A CULTURE-AGNOSTIC IMAGE AND TEXT CLUSTERING SYSTEM WITH MULTIMODAL AUTOENCODER ARCHITECTURE

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

John R. Kender
Department of Computer Science
Columbia University

January 1, 2024

ABSTRACT

The emergence of Large Language Models (LLMs) has introduced a new approach to video content abstraction, both visual and audio. Previous research has shown that vanilla Convolutional Variational Autoencoders (CVAE) can effectively reduce the dimension of video frames to enable clustering algorithms. In this work, we propose another Convolutional-Recurrent Variational Autoencoder (CRVAE) model, which incorporates encoder-decoder-structured fully-connected and LSTM layers to extend the modality of the previous model. Further, an end-to-end video understanding pipeline is built, including procedures of frame-caption alignment, latent space vector clustering, and LLM-based cluster interpretation. Experiments are designed to evaluate and compare the model performances, and the system is validated on videos with different cultural orientations.

1 Introduction

The Big Data era has witnessed the explosive development of online multimedia as an efficient source of information, among which videos convey information with high temporal density to the audience through both images and audio. 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 video content extraction and abstraction are gradually gaining their significance. For this work, we are primarily motivated to employ modern Computer Vision (CV) and Natural Language Processing (NLP) techniques to generate video tags or short descriptions without human supervision, to briefly understand a video without watching every second.

It is our intuition that the information density of videos comes in temporal peaks – i.e., key information may be contained in a short series of frames and captions – so our initial aim was to extract these slices of the video of interest by performing clustering techniques. 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. Upon studying, the ideas from previous work [Onder, 2021] have inspired us that Variational Autoencoders with only Convolution layers (Pure CVAE, see Figure 1) can reduce the dimension of images low enough for popular Machine Learning algorithms to handle.

It is worth admitting that pure CVAE is, by every means, the state-of-the-art Neural Network model for encoding 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 Linear layers, also following the encoder-decoder structure (Dense CVAE, see Figure 2). Experiments are then designed to show the superiority of dense CVAE over pure CVAE, along with the theoretical reasoning to support such arguments.

Enlightened by the idea that ignoring the first and last convolution layers, the dense neural network layers in the center can, by themselves, act like a vanilla Variational Autoencoder and learn the important features and characteristics, independent of the format of input vectors, the researchers then propose to incorporating NLP elements into the original Computer Vision-forwarded CVAE model and presents the CRVAE (Convolutional-Recurrent Variational Autoencoder)

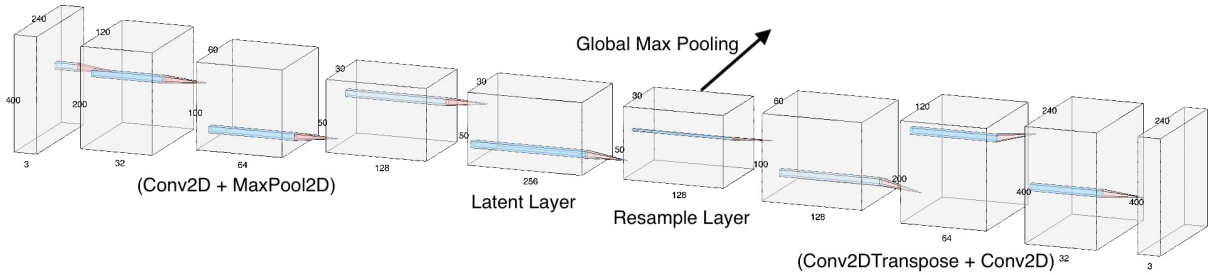


Figure 1: Pure CVAE Architecture

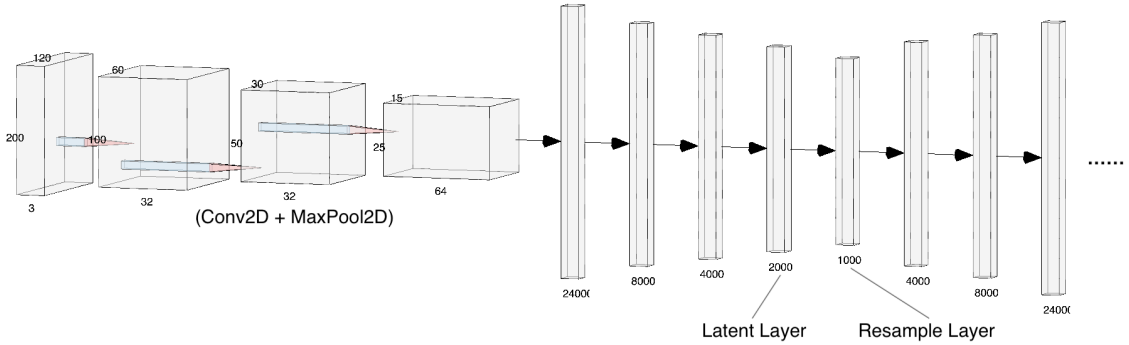


Figure 2: Dense CVAE Architecture (duplicated Convolution layers at the end are omitted)

model, to handle both texts and images. In a practical sense, it is worth notifying that the audio channels in our news videos mainly consist of clear speech of organized sentences. In such consideration, we decide to directly transform the audio data into natural languages in text format, instead of keeping a time series of audio inputs.

Besides the CRVAE model and necessary preprocessing steps, our full pipeline (see Figure 3) also consists of clustering methods in the pre-neural network era (especially the K-means algorithm), followed by visualized and verbalized evaluation metrics to determine the optimal set of clusters. We use the "elbow method" to select the most suitable k for the inter- and cross-cluster distances. To understand the real-world meanings (if any) of each cluster, we further apply the Bootstrapping Language-Image Pre-training (BLIP) model to caption each image frame and the LLaMA model to generate tags for each cluster (including texts and images). The above pipeline is tested on COVID-19 news videos in both China and the United States, and in a high-level overview, it takes a YouTube video link as input and returns a certain number of clusters of {`frame`, `caption`} pairs, with 10 tags as a description of each cluster's content.

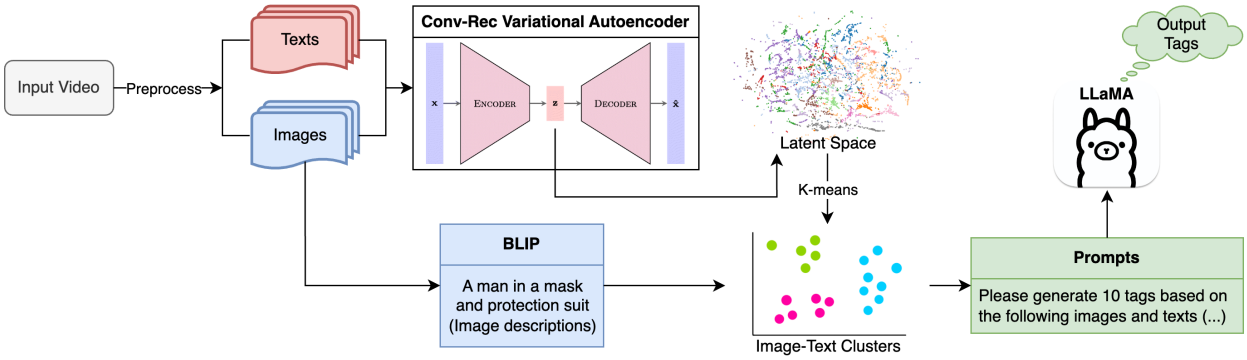


Figure 3: System Overview

The project's source code is available on GitHub `https://github.com/Anemonee1212/cvae_video_cluster` and `https://github.com/Anemonee1212/crvae_video_cluster`.

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.

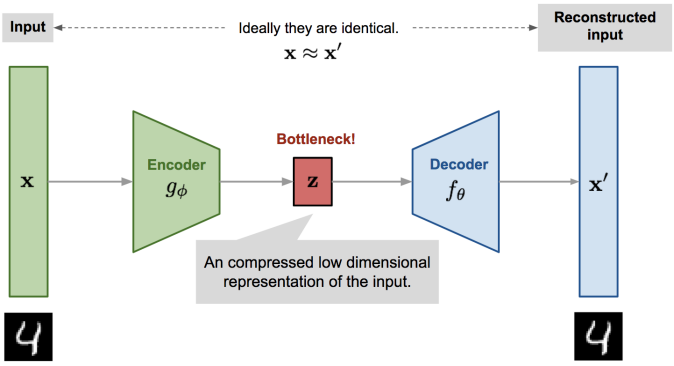


Figure 4: General Structure of Autoencoders

Figure 4[1] 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.

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 encoding 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

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

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,

$$\begin{aligned}
&\underset{\theta,\phi}{\operatorname{argmin}} \|x - x'\|_2, \text{where } x \text{ is the input data, } x' \text{ is defined by} \\
&\begin{cases} z = E_\theta(x), x' = D_\phi(z), & \text{for AEs} \\ z = E_\theta(x), z' \sim p(z|x), x' = D_\phi(z'), & \text{for VAEs} \end{cases} \\
&E_\theta, D_\phi \text{ are encoder and decoder networks with parameters } \theta \text{ and } \phi, \text{ and} \\
&p(z|x) \text{ is the Normal distribution function with parameters } \mu, \sigma \text{ to be learnt}
\end{aligned}$$

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.

