AN IMPROVED AUTOENCODER STRUCTURE FOR IMAGE DIMENSION REDUCTION AND CLUSTERING

Tiancheng (Robert) Shi
Data Science Institute
Columbia University
ts3474@columbia.edu

Supervisor: John R. Kender
Department of Computer Science
Columbia University

January 1, 2023

ABSTRACT

Video frame clustering methods have introduced a new approach to video content analysis, besides traditional, sequential processing. Previous research has shown that using fully Convolutional Variational Autoencoders (CVAE) to perform dimension reduction on images ensures the efficiency of clustering. In this work, we propose another CVAE structure with Dense Neural Network layers and show its improvement from both theoretical and practical aspects. Further, multiple clustering methods and analyses are performed on the video frames based on the new dimension reduction process to provide real-world insights.

1 Introduction

The Big Data era has witnessed the explosive development of online multimedia as a source of information. For example, various news videos on YouTube can oftentimes provide an excessive amount of information for one to absorb. Thus, it is a general trend that briefly understanding the main opinion of a video – sometimes without watching every second of it – is gradually gaining significance. For this project, we are primarily motivated to use modern Computer Vision (CV) techniques, especially Neural Networks, to help the audience quickly capture the keyframes of a video, thus having an overview of its topic.

As part of the NFS Information and Intelligent Systems sponsored "Tagging and Browsing Videos According to the Preferences of Differing Affinity Groups" Project, the initial aim of this work is to perform clustering techniques on frames (images) sampled from videos of interest. Yet it comes to the researchers that video frames with high resolutions – typically encoded as $Height \times Weight \times Channel$ (i.e., RGB's) – are associated with an extremely high dimension that makes it impossible to determine distances (dissimilarities) during clustering. The issue of dimension explosion lays emphasis on reducing the dimension low enough for popular Machine Learning algorithms to handle, at the cost of losing redundant or unnecessary information hidden in images.

Upon studying multiple dimension reduction models for images, we decide to apply an Autoencoder-based, unsupervised Deep Learning model. Previous work [Onder, 2021] exists on using Variational Autoencoders with only Convolution layers to reduce the dimension to 128. It is worth admitting that CVAE is, by every means, the state-of-the-art Neural Network model for dimension reduction on large image datasets with high resolutions, but the use of the Global Max Pooling layer on top of the bottleneck layer seemingly undermines its result. In such consideration, we propose to substitute the potentially insensible Global Max Pooling structure with multiple Dense Neural Network layers in the encoder-decoder structure.

Experiments are then designed to compare the performance of pure CVAEs and dense CVAEs – i.e., CVAEs with Dense layers. The experiments are performed on image datasets generated from news videos. These videos are chosen on the topic of COVID-19, which involves multiple news sources from various cultural backgrounds, including China and the United States.

After dimension reduction, clustering methods based on pre-neural-network era Machine Learning techniques are applied, including K-means and Gaussian Mixture Models (GMMs), to find the similarity and connection between data points. Keyframes can thus be reconstructed from cluster centers (for centroid-based or mixture-based methods) using the decoder network. Besides extracting keyframes from a single-source dataset, the method is also capable of finding the similarities and differences between the frames from multi-source datasets. This outcome provides insights into how certain features (e.g., style, format, affinities, etc.) of news videos differ with respect to cultures.

This project's source code is available on GitHub `https://github.com/Anemonee1212/cvae_video_cluster` upon the submission of this report.

2 Related Works

2.1 Autoencoder

With the rapid increase of computation power, the emergence of modern Artificial Intelligence also brought about the renaissance of Autoencoders [Rumelhart et al., 1987]. By taking advantage of the Gradient Descent optimization algorithm, it has long been the state-of-the-art unsupervised representation learning model for large-scale dimension reduction tasks.

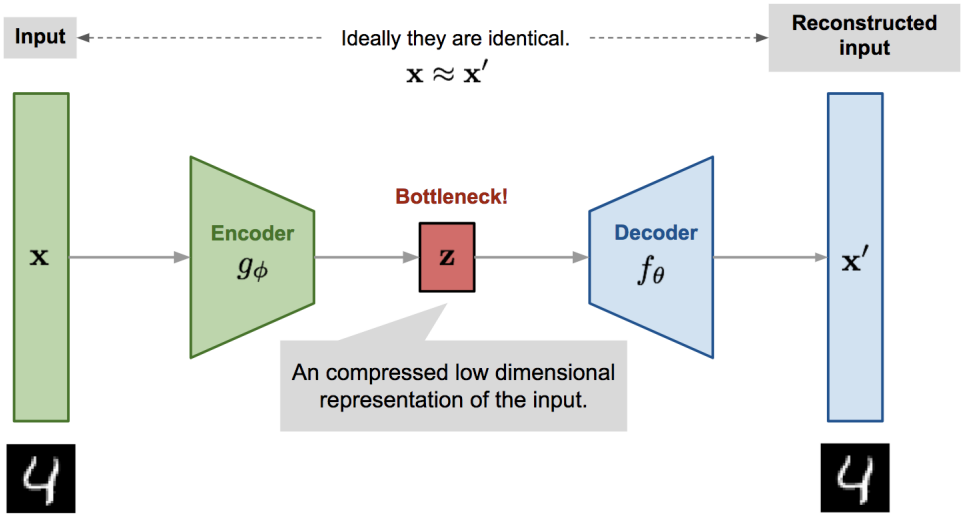

Figure 1: General Structure of Autoencoders

The image[1] above illustrates the encoder-decoder structure of a generic Autoencoder. The Encoder network maps the input space into a lower dimension subspace, defined as latent space, while the Decoder network reconstructs the data points in latent space back to their original dimension. The entire model is trained to minimize the dissimilarity between input data and reconstructed output data, so in ideal cases, a well-trained Autoencoder can achieve lossless data compression – all information of the input data can be fully recovered if "unzipped" properly. In this way, the data in the latent layer (or colloquially, the bottleneck) is successfully "learnt" from the training data without explicit supervision labels needed.

Note In practice, the Encoder network typically consists of several Dense Neural Network layers with a monotonously decreasing number of neurons in each layer, while the Decoder network is oftentimes symmetric to the Encoder. To avoid confusion of information within deep, fully connected neural networks, which leads to difficulty in restoring data, the number of hidden layers in each network of a vanilla Autoencoder is typically limited to 2 (excluding the input and latent layers).

2.2 Variational Autoencoder

Admittedly, from its very beginning, Autoencoders marked a cutting-edge milestone for unsupervised representation learning, especially for dimension reduction tasks; meanwhile, the desired, almost lossless dimension reduction

[1] Source: `https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`

approach comes with a cost – the latent space oftentimes experiences "overfitting-like" behavior due to the loss of structural information.

