Make It Move: Image To Video Generation For Memes

Fengyu Li, Yolanda Wang, Zichen Wang

Keywords

- Computer Vision
- Multimodal Learning
- Generative Modeling
- Finetuning for Domain Generalization
- LLM In-Context Learning

Application Settings

Our project seeks to animate static meme images for better visual effects and to better convey emotions in memes. Given that at present GIF memes are hard to create and locate, we aim to address this in our project by developing an image-to-gif meme generator that enhances the humor of such memes.

Code

Video Presentation

Project Description

To efficiently describe our project, challenges, and outcomes, we first described the original plan, then briefly detailed the built project and its differences. Next, we discuss difficulties faced, resolution attempts, and finally describe in detail the components of our final version.

The Original Plan

The initial project plan involved fine-tuning a general-purpose model for our specific task through stages including data collection via web scraping, preprocessing, transfer learning, and experimentation. Our dataset aimed to comprise numerous meme GIFs from sources like /r/gifmemes subreddit and other websites, forums, and social media sites.

After dataset acquisition, we planned standardization by adjusting image size, frame count, frame rate, and color space. We sought to select representative keyframes from memes, with original memes being labels. The dataset would then be partitioned into training, validation, and testing segments for balanced representation of meme categories.

The AI algorithm would use a deep learning model, incorporating self-attention mechanisms, residual connections, and potentially novel activation functions. We considered using architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or Generative Adversarial Networks (GANs).

Base architecture selection would rely on an analysis of computational resources, including processing power, memory constraints, and training time requirements, helping us decide on optimal model complexity for streamlined training and maximum efficiency.

In conclusion, the original plan emphasized a comprehensive approach, focusing on leveraging deep learning advancements to create a single tailored model. This involved a thorough data collection and preprocessing pipeline, transfer learning techniques, and an optimal base architecture for performance results.

Method

Overview

Our project involves three major components as we eventually decided to break down this image-to-video task into three subtasks due to the difficulties explained in the section below. The three components are:

1. An image captioning (image-to-text) model with visual question answering (VQA) capabilities that detect the content of the image meme.

2. A large language model (such as ChatGPT) that synthesizes the answers given by the VQA models and provides a comprehensive description of the relevant information in the image meme.
3. A video generation (text-to-video) model that is fine-tuned on a corpus of meme videos and texts. Its input is the synthesized caption from the large language model, and it outputs a video that attempts to capture the same theme and meaning behind the image meme.

All three modules are developed in Python, PyTorch, and HuggingFace transformers. They are chained together through function calls, thus giving the user the experience of a unified image-to-video system.

Difficulties with Image to Video

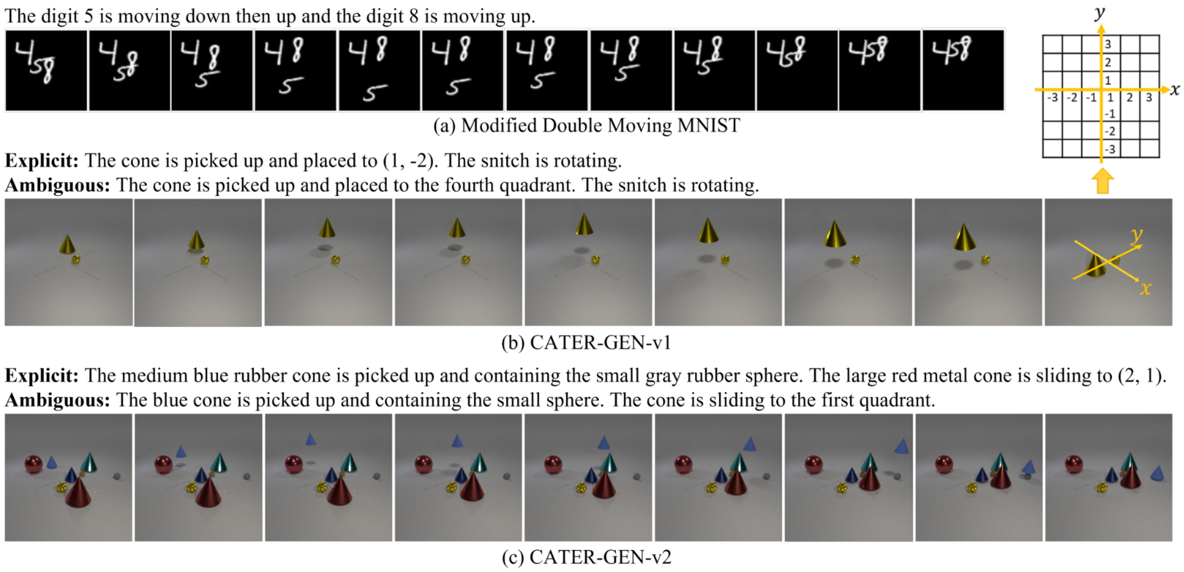
We have faced a significant challenge during our research about direct image-to-video models:

