Frame Comparison and Frame Clustering with Vision Transformer and K-Means on COVID-19 News Videos from Different Affinity Groups

Zhepeng Luo
Department of Computer Science
Columbia University
zl3092@columbia.edu

Directed Research Report - Fall 2022
Advisor: Prof. John R. Kender

01/02/2023

Abstract

Key frame extraction and comparison are essential in video analysis. In this paper, we examine a new method on frame similarity comparison based on context and apply clustering method to extract key frames in news videos from two different affinity groups(English and Chinese) in the same context(COVID-19). With vision transformer gaining more popularity in recent computer vision research, we developed a system to automatically extract key frames and analyzed the capability of vision transformers(ViT)[1] on feature extraction and video comparison.

1 Introduction

In project 'TAGGING AND BROWSING VIDEOS ACCORDING TO THE PREFERENCES OF DIFFERING AFFINITY GROUPS'[1], earlier work has been done by other students in effort to automatically compare frame similarity and extract key frames. Xu Han[2] has developed an unsupervised method to extract feature vector by applying VGG-19 and hashing tools followed by L1,L2 and cosine similarity comparison. Based on Han's work, Omer Onder[3] developed an unsupervised approach to learn the representation of news videos, replacing general purpose VGG-19 with task specific Variational Autoencoder(VAE), followed by clustering algorithm to extract key frames. This

[1]"Tagging and Browsing Videos According to the Preferences of Differing Affinity Groups" is a NSF IIS-Sponsored Project

approach has demonstrated promising result in some scenarios, encouraging further research on improving embedding and clustering algorithm.
Inspired by the excellent work from previous students, this paper proposed a new approach for video similarity comparison with ViT as video frame encoder.

2 Methodology

2.1 Data

News videos on the same topic (COVID-19) are collected to analyze the difference between two different affinity groups: English news videos and Chinese news videos. All videos are sourced from official channels of different news medias on Youtube, including CCTV, CBS, SETN and NBC. News reports covers a wide range of topics including vaccines, lock downs and public health. There is at least one video from each affinity group on every topic.
All videos are sampled into frames at frequency of 0.5 Hz, which would results in some information loss but the probability of completely missing one key frame is relatively low, since the average shot length in television news videos is above 3 seconds[4]. Then the frames are scaled to 384×384 to match the input size of the transformer. All videos are used to evaluate the performance of ViT encoding and K-mean clustering.

2.2 Vision Transformer

ViT (See Figure 1) is applied to extract the high dimensional feature vector from the sampled frames. We acquired the Pytorch implementation of the pre-trained ViT[5] on image classification. The models are pretrained on ImageNet21k[6], some models are fine tuned on ImageNet1k. The API also offers two different floating point precision and two different model scales. We examined the representation of different models by checking the clustering result as well as the similarity comparison result. Additionally, we tested the performance of the models with various reduced feature vector sizes.

2.3 Frame Similarity Analysis

The cosine similarity on the feature vectors is calculated to detect frames of high similarity within one video. The result is visualized in a cosine similarity matrix. We have also derived the similarity matrix between two videos.

2.4 Frame Clustering

K-Means[7]–[9] clustering models were trained for each videos with number of clusters sweeping from 2 to 20. We tested both euclidean distance functions and cosine similarity distance function. Since we were using the classification layer of the model, the output feature vector is already normalized to values between