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

Concrete progress has been made by Onder [2021] and other researchers toward the encoding and 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 work started, the original AlphaGo news dataset and the code for CVAE training, 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.

2.4 Recurrent Neural Network and its variants

RNN and its descendants had long been the state-of-the-art method for NLP tasks, because of their memory property when dealing with sequential data. Specifically, for each cell in a sequence, a hidden state vector is kept, incorporating all the information from the beginning of the sequence up to this cell. Such hidden state is then concatenated with the input value and passed through a forward layer. In mathematical terms,

$$y_t = h_t = \tanh(W_h h_{t-1} + W_x x_t + b)$$

The initial task of RNN is to perform Language Modeling, i.e., to model special patterns like grammar or phrases of the given language. The method of training is to minimize the cross-entropy error of predicting the next word

from the existing words, e.g., given "<Start> Hello world", predict "Hello world <End>". To solve this task, the original RNN model adopted an encoder-decoder structure, which can intuitively be taken advantage of by the Autoencoder structure in our task. For this reason, we are inspired by the idea of combining RNN-based networks with our existing CVAE.

Admittedly, the idea of using recurrent cells to handle sequences of natural languages (oftentimes with variable lengths) is quite impressive. Yet it is still worth pointing out that various issues prohibited vanilla RNN models to achieve satisfactory performance. The most outstanding one is the gradient vanishing issue in long sequences – the gradients of the loss function with respect to parameters in early cells are raised to extremely high power, resulting in infinite or infinitesimal values (in most cases).

Long Short-Term Memory [Hochreiter and Schmidhuber, 1997] (LSTM) model is the most commonly used variant to address the gradient vanishing issue. By taking advantage of another "cell state", in addition to the original hidden state, LSTM avoids exponential products which directly causes gradient vanishing. In mathematical terms, the model of LSTM looks like

$$\begin{aligned}(f_t, i_t, o_t, g_t) &= W_h h_{t-1} + W_x x_h + b \\ c_t &= \sigma(f_t) \odot c_{t-1} + \sigma(i_t) \tanh(g_t) \\ h_t &= \sigma(o_t) \odot \tanh(c_t)\end{aligned}$$

where f_t, i_t, o_t are forget, input, and output gates, σ is the sigmoid activation, and $\odot$ is the element-wise matrix product, i.e.,

$$\begin{aligned}\sigma(x) &= \frac{1}{1+e^{-x}} \\ A \odot B &= [a_{ij} b_{ij}]_{ij}\end{aligned}$$

Thanks to the forget gate and cell state features, LSTM models, in practice, achieved much higher performance in all major NLP tasks. Empirically, LSTM shows an effective memory of previous information in a sequence of around 100 cells, while the capability of vanilla RNN is only 7 cells.

Gated Recurrent Unit (GRU) can be viewed as the intermediate form of RNN and LSTM. It is more complex than RNN by adding another reset and update gate to control the flow of how hidden states are passed along and updated at each cell of the sequence. On the other hand, it is simpler than LSTM because it does not have a separate cell state. In mathematical terms,

$$\begin{aligned}(r_t, z_t) &= W_{gh} h_{t-1} + W_{gx} x_t + b_g \\ g_t &= W_h \left(\sigma(r_t) h_{t-1}\right) + W_x x_t + b \\ h_t &= \sigma(z_t) \odot \tanh(g_t) + (1 - \sigma(z_t)) \odot h_{t-1}\end{aligned}$$

Figure 5[2] is a clear illustration of these three models.

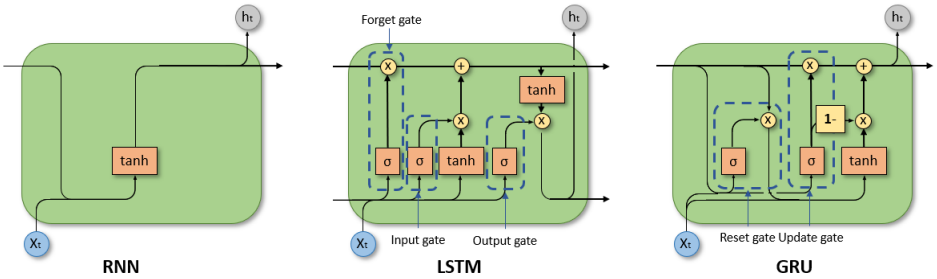


Figure 5: RNN, LSTM, and GRU cells

It is also worth emphasizing that even though vanilla RNN models have inherent directions among the sequence, we can stack another layer of RNN cells with the opposite direction to make the model bidirectional so that it can learn the language patterns in both directions, as illustrated in Figure 6.

[2] https://towardsdatascience.com/a-brief-introduction-to-recurrent-neural-networks-638f64a61ff4

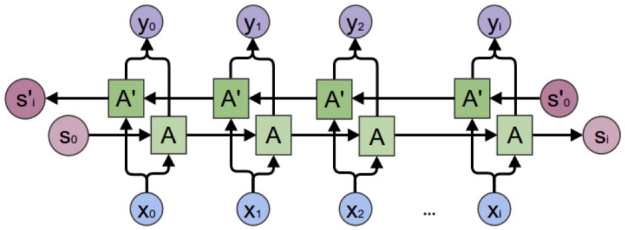


Figure 6: Bidirectional RNN (LSTM) Cells

2.5 Bootstrapping Language-Image Pre-training (BLIP)

BLIP [Li et al., 2022] is a pre-trained language-vision model which bridges the visual concepts of images and natural languages. In the paper, Li et al. [2022] proposed a multimodal framework that combines Vision Transformers (ViT) image encoder and self-attention text encoder and decoders. (See Figure 7.) BLIP achieved state-of-the-art performance on multiple tasks, including Image-Text Retrieval, Image Captioning, Visual Question Answering (VQA), and NLVR2.

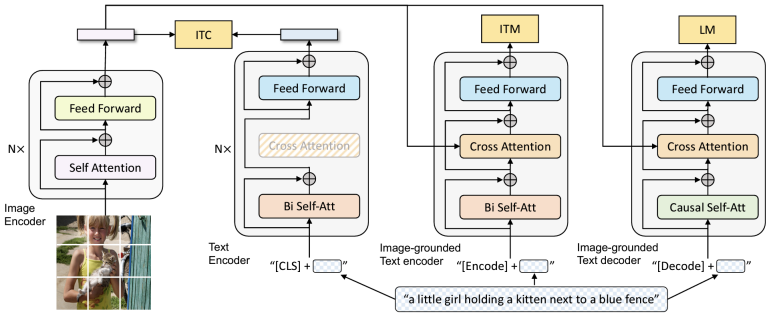


Figure 2. Pre-training model architecture and objectives of BLIP (same parameters have the same color). We propose multimodal mixture of encoder-decoder, a unified vision-language model which can operate in one of the three functionalities: (1) Unimodal encoder is trained with an image-text contrastive (ITC) loss to align the vision and language representations. (2) Image-grounded text encoder uses additional cross-attention layers to model vision-language interactions, and is trained with a image-text matching (ITM) loss to distinguish between positive and negative image-text pairs. (3) Image-grounded text decoder replaces the bi-directional self-attention layers with causal self-attention layers, and shares the same cross-attention layers and feed forward networks as the encoder. The decoder is trained with a language modeling (LM) loss to generate captions given images.

Figure 7: BLIP Framework

Specifically, for Image Captioning tasks, the text encoder block allows conditional image captioning, in which case BLIP additionally takes a text segment as a prompt and generates subsequent text that describes the image. This additional prompt serves as a hint to the model. For example, a common image prompt may be "*A photo of ...*". In contrary, unconditional image captioning does not have such inputs.

2.6 Llama 2

Llama 2 [Touvron et al., 2023] is a significant contribution to the field of language models, building upon the original LLaMA model and introducing several innovations that enhance the model's performance. The authors proposed an open foundation for chat models, which enables fine-tuning on a wide range of tasks, including text classification, sentiment analysis, question answering, and more. This approach allows for greater adaptability and customization of the model to specific domains and applications.

Llama 2 also introduces several new techniques for improving the performance of the model, including:

- A novel attention mechanism that allows the model to focus on different parts of the input sequence, improving its ability to capture long-range dependencies and contextual information,
- A fine-tuning framework that enables quick adaptation of the model to new tasks without requiring significant changes to the underlying architecture, and
- A safety analysis mechanism that evaluates the model's performance under various adversarial attacks (e.g., toxicity, discrimination, etc.), providing insights into its resistance to these threats.