Intuitively, since the loss function of vanilla Autoencoders only involves optimizing the encoded data points in latent space, the regularization of how data points are represented (or in a statistical term, distributed) in the latent space is not taken into consideration. That says, the resulting latent space may be so unorganized that, even though a certain data instance can be reconstructed into a meaningful output similar to its input data, its neighboring point (not directly mapped from the training set) is very likely to be decoded into random noises. Such patterns are quite analogous to overfitting in supervised learning in the way that the model performance is guaranteed only on the training set, regardless of data instances outside the dataset seen.

However, it is worth emphasizing that in our context, such "overfitting" patterns cannot be ignored or compromised. All mainstream clustering algorithms which can potentially be applied to the compressed image data require some sort of distance metrics, if not solely depends on them. For this reason, we demand the continuity and completeness of the latent space, and that it admits distance calculation as a representation of dissimilarity between data points – the closer two images are embedded in the latent space, the more visually similar they should be. In such consideration, we adopt the Variational Autoencoder (VAE) model [Kingma and Welling, 2013], one of the direct descendants of traditional Autoencoders, to better address this issue.

The primary improvement of VAE over plain AE is that, instead of using the encoded data instances to reconstruct the output data, we perform a random sampling of a certain distribution (say, Gaussian distribution) in the latent space pre-defined around the encoded data point, and pass this random sample into the Decoder network. Or in mathematical terms,

$$\underset{\theta,\phi}{\operatorname{argmin}} \|x - x'\|_2, \text{where } x \text{ is the input data, and } x' \text{ is defined by}$$

$$\begin{cases} z = E_\theta(x), x' = D_\phi(z), & \text{for AEs} \\ z = E_\theta(x), z' \sim p(z|x), x' = D_\phi(z'), & \text{for VAEs} \end{cases}$$

Even though by intentionally introducing randomness, we sacrifice some accuracy in reconstructing the original data, this "variation" approach significantly improved the robustness and interpretability of Autoencoders by, at least in some aspects, supporting the regularization of the latent space.

Note In practice, Normal (Gaussian) distribution are typically used for resampling, with both parameters (mean μ and standard deviation σ) to be learnt.

2.3 Convolutional Variational Autoencoder

Within the scope of the "Tagging and Browsing Videos According to the Preferences of Differing Affinity Groups" Project, Onder [2021] and other researchers have made concrete progress toward the dimension reduction (as well as the following clustering) algorithms of processing video frames with various cultural orientations. Specifically, the use of Convolutional Variational Autoencoders (CVAEs) has been proposed and deployed to a historical image dataset (sampled from multiple news videos about the competition in the ancient Chinese game "Go" between Ke Jie, the world champion, and AlphaGo, an AI by Google DeepMind).

The idea of using fully Convolution layers in VAE is quite impressive, under the common consensus that for an object shown in an image, once its approximate position is given, its precise, pixel-wise position is less of interest. Yet it is also worth emphasizing that for images with high resolutions, simply using 1-strided Convolution layers, followed by Max Pooling layers with 2×2 pool size, is insufficient to reduce the extremely high dimensions (sometimes even in millions) to an approachable level, while using larger strides and/or pool sizes will result in fuzziness during reconstruction using Transposed Convolution layers (so-called de-convolution layer).

The original method to resolve this dilemma was to use a Global Max Pooling layer to the latent layer, probably inspired by the way in which Convolutional Neural Networks (CNNs) without fully connected layers handle traditional image classification tasks. In the meantime, multiple external researchers (including Pu et al. [2016]., Fan et al. [2020], etc.) have shown the solid theoretical foundation, as well as the potential practical development, of using Dense Neural Network layers between the Convolution layers and the latent layer.

Note At the point when this project started, the original AlphaGo news dataset and the code for CVAE training and further clustering and analyses are no longer available. Though tremendous efforts have been made to replicate the original structure, framework, training metrics, and hyperparameters, it is still worth mentioning that minor differences

in model performance may exist. The results in this report should be viewed only as a reflection of Onder's work, and be used for cross-comparison.

3 Dataset

To deploy and evaluate our model on a topic with popular and impactful insights, we choose to focus on the news videos related to COVID-19. This topic is selected due to its ongoing, popular trend, as well as global attention and awareness, universal to all cultures. We believe that compared with the outdated AlphaGo-beats-human debate, the worldwide impact of COVID-19 indeed arouses more public interest of the time. Specifically, the following two news videos come into our sight.

3.1 Chinese Vaccine Video

This video[2] is published by China Central Television (CCTV) on YouTube. The primary purpose of this video was to encourage (or demand) senior Chinese citizens to take vaccines for COVID-19, while also reporting the current situation of the pandemic and the progress of disease control.

The images from the video are sampled at the rate of 1 frame every 2 seconds, i.e., 30 frames per minute. The video length is 15 minutes and 8 seconds in total, which generates 378 image frames, excluding irrelevant scenes at the beginning and end. Some manually labeled typical frames include scenes of Chinese citizens queuing to take vaccine shots, as well as medical experts or government officials talking to the camera.

Figure 2: Chinese officials talking about COVID-19

Figure 3: Chinese elderly taking a vaccine

3.2 US New Variant Video

This video[3] is published by CBS Mornings on YouTube. The main content of this video was about the resurgence of COVID-19, BA.5 (Omicron) variant, and its spread across the US. Dr. Agus also discussed the corresponding reactions and restriction policies in large cities like New York and Los Angeles.

The image dataset is created in the same manner of 1 frame every 2 seconds. This video is relatively shorter than the previous Chinese Vaccine Video, with only 3 minutes and 40 seconds, so it generates 109 image frames in total. Typical frames involve the host and/or the medical expert talking to the camera, and a presentation slide.