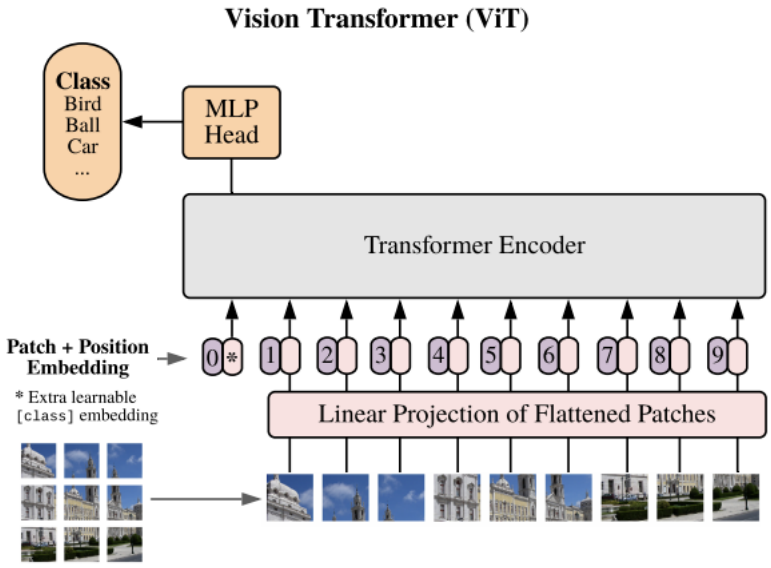


Figure 1: ViT Architecture.

0 and 1. The cosine distance function is not implemented in scikit-learn[9], but we could use euclidean distance function to calculate cosine distance by normalizing the feature vector by its euclidean norm.
Elbow method[10] was originally used to determine the optimal cluster number in Omer's work, but the optimal solution provided by this method is ambiguous when the curve is relatively smooth. Therefore, In addition to elbow method, we calculated the average silhouette score[11] of each number of clusters, which also measures the quality of a clustering, and removes the ambiguity by taking quantitative measurements.

3 Result

3.1 Vision Transformer

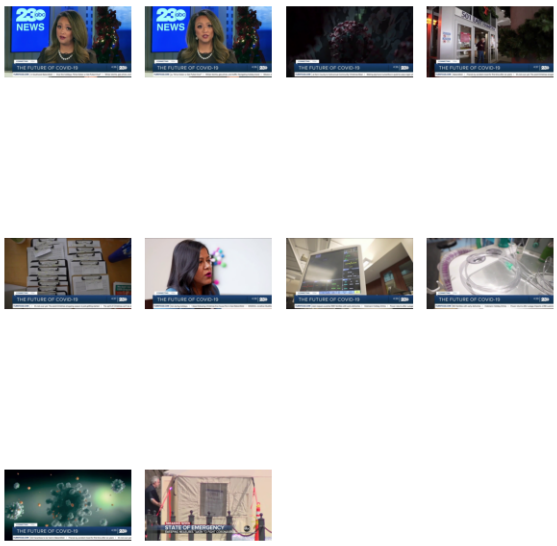


Figure 2: Clustering on 2 videos

The ViT yield the best performance when the classification layer was not modified. As we further reduced the feature vectors, the feature vector became more ambiguous and introduced more errors into the clustering algorithm. This resulted in cluster demonstrated in Figure 2 which contains many irrelevant frames.

3.2 Frame Similarity Analysis

The similarity comparison yielded both positive and negative results.

positive As shown in Figure 3, the frames with the highest similarity were frames from a continuous shot. These near-duplicate frames consist of most of the high similarity pairs. The frames with the lowest similarity (Figure 3c) shares little contextual similarity. The similarity matrix (Figure 3a) indicates

(a) Cosine Similarity (b) High similarity (c) Low similarity

Figure 3: High Similarity Group

Figure 4: High contextual Similarity

that some frames are similar across the whole video, these frames are most likely the intermediate transition between major events or the shots with reporters (Figure 3b). Additionally, some frames with high contextual similarity are correctly grouped together (Figure 4), which indicated the capability for ViT to output semantically related representations of video frames.