Llama 2 is made open source by Meta with 3 different sizes – 7B, 13B, and 70B – each representing how many billions of parameters. In practice, the template for prompting LLaMA is as follows:

Prompt Template: `<s>[INST] <<SYS>>` { *system_prompt* } `<</SYS>>`
{ *user_prompt* } `[/INST]`

where the *system_prompt* is the general instructions on how LLaMA should respond, and the *user_prompt* is specific conversations that user interacts with LLaMA. Multiple user prompts (and corresponding responses) can be provided subsequently.

3 Dataset

To deploy and evaluate our pipeline on a topic with popular and impactful insights, we chose to focus on the news videos related to COVID-19. This topic was selected due to its global attention and awareness, as well as the potential cultural difference reflected by how official news agencies report and propagate. During the exploration phase of our research, we believed that compared with the outdated AlphaGo-beats-human debate used in previous work, the worldwide impact of COVID-19 could indeed arouse more public interest.

In our context, in news videos, images will typically include outdoor scenes recorded by the reporter and indoor scenes of hosts in the studio, while audio will include words of the hosts and guest speakers, or sometimes even background music. The raw image dataset is sampled at the rate of 1 frame every 2 seconds (i.e., 30 frames per minute). Since we introduced multimodality to our model, the preprocessing step is now more effort-taking, in order to align segments of text data with each image frame.

3.1 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 video length is 3 minutes and 40 seconds in total, which generates 109 image frames, excluding irrelevant scenes at the beginning and end. Some manually labeled typical frames include the host and/or the medical expert talking to the camera, and a presentation slide.

The text data of US videos are collected from the YouTube auto-generated closed caption (transcript/subtitles/etc.) in English, enabled by the Python `youtube-transcript-api` package. We manually inspect the text to ensure its correctness. Since the autogenerated caption does not include punctuations, we can simply use the Basic English Tokenizer in the PyTorch `torchtext` package. We use a pre-trained GloVe (Global Vector) Embedding with a dimension of 300 to embed every word (token).

The Transcript API segments the whole script paragraph into 96 segments of text, which gives approximately 10 to 20 words in each segment. In addition, it also provides an accurate timestamp at which time it appears and disappears, which shows that the period between 2 neighboring segments are roughly 2 seconds (which corresponds to the rate at which we sample the frames). Given such detailed information, we aligned the images to the text segments by selecting the frame closest to their starting time.

Sample text and images are given as follows:

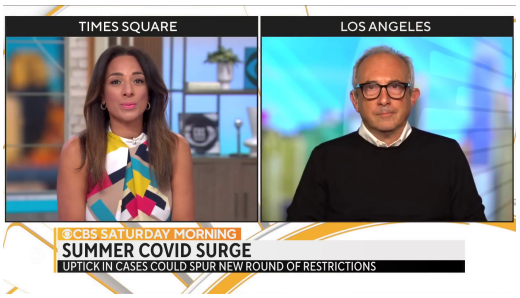
```
"health officials here in new york city",
"and in los angeles are sounding the"...
```

3.2 Chinese Vaccine Video

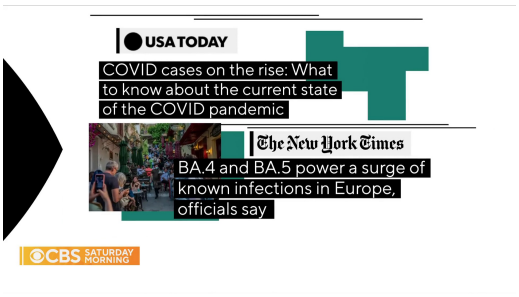
This video[4] is published by China Central Television (CCTV) on YouTube. The primary purpose of this video was to encourage (or demand) senior Chinese citizens to take vaccines for COVID-19, while also reporting the current situation of the pandemic and the progress of disease control. This video is relatively longer than the previous US New Variant Video, with 15 minutes and 8 seconds, so it generates 378 image frames in total. Typical frames involve the host and/or the medical expert talking to the camera, and a presentation slide.

Since YouTube does not provide transcripts for Chinese videos, we have to resort to external audio-to-text converters. However, the source videos involve some conversations in Chinese dialects (which are pretty different from Mandarin),

[3]Video link: `https://www.youtube.com/watch?v=doP5UacBlt0`
[4]Video link: `https://www.youtube.com/watch?v=xcWeBCOMoiU`



(a) US host and official talking



(b) US presentation slide

Figure 8: US Sample Frames

(a) Chinese officials talking about COVID-19

(b) Chinese elderly taking a vaccine

Figure 9: China Sample Frames

so the accuracy of the AI converter is not ideal. We apply careful manual inspection to correct its mistakes and add timestamps with a segment every 10 seconds. This 10-second period is chosen with the consideration that the information density of Chinese is relatively larger than that of English. If we stick to the 2-second period, the segments will be too short for the model to capture any valuable information.

The manual process of text data gives us 90 text segments, which need to be aligned to the 378 image frames. The average sample rate is 4.2 frames/segment, so we uniformly sample 5 images out of a series of 21 image frames, under the assumption that within a short period of time, the speech of Chinese words (or characters) is also uniformly distributed. We further use Jieba Chinese word tokenizer as well as Chinese Word Vectors embedding [Li et al., 2018], which also embeds Chinese words and characters into vectors of dimension 300. However, Jieba does not necessarily remove the punctuations during tokenization, so we need to explicitly filter out Chinese-style punctuations like “,”, “.”, and “?”, in order to avoid these meaningless embeddings affecting the model training.

Sample text (translated to English) and images are given as follows:

```
"Should the elderly people take COVID vaccines? Yes, elderly people must take vaccines.
It can prevent severe illness and deaths with roughly 90%..."
```

Note Due to the character-based nature of Chinese language, and the shortcoming that Jieba and Chinese Word Vectors (CWV) do not cooperate, there exist some Chinese words (approximately 6%), as defined by Jieba, that are not recognized by CWV. An outstanding example is the Chinese abbreviation of the word “coronavirus”. In such cases, we have to either handle the word as unknown tokens (`"<UNK>"` embedded as a vector of zeros) or use a more fragmentary tokenizer to further split the word into single characters (e.g., the word “coronavirus” can be split into “new” and “crown” in Chinese).

4 Methods

4.1 Pure CVAE vs. Dense CVAE

In the previous work [Onder, 2021], where only images from AlphaGo-beats-human news are considered, the author stated that applying a Global Max Pooling layer on top of the 128-channel latent layer (see Figure 1) 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 used in traditional CNN for classifications. However, 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 to encode images, 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 propose the dense CVAE model: CVAE with Dense neural network layers in the middle. (See Figure 2.)

Specifically, we use 3 2D-convolution layers (with kernel size 3×3, a stride of 1, and a Max Pooling layer of 2×2) followed by 3 fully connected layers for the encoder network, and symmetrically for the decoder network, only that we use Transposed Convolution layers (with stride 2) followed by another Convolution layer of same dimension to restore the original dimension (or so-called "de-convolution"). The model setup and training details of both pure and dense CVAE are provided in Appendix 7.1 through 7.4.

4.2 Convolutional-Recurrent Variational Autoencoder

Based on the dense CVAE framework and new data with multimodal aspects of text and images, we propose our new CRVAE model architecture as shown in Figure 10, which takes an image and a sentence as input each time, processes them in parallel using CNN and LSTM respectively, and finally combines the multimodal input vectors through fully connected layers. It generally follows the encoder-decoder structure, determined by a latent layer in the center. The final output of this model is the vectors in the latent space, with lower dimensions compared to the input images and text.

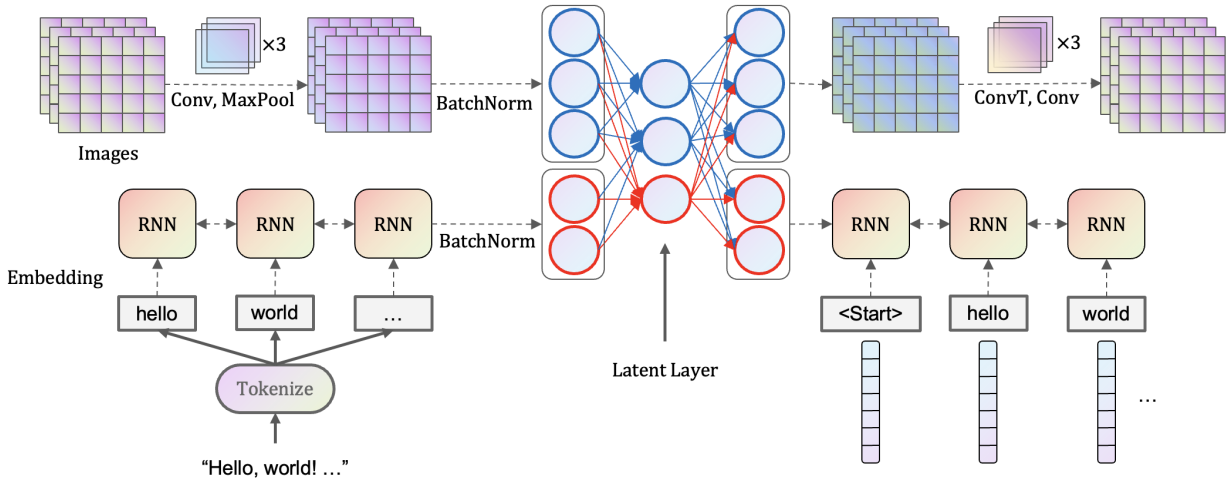


Figure 10: CRVAE Architecture

4.2.1 Encoder

Similar to the dense CVAE model, we set the dimension of input images as $200 \times 120 \times 3$, with a batch size of 16, and 3 represents the RGB channels. The framework of Convolution and Max Pooling layers remains the same, only that the

numbers of filters (channels) are set to be uniformly 32. The motivation to use fewer channels in the last convolution layer (decreased from 64 to 32) is that after flattening, we will have $32 \times 25 \times 15 = 12000$ neurons, instead of 24000, which significantly reduces the model size.

