UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Introduction

Introduction

1. What is Video Data?

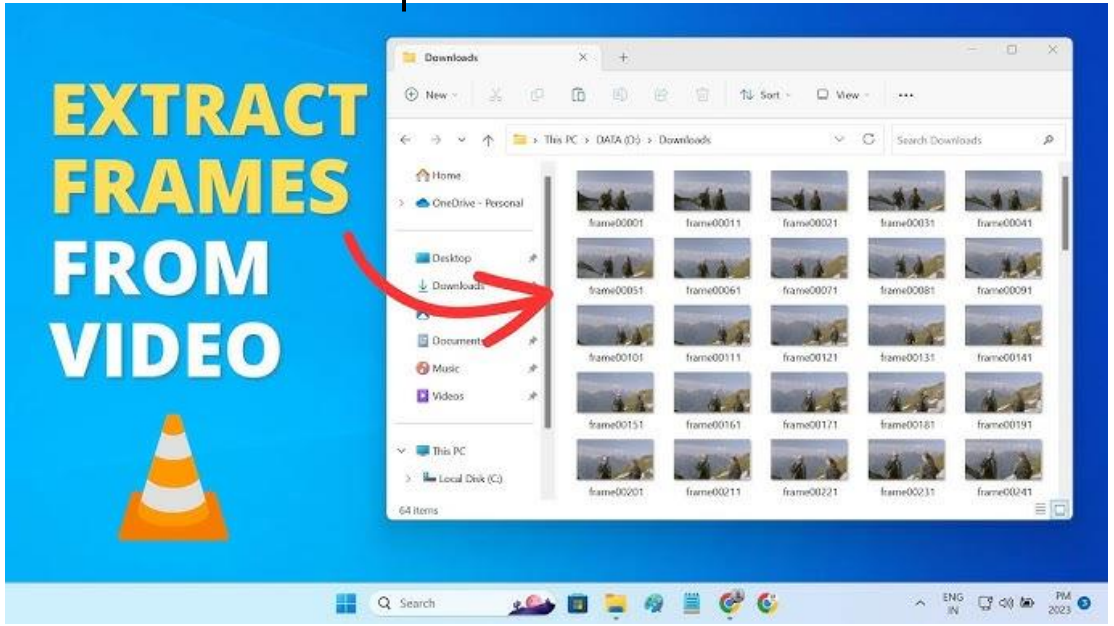
Video data refers to a sequence of images (frames) displayed in rapid succession, typically accompanied by audio, to create the perception of motion. Each frame in a video is essentially a still image, and when played at a certain frame rate (e.g., 30 frames per second), it produces a continuous visual experience.

Introduction

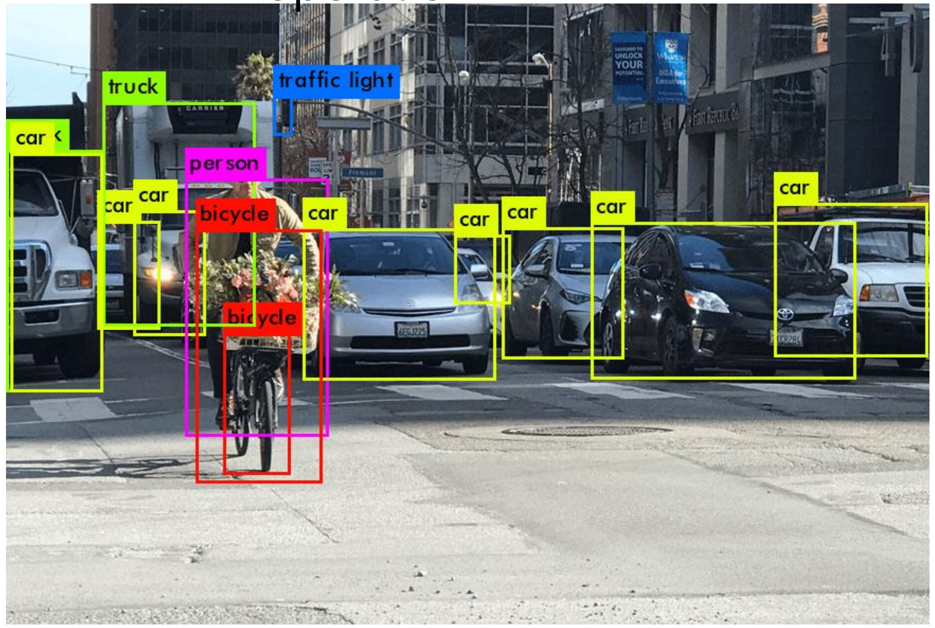
2. What is Video Data Processing?

Video Data Processing refers to the techniques and methods used to analyze, enhance, manipulate, and extract information from video data. This involves operations ranging from basic ones, such as frame extraction, compression, and filtering, to more complex methods like object detection, tracking, and action recognition.

Basic operation



Complicated operation



Introduction

3. Why is Video Data Processing Needed?

Video Data Processing helps transform raw video into useful information, serving various fields.

- Feature Extraction: Helps extract key information like motion patterns, object positions, and scene changes.
- Object Detection & Tracking: Enables applications in surveillance, autonomous vehicles, and sports analytics.
- Action Recognition: Identifies human activities for security, healthcare, and human-computer interaction.
- Video Enhancement: Improves quality through stabilization, noise reduction, and contrast adjustments.
- ...

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Challenges

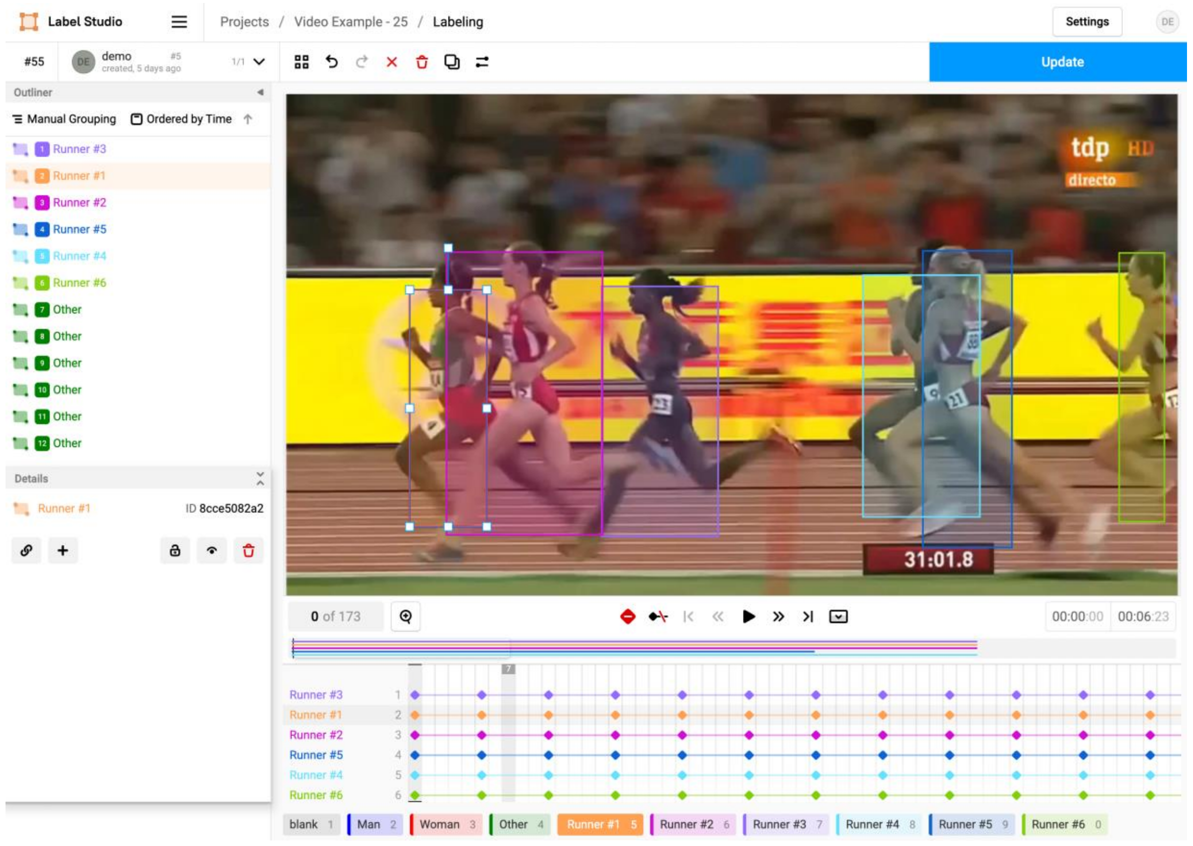
UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Challenges

1. Data Collection and Annotation Time

As presented, video data consists of consecutive frames, making the complexity of collecting and annotating video data significantly higher than that of image data. The challenges scale up several times due to the temporal nature of videos. For example:

- In object classification: Annotation is needed for only a single image per data point.
- In video classification: Annotators must watch the entire video and determine its overall content, making the process much more time-consuming and demanding.