[2] https://www.youtube.com/watch?v=xcWeBCOMoiU

[3] https://www.youtube.com/watch?v=doP5UacBlt0

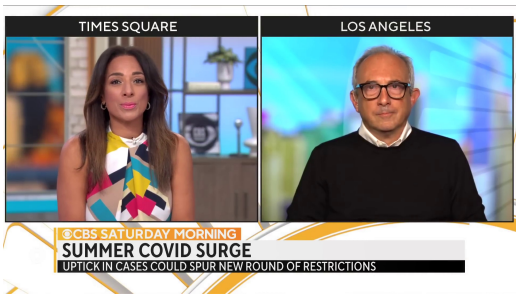

Figure 4: US host and official talking

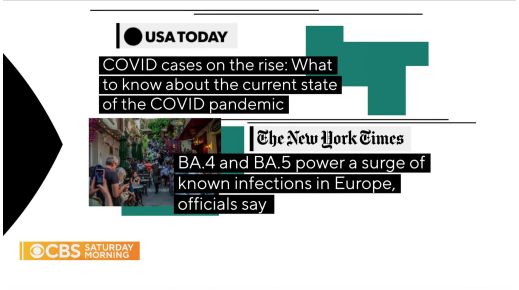

Figure 5: US presentation slide

4 Methods

4.1 Pure CVAE (without Fully Connected Layers)

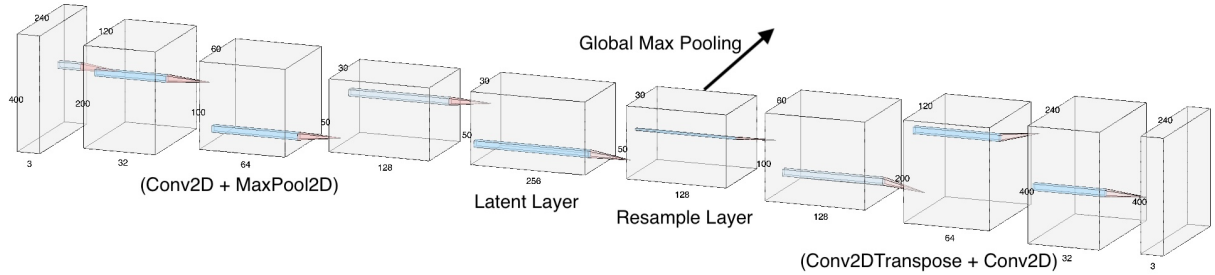

Figure 6: Pure CVAE Architecture

The architecture of pure Convolutional Variational Autoencoder is intentionally designed to emulate the model in the work of Onder [2021].

The Encoder network has an input shape with a height of 240 pixels, a width of 400 pixels, and 3 channels representing RGB. All 3 Convolution layers have the same kernel size of 3×3 and a stride of 1, and they have 32, 64, and 128 filters (channels) respectively. ReLU activation is applied to Convolution layers. Each Convolution layer is followed by a Max Pooling layer of pool size 2×2. That says, the size of each filter reduces by $\frac{1}{2}$ in both height and width after passing through a Convolution-Max Pooling combination.

The latent dimension is set to be 128, so the number of channels in the latent layer is $128 \times 2 = 256$: 128 of which are trained to be the mean, while the 128 channels remaining are the variance, as per the parameters of Normal (Gaussian) distribution. After that, random samples are made accordingly, which forms the resample layer.

The Decoder network can be viewed as the opposite of the Encoder, where the Convolution layers are constructed in exactly the same manner. The Max Pooling layer used for downsampling, on the other hand, is reconstructed by

Transposed Convolution layers with the same shape, stride of 2, and kernel size of 3×3. Such 2-strided layers can upsample the channels and thus double the height and width. All layers in the Decoder are activated by ReLU.

The pure CVAE model is fitted on China vaccine data, US new variant data, as well as their mix (multi-source) data, in a batch size of 32. Pixel-wise Mean Squared Error (MSE) is set as the loss function. We use an Adaptive Momentum (Adam) optimizer with a learning rate of 1×10^{-4}. A 100-epoch training session on our local device takes approximately 10 minutes.

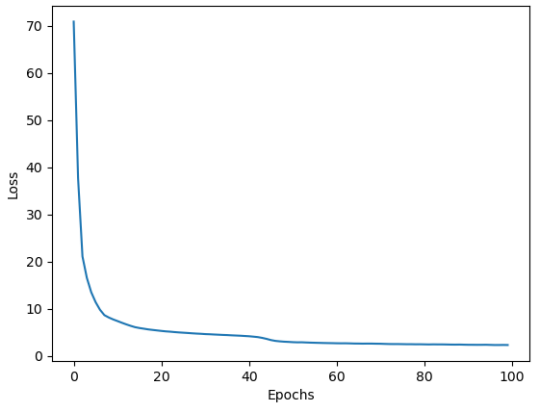

Figure 7: MSE of Pure CVAE on China vaccine dataset

From the plot above, we can clearly see that the model converges quickly within 100 epochs. The MSE loss at epoch 100 is 2.338, and sample reconstructed images are shown below. The output [8] of CVAE is impressive, as it almost successfully recovered all major parts or objects in the image, except for a few minor bright noises in a white or relatively lighter background. Similar procedures are performed on US new variant dataset and multi-source dataset, and results are also obtained. (See Appendix 21.)

4.2 Dense CVAE (with Fully Connected Layers)

After fitting the previous pure CVAE model on the training set, Onder [2021] stated that applying a Global Max Pooling layer on top of the 128-channel latent layer would help reduce the dimension of the images to 128. This hypothesis might sound acceptable at first glance, as a mimic of the Global Average Pooling layer (to substitute the Dense layers with Softmax activation) used in traditional CNN for classifications. However, in our dimension reduction task, this almost brute-force method is neither theoretically sensible nor practically reasonable.

- The obvious and most outstanding drawback of this method is that, by forcing a full, 2D feature map to compress to 1 single value, we are indeed losing all spatial information about objects, structures, etc. within this convoluted image, which is opposed to our initial goal of clustering images based on their relative similarities. In other words, keeping track of an extremely "bright spot" (the pixel that is most activated by the Convolution layers) in a feature map does not necessarily provide information on the content of the image according to human intuition.
- On the other hand, it should not be neglected that during the actual training process, the "latent vector" of length 128 after Global Max Pooling is, in fact, not involved. It is neither the direct output for which the Encoder network is optimizing, nor can it be used to reconstruct the output image – the Decoder network needs to take the whole feature map of $30 \times 50 \times 128$ into consideration, so merely 128 elements cannot necessarily represent the original image in the lower-dimension subspace.

For these reasons, we conclude that pure CVAE is insufficient in our dimension reduction task, because the step where the harshest compression of $\frac{1}{150}$ happens, i.e., the Global Max Pooling layer, does not belong to the encoder-decoder architecture through the Gradient Descent optimization algorithm. To address the problem of reducing the dimension into an approachable level without extra pooling, we proposed the dense Convolutional Variational Autoencoder model: a CVAE model with Dense Neural Network layers in the middle, i.e., the deep layers of the Encoder network and the shallow layers of the Decoder network.