The text encoder network is an RNN-based model, with the input of a sequence of embedded text (in dimension 200). Note that the pre-trained embedding weights (Chinese or English) are fixed and are not optimized during Gradient Descent. The model is selected from either vanilla RNN or LSTM, both of which have a hidden size of 512 (i.e., the length of the hidden state vector) and 2 stacked layers of bidirectional recurrent cells. The results show that the LSTM model has higher performance than vanilla RNN on all tasks. GRU is not implemented or evaluated in our model, because we know that it is an intermediate form between RNN and LSTM, so we tend to assume that LSTM will always have better performance than GRU.

It is also worth emphasizing that since 2017, the Transformer [Vaswani et al., 2017] model (and its variants), thanks to the self-attention mechanism with an absolute advantage in long-range inter-sentence dependencies, have exceeded the state-of-the-art performance of LSTM in all major NLP tasks. Indeed, the initial thought of the authors *was* to incorporate the Transformer's encoder and decoder into the CRVAE model, but the following shortcomings, after taking into consideration, make Transformers less ideal than LSTM in our context:

- Transformer cells, which include Multi-Head Attention, Feed Forward Neural Network, and Residual Connection layers, are more computationally complex than LSTM. The reasons why Transformer-based models typically train faster are that they are more suitable for pre-training, and that parallelize better by avoiding the sequential operations of RNN-based models. But in our case when pre-training is less significant, and when the sequences are relatively shorter (segments of sentences), such advantage in parallelization does not outweigh its disadvantage in model size.

- Again, since we are dealing with segments of sentences, the strict grammar and lexical structure of a language are oftentimes weakened or even broken. That says, the self-attention cells in Transformers are harder to learn the important language patterns of our text data, while the cross-segment attentions are impractical to train through neural networks.

CRVAE uses 2 fully connected neural layers in the middle. The outputs of the Convolution layers and LSTM layers are flattened and normalized (using the Batch Normalization layer) to ensure that they are approximately on the same scale. The resulting vector, after concatenation, has the dimension of $32 \times 25 \times 15 + 2 \times 2 \times 512 = 14048$, where 2×2 represents 2 layers of LSTM, both in 2 directions. It is then narrowed down to 1 layer with 4000 neurons, and the final latent layer with 2000 neurons (with which 1000 for the latent mean and the other 1000 for the latent standard deviation in logarithm scale).

The latent mean μ and latent standard deviation $\ln \sigma$ are then used to resample (in Normal distribution) across the latent space, and the resulting vector is passed to the decoder network for the reconstruction of images and text. After the training session, the latent mean μ is finally calculated as output.

4.2.2 Decoder

The decoder network is roughly symmetric to the encoder network, as the 1000-dimension input is gradually rebuilt to 4000 and 14048-dimension. It is then spitted into 2 parts and trained to reconstruct the images and text separately.

In contrast to the encoder where images are downsampled by the Max Pooling layer, the image decoder uses Transposed 2D Convolution layers with 3×3 kernel and a stride of 2 to "upsample" an image channel, resulting in doubled width and height. Each of these Transposed Convolution layers is followed by a Convolution layer with the same kernel size (3×3) and number of channels (32). After 3 (Transposed Convolution, Convolution) blocks, the tensor is reconstructed into the original resolution with 3 channels, representing the Red, Green, and Blue pixels of an image. We use the pixel-wise Mean Squared Error (MSE) between the input image and the reconstructed images as the loss function.

The text decoder in CRVAE is different from the traditional LSTM decoder for NLP tasks because we would require the model to handle multilingual inputs. Generally, we would map the neurons to a layer with the same length as the vocabulary size and assign a SoftMax activation, which corresponds to the Cross-Entropy loss. However, due to the different natures of Chinese and English, this approach will not work. Instead, we now require the decoder to predict the embedded text as tensors and optimize the MSE loss between the original and reconstructed embedded text. This explains why we "freeze" the embedding layer during the encoder network – the word embeddings are not involved in the training session. In response to this change, some additional features of the text decoder that are worth mentioning are as follows.

- We explore the effect of the Teacher Forcing algorithm on our text decoder network. By using Teacher Forcing during decoding, at each LSTM cell, we predict the next word by the input of *ground-truth* previous word embeddings, while without Teacher Forcing, the input is the *predicted* previous embedding. The name "Teacher Forcing" of this algorithm is an analogy of a teacher instructing the student step by step to solve problems.

  We experienced a significant performance increase with this algorithm, and such use is also theoretically reasonable. This is because the task of our network is not training a Language Model to generate texts. Instead, our purpose in building the text decoder is to train the Autoencoder to learn important features and characteristics of the given text, and these types of information (encoded in the latent layer) are passed on to the decoder through hidden and cell states. That says, the input of each cell should indeed be correct words to avoid "misunderstanding".

- Upon using the MSE loss, our final reconstructed text segments are actually tensors of numeric values, instead of words (in Chinese or English). Yet we, humans, cannot necessarily interpret these vectors, so we still need some methods to "visualize" (or more exactly, "verbalize") the vectors. In a practical term, we search for the nearest neighbor of an embedded word in the vocabulary, and further "decode" this vector as the word which has the closest embeddings. Note that this nearest neighbor verbalization is not part of the training process either.

4.3 CRVAE Model Configurations

By constructing our model based on the above architecture [10], we can formulate 2 types of MSE loss – image loss and text loss. The final loss function is $L = ImageLoss + \lambda\, TextLoss$, where λ is a ratio hyperparameter to balance the reduction of losses during the training session. In practice, we set $\lambda = 3$.

All intermediate layers in the model are activated by ReLU (Rectified Linear Unit) function, and an Adam (Adaptive Momentum) optimizer with a learning rate of $\alpha = 10^{-4}$ is used. The model is trained for 500 epochs on local devices[5], and a typical training session lasts for around 60 minutes, which is similar to the training time of a 300-epoch dense CVAE model.

4.4 Clustering and Cluster Interpretation

After encoding each {`frame`, `caption`} pair into a latent space of dimension 1000, we perform K-means clustering on all 1000D vectors generated from a certain video. We first traverse through a series of numbers of clusters k to determine an approximate range of optimal 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.

Besides these quantifiable metrics, we also propose a quality evaluation in an intuitive manner. Traditional opinions are that clustering algorithms – in contrast to classification – group the elements only based on their similarities, with no consideration of their real-world meanings. However, by taking advantage of the deep contextual understanding capability of LLMs, we present a way to interpret the meanings of each cluster. Specifically, we choose to prompt the LLM to generate short, English tags for the cluster of {`frame`, `caption`} pairs, considering the conciseness and popularity of tags as description of videos.

We first use the BLIP-base model on HuggingFace (`blip-image-captioning-base`) in full, `float32` precision to "caption" each image frame. We experiment both conditional and unconditional image captioning, and the image prompt provided for conditional one is given below.

Image Prompt: *A news photo of ...* (BLIP will generate subsequent text.)

Note To avoid the misunderstanding between "video caption" in news videos and "image caption" generated by BLIP, we refer to the latter one as "image description" or "frame description" from now on.

We then use the Llama 2 model with 7B parameters on HuggingFace (`Llama-2-7b-chat-hf`) to perform generative language modeling. Because text elements can be easily concatenated, we combine all information of frames and captions in a cluster into the user prompt, in addition to the following system prompt.

[5]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.

System Prompt: *Please generate 10 short tags for a series of frames sampled from a YouTube news video based on the images and captions provided. Please avoid generic words that describes the whole video, but to emphasize the unique characteristics of these frames. You may need to implicitly infer the meanings of the objects in the image description according to the video context.*

The purpose of the second sentence is to avoid common keywords that appear in all clusters (e.g., "coronavirus" or "COVID-19"), and the purpose of the last sentence is to emphasize the contextual understanding between image frames and text captions. In a practical term, in our context, a sample image description can be "a man in a mask and protection suit", and we want LLaMA to infer that this man is likely a medical staff performing examinations for COVID-19. The final prompt looks like the following.

