"Motion Detection Video Analytics in NBA Videos"

DEVANSH PUROHIT, Courant Institute of Mathematical Sciences, New York University, USA
EUIJAE KIM, Courant Institute of Mathematical Sciences, New York University, USA

ACM Reference Format:
Devansh Purohit and Euijae Kim. 2024. "Motion Detection Video Analytics in NBA Videos". *ACM Trans. Graph.* 37, 4, Article 111 (August 2024), 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 ABSTRACT

The Authors for this paper, Devansh and Euijae share a deep passion for basketball. We find immense enjoyment in watching the game due to its unparalleled action and intensity, which surpasses that of any other major league sport. The rapid pace of the NBA, where ball possession switches every 24 seconds if a team fails to make a shot attempt, adds to its allure, making it incredibly fast-paced and dynamic to witness. Recently, we began contemplating if there exists a method to analyze basketball videos and automatically catalog shots and game events without the need for manual observation. This curiosity sparked the inception of our research journey. The final result uses a fine tuned pre-trained Video Masked Auto-Encoder [Tong et al. 2022] to classify videos with 100% precision and accuracy for field goal attempts, and significant accuracy for multi-angle free throw shot classification.

2 INTRODUCTION

In this research project, we investigate the process of acquiring NBA video data and ensuring it's compatible with our video analysis model. We then proceed to predict whether shots are successful or missed in NBA video clips by employing pre-trained models like `VideoMAE`[Tong et al. 2022] and `PySlowFast's X3D`[Ge et al. 2021]. Our aim is to delve deeper into understanding how these models can enhance our ability to analyze basketball footage and extract meaningful insights.

We experiment with various video classification algorithms, and even an object detection algorithm to figure out which is the best fit for this use case. Pre-trained models with our downloaded video dataset is then used to train pre-trained models for Video Masked Auto-Encoders [Tong et al. 2022], and is tried on X3D (PySlowFast)[Ge et al. 2021], and V-JEPA[Bardes et al. 2024] models. We have also finetuned the YOLO [Ge et al. 2021] fast object detection model to detect baskets on the fly.

Authors' addresses: Devansh Purohit, dnp9357@nyu.edu, Courant Institute of Mathematical Sciences, New York University, New York, New York, USA; Euijae Kim, ek3955@nyu.edu, Courant Institute of Mathematical Sciences, New York University, New York, New York, USA.

0730-0301/2024/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

3 RELATED WORK

Motion and object detection technology has widespread applications across numerous industries. One notable example is its integral role in Tesla's self-driving cars. These autonomous vehicles rely on motion detection to continuously monitor their surroundings, identifying various objects such as traffic lines, vehicles, pedestrians, and obstacles to ensure safe navigation. Another innovative application can be seen in Amazon Go, a revolutionary cashier-less store concept. Utilizing motion detector cameras, this store eliminates the need for traditional checkouts by accurately tracking the products customers place in their shopping carts as they move through the store. As advancements in motion detection and video analysis technologies continue to accelerate, their adoption and integration into various sectors are expected to grow exponentially.

The concept of *Auto-captioning basketball highlights with object tracking* stands out as the most pertinent reference ([Wu et al. 2022]) we have come across. This project endeavors to create a straightforward and budget-friendly solution for analyzing game footage and extracting vital game highlights, by employing a UniVL[Luo et al. 2020] model to concurrently understand and also give live captions for the game. Initially, our intention was to build a classifier model for various play types outlined below. However, upon further consideration, we recognized the potential value in exploring alternative methodologies to achieve our objectives.

(1) 3 points: 3 points shot made and missed
(2) 2 points: Lay-up, dunk, mid-range shoot, and missed shot.
(3) 1 point: Free throw
(4) Block
(5) Steal
(6) Turnover
(7) Fouls (defensive and offensive)

In each such classification, our current focus is to make every decision binary, and then create a system that is able to make classification decisions for multiple actions simultaneously. This enables us to build a modular system, and also allows relatively simple models to function efficiently. Hence, currently we focus on just whether a shot is being made or not. As we later see, we can further complicate the model to accept additional actions.