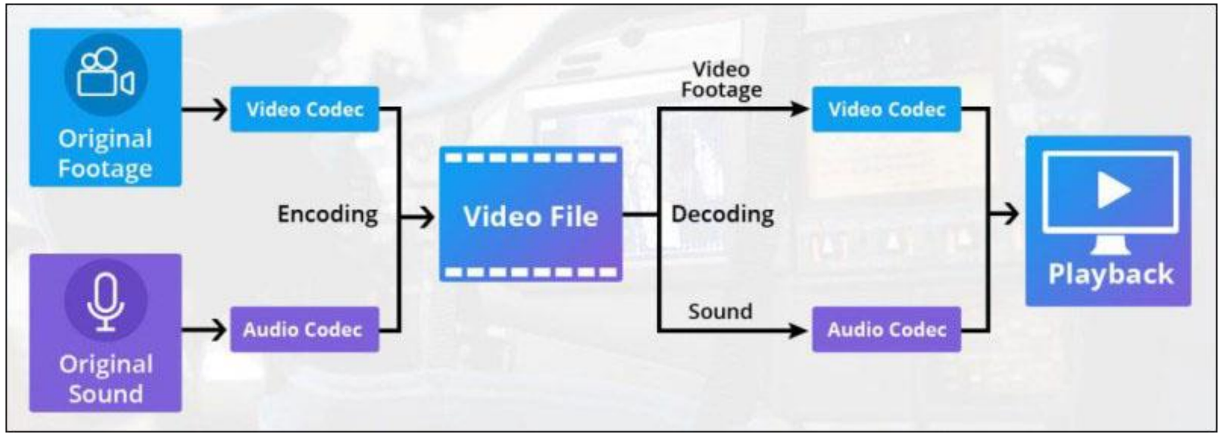


UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Challenges

2. Storage Space

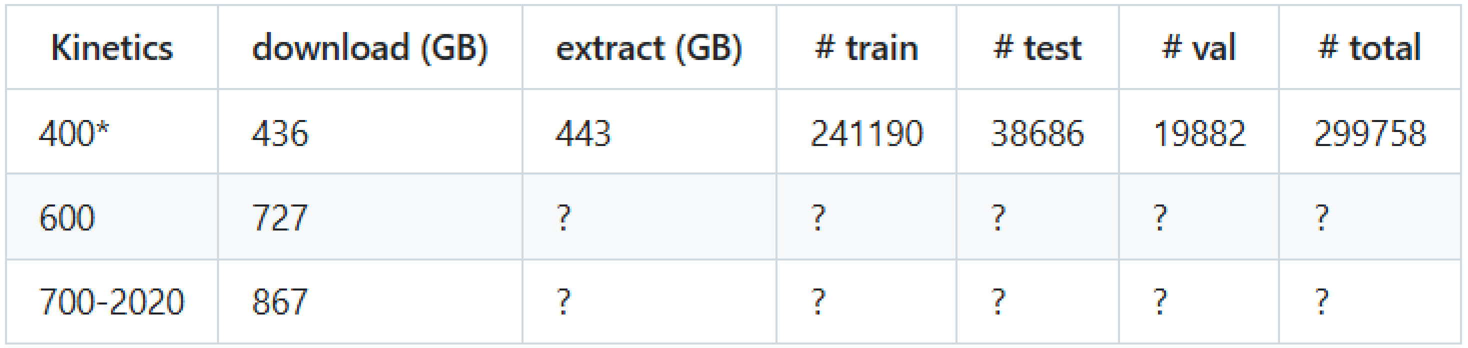
- Instead of storing a single image for each data point, we now need to store information for an entire sequence of video frames. This can range from hundreds to thousands or even hundreds of thousands of frames per data point.
- Although videos are compressed using codecs before storage, the required space is still significantly larger than that of image data (with the same number of data points).



Challenges

2. Storage Space

- Source: github comment



Kinetics	download (GB)	extract (GB)	# train	# test	# val	# total
400*	436	443	241190	38686	19882	299758
600	727	?	?	?	?	?
700-2020	867	?	?	?	?	?

- Source: Wikipedia

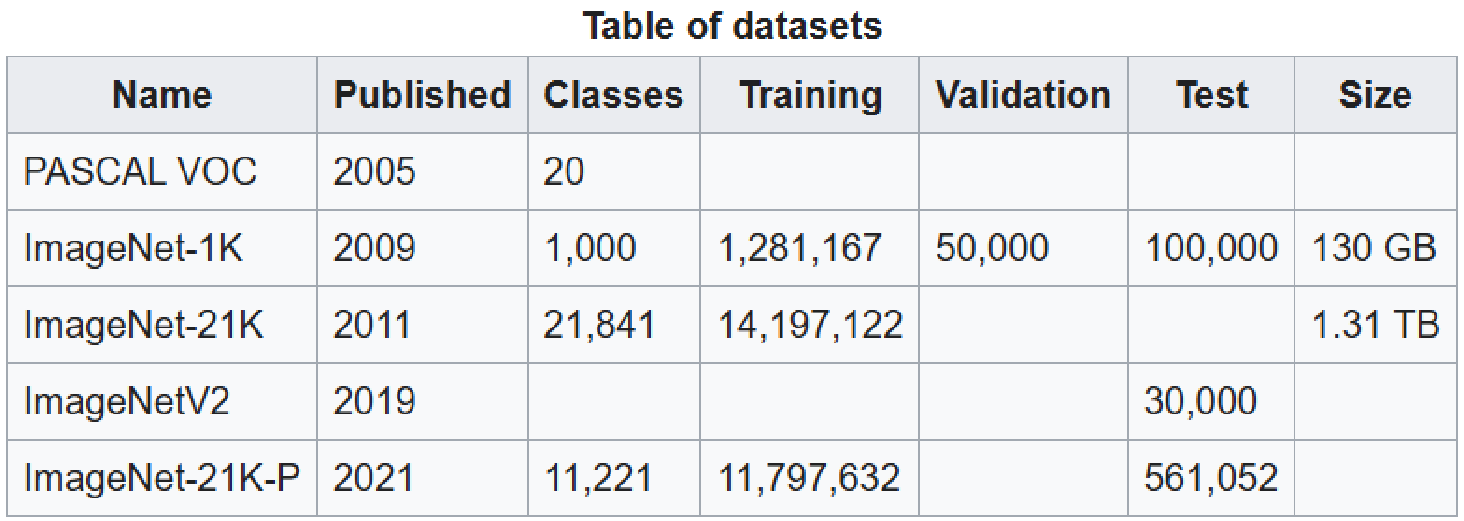


Table of datasets

Name	Published	Classes	Training	Validation	Test	Size
PASCAL VOC	2005	20				
ImageNet-1K	2009	1,000	1,281,167	50,000	100,000	130 GB
ImageNet-21K	2011	21,841	14,197,122			1.31 TB
ImageNetV2	2019				30,000	
ImageNet-21K-P	2021	11,221	11,797,632		561,052	

Challenges

3. Limited Availability of High-Quality Public Datasets

- Given the mentioned challenges in constructing video data, publicly available video datasets are rarely comparable in scale to image or text datasets.

- That's why we need to augment data for videos (which will be introduced in the next section).

ImageNet contains ~ 14M images

Wiki, Reddit, StackOverflow …

Kinectic-700 with only ~ 650k video

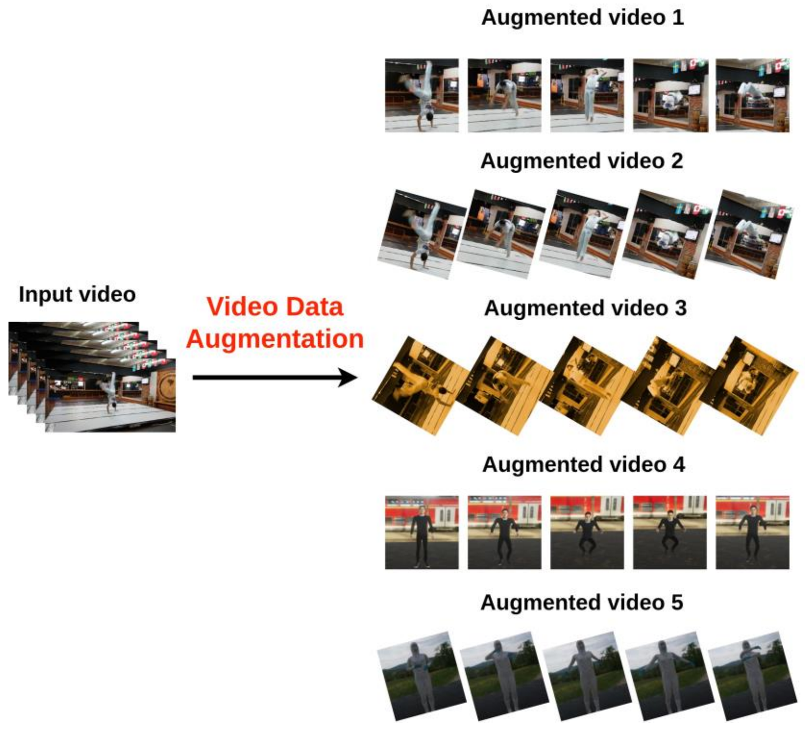
UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