Prompt Template: `<s>[INST] <<SYS>>`
Please generate 10 short tags for a series of frames sampled from a YouTube news video based on the images and captions provided. Please avoid generic words that describes the whole video, but to emphasize the unique characteristics of these frames. You may need to implicitly infer the meanings of the objects in the image description according to the video context.
`<</SYS>>`
`Text caption: ...` (some text captions in English or Chinese)
`Image description: ...` (some image descriptions in English)
`[/INST]`

The sampling process during LLaMA text generation is controlled by several hyperparameters, including `temperature`, `top_k`, `top_p`, etc. We experimented on different values of `temperature` $= t \in [0, 1]$. A general rule of thumb is that with small t, LLMs tend to respond in a conservative way, predicting text that most likely to follow, while with large t, LLMs tend to be creative. In practice, we set a large value of $t = 0.9$ for a diversified generation.

5 Results

5.1 Autoencoder Model Experiments

The performance of the CVAE and CRVAE models on image encoding task is listed below.[6] We conclude that dense CVAE is superior to pure CVAE.[7] We successfully prove the theoretical soundness of adding linear layers in the middle, and thus setting up a solid foundation of CRVAE.

Table 1: CVAE and CRVAE Model Performance

		Dataset	
Model	Loss Type	China	United States
Pure CVAE	Image	0.585	1.458
Dense CVAE	Image	0.382	0.662
CRVAE	Image	**0.068**	**0.133**

Table 2 is the comparison of performance across all subvariants of our CRVAE model. We conclude that the LSTM subvariant of CRVAE model with Teacher Forcing algorithm is superior to either RNN subvariant or without applying Teacher Forcing in all datasets, and we decide to go on with this. Further, we can see that our best model performs better in image reconstruction for Chinese data, and text reconstruction for US data. Such differences in behavior are somewhat intuitive, as the common consensus is that Language Modeling is much harder for Chinese than English.

The loss curves of the CRVAE (LSTM + TF subvariant) on the 2 datasets are shown in Figure 11. We can see that for the Chinese dataset, the loss from text is always larger than that from images, while in both datasets, the Image Loss decreases rapidly at first, and then converges slowly, together with the Text Loss.

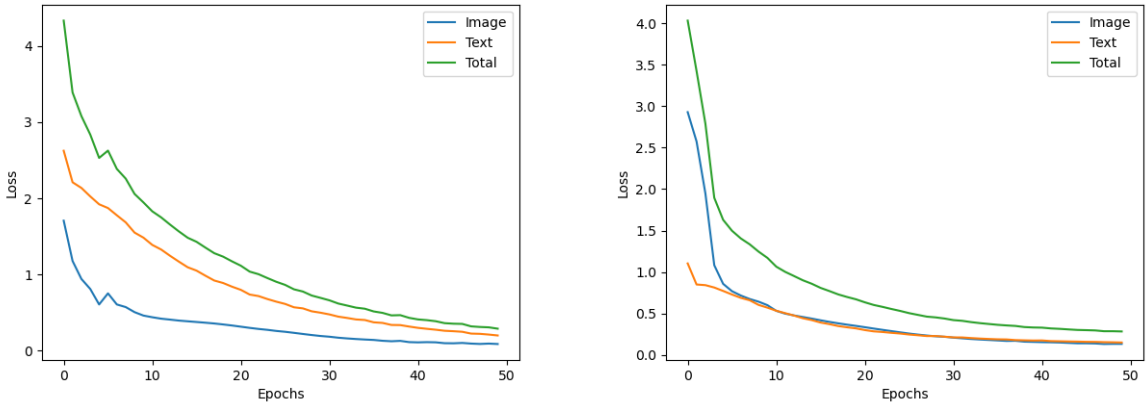
The sample reconstructed images of our model are also shown below (see Figure 12 and 26). The output images are generally fuzzier than those generated by CVAE (see Appendix 23), yet considering the fact that mixing information

[6] It is worth admitting that the number of epochs for which the 3 models are trained does not match. Pure CVAE is trained for 100 epochs and starts to converge at around Epoch 70; dense CVAE is trained for 300 epochs and converges at around Epoch 200; CRVAE is trained for 500 epochs. Yet the difference in training time will not affect the performance because of the convergence.

[7] The MSE losses of CVAEs are scaled in the same coefficient to account for the difference of image resolutions between CVAE and CRVAE, but the relativity remains unchanged.

Table 2: CRVAE Subvariants Model Performance

Subvariant	Loss Type	Dataset	
		China	United States
RNN + TF	Total	1.002	1.064
LSTM + No TF	Total	0.539	0.520
LSTM + TF	Image	**0.068**	0.133
LSTM + TF	Text	0.085	**0.050**
LSTM + TF	Total	**0.323**	**0.284**



(a) China vaccine dataset

(b) US new variant dataset

Figure 11: MSE Losses of CRVAE during training

from text into the neural network for dimension reduction will add confusion to the model, demanding it to learn and distinguish between information from both sources, such minor flaws are just a fair trade-off.

In addition, we manually examined some reconstructed images in detail, (see Appendix 24), and notice an outstanding pattern that, if the original image has a white or relatively lighter background, the reconstructed image will likely have minor bright noises of pure colors (e.g., red, green, or blue) in those areas. This common pattern is also shared in CVAE results. (See Appendix 7.4.) We tend to attribute this phenomenon to the fact that Neural Networks are mostly activated around 0, which makes them harder to predict large values (e.g., $R = 255, G = 255, B = 255$). If we are decoding white colors, it is likely that some pixels will lack (at least) one channel of colors, which will show us a "bright spot" in the image.

Despite all these shortcomings, we conclude that CRVAE is satisfactory in image reconstruction, as it almost successfully recovered all major parts or objects in the image. In addition, we verbalize the reconstructed text and clean up the padded tokens. We can see that the results are very accurate.

5.2 t-SNE Visualization

Even though the models encode high-resolution image frames and long text captions into an array in its lower-dimension subspace, the 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 8 for both datasets. We can see that most data points are sparsely distributed in the 2D space, with only a few loose clusters. We also notice several small but compact clusters for Chinese dataset, as well as a very extreme outlier for US dataset in Figure 14.