4 PROBLEM

In this study, our primary focus centers on two distinct categories of play types: shots made and shots missed. According to the rules of basketball, a successful goal is defined as "A legal field goal or free throw attempt shall be scored when a ball from the playing area enters the basket from above and remains in or passes through the net" [Silver 2022]. Conversely, if a field goal attempt fails to meet this criteria, it is deemed unsuccessful and classified as a missed shot.

The challenges we must tackle encompass the acquisition of suitable video clips and their alignment with our video analysis

model. Our first task involves sourcing video content that aligns with our research objectives, ensuring it possesses the necessary quality and relevance for effective analysis. Subsequently, we embark on the process of preparing these video clips to seamlessly integrate with our chosen analysis model, requiring careful consideration of factors such as format compatibility and data pre-processing techniques. Once our data is appropriately curated and formatted, we proceed to the next phase, where we deliberate over the selection of a baseline model. This critical decision involves evaluating various candidate models and their suitability for training to predict the desired categories of play types accurately.

4.1 Pre-processing Data

We fetch Play-By-Play data and Video data using the NBA stats API[Swar year], and by employing the NBA_API GitHub repository. The typical length of a video clip averages around 15 seconds, comprising approximately 375 frames per video. The size of each video varies between 9 MB to 15 MB. However, it's important to note that not every frame contains essential information crucial for video analysis. As such, the process of eliminating redundant frames significantly enhances the efficiency of video analysis training. Moreover, as we learned, reducing the number of frames in a video contributes to improved prediction accuracy, as the model can focus on the most relevant and informative frames during analysis. The time frame for the two classes also needed to be approximately the same in order to avoid flat classifications towards just one class.

4.2 Train Baseline Model

In order to ascertain whether a goal attempt depicted in a given video is successful or not, it is imperative to track the trajectory of the basketball, as its position serves as a key determinant in classifying the play type (shot made and shot missed). Initially, we opt for a model specifically designed to detect the presence of a basketball within a video frame. Subsequently, we meticulously monitor the movement of the basketball throughout the possession. Since our criteria for scoring aligns with the regulations outlined by the NBA, where a successful goal is defined by the ball entering the basket from above, our model must also be equipped to identify the basket within the video frame. Once the basketball object has been successfully detected, the next crucial step involves tracking its trajectory, as our ultimate objective is to discern whether the goal attempt has been converted or not. This multifaceted approach enables us to accurately analyze and classify the outcome of each goal attempt depicted in the video footage.

5 METHODS

Our study encompasses two key components: data handling and prediction. The first aspect involves data acquisition, which entails scraping videos from the official NBA (National Basketball Association) developer API and removing unnecessary frames using the `ffmpeg` package. The second aspect revolves around making predictions using pre-trained models. In this chapter, we will delve into the specifics of the REST API utilized for downloading video clips and the metrics employed for trimming them. Subsequently, we will explore the various models utilized in our study, providing insights into their selection and implementation.

5.1 Preprocessing Data

A plethora of basketball videos can be readily found online, presenting a rich source of data for analysis. However, it's essential to acknowledge that not all videos are captured from the same fixed angle. Some are filmed from ground level, while others adopt a bird's-eye view perspective. This variance in recording angles poses a significant challenge during model training, as it necessitates the inclusion of additional training data to ensure the model can accurately recognize and interpret different viewing angles. Consequently, acquiring video footage with a properly fixed angle becomes a pivotal aspect of the project, as it simplifies the training process and enhances the model's ability to generalize across various viewing perspectives.

Throughout this project, we have utilized a dataset comprising of more than 300 video clips across 30+ games for training the model. This comprehensive collection of videos serves as the foundational training data upon which our model learns to accurately analyze and predict various play types. By leveraging this diverse dataset, encompassing a range of scenarios and gameplay situations, we aim to equip our model with the requisite knowledge and insights needed to effectively interpret and classify basketball footage.