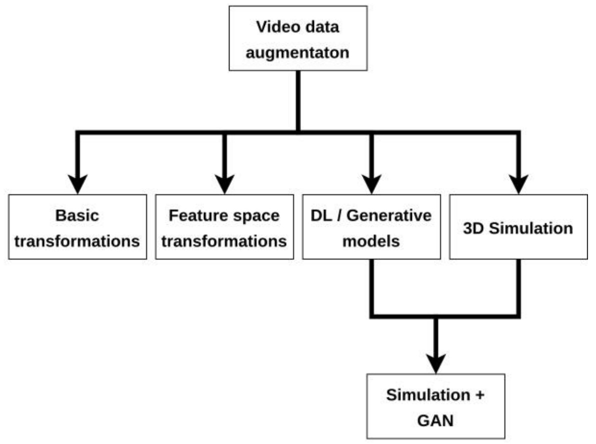
Introduction to Video Data Augmentation

Introduction to Video Data Augmentation

1. What is Video Data Augmentation?

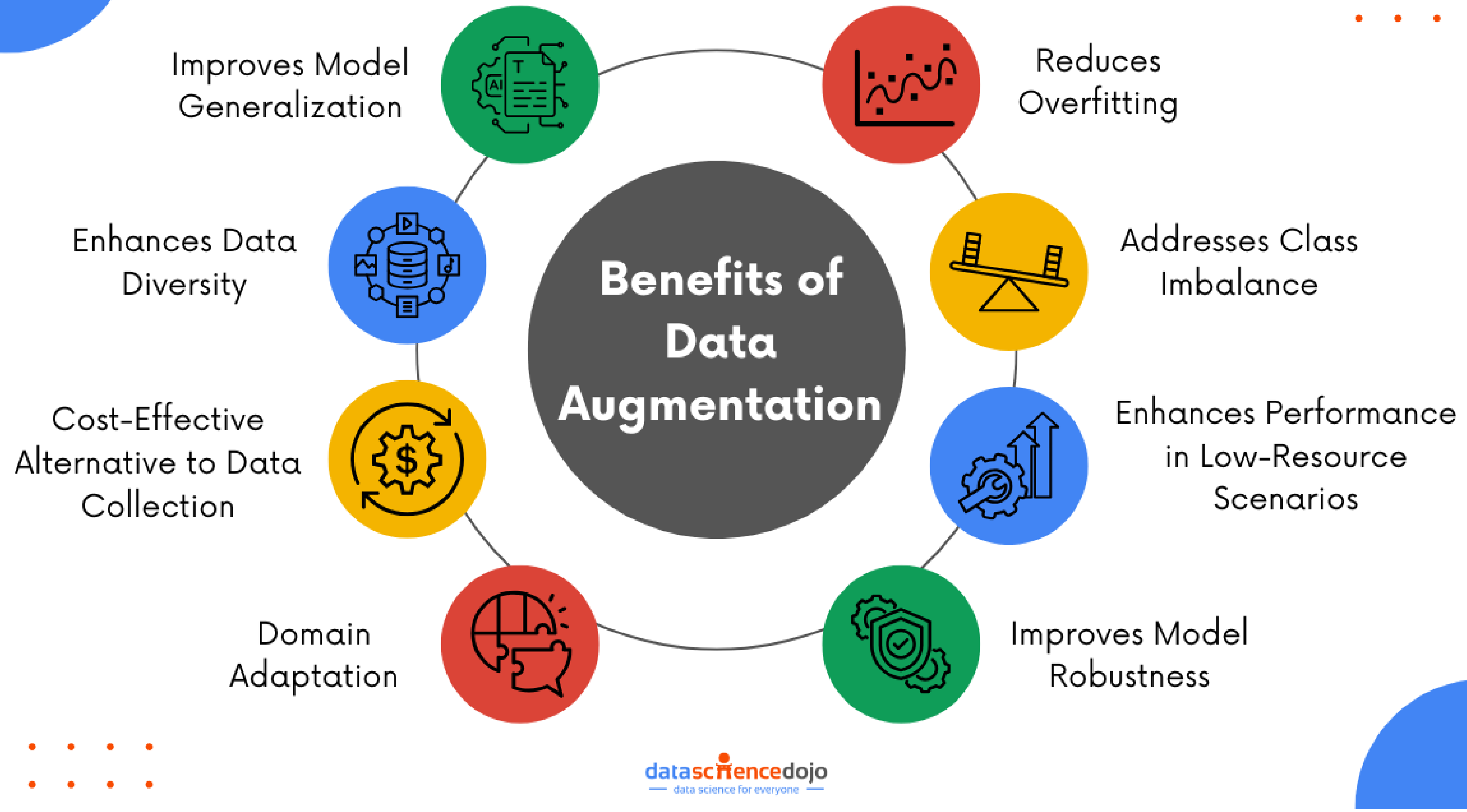
- Video Data Augmentation refers to the process of applying transformations or modifications to video data to create variations of the original dataset..
- Unlike image augmentation, video augmentation must consider temporal consistency, ensuring that transformations are applied smoothly across frames to maintain coherence.





Introduction to Video Data Augmentation

2. Why is Video Data Augmentation Needed?


Benefits of Data Augmentation
Improves Model Generalization
Enhances Data Diversity
Cost-Effective Alternative to Data Collection
Domain Adaptation
Reduces Overfitting
Addresses Class Imbalance
Enhances Performance in Low-Resource Scenarios
Improves Model Robustness
datasciencedojo
data science for everyone

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Basic Operations with Video Data

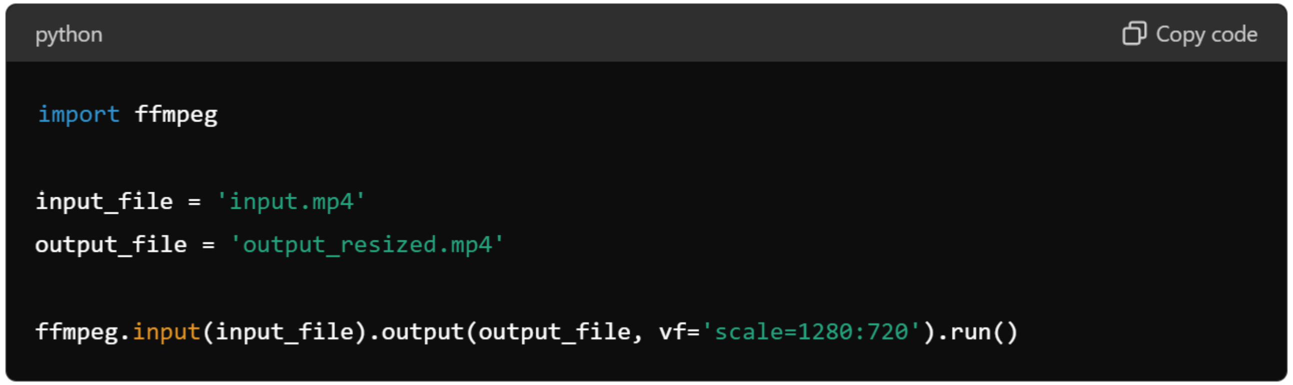
UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Basic Operations with Video Data

1. Tools (ffmpeg, OpenCV…)

- ffmpeg is a powerful open-source multimedia framework used for processing, converting, and streaming audio and video files. It supports a wide range of formats and codecs, making it a versatile tool for video editing, compression, and streaming. It is widely used in video processing applications, media servers, and machine learning pipelines.

- Using ffmpeg with a Bash script seems faster (and more common) than using it directly in Python, but this approach can be somewhat difficult to use.



```python
import ffmpeg

input_file = 'input.mp4'
output_file = 'output_resized.mp4'

ffmpeg.input(input_file).output(output_file, vf='scale=1280:720').run()
```

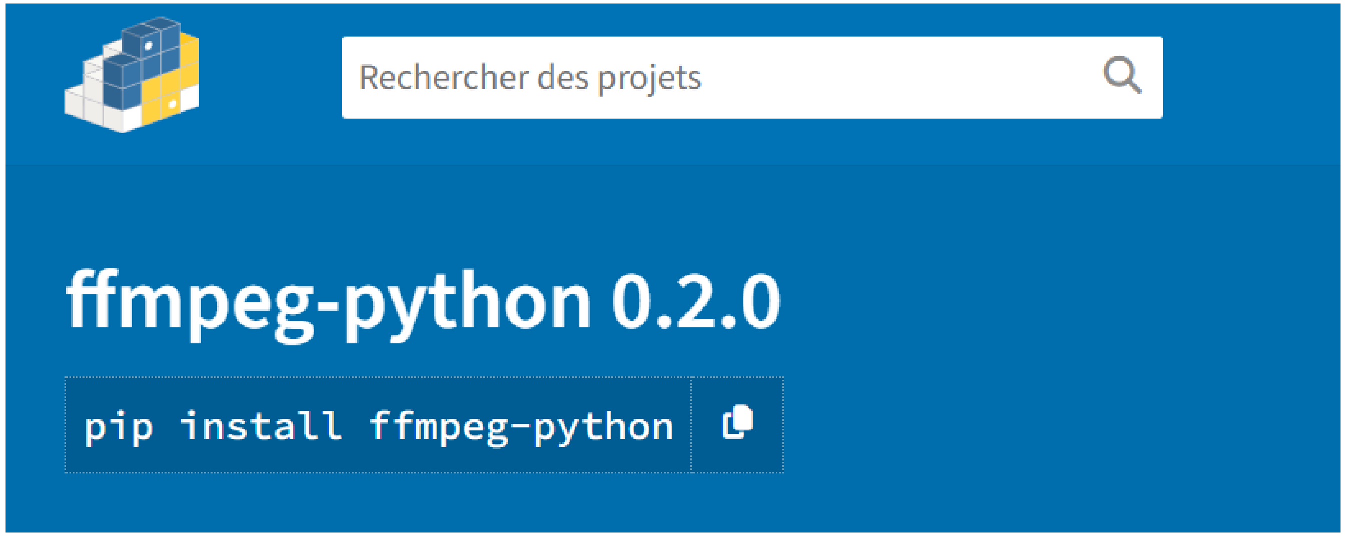
UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Basic Operations with Video Data

1. Tools (ffmpeg, OpenCV…)

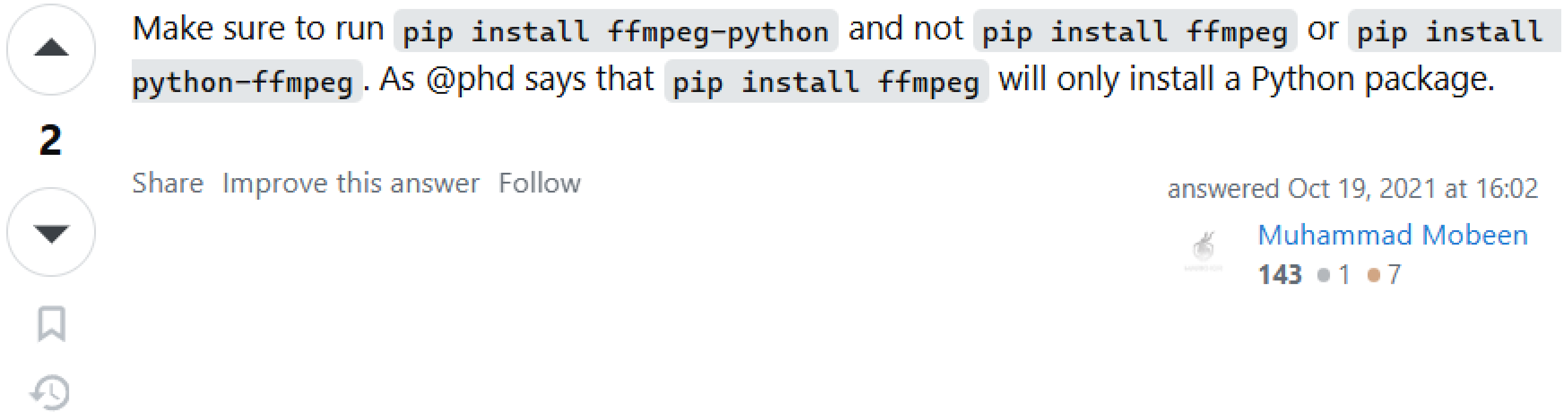
Make sure you installed the correct one

- conda install ffmpeg
- pip install ffmpeg-python


Rechercher des projets

ffmpeg-python 0.2.0

```
pip install ffmpeg-python
```


Make sure to run `pip install ffmpeg-python` and not `pip install ffmpeg` or `pip install python-ffmpeg`. As @phd says that `pip install ffmpeg` will only install a Python package.