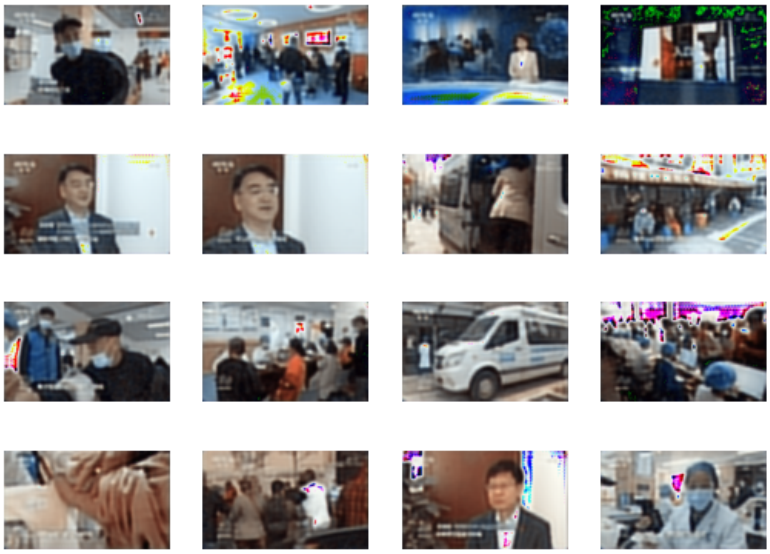

Figure 8: Sample output of Pure CVAE on China vaccine dataset

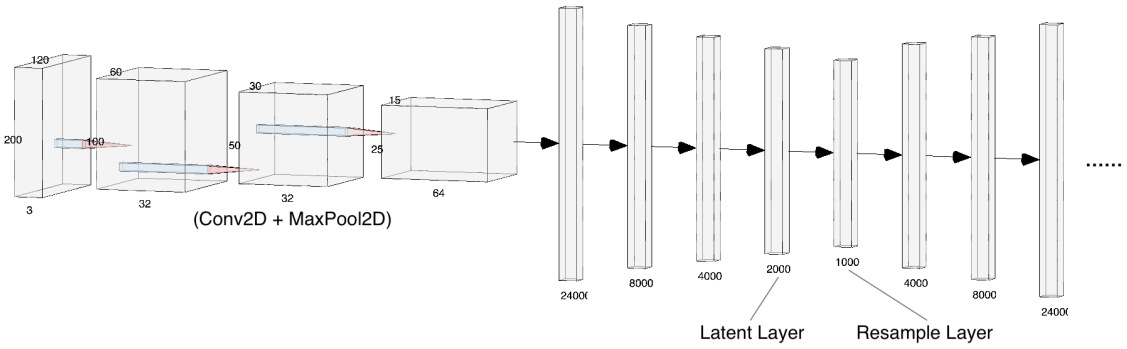

Figure 9: Dense CVAE Architecture

Note For simplicity, the duplicated Convolution layers at the end of the Decoder network are omitted.

Learning the prior experience of the pure CVAE model, we design the new structure to let the dimension reduce smoothly. First, the input shape is limited to $16 \times 120 \times 200 \times 3$, where 16 is the batch size. The framework of Convolution and Max Pooling layers remains the same, only that the numbers of filters (channels) are set to be 32, 32, and 64 respectively. That gives a vector of length $15 \times 25 \times 64 = 24000$ after flattening. Then in 2 Dense layers with decreasing number of neurons, the latent dimension is finally reduced to 1,000 (means and variances).

Similarly, the omitted Convolution layers in the Decoder network add back the feature map dimensions gradually. In the meantime, the number of channels reduces from 64, 32, 32, to 3, which represents RGB.

For reference, the dense CVAE model is also fitted on the same 3 datasets. Even though the input and output shape changed to $\frac{1}{4}$ of the pure CVAE model, the nature of pixel-wise Mean Squared Error (MSE) as a loss function still supports the cross-model comparison. The same hyperparameters are adopted, and the network starts to converge at

around 300 epochs. That says, a typical training session on China vaccine data requires almost 1 hour on local device[4], which includes both fitting the model and transforming the training images into reconstructed images.

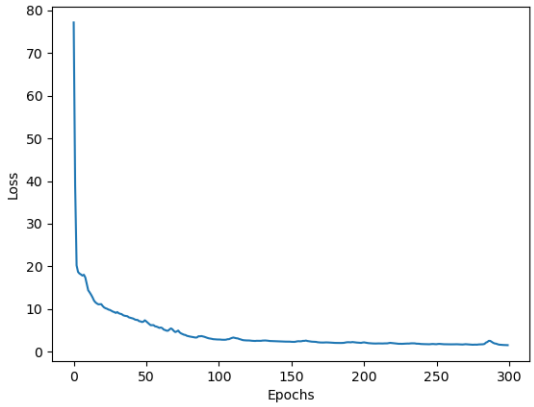

Figure 10: MSE of Dense CVAE on China vaccine dataset

Different from the pure CVAE, dense CVAE with multiple fully connected layers typically takes more epochs to converge. This is somewhat understandable, considering that complex matrix multiplications between flattened Dense layers indeed contribute to the majority of parameters in Neural Networks. Besides a tripled number of training epochs, the time complexity of each epoch also increases. Despite the increased cost in computation power, we still believe that it is a fair tradeoff, given the absolute advantage of how dense CVAE helps achieve our initial goal.

Figure 11: Sample output of Dense CVAE on China vaccine dataset

[4]The hardware that the researcher uses is NVIDIA GeForce RTX 3080 with 10 GB GPU Memory. If more GPU Mem is available, you may wish to try input images with higher resolutions and/or more complex neural networks.

The MSE loss at epoch 300 is 1.528. It is worth notifying that with this relatively lower MSE, dense CVAE significantly outperforms pure CVAE in most cases, but there still exists some extreme outliers, where dense CVAE generates totally wrong, highly contrastive color patterns, e.g., image 16 (lower right corner) of the sample reconstructed images [11]. This abnormal outcome is even more obvious in the US new variant dataset, where white or other light-colored backgrounds are more frequently observed in presentation slides. (See Appendix 27.) We suppose that when encountering white colors (i.e., R = 0, G = 0, B = 0), due to the global "average" nature of the MSE loss function, the model tends to tolerate some errors in one of the colors of Red, Green, and Blue, as long as other two remains almost 0. This phenomenon, if correct, also explains why the loss function curves of dense CVAE are relatively more fluctuating as the training session goes on.

Despite such flaws in outliers, we tend to conclude that the performance of dense CVAE is promising. We would especially like to further address that questions may be raised about the effectiveness of fully connected layers, specifically about whether the complex, neuron-to-neuron connection will introduce fuzziness to the relative spatial (positional) information of pixels, edges, patterns, or objects in an image. In our opinion, even though the mechanisms of black-box Neural Networks cannot be clearly understood, it is for us, humans, to believe that Neural Networks can learn to reveal (and then retain) the important patterns hidden within the images, just like regular CNNs, and that whether the positional information is fully retained, or encoded into some uninterpretable vectors, or even ignored during dimension reduction, is just a tradeoff that our model has to make through its training.

5 Results

5.1 Model Experiments

The performance of the two models on three datasets is listed below. We conclude that dense CVAE is superior to pure CVAE, and we decide to go on with this. (The number of epochs for which the two models are trained does not match, yet pure CVAE indeed starts to converge at around 70 epochs, so the remaining 200 epochs will not affect the performance.)

Table 1: Model Performance

		Dataset		
Model	Epochs	China	United States	Multi-Source
Pure CVAE	100	2.338	5.832	2.551
Dense CVAE	300	**1.528**	**2.646**	**1.496**

5.2 t-SNE Visualization

Even though the dense CVAE model encodes a high-resolution video frame into an array in its lower-dimension subspace, its length of 1,000 still prohibits researchers to understand the distribution of data point embeddings in the latent space. Considering this, we utilize the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm to further reduce its dimension to 2D. It needs to be reiterated that all clustering processes below are still performed on the original, 1000D vector space; the 2D space is only used for visualization and illustration.