5.1.1 Download Video using REST API. The National Basketball Association (NBA) generously offers an abundance of REST API services tailored specifically for developers. Leveraging these APIs, we accessed a wealth of ground truth data essential for our research. Notably, each script was filmed from a standardized angle, ensuring consistency in resolution and duration across all recordings. Below, we provide a comprehensive overview of the API specification, including its detailed response schema, which facilitated seamless integration into our analysis framework.

```
// API Specification
GET /stats/videoeventsasset?GameEventID={p1}&GameID={p2}
```

```
// Example of response in JSON format
{
  "resource": "videoevents",
  "parameters": {
    "GameID": "0022101182",
    "GameEventID": 17
  },
  "resultSets": {
    "Meta": {
      "videoUrls": [
        {
          "uuid": "0240131a-6a04-b350-689f-e161b79fbf2e",
          "sdur": 13733,
          "surl": "https://videos.nba.com/.../<id>_320x180.mp4",
          "sth": "https://videos.nba.com/.../<id>_320x180.jpg",
          "mdur": 13733,
          "murl": "https://videos.nba.com/.../<id>_960x540.mp4",
          "mth": "https://videos.nba.com/.../<id>_960x540.jpg",
          "ldur": 13733,
```

```
        "lurl": "https://videos.nba.com/.../<id>_1280x720.mp4",
        "lth": "https://videos.nba.com/.../<id>_1280x720.jpg",
        "vtt": "https://videos.nba.com/.../<id>.vtt",
        "scc": "https://videos.nba.com/.../<id>.scc",
        "srt": "https://videos.nba.com/.../<id>.srt"
      }
    ]
  },
  "playlist": [
    {
      "gi": "0022101182",
      "ei": 17,
      "y": 2022,
      "m": "04",
      "d": "05",
      "gc": "2022-04-05/ATLTOR",
      "p": 1,
      "dsc": "Birch Bad Pass Turnover (P1.T1) #@#Capela STEAL (1 STL)"
      "ha": "TOR",
      "hid": 1610612761,
      "va": "ATL",
      "vid": 1610612737,
      "hpb": 3,
      "hpa": 3,
      "vpb": 2,
      "vpa": 2,
      "pta": 0
    }
  ]
 }
}
```


5.1.2 Trim Video. In the final stage of preprocessing our data, we undertake the task of trimming unnecessary frames from the video footage. It's important to note that not every frame contains essential information pertinent to video analysis. Therefore, the process of eliminating redundant frames plays a pivotal role in enhancing not only the efficiency of video analysis training but also the accuracy of motion prediction. To accomplish this, we manually cropped the videos to actually match the shot attempt and the ball being given to the opposing team's player, indicating a turnover. This meticulous approach ensures that our dataset is optimized for subsequent analysis, allowing us to derive meaningful insights with greater precision and reliability.

5.2 Data Augmentation

To enhance the robustness and generalization capabilities of our model, we implemented data augmentation techniques. These techniques involved randomly applying transformations to the video clips, such as horizontal flipping, rotations, and slight cropping. By introducing these variations, we effectively expanded our dataset, exposing the model to a wider range of scenarios and improving its ability to handle diverse situations during real-time video analysis.

5.3 Model - VideoMAE

VideoMAE [Tong et al. 2022] was presented in NeurIPS 2022, presenting a Data-Efficient Self-Supervised model for Action detection in Videos. It does this by training an Auto-Encoder on masked input data. Since the visual medium is much more contextually rich than the Lingual Medium, not a lot of data is lost by masking, and this technique is able to teach the model contextually rich information through relatively little contextual data. UCF101 is an action recognition data set of realistic action videos, collected from YouTube, having 101 action categories. This data set is an extension of UCF50 data set which has 50 action categories. We choose our pretrained VideoMAE[Tong et al. 2022] model to be trained on this dataset, since it provides a huge variety in environments, making finetuning much faster.

We run this model on two datasets - one contains 'made' and 'missed' Field Goal Attempts, and another contains 'made' and 'missed' Free Throws, retrieved programmatically. The FGA video dataset comprises of only 72 video clips, and is cropped to approximately match a player taking a shot, and the ball being passed behind the baseline. Since the camera angle is consistent with NBA matches, we don't require a lot of these clips. In comparison, we have 140 free throw clips.

Once we got 100% accuracy and precision on the above dataset, (discussed below), The Free throw dataset was fed into the model directly from the NBA API, without any manual cropping, in order to experiment with the versatility of this data. This means that both Made and Missed shots are sometimes shown from different angles, as shown below.