2

Share Improve this answer Follow

answered Oct 19, 2021 at 16:02

Muhammad Mobeen
143 ● 1 ● 7

Basic Operations with Video Data

2. Sample Frames

- As mentioned earlier, a video is a sequence of frames. Typically, when datasets are downloaded, they are compressed in formats such as .mp4 or .avi. To facilitate loading data for training models, we first extract individual frames from the video and save them into a folder with corresponding names.

- For example, the code snippet below extracts 55 frames per second from the input video (fps=55). This can be considered Uniform Sampling.

- See more at ffmpeg documentation ... or simply ask ChatGPT :D

```python
import ffmpeg

input_video = "input.mp4"
output_frames = "frames/frame_%04d.jpg"  # frame_0001.jpg, frame_0002.jpg, ...

# method chaining coding
ffmpeg.input(input_video).output(output_frames, vf="fps=55").run()
```

Basic Operations with Video Data

2. Sample Frames

- Additionally, there are many other options when extracting frames with FFmpeg, such as rotating images, resizing, adding blur, and more.
- Again, see more at ffmpeg documentation ... or simply ask ChatGPT :D

```
# method chaining coding
ffmpeg.input(input_video).output(output_frames, vf="fps=55").run()

# change image size to 640x480
ffmpeg.input(input_video).output(output_frames, vf="scale=640:480").run()

# change ratio of frame
ffmpeg.input(input_video).output(output_frames, vf="scale=iw/2:ih/2").run()

# rotate
ffmpeg.input(input_video).output(output_frames, vf="transpose=1").run()

# add blur
ffmpeg.input(input_video).output(output_frames, vf="gblur=sigma=10").run()

# stacking operations
ffmpeg.input(input_video).output(output_frames, vf="scale=640:480,format=gray,fps=30").run()
```

Be careful!

The above operations run **extremely fast**, and depending on your system configuration, they can extract **thousands of frames per second**. And if your PC is lagging, you, of course,have to wait unstill the end of the extracting process!!!!!!!!!!!!!!!!

Therefore, you should test them on a small video segment first to ensure they work as expected before letting them consume your memory!You definitely don't want to end up deleting tens of GBs of data just because of a coding mistake.

Basic Operations with Video Data

3. Basic Preprocessing

Additionally, we can apply some minor preprocessing steps to speed up the data loading process. Some examples include:

- Converting frames into NumPy arrays and saving them.
- Normalizing the size
- Normalizing colors (mean, std)
- Filtering out low-quality videos (low resolution, low FPS, etc.)

Disadvantages of the above approach:

- Additional storage is required to save the new version.
- If the original data is deleted, making modifications or reverting to the initial data will be difficult.

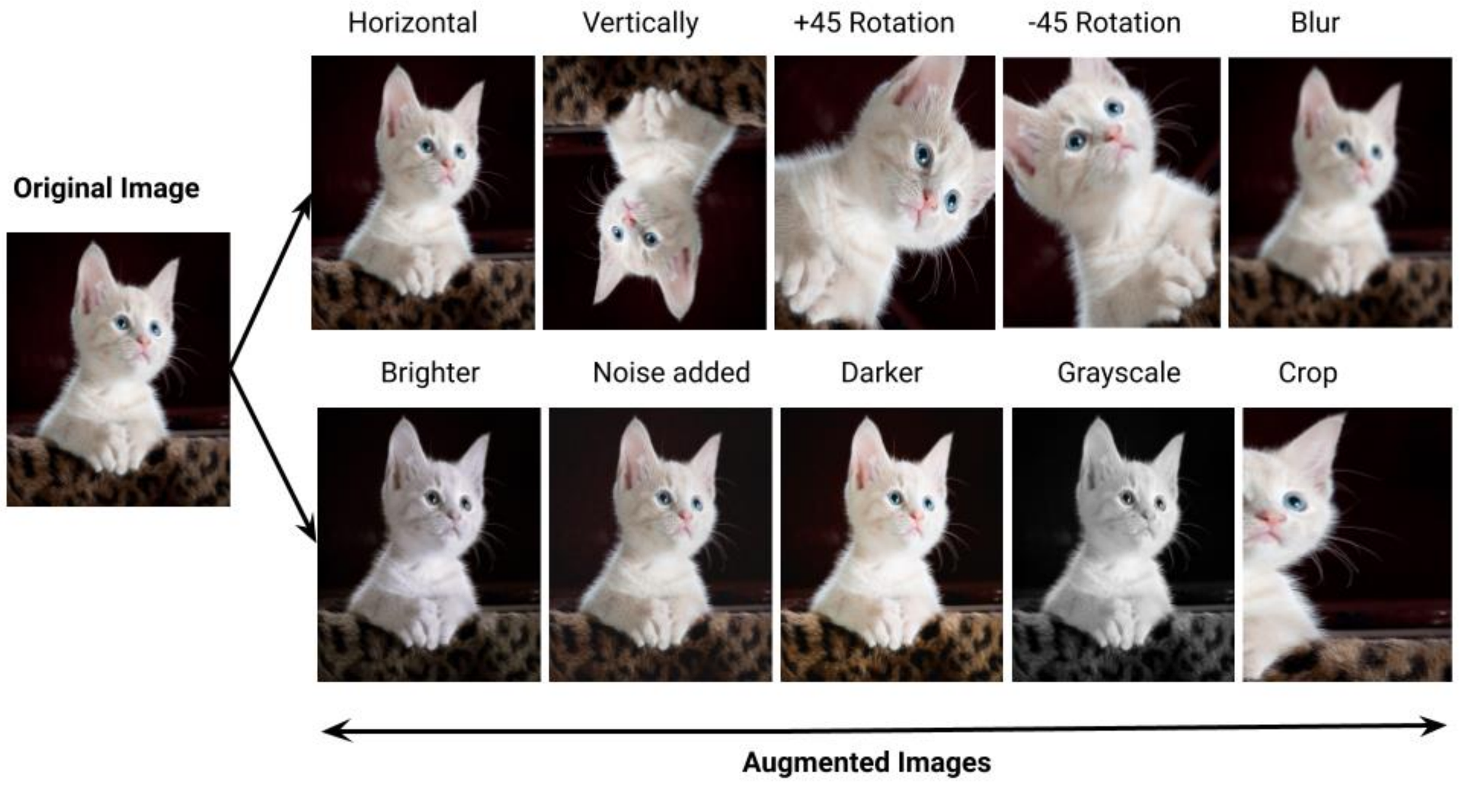
UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Image Data Preprocessing

Image Data Preprocessing

1. Introduction

Data Augmentation is a technique used to generate new data points from existing ones to increase both the quantity and generalization of the data. This technique is highly popular and essential in Computer Vision due to the high cost of collecting and labeling image or video data.



1. Introduction

For image data, there are various types of augmentation that can be categorized into three main groups [1]:

- In-network Augmentation
- Data-level Augmentation
- Data-level Mixing

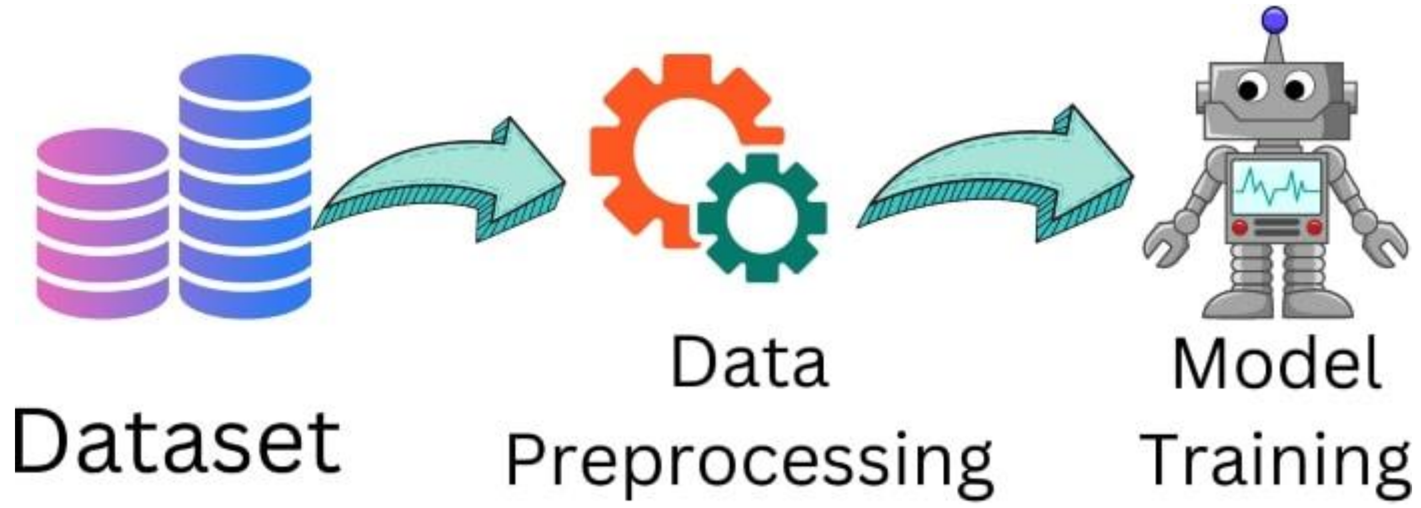


Image data preprocessing

2. In-network Augmentation

In-network Augmentation: This approach applies augmentation directly to the network’ s features during training, which is known for its ability to prevent the model from overfitting. Some common techniques in this category include:

- Dropout: Randomly deactivates activations during training (some implementations also apply it to backpropagation).
- Dropblock: Similar to Dropout, but drops neighboring regions together.
- Shake-Shake: Allows the network to use regularization on residual connections.
- Shake-Drop: An enhancement of Shake-Shake.

