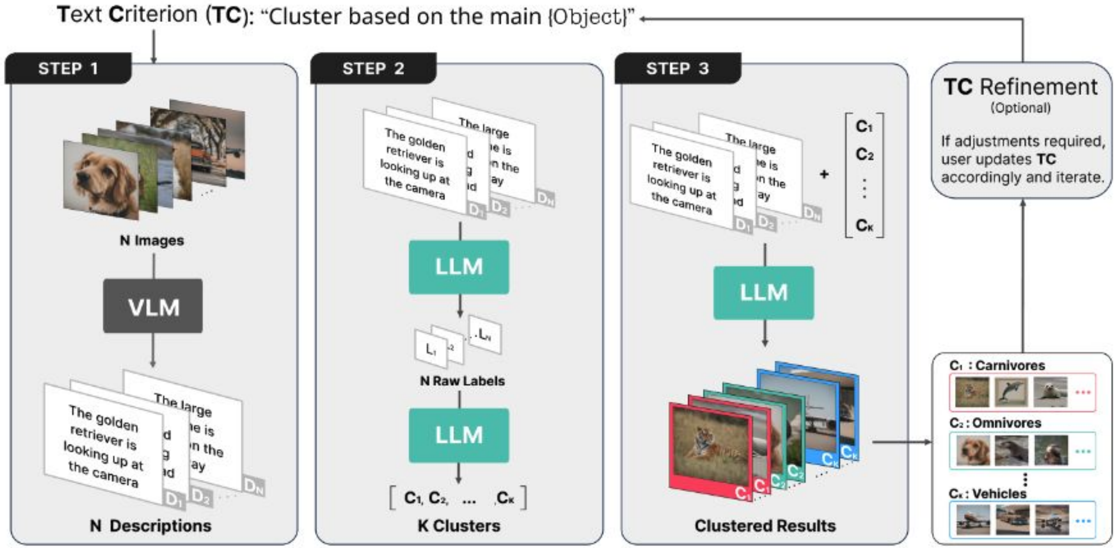


Figure 2: The IC|TC method. (Step 1) Vision-language model (VLM) extracts detailed relevant textual descriptions of images. (Step 2) Large language model (LLM) identifies the names of the clusters. (Step 3) LLM conducts clustering by assigning each description to the appropriate cluster. The entire procedure is guided by a user-specified text criterion (TC). (Optional TC Refinement). The user can update the text criterion if the clustering results are unsatisfactory. See Appendix B.4 for an unabridged sample output.

But there were some problems…

Cons about the paper… 🧐

- Computationally **VERY expensive**
- Need to run every step for every query
- For **every query**:
    - Caption all images in the database using VLM
    - Cluster those captions using LLM
    - Put each image to corresponding cluster using LLMs

The goal of my project is…

- To make this process more efficient!
- Why I am interested in this?
    - This is a new problem in clustering
        - That paper is the only paper
    - This is an open question, whose answer I don’t know
        - It’s not about making 0.2% improvement

How to do this?

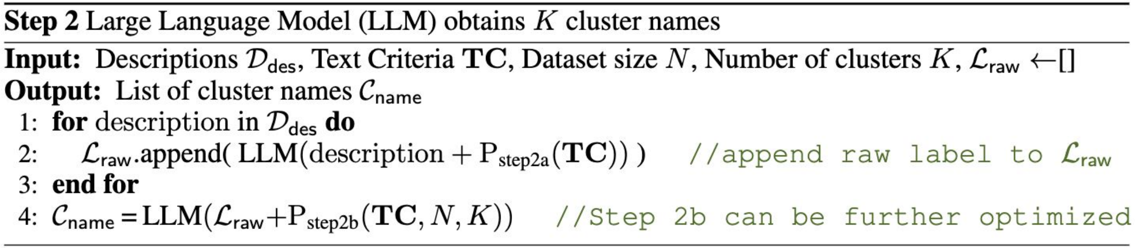
Notice the three steps:

- For **every query**:
  - Caption all images in the database using VLM
  - Cluster those captions using LLM
  - Put each image to corresponding cluster using LLMs

Can we replace these LLM calls?

How to do this?

- Step 2 & 3: Cluster the captions using LLM
- My idea:
  - Can we use CLIP embeddings?
  - Then do unsupervised clustering on embedding space?
    - Hierarchical clustering
    - DBSCAN
    - K-means



Step 2 Large Language Model (LLM) obtains K cluster names

```
Input: Descriptions $\mathcal{D}_{\text{des}}$, Text Criteria $\mathbf{TC}$, Dataset size $N$, Number of clusters $K$, $\mathcal{L}_{\text{raw}} \leftarrow []$
Output: List of cluster names $\mathcal{C}_{\text{name}}$
for description in $\mathcal{D}_{\text{des}}$ do
    $\mathcal{L}_{\text{raw}}$.append( LLM(description + $\text{P}_{\text{step2a}}(\mathbf{TC})$) )    //append raw label to $\mathcal{L}_{\text{raw}}$
end for
$\mathcal{C}_{\text{name}} = \text{LLM}(\mathcal{L}_{\text{raw}} + \text{P}_{\text{step2b}}(\mathbf{TC}, N, K))$    //Step 2b can be further optimized
```

If previous idea doesn’t work

- Can we go with traditional text clustering?
    - Topic modeling
    - Or purely text embedding clustering?
        - Slightly different from CLIP embedding
    - Try other modes of text clustering from NLP

Project Plan

- The paper code is available at: https://github.com/sehyunkwon/ICTC
- Joint embedding can be found using CLIP models
- Text embedding can be found using OpenAI models
- Clustering implementations are from sklearn

Full replication is not possible because of API costs

I will mostly work with CIFAR and Stanford 40-Action dataset

Timeline

- **Week 11:** Replicating the original paper on CIFAR dataset
- **Week 12:** Implementing the text clustering algorithm for CIFAR dataset
- **Week 13:** If the results are good, extend to other datasets (Stanford 40 actions
- **Week 14:** Write report

Task distribution: Single person project

If you have any questions, please leave in the comment box. I will reply!

Thanks for listening 🤗