5.4 Model - YOLOx-v8

Initially, before we spent time manually cropping our dataset to give much richer contextual and temporal meaning to our data, we planned on another strategy involving cropping out the basket and running action detection algorithms on a cropped feed of just the basket at all times. This strategy is similar to Efficient Video Action Detection with Token Dropout and Context Refinement (Chen et. al), where we aim to isolate the action in place, albeit it's more decoupled that the EVAD architecture [Chen et al. 2023].

We finetuned the YOLOX-v8 model [Ge et al. 2021] on a dataset of 300 images, which we manually cropped to feed into an automated annotation service, which created the relevant annotations required for fine-tuning our base YOLO-v8 [Ge et al. 2021] model. Using these annotations, as we see, we're able to dynamically crop out the basket, which we can use for a host of downstream applications, including training Region Prediction Networks, EVADs [Chen et al. 2023], and other architectures which we can use in order to facilitate parallel action detection. We didn't employ this in our model as we realized that other context on the court (players returning to baseline on point being scored) was also really important.

6 IMPLEMENTATION AND RESULTS

6.1 Preprocessing Data

In this section, we will provide a concise overview of the various implementations utilized to prepare for the preprocessing of data. We will briefly discuss the methodologies and tools employed to ensure that the data is appropriately formatted and optimized for subsequent analysis. This includes outlining the steps involved in acquiring and organizing the raw data, as well as the techniques utilized to clean, filter, and preprocess the dataset to enhance its

Fig. 1. Comparison of Free-Throw Angles

quality and suitability for analysis purposes. Additionally, we will touch upon any specific algorithms or software packages utilized in this process, highlighting their role in facilitating the preprocessing tasks.

6.1.1 Download Video. Python serves as the primary programming language driving our project forward. When it comes to downloading video content, we rely on the `http` package to interact with the NBA's REST API seamlessly. The API endpoint operates using the GET method and requires two query parameters: `GameEventID` and `GameID`. These identifiers are pre-defined and readily accessible online, enabling us to retrieve the specific video content needed for our analysis. Leveraging Python and the `http` package simplifies the process of accessing and downloading the requisite data, streamlining our workflow and enhancing the efficiency of our project implementation.

```
import requests

// Set GET method API call
r = requests.get(
    url,
    headers={
        'Accept': 'application/json, text/plain, */*',
        'Origin': 'https://www.nba.com',
        'Host': 'stats.nba.com',
        'User-Agent': random_agent,
        'Referer': 'https://www.nba.com/',
    },
    params={
        // Pass parameters
        'GameEventID': str(eventID),
        'GameID': gameID
    }, timeout=6
)

try:
    r.raise_for_status()
except requests.exceptions.HTTPError:
    // Handle Exception

try:
    json_data = r.json()
except json.decoder.JSONDecodeError:
    // Handle Exception
```

6.1.2 Trim Video. To seamlessly integrate our video content with our video analytics model, we employed the versatile `ffmpeg` tool for video trimming purposes. `ffmpeg` stands out as a comprehensive, cross-platform solution renowned for its ability to record, convert, and stream audio and video content with utmost efficiency and reliability. The implementation process involved the creation of a bash script, designed to automate the trimming process. Initially, the script calculates the duration of the video in seconds, allowing for precise manipulation of the footage. Subsequently, it strategically identifies and extracts a five-second segment from the end of the video, ensuring seamless alignment with the requirements of our video analytics model. This meticulous approach to video trimming ensures that our dataset is optimized for analysis, thereby enhancing the accuracy and effectiveness of our model's predictions.

We did not need to optimize the starting point of the clips, but we found that ending the clips as soon as a turnover is about to begin, is a great way of ensuring that the pre-trained model correctly picks up the cues for a basket.

```
// Function to process each video file
process_video() {
    local input_path="$1"
    local output_path="$2"

    // Find the running time of a video.
    duration=$(ffprobe -v error -show_entries format=duration
      -of default=noprint_wrappers=1:nokey=1 "$input_path")

    // Find the start time
    start_time=$(echo "$duration - 5" | bc)
```

