TransNetV2 and BLIP: technology, use cases, examples

Andrii Zhukov

August 2024

Introduction

Artificial Intelligence has experienced remarkable growth over the past few decades. It seemed like a fairy tale and is now used almost in all aspects of our lives. Now we are entering a new era where AI enables machines to perform tasks that were once considered the exclusive domain of human intelligence. One area where AI has made significant success is in video processing and analysis. It is a huge field that includes topics, such as facial recognition, object detection, motion analysis, and many more. In this report, we will concentrate on the influence of AI on some of the most critical tasks in video processing video transitions as well as on video understanding and generation. Specifically, we will discuss TransNet V2 and BLIP technologies. With the use of deep learning algorithms, especially convolutional neural networks (CNNs), it managed to achieve state-of-the-art performance and bring video analysis to a new level. This report dives into the architecture, functionality, and practical applications of TransNet V2 and BLIP, providing detailed insights and code examples to demonstrate their use.

TransNet V2

TransNet V2 is a deep-learning model designed to detect shot transitions in videos. Identifying points in the video where one scene ends and another begins is not as trivial a task as it might seem at first glance, as human intelligence is pretty good in tasks like that, but computers think in a completely different manner, usually unknown to many people. Moreover, the task is complicated by the fact that there are both hard (abrupt) cuts and gradual ones. To achieve better performance over its predecessors like TransNet, Hassanien et al., or ResNet baseline, which relied on heuristics and handcrafted features, developers of TransNet V2 came up with an idea to use convolutional neural networks to improve the accuracy by learning features directly from the video data. We will discuss how it works and where it is used later in this section.

How it works

Now we discuss the detailed algorithm of how the TransNet V2 model works:

- First of all, the model takes a video as a sequence of frames for an input. It resizes it to a lower resolution leaving all necessary details but significantly making the process faster and more efficient.
- Then the input frames are processed through a series of DDCNN cells. Each cell captures different levels of temporal and spatial features, with increasing levels of abstraction as you move deeper into the network. These cells use convolutions to ensure that the model can consider a temporal context, which is important for identifying gradual transitions.
- After the model has passed through the DDCNN cells, using both learned and handcrafted features it finds between frames and calculates scores based on it. These scores help the model decide whether the changes between frames are significant enough to indicate a transition.
- At the very end, two classification heads predict the presence of a transition based on features and similarity scores. The single-frame head predicts the exact frame where a transition occurs, while the all-frame head considers the entire sequence. After all the predictions are combined to make the final decision about the point in the video where the transition occurs.

Key components

To make this algorithm possible, the model uses a specific architecture:

- **Dilated Deep Convolutional Neural Network Cells**: They are the main part of the whole model as they process a sequence of video frames by using convolutional operations. These operations take information about both the spatial dimensions and the temporal dimension. Moreover, they take only the needed part of the input skipping certain elements. It allows the model to capture a larger temporal context without significantly increasing the number of parameters. After all, the batch normalization is used inside each cell to stabilize the training process and add regularization. In this way, it improves model performance and training efficiency.

- **Convolution Kernel Factorisation**: This is the crucial point in the optimization. In the traditional way, the model would use 3D convolutions taking both 2D spatial and 1D temporal features at once. However, in TransNet V2, it separates operations of convolving spatial and temporal dimensions. Such an action achieves two goals: learning spatial features first and then understanding how they change over time and preventing overfitting by reducing the number of parameters.

- **Multiple Classification Heads**: The model has two classification heads. It allows concentration both on the exact point of transition and the overall sequence of frames in transition. In this way, the developers achieved a better understanding and higher accuracy in detecting transitions.

- **Frame Similarities as Features**: The model compares the frame to its predecessors and successors in the sequence. This is done to detect transitions using frame similarities as features. TransNet V2 uses both handcrafted and learned features and then compares them using the cosine similarity metric. This is the metric that considers only the direction of the vector instead of taking into account both magnitude and direction. In this way, the model can identify where the transition occurs.

Training methodology

Now we discuss how the model is trained. Training is one of the most important aspects as it directly influences performance.

