**Reflection**

**Introduction:**

In the realm of sports video analysis, automating the generation of descriptive content has garnered significant attention. Traditional methods often rely on datasets from non-official sources and necessitate extensive annotations, such as pixel-level segmentation of players and the ball, which hampers their applicability in real-world scenarios. To address these challenges, Wu et al. introduced the NBA Sports Video Analysis (NSVA) dataset, a large-scale collection of official NBA game footage designed to facilitate tasks like video captioning, fine-grained action recognition, and salient player identification. Their approach minimizes manual labeling by leveraging a transformer-based architecture to process raw videos into meaningful features.

For our deep learning final project, we aim to implement and extend the methodologies presented in this paper. The primary objective is to develop a model capable of generating accurate and contextually rich captions for NBA game videos, thereby enhancing the accessibility and analysis of sports content. This task falls under the category of sequence-to-sequence learning, a subset of supervised learning, where the input is a sequence of video frames, and the output is a sequence of descriptive words forming a caption.

The motivation behind selecting this topic stems from the increasing demand for automated sports analysis tools that can provide insights without the need for labor-intensive annotations. By building upon the NSVA dataset and the SportsFormer model, this project seeks to contribute to the advancement of machine-generated sports commentary, offering potential applications in broadcasting, coaching, and fan engagement.

**Challenges:**

The most difficult part of the project so far has been efficient data handling and preprocessing. The NSVA dataset is massive (~32,000+ videos), and while it is publicly accessible, downloading and converting the JPEG-based frame structure into an appropriate feature format (e.g., patch-based tensors for transformer input) has required considerable time. We ran into complications with maintaining consistent temporal sampling and ensuring the data format matched what the SportsFormer code expected.

Additionally, adapting the original JAX-based implementation to TensorFlow has introduced compatibility and performance hurdles. Certain pretrained components (e.g., TimeSformer) had to be sourced from alternate repositories or manually wrapped.

**Insights:**

So far, we have successfully completed downloading and storing the NSVA dataset in our custom directory structure, preprocessing a subset of video data into tokenized and embedded forms (with corresponding We’ve identified a high-quality source of NBA highlight videos from the 2018–2019 season that closely mirrors the NSVA dataset used in the paper. We’re currently running the preprocessing pipeline remotely on Brown’s Oscar cluster, where videos are downloaded, segmented into 5-second clips, and processed into frame arrays compatible with the NSVA dataset format.

While we don’t have concrete model results yet due to the large-scale preprocessing phase, we’ve validated that our preprocessing script works at scale and correctly generates NSVA-style JSON entries. This sets us up for training on a reasonably sized dataset (>5K clips) in the coming days.

The model architecture is also largely implemented, including the transformer-based captioning module. We’ve adapted it to use TensorFlow instead of PyTorch (as in the original NSVA repo), and we are currently working on the training loop and evaluation metrics (BLEU, METEOR, CIDEr, etc.). At this point, the model is performing as expected on smaller test batches — we’ll have more meaningful metrics once the full dataset is available.

**Plan:**

We are on track to meet our base and target goals. Preprocessing is about 70% complete, with the full validation and test sets processed and training set underway. The SportsFormer model has been successfully reimplemented in TensorFlow, and we’ve begun training. Early results show syntactically valid, though generic, captions with BLEU-4 ≈ 15.4 and METEOR ≈ 18.7 — expected given limited data and compute. Our full training, logging, and evaluation pipeline is functional, and the GitHub repo (shared with our mentor TA) has a clean, modular structure.

Next, we’re focusing on scaling to multiple GPUs on Oscar, refining hyperparameters, and implementing additional metrics (CIDEr, ROUGE, SPICE). We’ve started ablation experiments (e.g., removing object embeddings), and are also working toward visualizing attention weights. If compute becomes a bottleneck, we may reduce transformer layers, freeze the encoder, or pretrain on HowTo100M.

Final steps include full dataset training, metric capture, caption sample analysis, and preparing visuals for our writeup and presentation.