Lack of Open-Source Research

We originally planned to re-produce and then fine-tune an advanced image-to-video model on a meme dataset, but later we discovered that there is an extremely limited choice of such models. We have identified many candidate models in our plan and during our development, but none of them turned out to be of satisfactory quality. These are:

1. MAGVIT, a state-of-the-art image-to-video transformer model. When we worked on our plan its website said "code coming soon", but they ended up not releasing their code or their pre-trained model. Besides, MAGVIT is a relatively outdated model that yields less satisfying performance than newer models, failing to satisfy our needs to generate non-trivial memes with complex context.
2. Facebook's Make-A-Video project. This is an excellent example of image-to-video whose model is unfortunately never released.
3. Video Diffusion Model, a new general purpose diffusion model proposed by Jonathan Ho, focusing on Text-to-Video diffusion and video inpainting. We are very inspired by this work and have applied its ideas for our project. Unfortunately it does not support the Image-to-Video generation task.

(See below) Figure 1. Video Generation by Make-It-Move. The images show different frames of the generated video, with the leftmost image the starting input.



(a) Modified Double Moving MNIST

(b) CATER-GEN-v1

(c) CATER-GEN-v2

4. Make It Move: Controllable Image-to-Video Generation with Text Descriptions. This paper has a name that coincides with ours and they did release open source code. However, their model focuses merely on basic movement of limited objects, as shown on the above image. The model performs poorly on a real-world image/video dataset. Our test-tunes show that their model does not have even the basic potential in recognizing real-world entities.

In this last paper, the authors mentioned that even the state-of-the-art models nowadays cannot satisfactorily handle the complexities of real-world videos and animations, resulting in poor qualities in the generated output. Currently, the only way to partially address this problem is to use a massive dataset along with a huge neural network that's well beyond what a non-commercial group can afford.

Conclusion

In conclusion, we decided to avoid using a direct image-to-video generation model and instead decompose this task into several subtasks. We'll explain them in detail below.

Data Collection

Here we only briefly explain our data collection process as it is less AI-related. We used selenium to scrape the reddit channel r/gifmemes, where we eventually downloaded more than 200 memes in .mp4 format. Then, we used the open-source software ffmpeg to select one important frame from each video, which we used as the image meme. Finally we link the image memes to their corresponding video memes, creating many image-video pairs.

We realized that it is not necessary to carry out data normalization on our own, as all the models we used already have a corresponding input processor that carries out data normalization, as well as other encoding tasks.

Mixture-of-Experts Image Captioning

Here we explain the first and second components of our project which involves an image-to-text model giving various outputs and a large language model synthesizing the texts (similar to a mixture-of-experts technique).

