BPM (Bron Performance Metric)

By: Angel Alvarado Reyes, Wali Siddiqui, Robayet Hossain, Alan Lucero
CSCI 1470: Deep Learning, Mentor TA: Johnny Elias

Background + Data

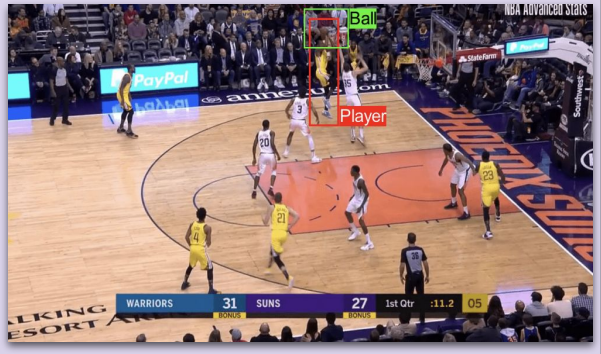
Introduction

To enhance the closed captioning of NBA video footage, we implemented techniques from the paper *Sports Video Analysis on Large-Scale Data*. Our model improves captions from generic outputs like "Player shoots ball" to specific descriptions such as "LeBron shoots a 25-foot 3-pointer," by integrating player identity and spatial information. We selected this paper because it combines our passion for LeBron James with course concepts such as Transformers and Convolutional Neural Networks (CNNs).

Data

Our dataset consists of NBA highlight videos sourced from YouTube, specifically the 2018-2019 season highlights comprising 27.5 hours of footage and ~40,000 captioned phrases from 165 games. The dataset is divided into training, validation, and test sets, with a subset of games excluded from training to ensure evaluation diversity.

Data processing was attempted through the Oscar cluster, though, we ended up using Google Colab. The original project reduced resolution and frame rate when analyzing footage that it deemed less significant, but we were unable to replicate this in our model. Instead, we use a consistent 8fps when extracting the features and maintain the original resolution.



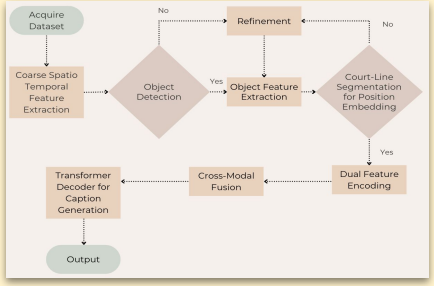
Example footage that we are generating captions for, with highlighted feature detection that we use to generate captions

Architecture + Pipeline

Methodology

The model architecture leverages four key transformer components: caption decoder, cross-attention module for feature fusion, fine-grained feature encoder, and a coarse-grained feature encoder.

Video features are processed using a pre-trained YOLOv5 model. Features such as players, basket, and ball are extracted. These are processed by a vision transformer to model fine-grained regional features. The system uses two encoders: a coarse transformer encodes overall video context, while a fine-grained transformer encodes object-level and positional information.



Model architecture that we are seeking to replicate in our project.

These outputs are fused using a cross-attention mechanism to produce a joint representation. A transformer-based decoder then autoregressive generates captions from this fused context. During inference, captions are generated using beam search to optimize fluency and accuracy.

Discussion/Conclusion

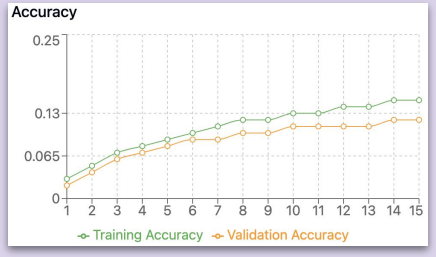
Our training and validation accuracy peaked at ~15%, indicating that our model was functional, yet inaccurate. Our loss began declining well, indicating that at the start of training, the model was certainly learning well, but it leveled off quickly, indicating that it was likely underfitting early on. When predicting the next word of a caption, our model did perform well, boasting a final validation accuracy of 93.4%, indicating that even while underfitting and with limited data, our model had potential to learn well.

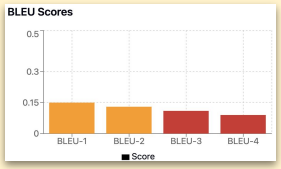
Results + Discussion

Results w/ Data Visualization

Training and validation loss lowers to around 4.5 within 15 epochs.

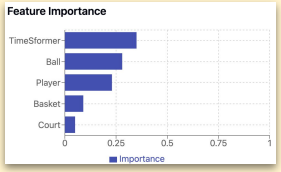
Both training and validation accuracies peak at around 15%.





BLEU scores measure the similarity between generated captions and reference captions.

TimeSformer visual encoder contributes most to the model's predictions.



Final Validation Loss	0.67
Final Training Accuracy	0.83
Final Validation Accuracy	0.93

```
Example 3:
Ground truth: he's getting there hey that's a break
Prediction: he's almost through now hey what a play

Example 4:
Ground truth: curry and he's going to the line looking
Prediction: curry goes to the line and he's focused in

Example 5:
Ground truth: [ applause ]
Prediction: crowd at at at at at at at at
```

Model's validation loss, training accuracy, and validation accuracy when predicting the next word of the phrase or sentence of the caption