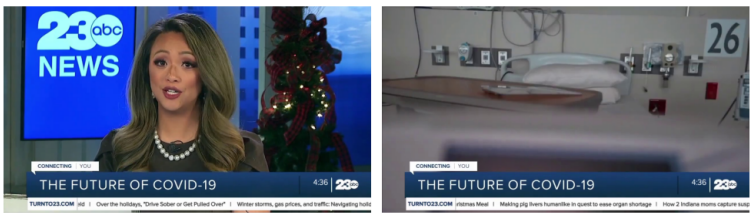
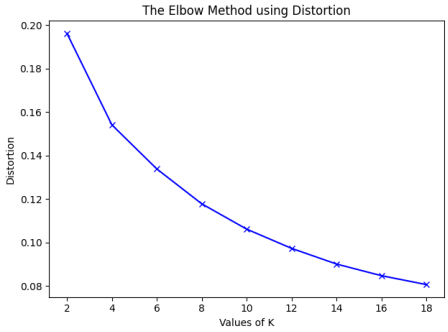


Figure 5: Low Contextual Similarity

negative Some high similarity groups shared little contextual similarity (Figure 5). Some visual artifacts, including large icons, text blocks with background color and transition effects, would dominate the feature vector expression so that frames containing these features are grouped together even if they are poorly correlated.

3.3 Clustering



(a) Distortion

(b) Silhouette Score

Figure 6: Chinese Lockdown Video Report

Single video The elbow method on Chinese lockdown video report (Figure 6a) had a relatively smooth curve, and it started to converge at around 10 - 14 clusters. On the other hand, the silhouette score clearly indicated that the optimal score is achieved with 10 clusters, if the score of 2 clusters is ignored. The high silhouette score of 2 clusters might indicated that the representation is not well learnt and some clusters does not have a clear boundary between them.

Two videos from different affinity group When videos from two affinity groups are combined and clustered together, a clear boundary could be found between the two videos (See Appendix A). In the very few overlapping clusters between the videos, we found some meaningful overlap between the two videos, for example, they both shared great amount of talking heads of interviewee or reporter wearing suit of similar color tone. There were also some errors inside

these clusters, which suggests that the learnt representation sometimes focuses more on visual effects introduced by editing and neglects the common content that shares between two videos.

4 Conclusion

In short, we explored the application of ViT on distinguishing news videos from two different affinity groups. We improved the automated process of comparing frames between two sets of videos. By examining the clusters and representation of the feature vectors, we concluded that the learnt feature vector could successfully distinguish videos from two affinity groups, but the representations are also greatly affected by visual effects.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy.

[2] X. Han, "Identifying near-duplicate frames from two sets of videos," *Tagging and Browsing Videos According to the Preferences of Differing Affinity Groups*, 2019.

[3] Omer, "Frame similarity detection and frame clustering using variational autoencoders and k-means on news videos from different affinity groups," *Tagging and Browsing Videos According to the Preferences of Differing Affinity Groups*, 2021.

[4] L. Kalev, "Using google's video ai to estimate the average shot length in television news," *Forbes*, Jun. 2019. [Online]. Available: https://www.forbes.com/sites/kalevleetaru/2019/06/03/using-googles-video-ai-to-estimate-the-average-shot-length-in-television-news/?sh=54c085ad8e3a/.

[5] L. Melas-Kyriazi, *Pytorch pretrained vit*, 2020. [Online]. Available: https://github.com/lukemelas/PyTorch-Pretrained-ViT.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

[7] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982. DOI: 10.1109/TIT.1982.1056489.

[8] J. MacQueen, *Some methods for classification and analysis of multivariate observations*. California, 1967, vol. 1, pp. 281–297.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[10] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, 1953.

[11] P. J.Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, 1987.