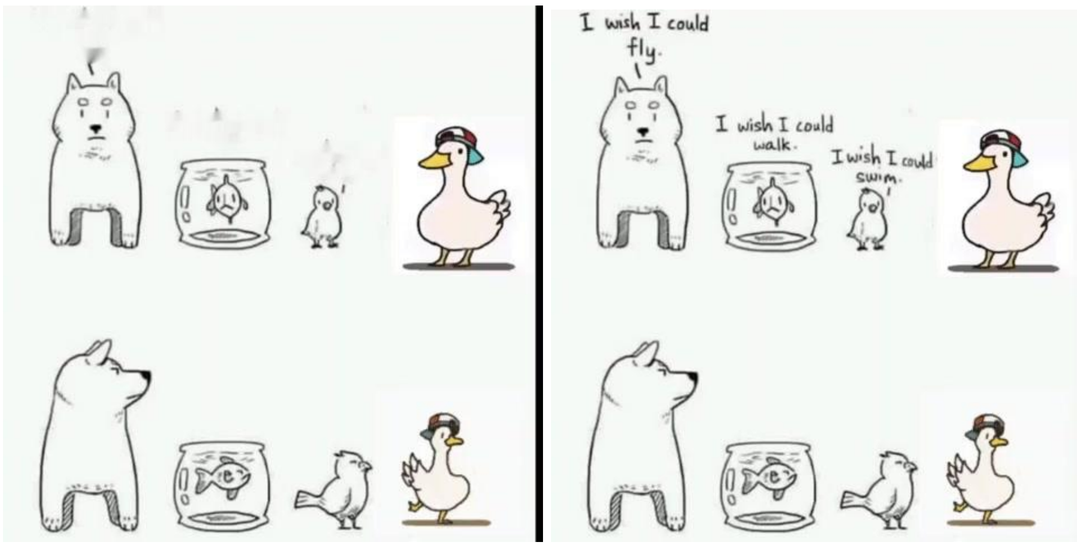


Figure 2. de-texting with Keras-OCR. The OCR outputs “I wish I could fly. I wish I could walk. I wish I could swim.”

de-texting with Keras-OCR

Before feeding the collected images to the network, we need to first remove the texts that often appear in memes. This is because if you feed directly untreated images, image-to-text networks such as BLIP2 would largely only pay attention to the texts in the image. Thus, we attempted to use Keras-OCR to first remove the text off the memes, and perform the task of text recognition and image captioning separately.

Keras-OCR is a Python library that provides an easy-to-use interface for performing OCR on images or scanned documents. It is built on top of Keras, a high-level deep learning framework, which allows for seamless integration with other deep learning models. The library combines pre-trained models with efficient image processing techniques to extract text from images accurately.

Keras-OCR leverages the power of convolutional neural networks (CNNs) to tackle text recognition tasks. The architecture of CNNs consists of multiple layers of interconnected convolutional filters that perform operations such as convolution, pooling, and non-linear activation. These layers allow the network to automatically learn and extract hierarchical features, starting from basic visual elements like edges and textures, and gradually progressing to more complex representations of characters and text regions. Thus, Keras-OCR can effectively capture and recognize the intricate patterns and variations present in characters, enabling accurate and robust text recognition in a variety of scenarios.

Image-to-Text with BLIP2

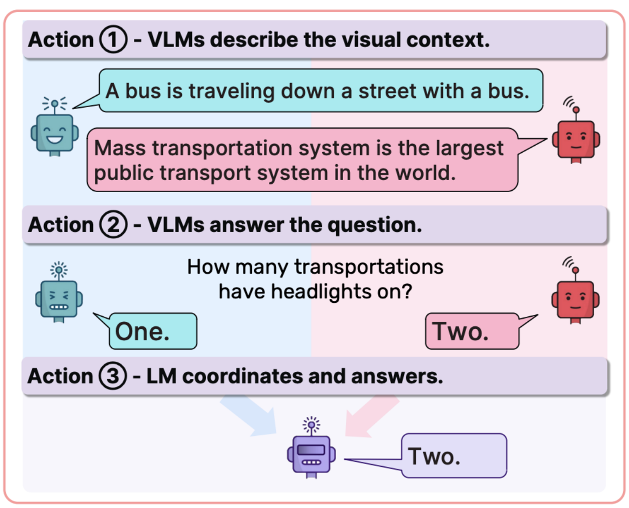
After de-texting, we need to extract key information into texts. This step is crucial in our project because it enforces the model to learn what parts in the meme are amusing and what parts in the figure are less important.