The most influential hyperparameter of t-SNE is perplexity, which controls how extreme or how "sharp" the t-distribution is. Typically, the smaller perplexity is, the more isolated the resulting data points will be. In practice, considering the data size, we set the perplexity at 15 for the China dataset, 20 for the US dataset, and 10 for the Multi-source dataset. We notice several small but compact clusters [12] outlying away from the majority in two single-source datasets. We can also see that the data points of the two classes are indeed very mixed in the multi-source dataset.

5.3 K-means Clustering

Based on our previous visualization, the initial try is the most popular K-means clustering algorithm. We first traverse through a series of numbers of clusters k to determine the best set of hyperparameters. The metrics used include average inter-cluster distances, average cross-cluster distances, and cluster robustness tests. Specifically, an ideal k will have comparably small inter-cluster distances (we want data points to be close to their centroids), large cross-cluster distances (we want centroids to be far away from each other), and robust clusters that do not change significantly as new centroids are introduced.

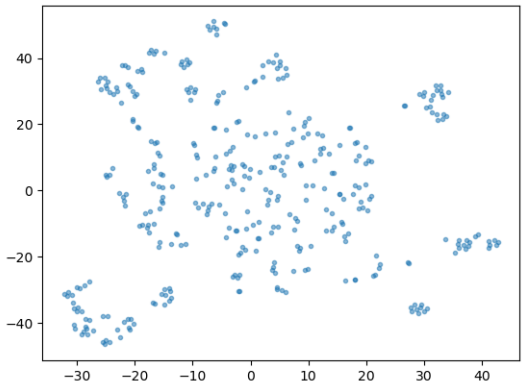

Figure 12: Distribution of China vaccine data in 1000D vector space

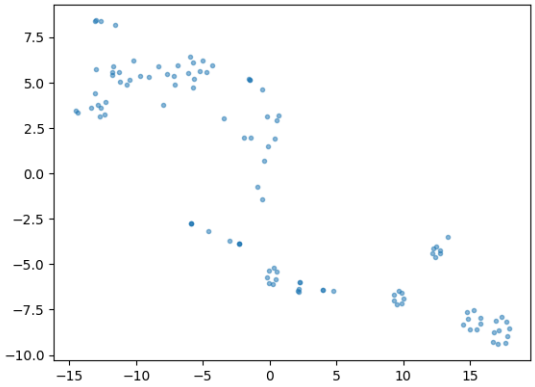

Figure 13: Distribution of US new variant data in 1000D vector space

US Data For example, we can see that the inter-cluster distance [15] drops smoothly, but the cross-cluster distance [16] shows an interesting pattern. More importantly, two very robust clusters (of respectively ~ 40 and ~ 30 elements) [17] are observed in US data. Even though the data is split into more clusters, these two groups seldom change. By retrieving a sample image from each cluster (see Appendix 32), we infer that they represent the scene of the medical expert talking to the camera and the conversation between the medical expert and the host, respectively.

From these analyses, we decide to choose $k = 4$ as the best number of clusters for the US new variant data, and the corresponding distribution of clusters seems reasonable in the 2D plane [18]. (To maintain the conciseness of this report, the visualizations for distribution of different k's are not listed, but they can be found in the GitHub `output/xx/cluster/` directory.)

China Data The performance of K-means on the Chinese vaccine data is not as expected. The "elbow" method does not work out on either the smoothly dropping inter-cluster distance curve or the smoothly climbing cross-cluster distance curve. Moreover, robust cluster centers are not observed until we split the data into very small pieces. In this context, we tend to set the optimal k at 12 through manual inspection of the cluster distribution on the 2D plane. (All visualizations see Appendix 34.)

Multi-source Data Initially, we propose to split the multi-source data into 2 or 3 clusters, which correspond to Chinese videos, US videos, and potentially another cluster for unclassified, mixed images. Yet considering the mixed distribution under t-SNE previously mentioned [14], it is not likely for our beloved K-means algorithm to effectively discriminate one source from the other.

5.4 Gaussian Mixture Model Clustering

In fact, this compromised performance of the K-means algorithm is not beyond our expectations. Our reasoning comes from the distance-based nature of K-means: even after CVAE processing, dimension explosion is still likely to happen

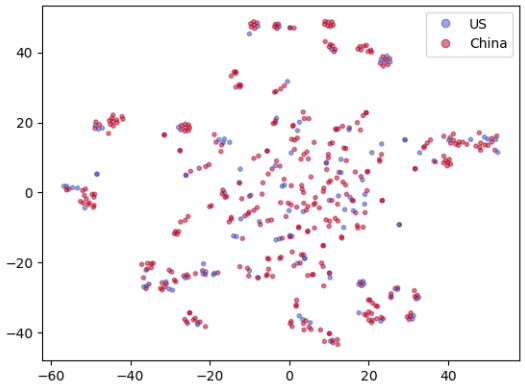

Figure 14: Distribution of Multi-source data in 1000D vector space

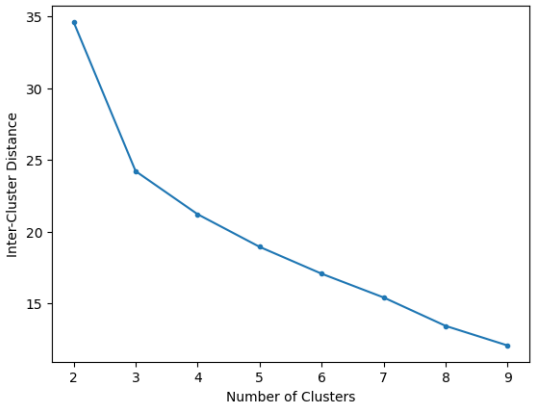

Figure 15: Inter-Cluster Distance of US data

when adding up long vectors during Euclidean distance (2nd order norm) calculation, resulting in an extremely sparse distribution of 487 elements in a 1000D subspace. In mathematical terms,

$$dist(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

The distance will increase monotonously as the dimension n increases because $(x_i - y_i)^2$ is non-negative.

To alleviate the issue of dimension explosion, we want to bypass the summation step during the distance calculation. Denying all such mainstream distance-based clustering algorithms indeed motivates us to explore Gaussian Mixture Models (GMMs). Practically, we believe that GMM has the following two advantages:

- GMM assumes an independent Normal (Gaussian) distribution of data across each of its dimensions. That says, the clustering is performed by taking each dimension separately into consideration. This no-aggregation characteristic is most desired in our context, as 1,000 relatively small differences are apparently more advantageous for clustering than 1 large value.
- GMM provides a "soft" prediction of clusters, i.e., different from K-means where each instance is assigned to either cluster A or cluster B, the instances in GMM are, instead, assigned with a probability of belonging to each cluster. Such fuzziness admits a boundary of uncertainty between clusters, which, in our context, corresponds to the mix of similar features shared between the video from both sources.

With a solid foundation on our prior knowledge of K-means, we also select 12 clusters for GMM. The centroids (means) are initialized to the 12 centroids given by the K-means algorithm, and every dimension is associated with an independent covariance matrix. Expectation-Maximization (EM) algorithm is used for stepwise optimization, and the iteration converges when the average gain is less than 1×10^{-3}, within 1,000 epochs.