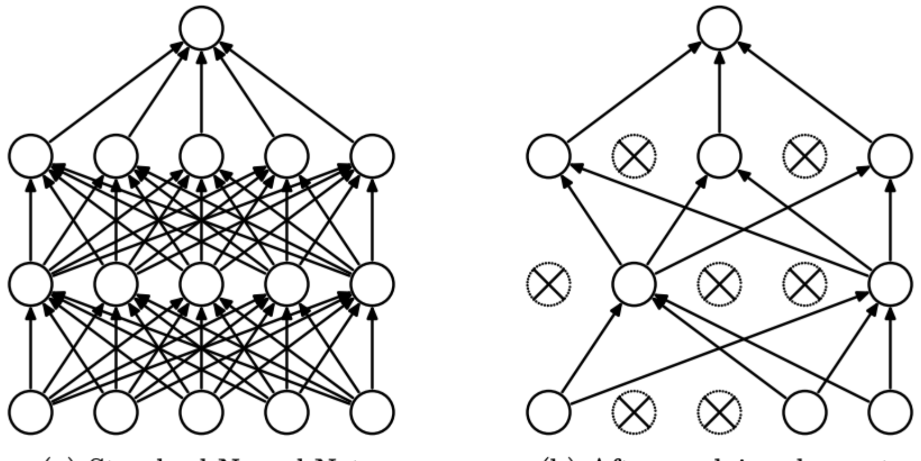


Image data preprocessing

3. Data-level Augmentation

Data-level Augmentation: This method applies augmentation directly to input images, increasing data diversity and generalization, helping the model learn invariant features. This approach can be divided into three subcategories:

- Geometric Transformation: Includes cropping, flipping, rotating, shearing, translating, etc.
- Photometric Transformation: Adjusts brightness, contrast, color, random hue, etc.
- Localizable Features Augmentation: Techniques such as CutOut, Hide-and-See, etc.

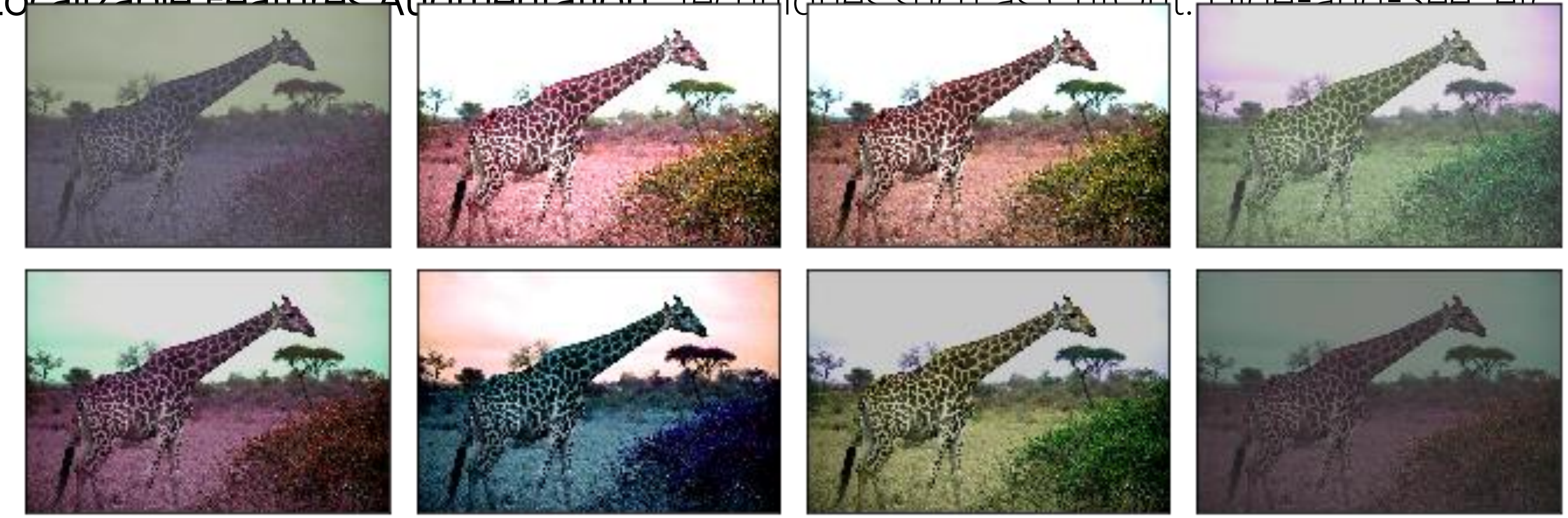
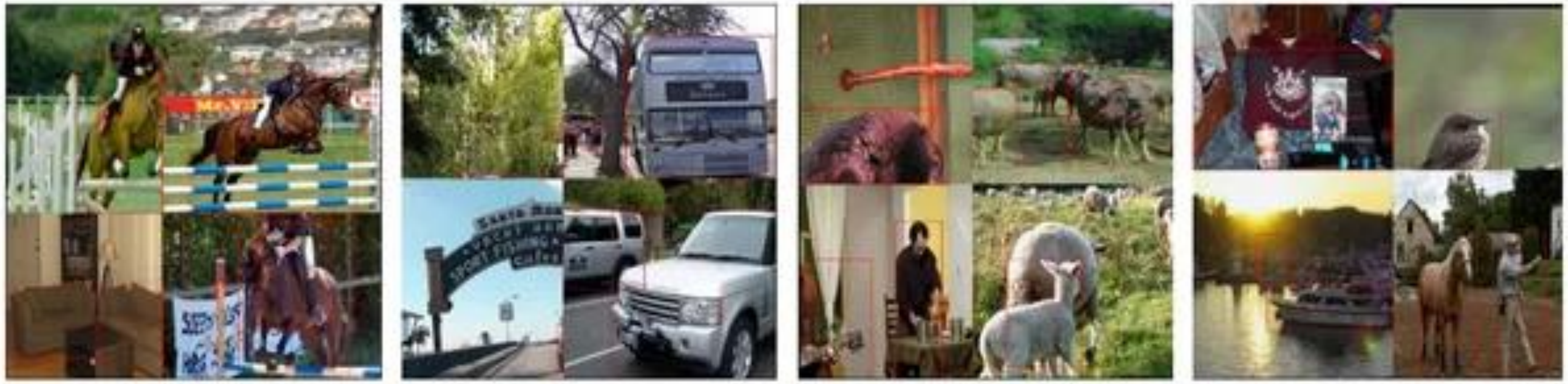


Image data preprocessing

4. Data-level Mixing

Data-level Mixing: This technique combines multiple images to prevent the model from relying too much on context. Some notable variations include:

- CutOut: Removes a portion of the image.
- CutMix: Replaces the removed region with a patch from another image.
- Mosaic: Mixes multiple images together to create a new one.
- CutBlur: Cuts and pastes between a low-resolution and a high-resolution



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Video Data Preprocessing

Video Data Preprocessing

1. Introduction

The reason we mention these image augmentation techniques is that, typically and very commonly, video data augmentation is implemented by extending image augmentation techniques to videos (applied to individual frames). Additionally, many other techniques can be used, which can be categorized as follows:

- Image-based Extended: Similar to image augmentation techniques but now extended to be applied to videos.
- Gen AI: Uses generative video models for augmentation.
- Simulation: Utilizes real-world simulation programs to generate videos.
- Ad-hoc.

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Video Data Preprocessing

2. Image-based Extended

Image-based Extended: There are multiple approaches to applying transformations:

- Applying the same transformation to all frames.
- Applying different transformations to each frame while maintaining a meaningful sequence of transformations.
- Applying completely different transformations to each frame independently.

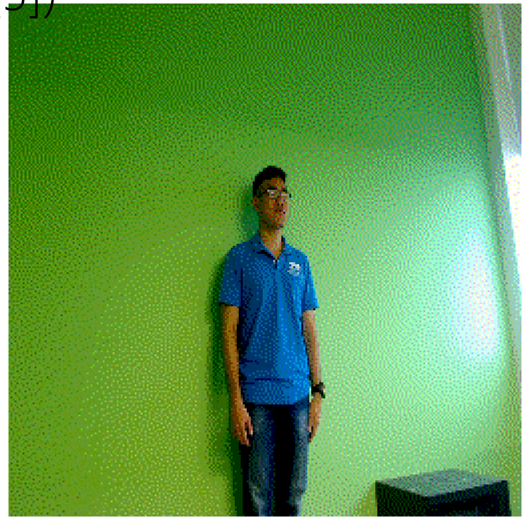
Additionally, one can combine multiple augmentation methods, as mentioned above, to create an augmentation pipeline suitable for each specific problem and design approach [2, 3, 4 ,5].

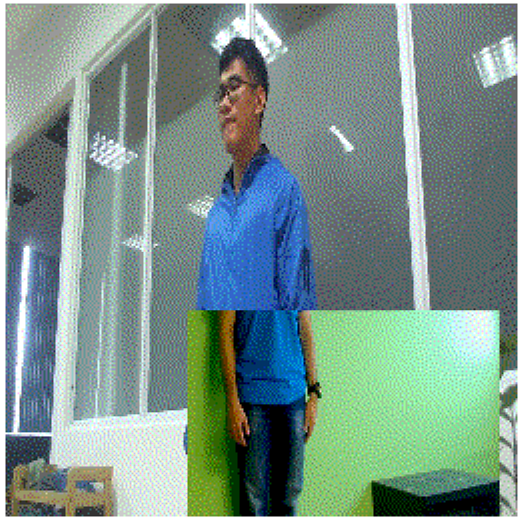
Video Data Preprocessing

2. Image-based Extended

- Examples: Applying the same transformation to all frames.