We used BLIP-2, a pre-training strategy for vision-and-language models that aims to reduce the computation cost associated with end-to-end training of large-scale models. BLIP-2 leverages off-the-shelf frozen pre-trained image encoders and large language models (LLMs) to bootstrap vision-language pre-training in a computationally efficient manner. It is computationally efficient compared to existing methods and can perform zero-shot image-to-text generation. By using frozen unimodal models and a lightweight Querying Transformer, BLIP-2 offers a more efficient approach to vision-language pre-training.

We proposed an outline of questions to query the model, including "What is the main character of the meme?", "What is the artistic style?", "What does the background contain?"... With specific questions as such, we are able to ask BLIP2 to output specific and accurate information about the meme, conveying as much information as possible.

Since BLIP2 is quite large, we perform inference on a Nvidia A6000 GPU.

Synthesizing Captions by a Large Language Model



Now that we are able to use BLIP2 to answer various questions regarding the meme, we want to automate the process of bringing all these answers together, forming one line of text that can be inputted to the text-to-video generation model. We took inspiration from the recent paper *Language Models are Visual Reasoning Coordinators*, where the authors proposed that a large language model like ChatGPT is capable of combining the outputs from multiple expert models and generating a single conclusion. This is illustrated by the figure to the left.

The idea here is that we feed a set of questions Q = {q1, q2, … qn} to BLIP2 and gather the responses R = {r1, r2, …, rn}. We then directly feed zip(Q, R) to a large language model and ask it to output two things: 1) a detailed description of the meme and 2) an exact transcription of the text that appears on the meme image, if any. In all of our experiments, we use n = 5 and OpenAI GPT-4 as the large language model.

In the Evaluation section, we have included a detailed example to walk through this process.

Fine-tuning for Text to Video Generation

We based our model on the open-source [Text-to-Video transformer of Damo-ViLab](#).

The text-to-video generation diffusion model consists of three sub-networks: text feature extraction, text feature-to-video latent space diffusion model, and video latent space to video visual space. The model is based on the Video Diffusion Model proposed by Jonathan Ho's team. The overall model parameters are about 1.7 billion. This model adopts the Unet3D structure, and realizes the function of video generation through the iterative denoising process from the pure Gaussian noise video.

We made use of the LoRA finetuning framework and fine-tuned the model with our dataset of video meme data and corresponding generated captions. With the convenient dimension reduction and layer freezing techniques, we are able to modify the baseline model and perform experiments with the use of reasonable time and resources.

The results were not satisfying with our previous attempts of BLIP summary, and original texted meme captioning. We were able to improve our results by improving our captioning methods. The comparison of the results with the primitive model, and various method of captioning is listed in the evaluation section.

During experimentation, we also encountered the problem that videos are generated with watermarks around the bottom section. After some literature review, we found out that this is largely due to consistent watermarks on the training set of the pretrained baseline model. We were not able to afford retraining the entire model with large datasets, so we attempted to employ the lama-video-watermark-remover. The quality of generated video improved from this attempt, but unfortunately some portions of the watermark still remain.

Key Aspects of AI

Based on all of our discussions so far, we may now identify the key aspects of AI in this project:

- Multimodal Learning: We used models that bridge modalities (image-to-text and text-to-video), such as BLIP2 and VideoFusion.
- Generative Modeling & Finetuning for Domain Generalization: We fine-tuned a generative model that creatively builds up videos.
- Computer Vision: We used multiple classic computer vision tools, such as the de-text feature in Keras-OCR.
- LLM In-Context Learning: This is an area of research that's especially popular this year. We leveraged GPT-4's summarization ability by designing prompts such that it learns about the meme image in context.

Project Evaluation

To illustrate the entire process with more clarity, we use the example of the following Winnie the Pooh meme throughout our evaluation section.

The example meme image:

Keras-OCR

The OCR module detects the text within the image and performs text removal to facilitate further image captioning. For example, the Winnie the Pooh meme produces this OCR result:

The knowledge of road laws leaving my body as I purchase a bmw.

The de-texted meme looks like this. We can see that text removal was not perfect since some trace of text was left on the image. However, it is adequate to help BLIP2 to avoid the trap of recognizing texts.