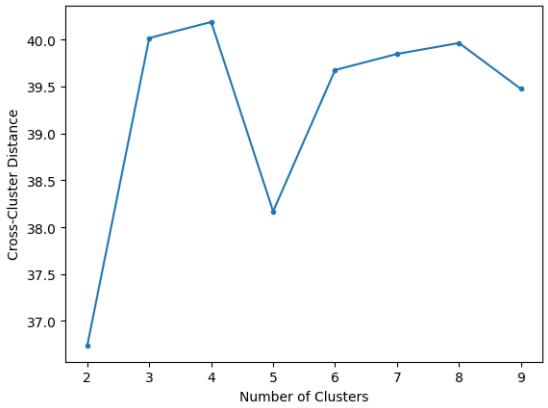

Figure 16: Cross-Cluster Distance of US data

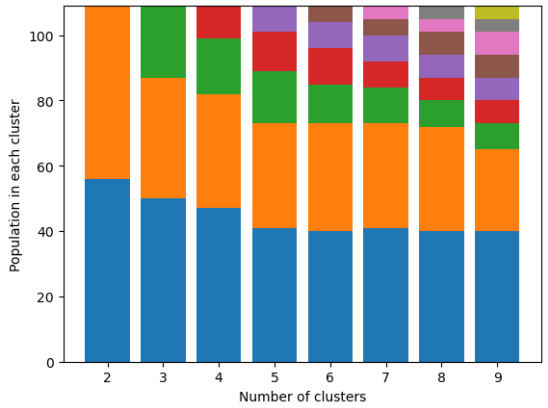

Figure 17: Population of each cluster of US data

The figure [19] shows that different from K-means, the clusters are now roughly of ellipsoid shape on the t-SNE plot. The overlap between clusters is reasonable, considering the additional dimension reduction into a 2D plane.

Insights Even though the labels (China or US) are still quite mixed, interesting patterns have been discovered through manual inspection. For example, the purest cluster mainly represents outdoor scenes (samples see Appendix 38), in which only 13% of the elements are from US data (as compared with the average of 22.4%). This is because the China dataset contributes to most of the street view images – there are only several frames of US street views. On the contrary, the scenes of the US medical expert (see Appendix 32) are generally classified into another cluster, in which only 27% are Chinese images. Mixed, fuzzy clusters also exist, where the proportions of two sources are close to the overall average distribution. It appears that the model cannot necessarily distinguish images with similar color patterns, especially those with very bright noises of red or blue.

Disadvantage It is worth admitting that the soft clustering feature of GMM is not well utilized in this analysis because the number of data instances (487) is less than the dimension of the feature vector (1,000). The clustering algorithm is actually overfitting on the given data, and thus predicts probabilities of either 1 (belongs to this cluster) or 0 (does not belong). With more images in the training data, we may consider fitting it into 2 clusters and analyzing the uncertain instances with intermediate probabilities ($\frac{1}{2}$).

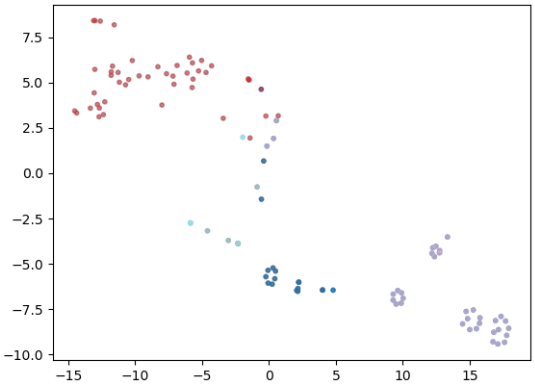

Figure 18: Distribution of 4 clusters in US data

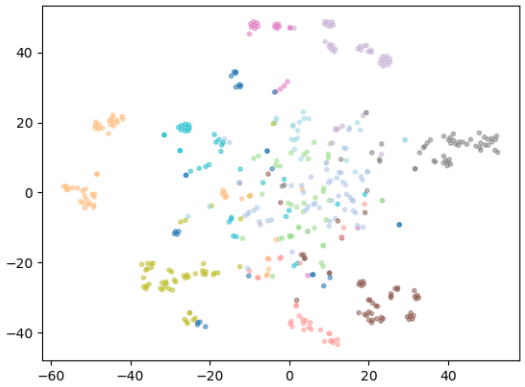

Figure 19: Distribution of 12 clusters by GMM in Multi-source data

6 Conclusions

In this project, we presented a full pipeline of Computer Vision tasks on videos, which includes in-depth theories of Neural Networks and Artificial Intelligence, Dimension Reduction approaches on image frames, Machine Learning-based Clustering algorithms to extract keyframes, and comparison and contrast of features and characteristics across videos from different cultural backgrounds.

We successfully proved and demonstrated the effectiveness of Dense Neural Network layers in Convolutional Variational Autoencoders with both conceptual inferences and real-life examples. Our analysis through Gaussian Mixture Models further revealed the strong correlation between cultural orientation and affinities in news videos, specifically on the topic of COVID-19. We are excited to see the potential of this framework to be well-adapted to the cultural variety topic, or other more generalizable ones.

Future Works This project indicates a future research area of collecting labeled data about detailed topics associated with each video, on which supervised learning can be performed on the data point embeddings in the latent space. In addition, without ground truth supervision labels given, semi-supervised or self-supervised learning algorithms can be adopted, which focus on whether a keyframe belongs to a certain video.

Acknowledgments I would like to express my gratitude toward fellow researcher Alan Luo at Columbia University for his great contribution to the collected datasets, as well as Prof. Joshua B. Gordon for his teaching and instruction on various Deep Learning topics, especially on concepts of Autoencoders. Beyond, thank you to Anne Wei for her love and support.

References

Omer F. Onder. 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. URL http://www.cs.columbia.edu/~jrk/NSFgrants/videoaffinity/Interim/21y_Omer.pdf.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Internal Representations by Error Propagation*, volume 1, pages 318–362. MIT Press, 1987. URL https://ieeexplore.ieee.org/document/6302929.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. 2013. doi:10.48550/arXiv.1312.6114. URL https://arxiv.org/abs/1312.6114.

Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. *Advances in Neural Information Processing Systems*, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/eb86d510361fc23b59f18c1bc9802cc6-Abstract.html.

Yaxiang Fan, Gongjian Wen, Deren Li, Shaohua Qiu, Martin D. Levine, and Fei Xiao. Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. *Computer Vision and Image Understanding*, 195, 2020. doi:10.1016/j.cviu.2020.102920.