First of all, we need to know which datasets were used. We need ones that would ensure that the model learns to accurately detect both hard cuts and gradual transitions. For this purpose, the TRECVID IACC.3 and ClipShots Dataset were used. ClipShots Dataset consists of 4039 videos with annotated real transitions, specifically 128,636 hard cuts and 38,120 gradual transitions. TRECVID IACC.3, in its turn, is used to generate synthetic transitions. For training TransNet V2, 85% of all transitions are synthetic. These are transitions that are generated during training by combining pairs of shots from the TRECVID IACC.3 and ClipShots datasets. The test of the model showed that using mostly synthetic transitions improves the model's performance significantly by giving a broader variety of transition types. 15% of the data for training were real. This is also valuable because it provides real-world examples of shot transitions. As mentioned above, all video frames are resized to a resolution of 48x27 pixels.

Then the process of preparing the training data starts. Arbitrary 100 frames with a known transition are chosen from the dataset with real transitions. For synthetic transitions, 300-frame segments are extracted from different parts of scenes. These segments are then randomly cropped to 100 frames and then they are joined with some transition of varying types. Sequences without any transitions are generally not used for training. This is because the model can identify non-transitions as transitions.

To make the model more robust and to prevent it from overfitting to the training data, several data augmentation techniques are used: flipping frames, adjusting color properties, and applying transformations. It helps the model handle a wide range of visual scenarios.

Use cases

In this section, we tried to explain how TransNet V2 works, but how can we use it?

First of all in video production editing. Video editing software can use TransNet V2 to automatically divide videos into separate scenes by detecting transitions. This can be helpful for editors to navigate to different scenes and make precise cuts.

It can also be used in media and entertainment. The model is able to help creators extract key moments from long video footage, for example, action scenes, to create summaries or highlights. Moreover, it can find an appropriate moment in the video to insert an advertisement.

One of the most useful cases is creating databases of videos and their individual scenes. TransNet V2 can detect shot boundaries and index different scenes. For example, such a use case is hugely important for events like Video Browser Shutdown (VBS).

Example 1: Basic Shot Transition Detection

```
from transnetv2 import TransNetV2
model = TransNetV2("path")
video_frames, single_frame_predictions, all_frame_predictions = model.predict_video("path")
scenes = model.predictions_to_scenes(predictions=single_frame_predictions)
print("Detected scenes:", scenes)
```

Example 2: Visualize Detected Transitions

```
visualization = model.visualize_predictions(
    frames=video_frames,
    predictions=(single_frame_predictions, all_frame_predictions)
)
visualization.show()
```

BLIP

BLIP (Bootstrapping Language-Image Pre-training) is a VLP frame designed to unify both understanding and generation tasks in multimodal AI. It combines a flexible Multimodal Mixture of Encoder-Decoder (MED) architecture with a novel Captioning and Filtering (CapFilt) method, which improves the performance across various tasks, such as image-text retrieval, image captioning, and visual question answering. Moreover, unlike traditional VLP models, which might be created for specific tasks, BLIP's architecture allows adaptability, making it well-suited for a broad spectrum of vision-language applications.

How it works

The core functionality of BLIP revolves around its two main components that work together to deliver state-of-the-art performance:

1. **Multimodal Mixture of Encoder-Decoder (MED) Architecture**: The MED architecture is a crucial element for BLIP's performance. This architecture allows BLIP to operate across different modes depending on the nature of the task—whether it requires understanding (e.g., retrieving images based on text) or generating (e.g., creating a caption for an image). This is achieved through a combination of flexible encoder-decoder structures that can switch between different modes.

    - **Unimodal Encoder**: In this mode, BLIP uses separate encoders for images and text. For the text, it uses a structure similar to the BERT model, where a special [CLS] token is applied to summarize the entire sentence. This approach aligns the feature spaces of visual and textual representations, which is particularly important for tasks that require understanding both modalities independently, such as image-text retrieval.
    - **Image-Grounded Text Encoder**: This mode involves the addition of cross-attention layers to model the interactions between vision and language. These layers effectively enter visual information into the text encoder, enabling the model to learn and represent the relationships between images and their associated textual descriptions. This ability is useful for tasks that require a deeper understanding of both visual and textual content at once, such as image-text matching. The combination of vision-grounded and language-grounded understanding improves the model's ability to reason about complex visual-textual relationships.
    - **Image-Grounded Text Decoder**: For generation tasks like image captioning, BLIP uses a text decoder that replaces the bidirectional self-attention layers typical of BERT-like architectures with causal self-attention layers. This modification is more suitable for generating coherent text sequences based on visual content. The decoder uses a [Decode] token to initiate the process and generate a sequence of text, allowing for detailed and contextually relevant descriptions that match the visual input.