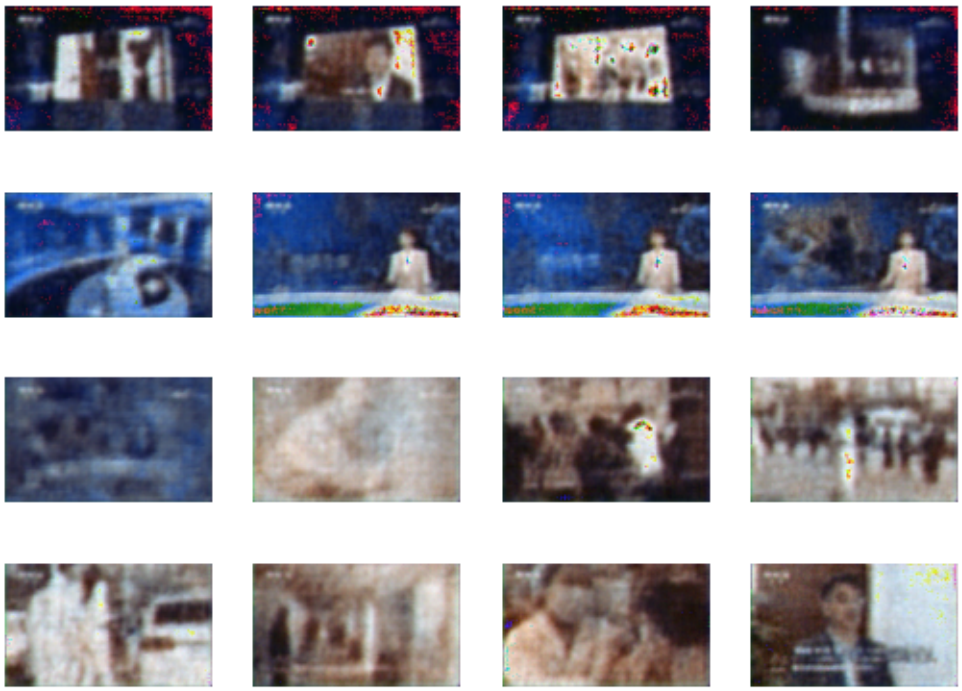


Figure 12: Sample output on China vaccine dataset

Chinese:
老年人 要 接种 新冠 疫苗 吗 老年人 一定 要 打 疫苗 他防 重症 和 防 死亡 能够 达到 90% 左右 的 保护 有 基础性 疾病 的 人 其实 更 应该 接种 这个 疫苗 应接 尽接
筑牢 免疫 屏障 婆婆 你们 都 打 了 疫苗 了 吗 打 了 我们 都 打 了 打上 我 家里 安全 省 着 你 担心 传染 对 大家 和 个人 都 有 好处 用 事实 说话 焦点访谈
你好 观众 朋友 欢迎您 收看 今天 的 焦点访谈 随着 新型 变异 毒株 奥密克戎 的 出现 全球 新冠 肺炎 疫情 的 第三阶段 形势 更加 严峻 中国 外防 输入 内防 反弹
的 压力 也 变大 了 要 战胜 疫情 疫苗 是 有利 的 武器 在 我国 的 14 亿 人口 当中 有 两个 多亿 是 60 岁 以上 的 老年人 目前 老年人 人群 的 疫苗 接种 情况 如何
对于 老年人 来说 接种 新冠 疫苗 安全 不 安全 有 哪些 注意事项 会 不会 有 不良反应 今天 我们 就 来 聚焦 这个 群体 的 疫苗 接种 问题 南非 新增 新冠 确诊 病例
超过 11000 例 发现 首例 奥密克戎 病毒 奥密克戎 变异 毒株 感染 奥密克戎 变异 病毒 奥密克戎 毒株 已经 几天 在 全球 至少 38 例 目前 新冠 疫情 仍 在 全球 肆虐
情况 不容乐观 尤其 近日 新 变异 毒株 奥密克戎 奥密克戎 截至 12 月 3 号 奥密克戎 毒株 以 波及 全球 38 个 国家 和 地区 这 给 我国 的 疫情 防控 工作 带来 巨大
压力 自 十月 以来 之初 一步 在 我国 辽宁 大连 黑龙江 内蒙古 北京 上海 等 多 地 都 出现 了 由 多个 不 关联 的 境外 输入 源头 引起 的 新一轮 疫情 在 这样 的
情况 下 疫苗 接种 工作 尤为重要 截至 12 月 2 号 31 个省 自治区 直辖市 和 新疆 生产 建设 周敏 累计 报告 接种 新冠 疫苗 二 五亿 3279.9 万 完成 全程 接种 的
人数 超过 11 亿人 而 在 尚未 进行 疫苗 接种 的 群体 当中 有 相当 一部分 是 老年人 我们 60 岁 以上 的 老年人 达到 了 2.64 亿 之初 二楼 我们 到 目前为止 的话
我们 大概 还有 接近 20% 的 老年人 也就是说 有 5000 万左右 的 老年人 没有 接种 新冠 疫苗 之初

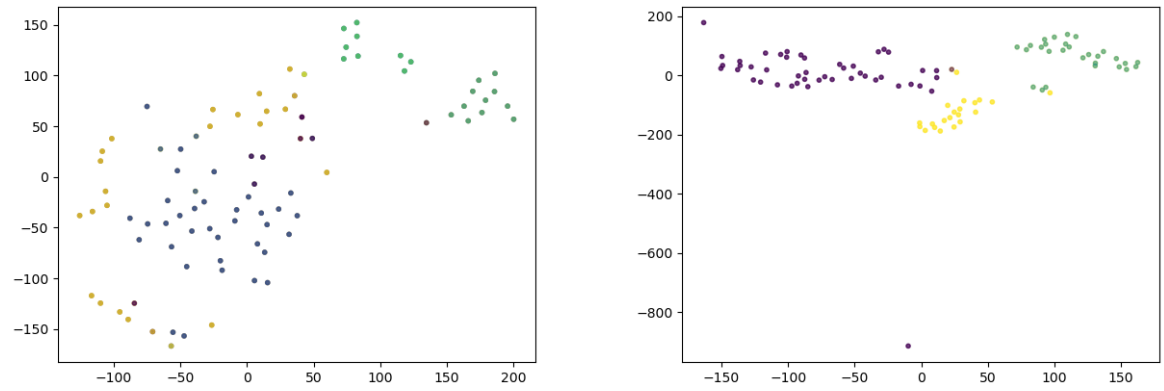
English:
officials here in new york city well in los angeles are sounding the well about a resurgence of the rising transmission rates in way angeles could
force a return to an well mask mandate new york city ' s well department is urging people to well masks in public indoor and indoor well and around
large outdoor well this comes as cases and well now on the rise way here to discuss this latest way is cbs news medical contributor dr way agus
he ' s in los angeles david way morning so it ' s called ba5 it ' s the way sub-variant that everyone seems

Figure 13: Sample text on China and US dataset

5.3 K-means Clustering

5.3.1 US Data

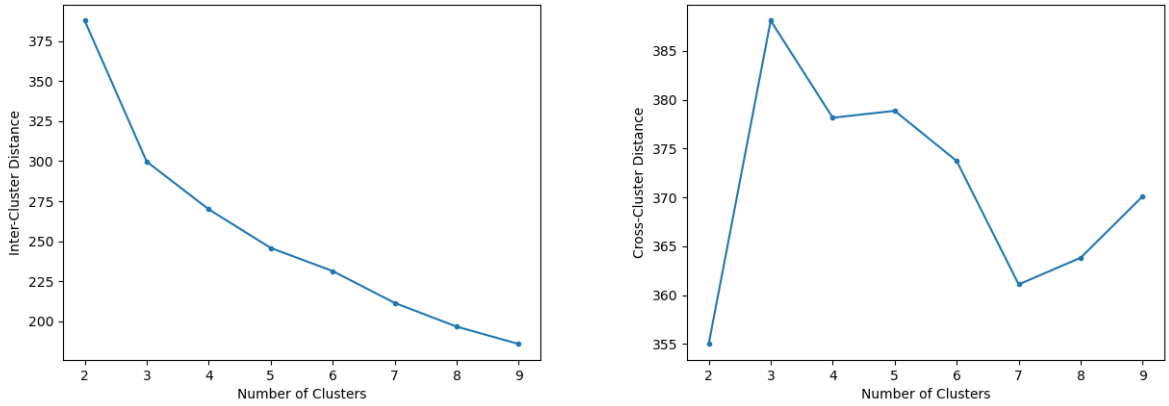
We can see that the inter-cluster distance shows a sharp "elbow" point at 3, and the cross-cluster distance reaches its maximum at 3 in Figure 15. These patterns are sufficient for us to choose $k = 3$ as the best number of clusters for the US new variant data, and the corresponding distribution of clusters seems reasonable in the 2D plane [14b]. (To maintain the conciseness, the visualizations for distribution of different k's are not listed, but they can be found in the GitHub `output/xx/cluster/` directory.)



(a) China data (5 clusters)

(b) US data (3 clusters)

Figure 14: Distributions of latent vectors in 2D space



(a) Inter-Cluster Distance

(b) Cross-Cluster Distance

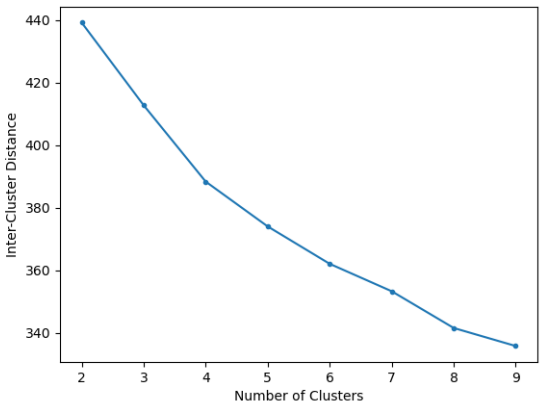
Figure 15: Cluster evaluation metrics for US data

5.3.2 China Data

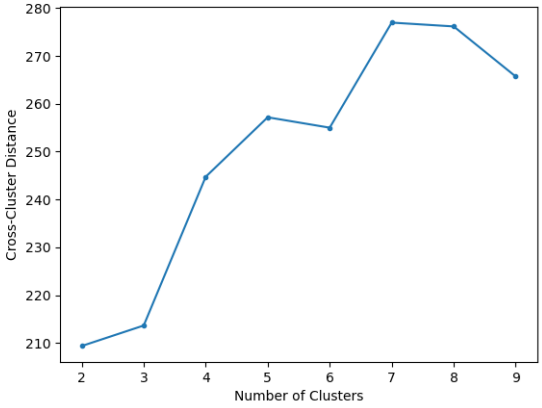
Choosing the optimal value of k on Chinese vaccine data requires more detailed analysis. The inter-cluster distance curve decreases smoothly, with a weak indication of "elbow"-like pattern at $k = 4$. However, the cross-cluster distance curve tells the other story, which shows that the distance between centroids are still on the increase at 4 clusters. (See Figure 16.) In this case, we resort to the cluster population plot in Figure 17b to see if any robust clusters turned out. We can see that as k increases from 3 to 5, the largest two clusters are pretty robust, while the smaller clusters are subject to flexible changes. That provides some evidence for us to select $k \approx 5$. The distribution of 5 clusters are shown in Figure 14a.

5.4 Cluster Interpretation

After encountering the uncertainty in selecting the optimal number of clusters for China data (only based on quantifiable metrics), we are further motivated to see the actual meanings of each cluster, hopefully without traversing through all images and texts in the dataset. Thus, we take advantage of the cutting-edge Large Language/Vision-Language Models, and their results are shown below.