7 Appendix

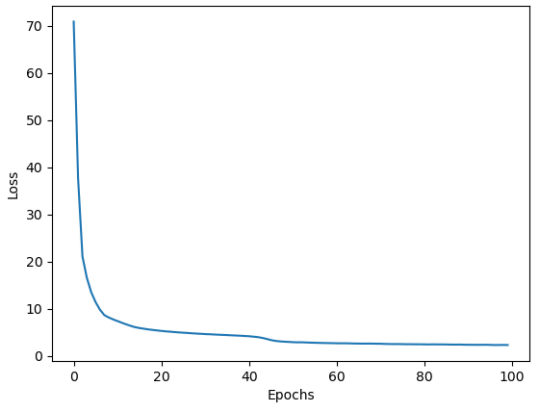

Figure 20: MSE of Pure CVAE on China vaccine dataset

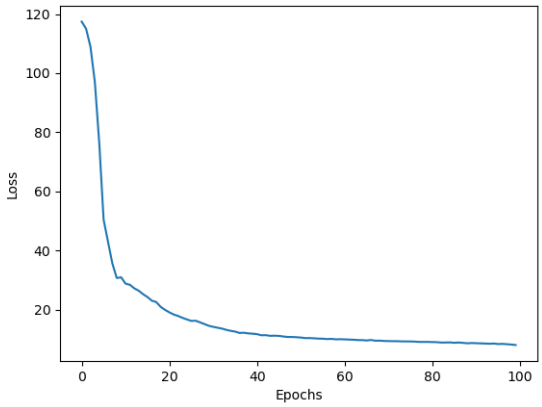

Figure 21: MSE of Pure CVAE on US new variant dataset

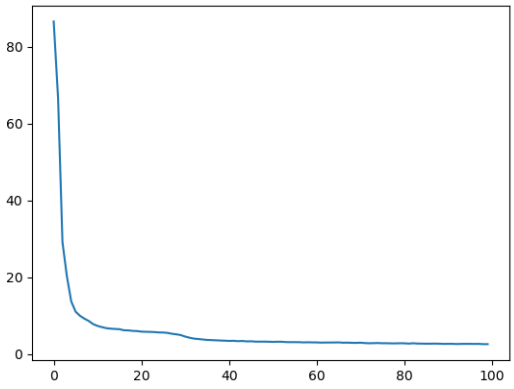

Figure 22: MSE of Pure CVAE on Multi-source dataset

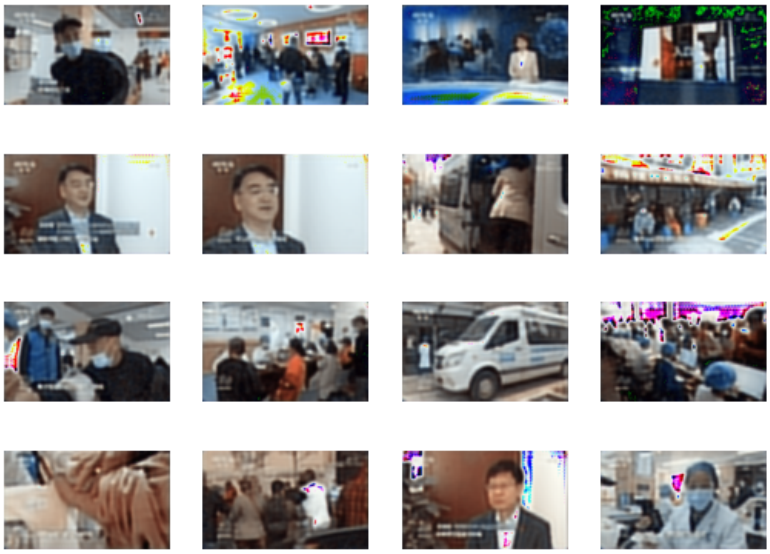

Figure 23: Sample output of Pure CVAE on China vaccine dataset

Figure 24: Sample output of Pure CVAE on US new variant dataset

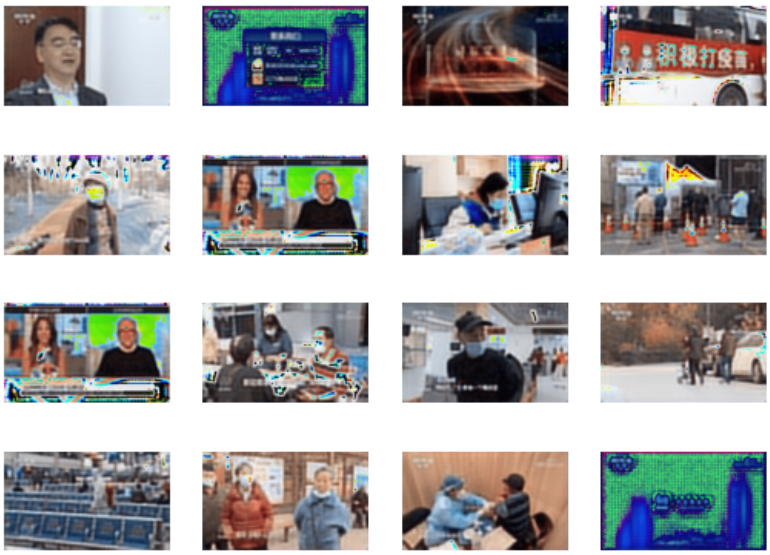

Figure 25: Sample output of Pure CVAE on Multi-source dataset

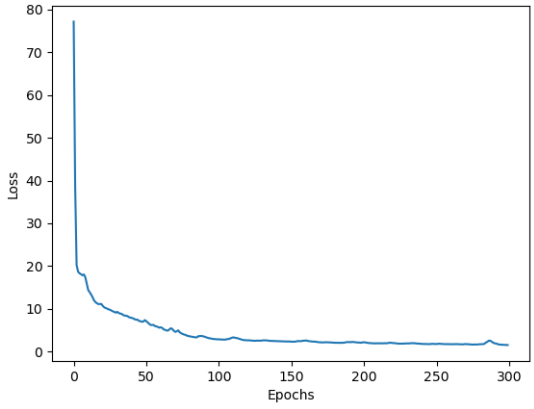

Figure 26: MSE of Dense CVAE on China vaccine dataset

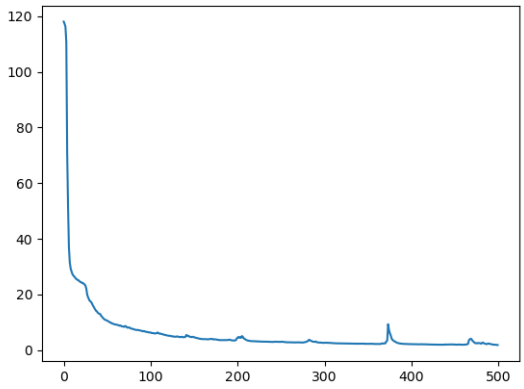

Figure 27: MSE of Dense CVAE on US new variant dataset

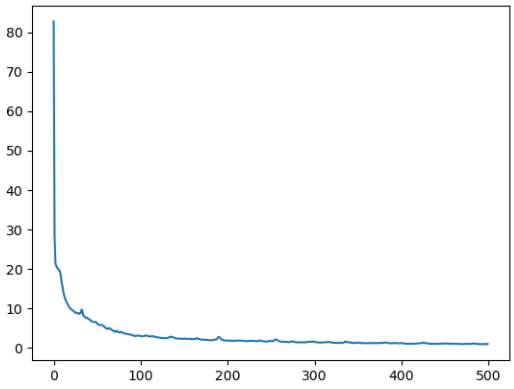

Figure 28: MSE of Dense CVAE on Multi-source dataset

Figure 29: Sample output of Dense CVAE on China vaccine dataset