We evaluated the OCR performance with two methods:

1. Manual sampling

   We assessed the OCR performance by randomly sampling 10% of our training data and visually evaluating the quality of text removal. This allowed us to analyze the accuracy and precision of the OCR system and identify areas for improvement. Visual evaluation serves as a quick and accurate evaluation method in the iterative development process of this module.

2. Testing with BLIP2

   During the final stages of developing the OCR module, we conducted a comprehensive evaluation using BLIP2. To assess the performance, we provided BLIP2 with de-texted images and requested it to recognize any text present. Remarkably, BLIP2 returned the response 'No text' for more than 95% of the de-texted training dataset. This result highlights the efficiency of the OCR module in effectively removing text from the images, confirming its robustness and accuracy.

BLIP2

We used BLIP2 to generate captions for the images with the following question prompts. We targeted a specific aspect of the image with each question to ensure information is retained as much as possible. The questions fed to BLIP2 and the corresponding answers for the Winnie the Pooh meme is listed here:

Who is the main character? Describe in detail.	Pooh.
What is the main character doing? Describe with complete sentence.	Pooh is sitting on the floor.
What is the color of the scene?	The scene is red
What is in the background? Describe with complete sentence.	The background is a room with a bed, a table, a chair, a lamp, a dresser, a wardrobe.
What is the style of drawing? Describe with phrases like cartoon style, realistic style, fantasy, and so on.	cartoon style
What is the emotion of the meme? Describe with phrases like, happy vibes, sad emotion, angry emotion, sarcastic humor, and so on.	happy

Here are some more examples of our image captioning results:

					
Character	me	kenshin	Kanye West	The main character is a boy who is playing badminton	Marceline
Action	driving	He is practicing kendo	He's watching the game	He is playing badminton	He's sitting on a chair
Color	The color of the scene is grey	The scene is blue	The color of the scene is red	red	blue
Background	A man in a suit and tie driving a car	A woman in a blue dress is practicing karate	The Cleveland Cavaliers	A badminton court	A chair
Style	Anime	The style of drawing is called "kabuki"	Kanyes	Badminton	Cartoon
Emotion	Fear	the emotion of the meme is anger	Kanyes	happiness	Sad

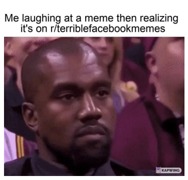
Large Language Model

We experimented with GPT-3.5-turbo (ChatGPT) and the new GPT-4 for our summary task, and found that GPT-4 yields better results. Thus, this is what we used for our model training and all further experiments.

The following is the summary generated by GPT-4 for the Winnie the Pooh meme.

The meme features a cartoon-style Pooh sitting on the floor in a red-colored room. The background includes a bed, table, chair, lamp, dresser, and multiple wardrobes. The overall emotion of the meme is happy.

The caption is of good quality and is accurate for the scene. The caption describes everything accurately except for the emotion. This is used as the input for our finetuning process and for inference of new test images. Here are some more examples of summarized caption according to the Q&A responses on the previous page.

	*Anime style meme, main character driving, grey scene, background features a man in a suit and tie driving a car, emotion of fear, text implying the main character is a terrible person.*
	*Main character Kenshin, practicing kendo, blue scene, woman in blue dress practicing karate in background, kabuki drawing style, emotion of anger*
	*Kanye West, red scene, watching the game, Cleveland Cavaliers in the background, Kanyes style drawing, Kanyes emotion.*
	*A happy boy playing badminton on a red badminton court in a badminton style drawing.*
	*Marceline, cartoon style, sitting on a chair, blue scene, sad emotion, text inside image unclear, chair in the background.*

A surprising side result

Interestingly, when we are designing the questions to ask the model, we discovered its ability to understand the deeper meaning behind the memes. Some of the experimented questions are: “Why is this meme funny?”, “Explain the meaning of this meme.” and so on. After synthesis (as described in the section below), we found that our model is able to give a highly accurate explanation to the reference and context of a provided meme.