CutMix on video (see also [5])





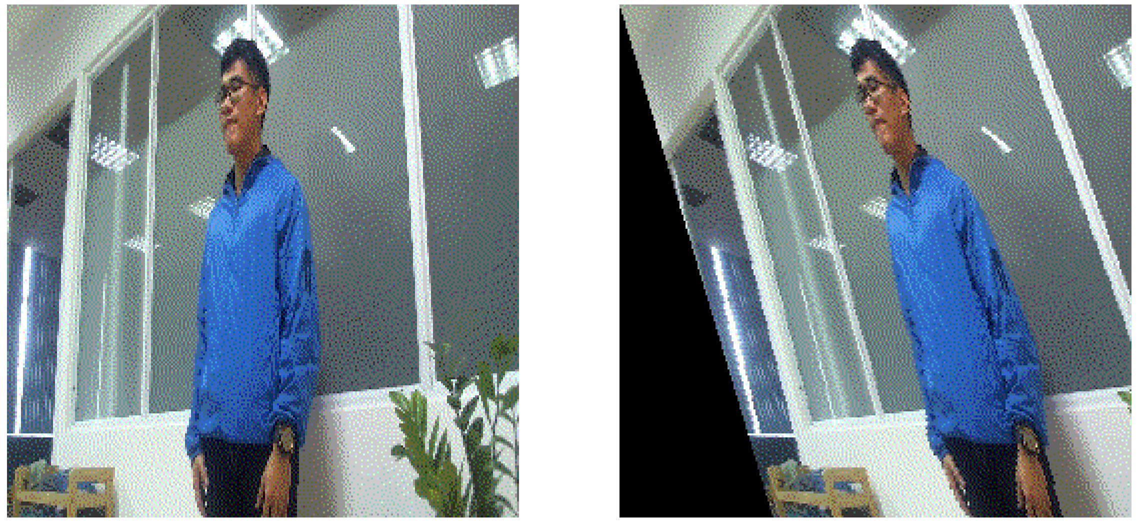
Source : Viblo blog

Video Data Preprocessing

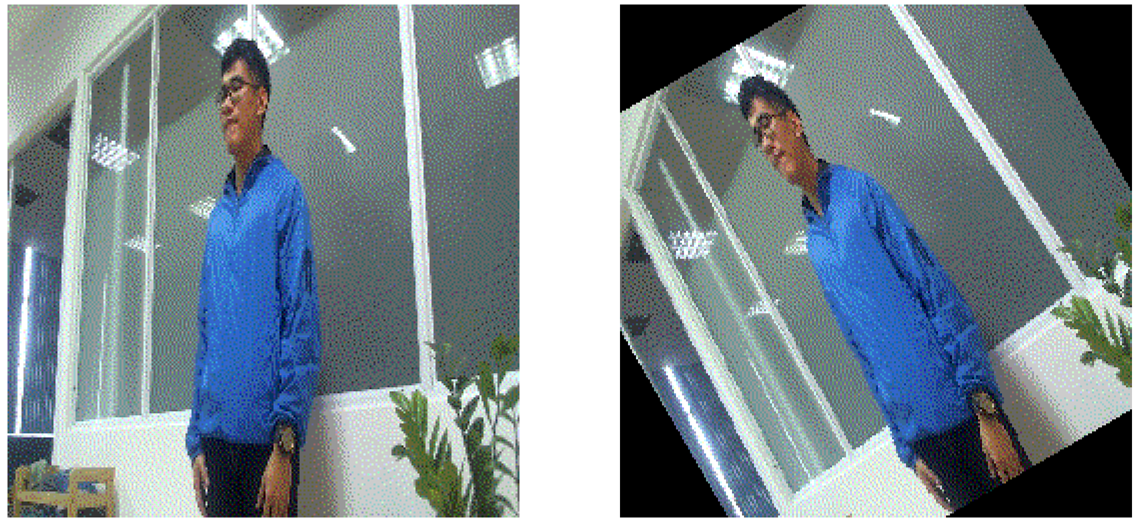
2. Image-based Extended

- Examples: Applying different transformations to each frame while maintaining a meaningful sequence of transformations.

Shear X



Rotate



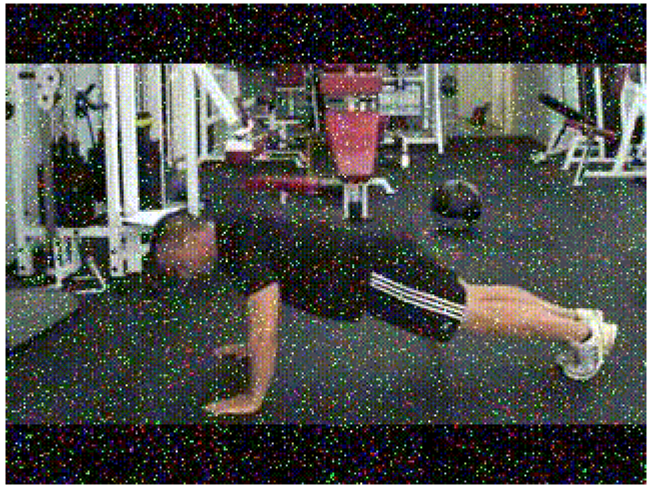
Source : Viblo blog

Video Data Preprocessing

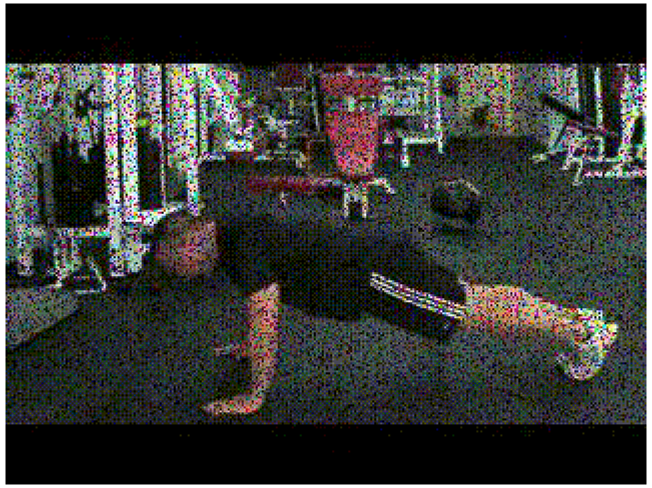
2. Image-based Extended

- Examples: Applying completely different transformations to each frame independently.

Peper



Salt



Source :
https://github.com/okankop/vidaug

Video Data Preprocessing

2. Image-based Extended

Library specialized for video data augmentation:

• Augly: A versatile augmentation library from Meta, supporting image, video, text, and audio transformations like speed changes, overlays, and distortions.
• VideoAug: A specialized video augmentation library offering spatial and temporal transformations such as flipping, cropping, speed variation, and frame dropping.

Library specialized for image data augmentation:

• Albumentations: A fast and flexible image augmentation library optimized for deep learning, providing geometric and color-based transformations with OpenCV acceleration.
• OpenCV: A widely used computer vision library with built-in image processing and augmentation functions, including resizing, blurring, rotations, and color adjustments.
• ImgAug: A powerful library for complex image augmentations, supporting geometric distortions, noise injection, blending effects, and pixel-wise transformations.
• torchvision: The official PyTorch library for image transformations, offering essential augmentations like cropping, flipping, normalization, and color jittering for deep learning.

Video Data Preprocessing

3. Gen AI

- With the development of Gen AI, many studies have applied it to enhance their video data and achieved promising results, notably such as in [7] or [8].

- Recently, the Sora model has impressed with its highly realistic video samples, paving the way for future research to leverage AI-generated videos by seeding them with an initial input image. This can be considered a form of Knowledge Distillation.

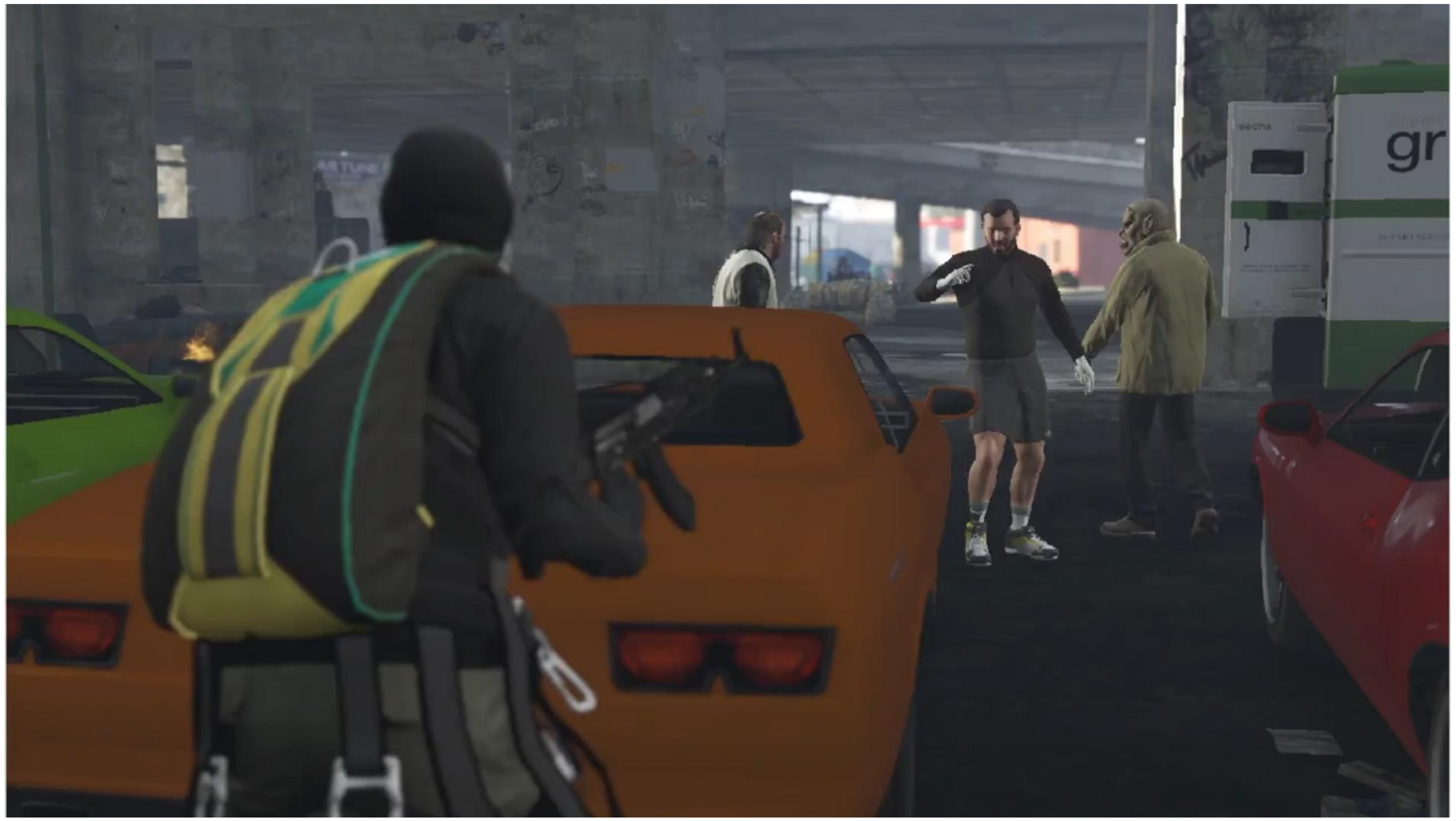
Source : SORA

Video Data Preprocessing

4. Simulation

- Additionally, in a more creative approach, many researchers have leveraged graphic rendering tools such as Unity, Unreal Engine [9, 10], or even games like Grand Theft Auto [11] to create simulated datasets.