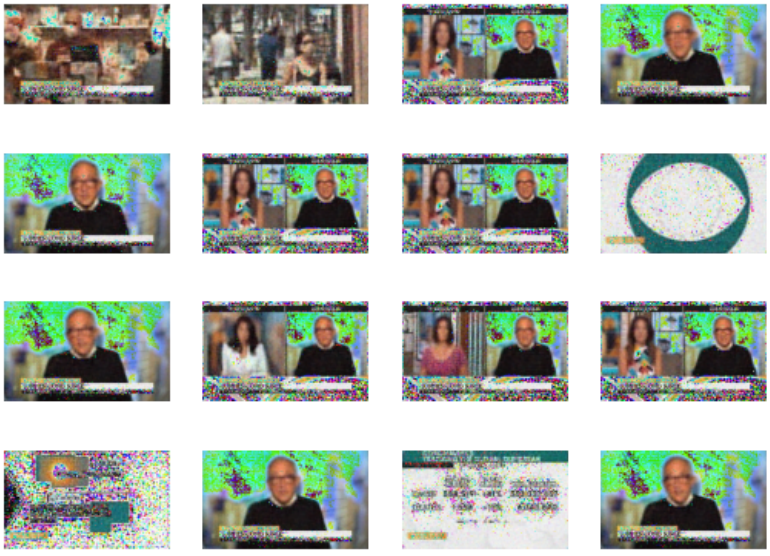

Figure 30: Sample output of Dense CVAE on US new variant dataset

Figure 31: Sample output of Dense CVAE on Multi-source dataset

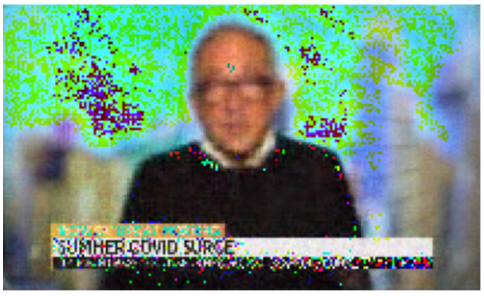

Figure 32: Sample from the largest cluster (reconstructed)

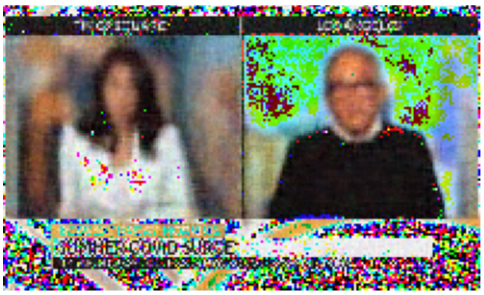

Figure 33: Sample from 2nd largest cluster (reconstructed)

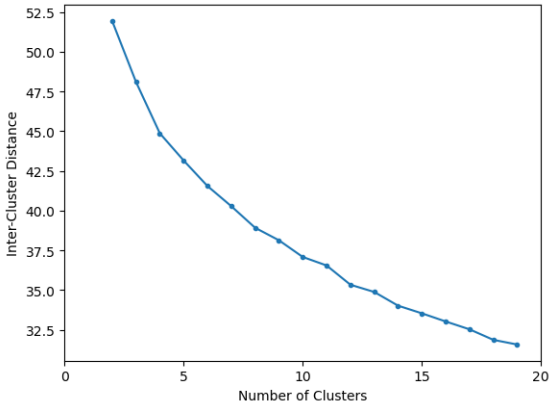

Figure 34: Inter-Cluster Distance of China data

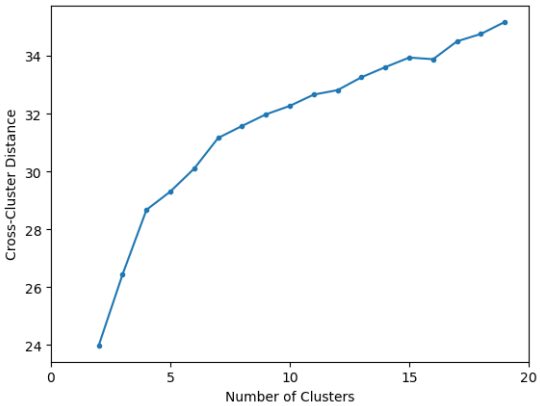

Figure 35: Cross-Cluster Distance of China data

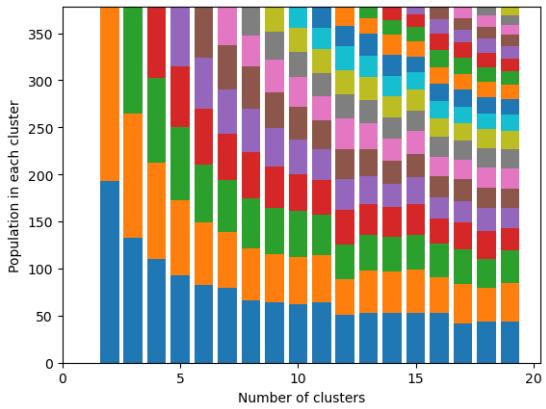

Figure 36: Population of each cluster of China data

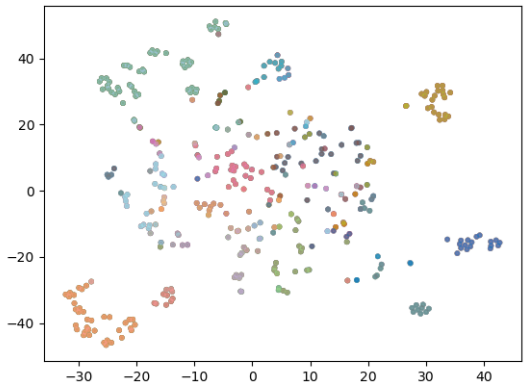

Figure 37: Distribution of 12 clusters in China data

Figure 38: Sample from cluster of outdoor view (reconstructed)

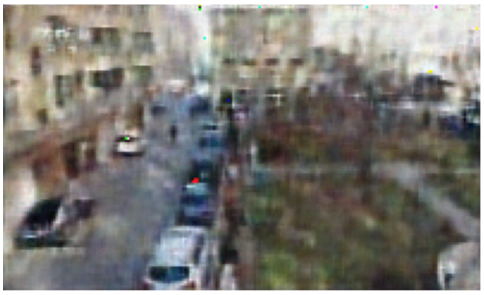

Figure 39: Sample from cluster of outdoor view (reconstructed)