For example, these questions asked to the Pooh meme above are synthesized to:

The meme is funny because it plays on the stereotype of BMW drivers and uses the innocent and lovable character of Winnie the Pooh to illustrate the change in attitude. The contrast between the two images of Winnie the Pooh adds to the humor, as the character is typically associated with kindness and humility, making the smug, tuxedo-wearing version an amusing and unexpected twist.

The description accurately explains the punchline and the context of the meme (with slight distortion like hallucinating a tuxedo). Some more examples are listed below:

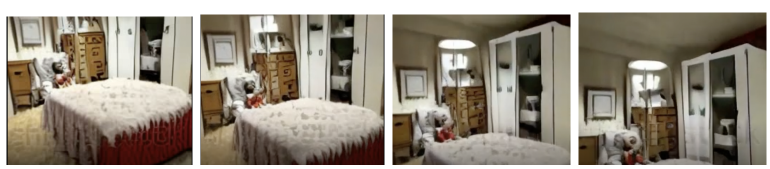
	“The meme is about the unexpected and humorous situation that occurs when the main villain in a movie dies, but there is still an hour left until the end of the movie. This creates a sense of confusion and anticipation for the audience, as they wonder what could possibly happen next since the primary antagonist has already been defeated. The meme likely features an image or character from a movie or pop culture that captures this feeling of surprise and uncertainty, making it relatable and amusing for those who have experienced similar situations in movies they've watched.”
	“The meme features the character Wario, known from the Mario video game series. The text inside the image reads, "Every repost Wario slaps a gacha kid to death." The humor in this meme comes from the idea that each time the meme is reposted, Wario will slap a gacha kid, which refers to a fan of Gacha Life, a popular game and animation app. The meme is meant to be shared and reposted, and the funny aspect is that with each repost, Wario continues to slap gacha kids. The image likely shows Wario in a slapping motion, adding a visual element to the humor.”

Text-to-Video Generation

Finally, we feed the texts given by GPT-4 into the text-to-video generation model finetuned from VideoFusion. Overall, our finetuned model seems to render more dynamic action and exaggerated camera motion, affected by the quality of gif memes. Also, we found that OCR recognizing text in the original memes negatively affects the generation process. We suspect that this is because of the vagueness and unpredictability of the analogies in the meme text.

In addition, we also attempted to do some prompt tuning to improve the performance of our model. We appended some magic phrases to the caption of each input image, including “high quality video”, “smooth action”, “exaggerated expressions”. We also experimented with providing video development hints for several selected images, such as, “camera moves up”, “main character gradually smiles”, …

Here we include a demo of ablation experiments with the aforementioned methods. We evaluated the quality of these generations both by our own visual evaluation and also by an oral questionnaire asking which generation is the most accurate. More than 70% of responses indicates that “Finetuned model w/ prompt” is the most accurate.

<table>
<tr><td>Original</td><td>
</td></tr>
<tr><td>Baseline model with summary + meme text</td><td></td></tr>
<tr><td>Baseline model with summary</td><td></td></tr>
<tr><td>Finetuned model with summary</td><td></td></tr>
<tr><td>Finetuned model with prompt tuning</td><td></td></tr>
</table>

References

Papers

1. Singer, Uriel, et al. "Make-a-video: Text-to-video generation without text-video data." arXiv preprint arXiv:2209.14792 (2022)
2. Ho, Jonathan, et al. "Video diffusion models." arXiv preprint arXiv:2204.03458 (2022)
3. Luo, Zhengxiong, et al. "VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation." arXiv preprint arXiv:2303.08320 (2023)
4. Chen, Liangyu, et al. "Language Models are Visual Reasoning Coordinators." ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models. (2023)
5. Hu, Yaosi, Chong Luo, and Zhenzhong Chen. "Make it move: Controllable image-to-video generation with text descriptions." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022
6. Yu, Lijun, et al. "MAGVIT: Masked Generative Video Transformer." arXiv preprint arXiv:2212.05199 (2022)
7. OpenAI. "GPT-4 Technical Report". arXiv preprint arXiv:2303.08774 (2023)

Softwares

1. Python
2. PyTorch
3. Tensorflow & Keras
4. HuggingFace Transformers
5. OpenAI ChatCompletion API
6. ffmpeg

Source of Data

1. Reddit