- Although the most noticeable drawback of this approach is that the realism is often not high, the advantage is that we can extract a vast amount of information from the environment since it is a simulation—something that is not possible when relying solely on video data.



Source : SAIL-VOS 3D dataset

Video Data Preprocessing

5. Ad-hoc

Video augmentation algorithms that do not fall into the previously mentioned categories can be grouped into an Ad-hoc category. Some commonly used techniques include using Optical Flow to generate an additional feature layer for the network input (along with RGB frames) [6] or simply down-sampling or up-sampling video frames to encourage the model to learn time-invariant features.

Source : Medium blog

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

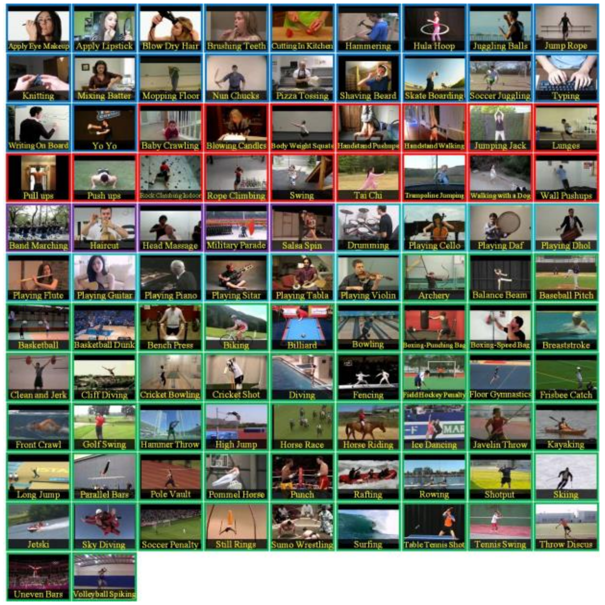
Some Related Problems

Some Related Problems

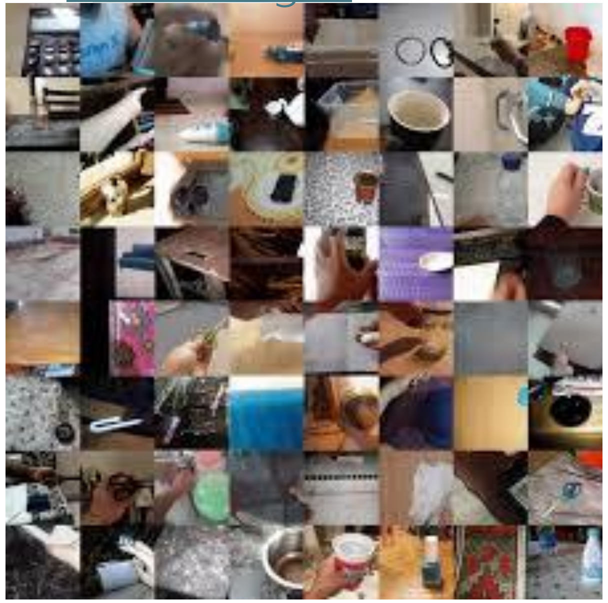
1. Action Recognition

"Action Recognition is a computer vision task that involves recognizing human actions in videos or images. The goal is to classify and categorize the actions being performed in the video or image into a predefined set of action classes." - Paperwithcode

UCF101



Something-Somethingv2



Kinetics

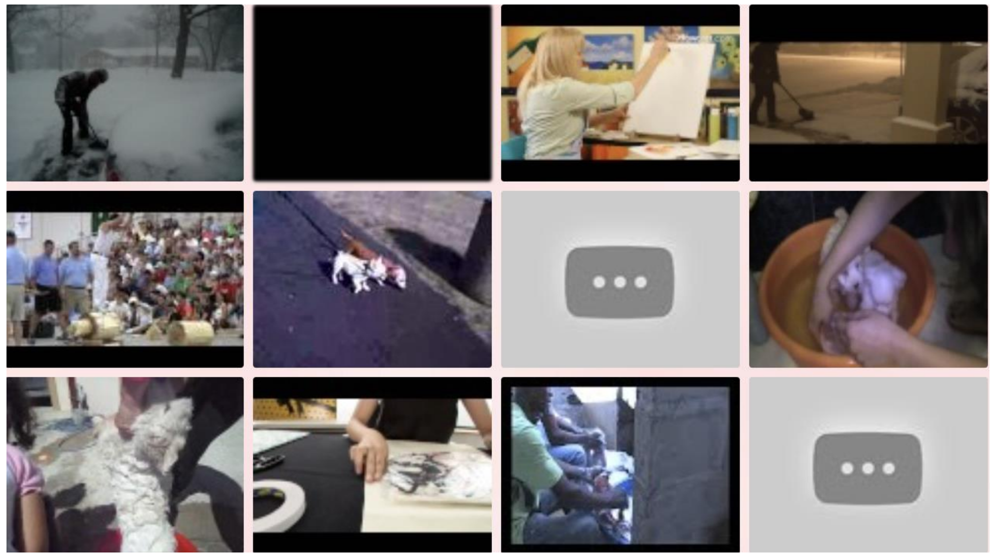
Some Related Problems

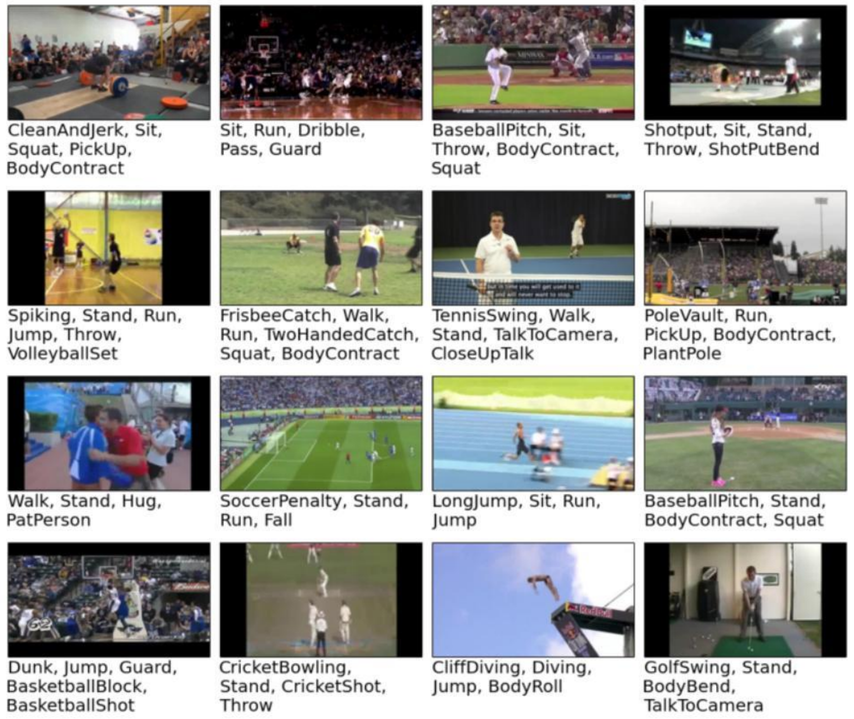
2. Temporal Action Localization

"Temporal Action Localization aims to detect activities in the video stream and output beginning and end timestamps. It is closely related to Temporal Action Proposal Generation." - Paperwithcode

ActivityNet

Multi-THUMOS




CleanAndJerk, Sit, Squat, PickUp, BodyContract
Sit, Run, Dribble, Pass, Guard
BaseballPitch, Sit, Throw, BodyContract, Squat
Shotput, Sit, Stand, Throw, ShotPutBend
Spiking, Stand, Run, Jump, Throw, VolleyballSet
FrisbeeCatch, Walk, Run, TwoHandedCatch, Squat, BodyContract
TennisSwing, Walk, Stand, TalkToCamera, CloseUpTalk
PoleVault, Run, PickUp, BodyContract, PlantPole
Walk, Stand, Hug, PatPerson
SoccerPenalty, Stand, Run, Fall
LongJump, Sit, Run, Jump
BaseballPitch, Stand, BodyContract, Squat
Dunk, Jump, Guard, BasketballBlock, BasketballShot
CricketBowling, Stand, CricketShot, Throw
CliffDiving, Diving, Jump, BodyRoll
GolfSwing, Stand, BodyBend, TalkToCamera

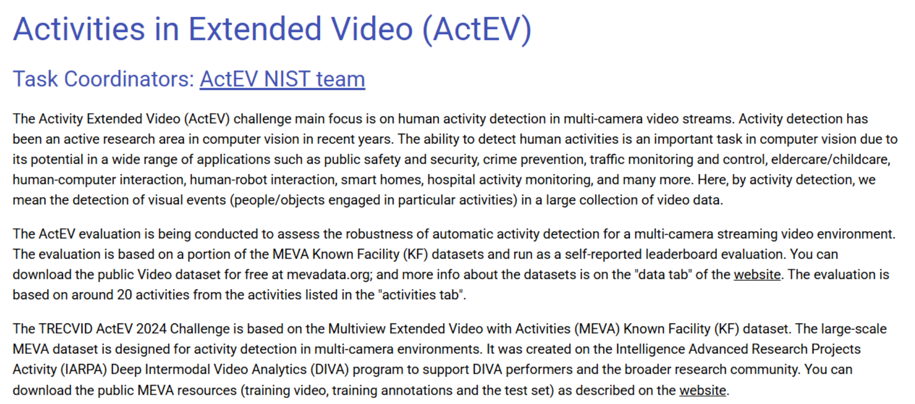
Some Related Problems

3. Human Action Detection and Recognition

"**Action Detection** aims to find both where and when an action occurs within a video clip and classify what the action is taking place. Typically results are given in the form of action tublets, which are action bounding boxes linked across time in the video. This is related to temporal localization, which seeks to identify the start and end frame of an action, and action recognition, which seeks only to classify which action is taking place and typically assumes a trimmed video. - Paperwithcode

AVAv2.2

TRECVID



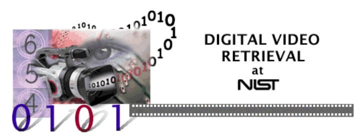
Activities in Extended Video (ActEV)

Task Coordinators: ActEV NIST team

The Activity Extended Video (ActEV) challenge main focus is on human activity detection in multi-camera video streams. Activity detection has been an active research area in computer vision in recent years. The ability to detect human activities is an important task in computer vision due to its potential in a wide range of applications such as public safety and security, crime prevention, traffic monitoring and control, eldercare/childcare, human-computer interaction, human-robot interaction, smart homes, hospital activity monitoring, and many more. Here, by activity detection, we mean the detection of visual events (people/objects engaged in particular activities) in a large collection of video data.