Table 1. Training Loss and Gradient Norm

Step	Loss	Grad Norm
0.06	0.7039	6.0619
1.02	0.6713	5.6039
1.08	0.7013	4.9959
2.05	0.5902	6.8378
3.02	0.5602	11.8895
3.08	0.779	3.7268
4.05	0.4429	5.7751
5.02	0.1808	1.7520
5.08	0.0391	0.2795
6.05	0.0057	0.0583
7.02	0.0588	0.0538
7.08	0.0641	0.0601
8.05	0.1446	0.0375
9.02	0.0095	0.2024
9.08	0.0125	0.2607

```
    // Extracts a 5 second segment from the end of the video
    ffmpeg -ss "$start_time" -i "$input_path"
      -t 5 -c copy "$output_path"
}
```

6.2 Model - VideoMAE

We use the VideoMAE[Tong et al. 2022] pretrained model as our base model, and finetune it on our dataset, giving it labels "Made" and "Missed". Selecting the lowest logits amongst the two, we get the correct classification. After training for a few epochs, the loss and grad norm fall off significantly, and it is able to distinguish between Made and Missed clips correctly, as shown in the CSV here, generated through
tqdm_inference.py in our code base.

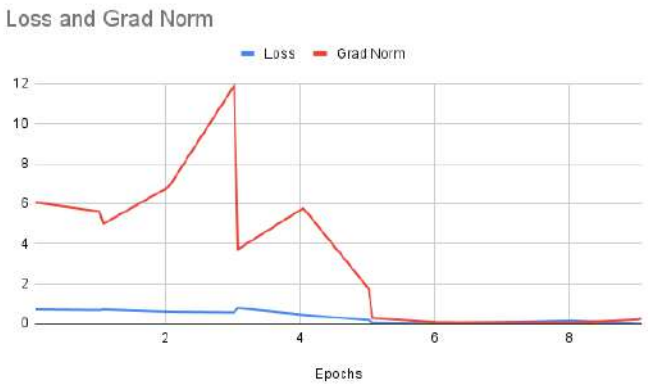

Fig. 2. Loss and Grad Norm while training. Learning rate: 10e-5

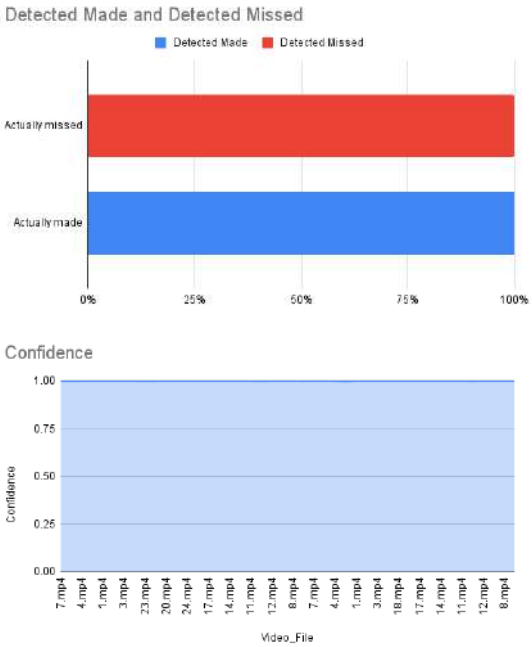

Fig. 3. Field Goal Attempt Classification. 100% accurate classifications were achieved, even on external data, with really high confidence.

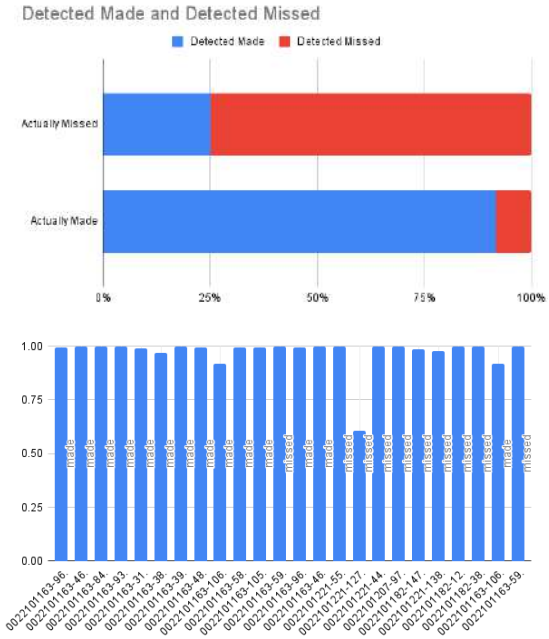

Fig. 4. Free throw attempt classification. 90% Precision and 83% Accuracy observed on Test Data, which is impressive considering the dataset wasn't cropped and had multiple angles.