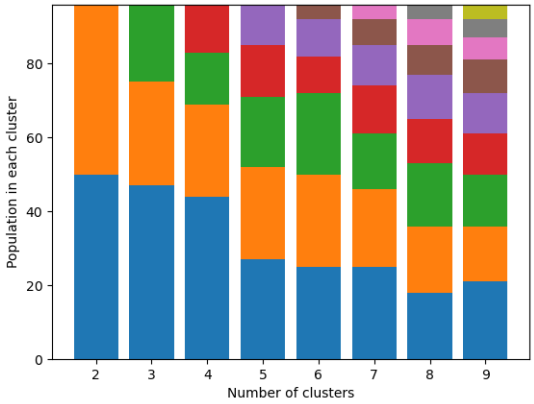


(a) Inter-Cluster Distance

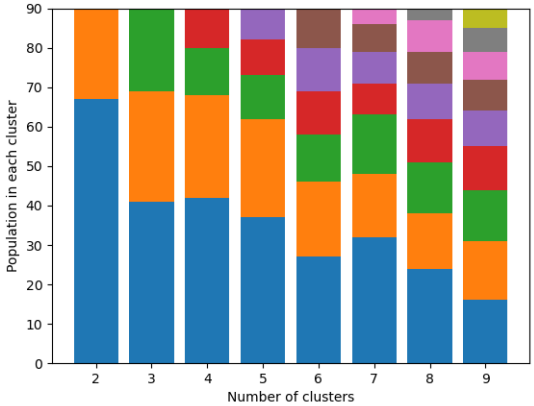


(b) Cross-Cluster Distance

Figure 16: Cluster evaluation metrics for China data



(a) Population of each cluster of US data



(b) Population of each cluster of China data

Figure 17: Cluster evaluation metrics for US data

5.4.1 BLIP-Generated Image Descriptions

Overall, the results of image descriptions generated by BLIP are satisfying. It can successfully detect persons, objects, scenes, etc. in a frame, and make inferences to some extent. For example, in Figure 18, BLIP can detect the "lab" settings in China data, and infer the occasion of the TV show in US data. However, BLIP shows disability in Optical Character Recognition tasks. Once obvious texts exist in the image, the model starts to perform weirdly. For example, in Figure 19, it hallucinates Chinese words "priority pass for people aged 60+" into "no to the government", which is never mentioned or even indicated in the image. Similar to the US data, it cannot understand the way statistics are presented, but instead returns a meaningless repetitive sequence of "corona" and "covids".

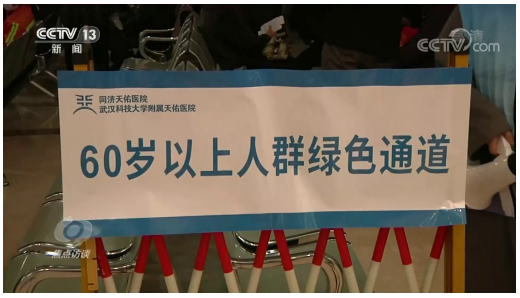
5.4.2 LLaMA-Generated Tags

We experiment with the tag generation process using Llama 2 for 3 and 4 clusters for the US video, and 4 and 5 clusters for the Chinese video (10 tags for each cluster). After manual inspections, we conclude that the results with 3 clusters are most sensible for US video, as shown in Table 4, while those with 5 clusters are most sensible for Chinese video, as shown in Table 3. In the meantime, it is also worth notifying that overall, the LLaMA model performs better on US video than Chinese video. We expect that such better performance is due to the Chinese captions, considering the fact that the training corpus of Llama 2 is mainly in English.

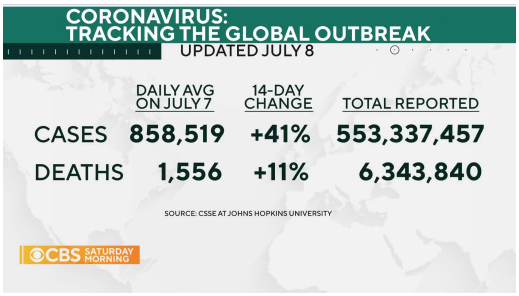
(a) "A person an a lab holding a bottle"

(b) "A woman in a colorful dress is on the set of the today show"

Figure 18: BLIP is good at detecting (and inferring) figures/objects



(a) "A sign that reads 'no to the government'" – it actually means "priority pass for people aged 60+"



(b) "corona corona covids, covids, ..." – it is a slide of COVID statistics
presented in the TV show

Figure 19: BLIP is bad at detecting texts, especially Chinese characters

We manually label the output tags by whether they correspond to the information in image frames or text captions.[8] During this labeling process, common tags throughout the whole video (which are typically shared across all clusters) are ignored. In either video, most tags are meaningful, i.e., reflect the information from images or texts. Yet there are still less than 20% of tags that contain generic, mistaken, or made-up information. Comparing with the human labor cost to going through all clusters and searching for the corresponding frames and captions, the cluster interpretation pipeline significantly increases the efficiency by only requiring to verify the 10 tags generated.

Table 3: Tags of China Vaccine Video (5 Clusters)

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
TV host*	Monitoring	COVID-19	COVID-19	Older Adults†
Masked individual*	Medical Staff*	Pandemic	Immune System†	Safety
Hospital scene*	Patient*	Vaccine†	Elderly†	Influenza
Remote work	Treatment†	Protection†	Healthcare	Healthcare Workers*
Health concerns†	Medicine*	Elderly	Medical Professional*	Medical Examinations*
Dental care	Diagnosis†	Health	Vaccination	Government†
Fitness	Testing*	Prevention	Public Health	Information
Age-related health issues†	Quarantine	Risk†	Infection Control	Technology
Global pandemic†	Global Response†	Immunity†	Virus†	Collaboration
Community protection	Infection Control†	Medical	Pandemic	Social Responsibility†

5.5 Insights

During the manual inspection of CRVAE clusters, we notice an time-sequential pattern upon we validate with the results from CVAE clusters. Different from CVAE which only takes images into consideration, the new CRVAE model

[8]Tags marked by $*$ reflect information in image frames. Tags marked by † reflect information in text captions.

Table 4: Tags of US New Variant Video (3 Clusters)

Cluster 1	Cluster 2	Cluster 3
COVID-19 news update	Ba5 Variant	group of people sitting at a table*
New York City health department alert	Medical Contributor†	covid-19 on the rise†
Coronavirus transmission rates surge†	Latest Wave of Infections†	city of Vancouver
Indoor mask mandate possible†	Health Risks†	sun setting over horizon
Hospitalizations due to COVID-19†	Repeated Illnesses†	green and white circle with words*
Booster shots and treatments crucial†	New Normal†	graphic*
New COVID-19 variant identified†	Suit*	symptoms of micro fage
Ba.2.75: the latest COVID-19 wave	Women*	percentage of coronavirus in the US*
Living with the virus†	Dress*	corona corona seen in graphic
Health officials sound the alarm†	Set of the Today Show*	corona

shows a more significant pattern – neighboring image frames and text segments in the videos are mapped to closer data points in the latent space, while in CVAE, the formation of clusters heavily relies on objects and color styles of images. We expect that such difference is because the high correlation between neighboring text captions is encoded into the CRVAE model, but ignored by the CVAE model.

It is also worth emphasizing that some interesting correlations between words in text and objects in images are also discovered. For example, in the China vaccine video, images with a certain guest speaker frequently co-occur with a Chinese word that means "the elderly". In the US new variant video, on the other hand, the word "Omicron" is oftentimes associated with medical expert Dr. Agus. This unexpected finding may imply the effectiveness of our model in processing and extracting information from multiple sources.

Disadvantages Even though not intentionally designed, the LSTM layers in our model inevitably learn Language Modeling information, which is absolutely different for Chinese and English. For this reason, cross-cultural element comparison seems to provide less real-life insights than traditional CVAE.

6 Conclusions

In this work, we constructed a full pipeline of CV, NLP, and LLM Prompt Engineering tasks on videos, integrating and processing data in different formats, including videos, audio, images, and plain texts. We designed a brand new Convolutional-Recurrent Variational Autoencoder architecture, which was established upon the combination of CVAE and LSTM while maintaining the encoder-decoder structure of both models. Fundamental theories of Neural Networks and Artificial Intelligence were applied to argue for the correctness and superiority of the model, compared with vanilla CVAE in previous work.

Advanced pre-trained Large Language Models and Large Vision-Language Models (i.e., Llama 2 and BLIP) have also been adopted to add explainability to the clustering process with no ground-truth supervision labels. High-quality, human-interpretable tags for video clusters provide concrete evidences to support the effectiveness of our system in the abstraction and extraction of key information in multimodal data. We are excited to see the potential of this framework to be applied to other cultural affinity studies in videos.

Future Works During our research phase, new multimodal LLMs emerged and indicated the potential integration of BLIP and LLaMA. For example, LLaVA-1.5 and GPT-4 can receive one single image (together with text) as inputs. Although these models did not satisfy our requirement of multiple image inputs, the development of LLMs may, one day, allow to bypass the image-to-text translation.