The ActEV evaluation is being conducted to assess the robustness of automatic activity detection for a multi-camera streaming video environment. The evaluation is based on a portion of the MEVA Known Facility (KF) datasets and run as a self-reported leaderboard evaluation. You can download the public Video dataset for free at mevadata.org; and more info about the datasets is on the "data tab" of the website. The evaluation is based on around 20 activities from the activities listed in the "activities tab".

The TRECVID ActEV 2024 Challenge is based on the Multiview Extended Video with Activities (MEVA) Known Facility (KF) dataset. The large-scale MEVA dataset is designed for activity detection in multi-camera environments. It was created on the Intelligence Advanced Research Projects Activity (IARPA) Deep Intermodal Video Analytics (DIVA) program to support DIVA performers and the broader research community. You can download the public MEVA resources (training video, training annotations and the test set) as described on the website.



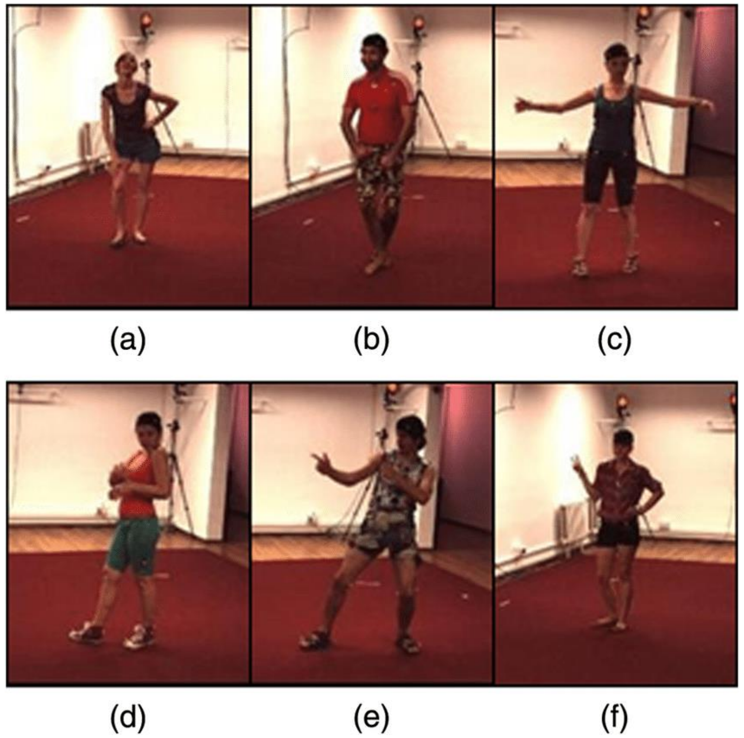
News magazine, science news, news reports, documentaries, educational programming, and archival video

Some Related Problems

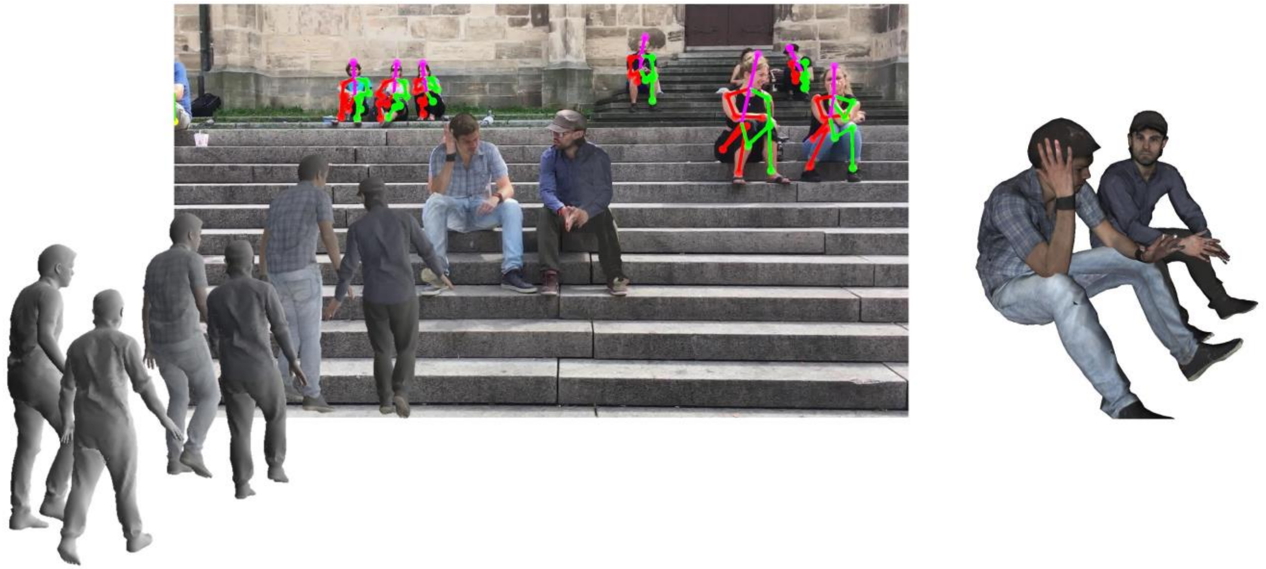
4. 3D Human Pose Estimation

"3D Human Pose Estimation is a computer vision task that involves estimating the 3D positions and orientations of body joints and bones from 2D images or videos. The goal is to reconstruct the 3D pose of a person in real-time, which can be used in a variety of applications, such as virtual reality, human-computer interaction, and motion analysis." - Paperwithcode

Human3.6M



3DPW

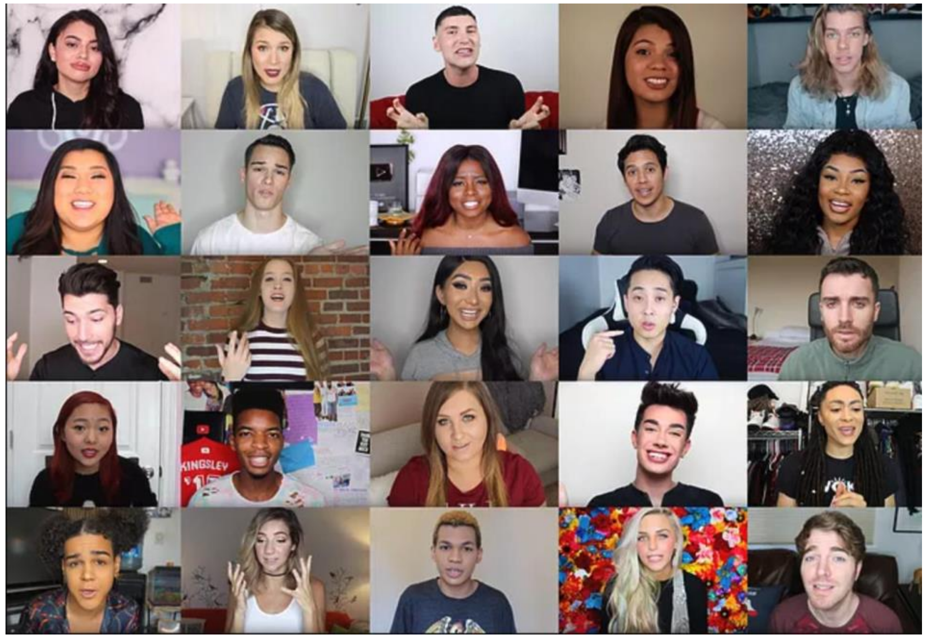


Some Related Problems

5. Multimodal Problems

IEMOCAP: Emotion Recognition in Conversation

CMU-MOSEI: Emotion Recognition in Conversation



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

References

References

[1] N. Cauli, D. Reforgiato Recupero. "Survey on Videos Data Augmentation for Deep Learning Models," in *Future Internet*, vol. 14, no. 3, 2022.

[2] Wang, L., et al. "Three-stream CNNs for action recognition," in *Pattern Recognition Letters*, vol. 92, pp. 33-40, 2017.

[3] Ur Rehman, M., et al. "Dynamic Hand Gesture Recognition Using 3D-CNN and LSTM Networks," in *Computers, Materials & Continua*, vol. 70, pp. 4675-4690, 2022.

[4] Isobe, T., et al, "Intra-Clip Aggregation For Video Person Re-Identification," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 2336-2340.

[5] Sangdoo Yun, undefined., et al. "VideoMix: Rethinking Data Augmentation for Video Classification," in *ArXiv*, vol. abs/2012.03457, 2020.

[6] W. Wang, J. Shen, L. Shao. "Video Salient Object Detection via Fully Convolutional Networks," in *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 38-49, 2018.

References

[7] Wu, D., et al, "Adversarial Action Data Augmentation for Similar Gesture Action Recognition," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1-8.

[8] Zhang, Y., et al, "Self-Paced Video Data Augmentation by Generative Adversarial Networks with Insufficient Samples," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1652–1660.

[9] De Souza, C., et al, "Procedural Generation of Videos to Train Deep Action Recognition Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2594-2604.

[10] Hwang, H., et al. "ElderSim: A Synthetic Data Generation Platform for Human Action Recognition in Eldercare Applications," in *IEEE Access*, vol. 11, pp. 9279-9294, 2023.

[11] Hu, Y., et al, "SAIL-VOS 3D: A Synthetic Dataset and Baselines for Object Detection and 3D Mesh Reconstruction from Video Data," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 3359-3369.