6.3 Model - YOLOx-v8

We used AutoGluon to bootstrap this process, using their trainer to finetune the object detection YOLOx-v8 model[Ge et al. 2021]. Before we reached this step, however, we needed a way to get annotation data for a significant set of frames from the broadcast video. We used an automated annotation service, RoboFlow, for this task. Manually drawing rectangles to highlight the basket, we manually annotated more than 200 such images with accurate locations of the basket. The service then generated Annotations for the bounding box for us.

We then were able to finetune the base pre-trained YOLOx-v8[Ge et al. 2021] model on our set of basketball images. As seen here, we get an accurately placed rectangle around the basket successfully.

6.4 Model - PySlowFast

Training the pre-trained PySlowFast X3D model was attempted, but it introduced added dependency management complexity, so we limited our project to only consider the VideoMAE[Tong et al. 2022] and the Object Detection models. The X3D framework is highly suitable for realtime use cases, so this is the model we will try training next.

6.5 Model - V-JEPA

Training the pre-trained V-JEPA [Bardes et al. 2024] action detection model was attemped, but to no avail. We weren't getting promising results, and we would like to continue attempting using V-JEPA[Bardes et al. 2024] for action detection.

7 CONCLUSIONS

Our research demonstrates the feasibility of leveraging pre-trained video analysis models like VideoMAE[Tong et al. 2022] and YOLOx-v8[Ge et al. 2021] for automated basketball shot classification. By meticulously preprocessing the data, carefully selecting and fine-tuning our models, and rigorously evaluating their performance, we achieved promising results in identifying successful and missed shots in NBA video clips. The VideoMAE model, in particular, showed exceptional accuracy and precision in classifying field goal attempts, even on external data, with high confidence. The YOLOx-v8 model[Ge et al. 2021] exhibited impressive real-time basket detection capabilities, paving the way for potential future applications in areas such as player tracking and region prediction.

However, we acknowledge that there is still room for improvement. The performance of our models on free throw attempts, particularly those captured from different angles, was less consistent. The dataset structure needs to be formalized and expanded, offering ready access to any type of Basketball event. Furthermore, expanding our analysis to include a wider range of play types and game events remains a challenging but exciting avenue for future research. As video analysis technology continues to advance, we envision a future where AI-powered systems can provide comprehensive and insightful analysis of basketball games, enhancing the viewing experience for fans and providing valuable feedback to players and coaches alike. A promising venture would be pairing this with real-time conversational systems powered by LLMs, enabling live commentary for Basketball games.

Fig. 5. Random Basket detection frames.

REFERENCES

Adrien Bardes et al. 2024. Revisiting Feature Prediction for Learning Visual Representations from Video. *arXiv.org* (Feb 2024). https://arxiv.org/abs/2404.08471 Accessed: 14 May 2024.

Lei Chen et al. 2023. Efficient Video Action Detection with Token Dropout and Context Refinement. *arXiv.org* (Aug 2023). https://arxiv.org/abs/2304.08451 Accessed: 14 May 2024.

Zheng Ge et al. 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv.org* (Aug 2021). https://arxiv.org/abs/2107.08430 Accessed: 14 May 2024.

Huaishao Luo et al. 2020. UNIVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation. *arXiv.org* (Sept 2020). https://arxiv.org/abs/2002.06353 Accessed: 14 May 2024.

Adam Silver. 2022. Rule No. 4: Definitions. https://official.nba.com/rule-no-4-definitions/. Accessed: 14 May 2024.

Swar. year. Swar/NBA_API: An API Client Package to Access the APIs for NBA.Com. https://github.com/swar/nba_api. Accessed: 14 May 2024.

Zhan Tong et al. 2022. VideoMAE: Masked Autoencoders Are Data-Efficient Learners for Self-Supervised Video Pre-Training. *arXiv.org* (Oct 2022). https://arxiv.org/abs/2203.12602 Accessed: 14 May 2024.

Dekun Wu et al. 2022. Sports Video Analysis on Large-Scale Data. *arXiv.org* (Aug 2022). https://arxiv.org/abs/2208.04897 Accessed: 14 May 2024.