A Appendix

```
Cluster0
From Video 1
[ 42  43  44  46  47  48  49  50  97  98  99 101 102 103 104 105 106 107
 108 148 149 150 151 190 191 192 212 213 214 215]
From Video 2
[]
Cluster1
From Video 1
[]
From Video 2
[  3   5  13  14  28  31  42  43  44  46  47  61  63  64  82  84  85  86
  90  96  97  98  99 101 102 105 106 107 108 120 121 124 127 128 129 130
 141 148 149 168 169 180 181 195 198 199 208 214 215 216 217 218 219 220
 221 224 230 231 232 236 238 255 262 263 264 311 319 324 325 330 375]
Cluster2
From Video 1
[]
From Video 2
[100 112 123 134 145 156 167 178 189 267 277 279 280 286 287 288 290 300]
Cluster3
From Video 1
[  2   3   4   5   6   7   8  51  52  53  54  55  57  58  59  60  61  62
  63  64  65  66  68  69  70  71  72  73  74  75  76  77  79  80  81  82
  83  84  85 152 153 154 156 157 158 159 160 161 162 163 164 165 167 168
 169 170 171 193 194 195 196 197 198 200 201 202 203 204 206 207 208 209
 211 216 217 218 219 220]
From Video 2
[]
Cluster4
From Video 1
[]
From Video 2
[  4  35  36  37  38  39  40  41 131 132 133 135 136 137 138 155 157 158
 159 160 161 162 172 173 174 175 176 177 179 187 188 190 191 205 206 207
 244 246 336 337 338 339 340 341 368 369 370 371 372]
Cluster5
From Video 1
[]
From Video 2
[ 23 303 304 305 306 307 308 309]
Cluster6
From Video 1
[205]
From Video 2
```

```
[ 26  45 126 192 193 197 201 210 212 213 260 328]
Cluster7
From Video 1
[  1  12  23  34  45  56 188 199 210]
From Video 2
[ 15  60  83  87  88  91  92  93  94  95 122 144 146 165 203 204 209 226
 228 229 233 235 237 241 243 247 248 249 265 266 315 317 318 320 322 326
 327 332 342 343 344 349 373 374 376 377]
Cluster8
From Video 1
[]
From Video 2
[163 200 211 223 225 234 245 254 256 278 281 282 283 284 285 289 291 292
 293 294 295 296 297 298 299 301 302]
Cluster9
From Video 1
[]
From Video 2
[  7  76  79 116 269 270 271 272]
Cluster10
From Video 1
[]
From Video 2
[ 30  65 103 104 109 113 114 115 142 143 154 202 257 329 333 335]
Cluster11
From Video 1
[127 128 129 130 131 132 133 135 136]
From Video 2
[78 89]
Cluster12
From Video 1
[]
From Video 2
[  6  48  49  50  51  52  53  54  55  57  58  66 110 147 151 152 153 194
 196 347 350 351 352 353 354 355]
Cluster13
From Video 1
[]
From Video 2
[ 16  17  18  19  20  21  22  24  25  56 261 321]
Cluster14
From Video 1
[  9  10  11  13  14  15  16  17  18  19  20  21  22  24  25  26  27  28
  29  30  31  32  33  35  36  37  38  39  40  41  67  78  86  87  88  89
  90  91  92  93  94  95  96 100 109 110 112 113 114 115 116 117 118 119
 120 121 122 123 124 125 134 137 138 139 140 141 142 143 145 146 147 172
```

```
 173 174 175 176 178 179 180 181 182 183 184 185 186 187 189]
From Video 2
[]
Cluster15
From Video 1
[126]
From Video 2
[  2   8   9  10  11  27  29  33  59  77  80 117 118 119 139 140 164 227
 239 240 242 258 259 268 273 274 275 276 316 348]
Cluster16
From Video 1
[]
From Video 2
[ 32  62  68  69  70  71  72  73  74  75  81 250 251 252 253 357 358 359
 360 361 362 363 364 365 366]
Cluster17
From Video 1
[]
From Video 2
[125 150 166 170 171 182 183 184 185 186 310 313 314 331 346]
Cluster18
From Video 1
[  0 111 144 155 166 177]
From Video 2
[34]
Cluster19
From Video 1
[]
From Video 2
[  0   1  12  67 111 222 312 323 334 345 356 367]
```