2. **Captioning and Filtering (CapFilt) Method**: The CapFilt method is a new approach designed to maximize the usefulness of noisy web data for pre-training. It addresses the common challenge of data noise by selectively filtering out incorrect image-text pairs while preserving high-quality data. The CapFilt framework is made using two main components: a captioner and a filter, which work together to expand the dataset size and diversity while ensuring data quality.

- **Captioner**: The captioner functions as an image-grounded text decoder, trained on a small-scale, high-quality dataset such as COCO. Its role is to generate synthetic captions for web images, effectively expanding the available dataset with new, relevant examples that are not present in existing datasets. By doing so, the captioner helps the model learn from a wider variety of image-text pairs, which is crucial for model performance.
- **Filter**: The filter is an image-grounded text encoder that is trained to identify and remove noisy captions from both the original web texts and the synthetic captions generated by the captioner. Utilizing Image-Text Contrastive (ITC) and Image-Text Matching (ITM) objectives, the filter evaluates whether a text is a good match for an image, discarding those pairs that are considered irrelevant or incorrect. This filtering process ensures that the dataset used for pre-training is of high quality, thereby improving the model's ability to learn effective vision-language representations.

Training metodology

The pre-training of BLIP is guided by three primary objectives, each corresponding to the different functionalities of the MED architecture:

- **Image-Text Contrastive Loss (ITC)**: The ITC loss is used in the unimodal encoder mode to align the visual and textual feature spaces. This objective encourages positive image-text pairs to have similar representations while pushing apart negative pairs. It is a critical component for tasks such as image-text retrieval, where the goal is to retrieve relevant images based on textual descriptions.

- **Image-Text Matching Loss (ITM)**: The ITM loss is applied when using the image-grounded text encoder. It allows the model to learn detailed alignments between vision and language by differentiating between matched and unmatched image-text pairs. Hard negative mining is used to concentrate on the most difficult examples, improving the model's robustness and accuracy in comprehending complex relationships between visual and textual data.

- **Language Modeling Loss (LM)**: This loss function is used with the image-grounded text decoder to train the model to generate coherent textual descriptions based on visual inputs. The model optimizes for cross-entropy loss to maximize the probability of producing correct captions. This objective is particularly important for improving the model's ability to generate text that is not only accurate but also creative and contextually relevant to the visual content.

Key components

The success of BLIP lies in its innovative architecture, which integrates several core components to handle diverse multimodal tasks with efficiency and adaptability:

- **Multimodal Encoder-Decoder Structure**: This structure is universal, allowing the model to switch between encoding and decoding modes. This adaptability is important for tasks requiring different approaches, such as understanding versus generation.

- **Image-Grounded Text Modules**: These modules, which introduce cross-attention mechanisms, play a crucial role in enhancing both understanding and generation capabilities. They allow the model to capture more specific relationships between images and their corresponding textual descriptions.

- **Synthetic Caption Generation and Filtering**: The CapFilt method, involving both caption generation and filtering, helps improve the quality of training data. By generating synthetic data and then filtering out noise, BLIP creates a more robust dataset that better supports downstream tasks.

Dataset Bootstrapping with CapFilt

The CapFilt method significantly improves the quality and diversity of the training data, ensuring that the model is open to a wide variety of scenarios:

1. **Generating Synthetic Captions**: The captioner generates new captions for images scraped from the web. This process increases the quantity and variety of training data, providing new, relevant image-text pairs that can help the model generalize better.

2. **Filtering Noisy Data**: The filter component then processes both the original web data and the synthetic captions, identifying and removing pairs that do not accurately represent the visual content. This filtering step reduces noise and ensures that only high-quality data is saved, which is crucial for effective learning.

3. **Combining Data Sources**: The final dataset is combined with human-annotated datasets, such as COCO and Visual Genome, to create a comprehensive pre-training dataset. This enriched dataset allows the model to learn from a diverse range of image-text pairs, improving its overall performance.

Use cases

BLIP's architecture and training methodology enable it to help in a variety of vision-language tasks, demonstrating state-of-the-art performance:

The first task that comes to our mind is image-text retrieval: BLIP achieves high accuracy in retrieving relevant images from textual queries and vice versa. It is better than any other existing models, such as ALBEF and CLIP, particularly in tasks that involve fine-grained matching between images and text.

Moreover, image captioning is also a field where BLIP can be helpful. The model is highly effective at generating descriptive and contextually accurate captions for images. It is able to generate diverse and creative captions.