Our work also indicates a future research area to substitute the LSTM model with more prevalent Transformer-based models like BERT or GPT. In addition, the authors expect that besides CVAE which follows a strict encoder-decoder structure, other models with similar 2-network architecture, e.g., CGAN (Convolutional Generative Adversarial Network) might also be combined with LSTM or Transformers.

Acknowledgments I would like to express my gratitude toward fellow researcher Alan Luo, Zheng Hui, and Hui's former teammate Zihang Xu at Columbia University. Luo has contributed to the web-scraping of image data, and Hui shared with us his experience of Chinese word tokenizer and embedding.

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.

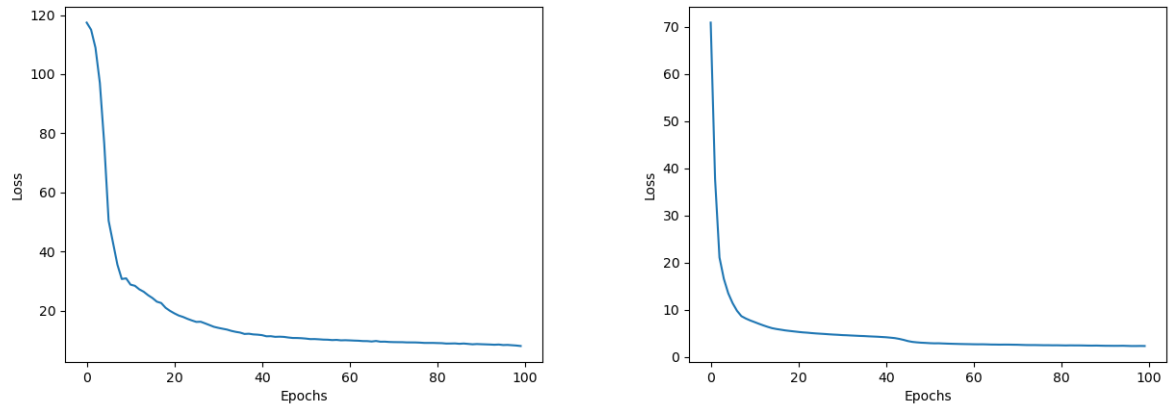
Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Proceedings of Machine Learning Research*, 2022. doi:10.48550/arXiv.2201.12086.

Hugo Touvron, Louis Martin, Kevin Stone, Amjad Albert, Peter amd Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2023. doi:10.48550/arXiv.2307.09288.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. Analogical reasoning on chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. doi:10.48550/ARXIV.1706.03762. URL https://arxiv.org/abs/1706.03762.



(a) US data

(b) China data

Figure 20: MSE Losses of Pure CVAE during training

7 Appendix

7.1 Pure CVAE Model 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.

7.2 Pure CVAE Training

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.

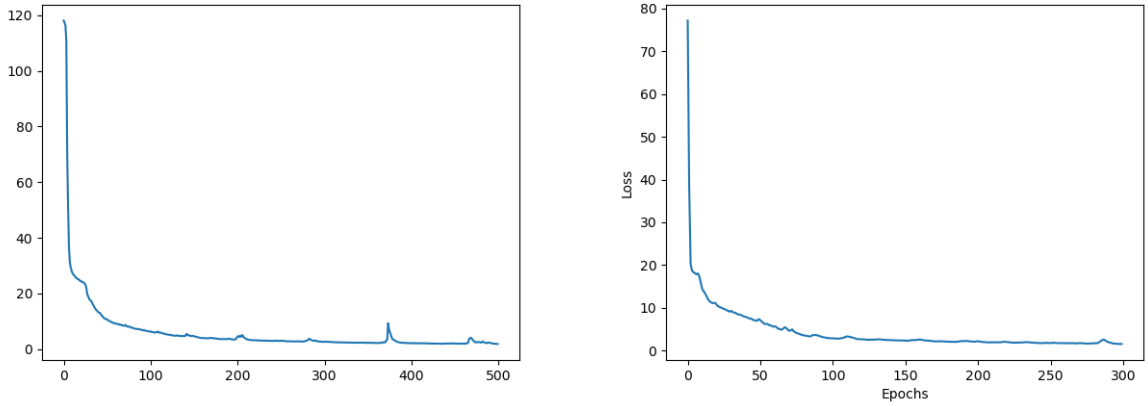
From Figure 20, we can clearly see that the model converges quickly within 100 epochs. The MSE loss at epoch 100 is 2.338, and sample reconstructed images are shown below. The output of CVAE (Figure 21a) is impressive, as it almost successfully recovered all major parts or objects in the image, except 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.

7.3 Dense CVAE Model Architecture

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,

(a) China data (b) US data

Figure 21: Sample outputs of Pure CVAE



(a) US data (b) China data

Figure 22: MSE Losses of Dense CVAE during training

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.

7.4 Dense CVAE Training

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, which includes both fitting the model and transforming the training images into reconstructed images.

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 trade-off, given the absolute advantage of how dense CVAE helps achieve our initial goal.

(a) China data (b) US data

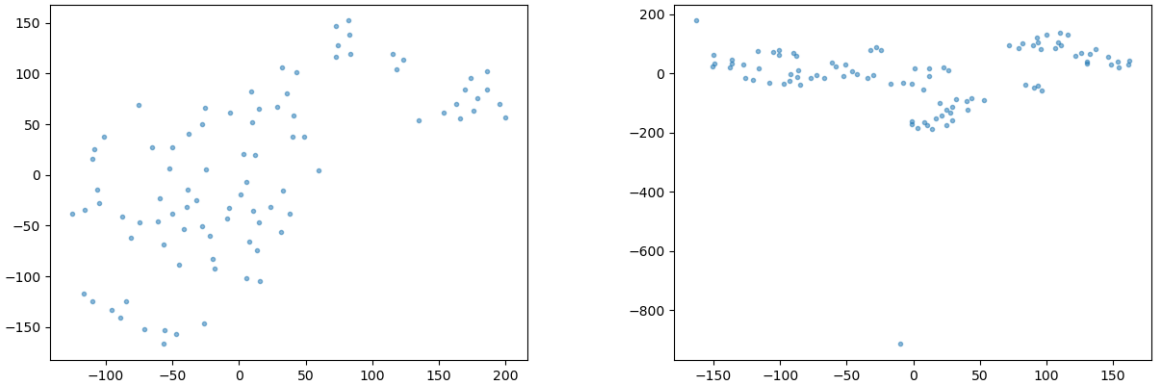
Figure 23: Sample outputs of Dense CVAE

Figure 24: Sample reconstructed images of CRVAE

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 contrasting color patterns, e.g., image 16 (lower right corner) in Figure 23a. 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 Figure 23b.) 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 trade-off that our model has to make through its training.

7.5 Supplementary Figures



(a) China data

(b) US data

Figure 25: Distribution in 1000D vector space

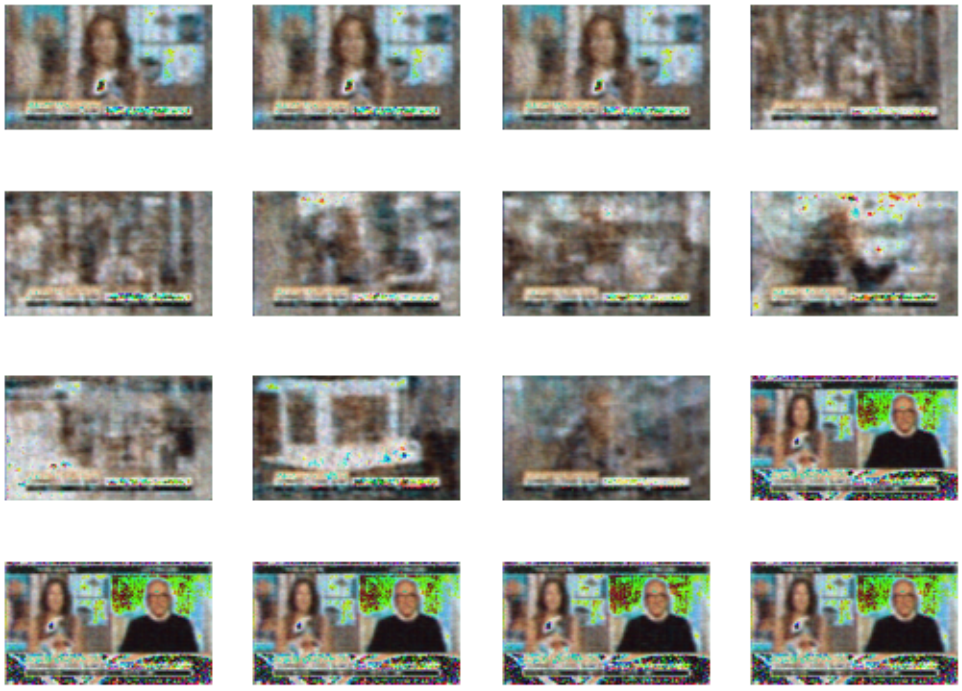


Figure 26: Sample output of CRVAE on US new variant dataset