Figure 7: Cluster 6

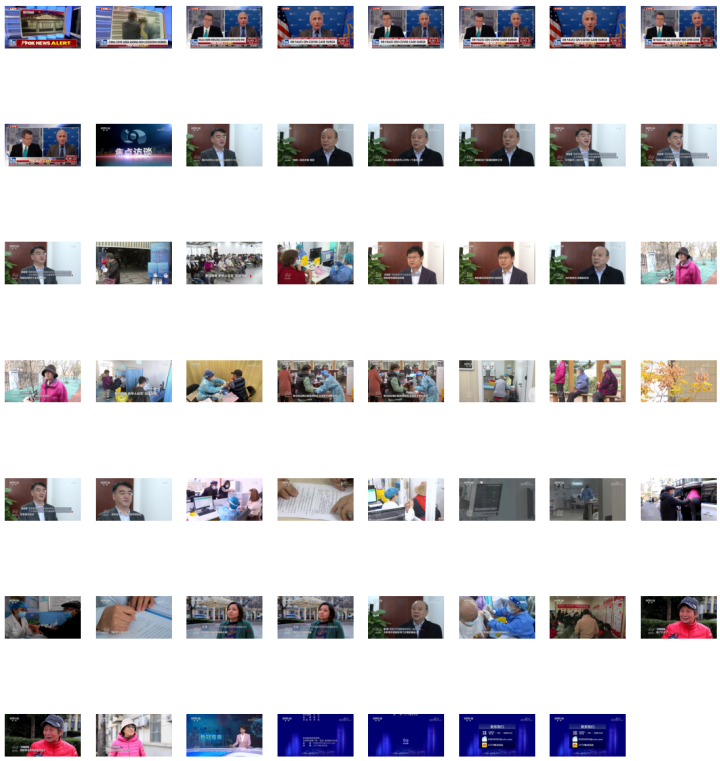


Figure 8: Cluster 7

Figure 9: Cluster 11

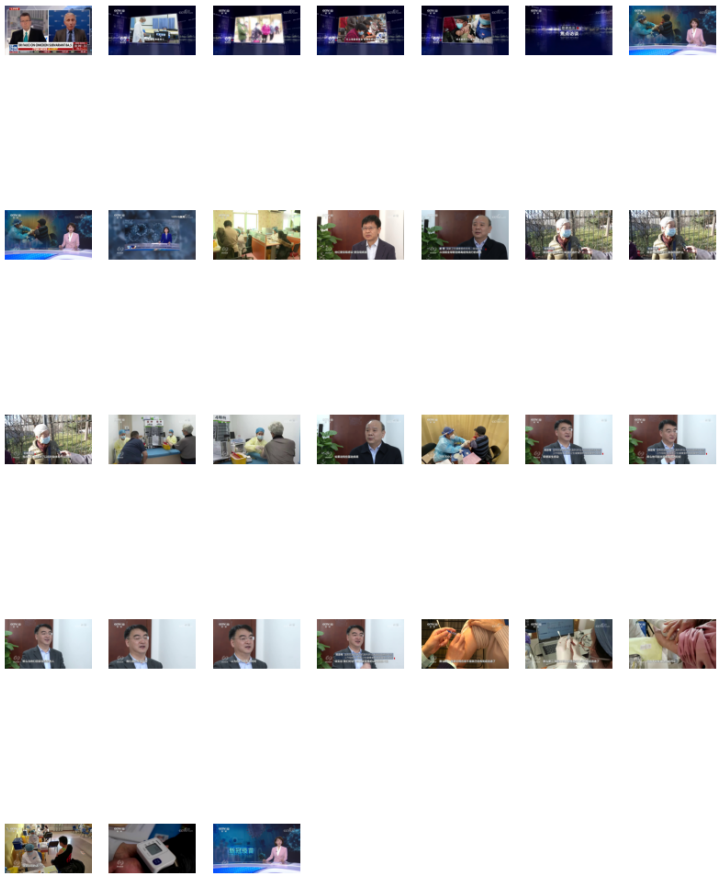


Figure 10: Cluster 15