Furthermore, BLIP can contribute to visual question answering. The model shows robust performance in answering questions related to visual content. Unlike traditional models, BLIP treats it as an answer-generation task, providing more nuanced and open-ended responses.

Finally, the model is extremely good at visual dialog and reasoning. It is useful in tasks that require understanding complex connections and interactions between images and text. Such an ability makes it suitable for advanced applications that require multimodal comprehension.

Example Implementations

Caption generation:

```
from blip import BLIPModel
with Image.open(image_path).convert(’RGB’) as raw_image:
    inputs = processor(raw_image,
    return_tensors="pt").to(device)
    out = model_blip.generate(**inputs,
    max_new_tokens=50)
    caption = processor.decode(out[0],
    skip_special_tokens=True)
return caption
```

Basic Image-Text Retrieval:

```
from blip import BLIPModel
model = BLIPModel("path")
image_text_pairs, retrieval_results = model.predict("path")
print("Retrieved pairs:", retrieval_results)
```

Conclusion

In this report, we explored two state-of-the-art AI models—TransNet V2 and BLIP—that are advancing the fields of video processing and multimodal AI.

TransNet V2 represents a significant evolution in shot boundary detection, achieving high accuracy across various video datasets. Its use of new technologies and deep learning-based approach allows it to efficiently detect both abrupt and gradual shot transitions. This model will significantly influence many fields, such as editing, media archiving, and content summarization.

On the other hand, BLIP (Bootstrapping Language-Image Pre-training) is a new framework that unifies vision-language understanding and generation tasks. Its ability to perform well across a wide range of vision-language tasks—including image-text retrieval, image captioning, and visual question answering demonstrates its adaptability and robustness.

Both TransNet V2 and BLIP exemplify how deep learning models can be adapted and optimized for specific domains, whether it's detecting fine-grained transitions in video or integrating textual and visual modalities for comprehensive AI understanding. The combination of these two models is particularly useful in an event, such as a Video Browser Shutdown, where it is needed to cut the videos in scenes and then describe them. To conclude, as AI is being improved, such models will play a crucial role in pushing the boundaries of automated video analysis and multimodal learning, providing more efficient, accurate, and scalable solutions to complex problems in multimedia and AI-driven applications.

References

[1] T. Souček and J. Lokoč, ”TransNet V2: An effective deep network architecture for fast shot transition detection,” arXiv preprint arXiv:2008.04838, 2020. Available: `https://arxiv.org/abs/2008.04838`.

[2] A. Hassanien, M. A. Elgharib, A. Selim, M. Hefeeda, and W. Matusik, ”Large-scale, Fast and Accurate Shot Boundary Detection through Spatio-temporal Convolutional Neural Networks,” arXiv preprint arXiv:1705.03281, 2017. Available: `https://arxiv.org/abs/1705.03281`.

[3] M. Gygli, ”Ridiculously Fast Shot Boundary Detection with Fully Convolutional Neural Networks,” arXiv preprint arXiv:1705.08214, 2017. Available: `https://arxiv.org/abs/1705.08214`.

[4] S. Tang, L. Feng, Z. Kuang, Y. Chen, and W. Zhang, ”Fast Video Shot Transition Localization with Deep Structured Models,” arXiv preprint arXiv:1808.04234, 2018. Available: `https://arxiv.org/abs/1808.04234`.

[5] L. Baraldi, C. Grana, and R. Cucchiara, ”A Deep Siamese Network for Scene Detection in Broadcast Videos,” in Proceedings of the 23rd ACM International Conference on Multimedia, 2015, pp. 1199-1202.

[6] Li, J., Li, D., Xiong, C., Hoi, S. (2022). BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. Proceedings of the 39th International Conference on Machine Learning (ICML), Baltimore, Maryland, USA.

[7] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning Transferable Visual Models from Natural Language Supervision. arXiv preprint arXiv:2103.00020.

[8] Li, J., Selvaraju, R. R., Gotmare, A. D., Joty, S., Xiong, C., Hoi, S. (2021). Align Before Fuse: Vision and Language Representation Learning with Momentum Distillation. Advances in Neural Information Processing Systems (NeurIPS).

[9] Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., Cao, Y. (2021). SimVLM: Simple Visual Language Model Pretraining with Weak Supervision. arXiv preprint arXiv:2108.10904.

[10] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In International Conference on Learning Representations (ICLR).

[11] Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y. (2020). The Curious Case of Neural Text Degeneration. International Conference on Learning Representations (ICLR).