For better quality view the presentation in Canva

DETECTION OF PLAYER CONTACT EVENTS IN NFL GAMES

USING ANALYSIS SYSTEMS

Kaggle competition

HTTPS://GITHUB.COM/ITZNXHIN/SAD---NAHIN-NICOLAS-AND-ANDERSON

INTRODUCTION

What is the reason of this?

American football is a high-impact sport characterized by constant physical contact between players. These collisions, while integral to the game, pose significant risks including concussions, musculoskeletal injuries, and long-term health conditions. Because of this, player safety has become a grow ing concern, prompting efforts to monitor and analyze contact events more effectively. Traditional methods such as manual video review are time-consuming and error-prone, leading to a demand for automated systems that can detect and evaluate contact with higher accuracy and speed.

Challenges

1.Data Quality: Variability in video quality, caused by camera positioning and coverage, hindered clear visibility of player interactions. Some views were too distant or at oblique angles, and zooming degraded resolution, introducing artifacts and reducing reliability.

2. **Synchronization: Aligning video and sensor data was complex due to differing frequencies and reference frames. Small misalignments could cause significant prediction errors, requiring timestamp interpolation and careful merging strategies.**

3. Class Imbalance and Rare Events: Contact events were rare compared to non-contact frames, creating an imbalanced classification problem. Techniques like class weighting, sampling adjustments, and domain-informed feature design were used to improve sensitivity to these rare events.

4. Multimodal Fusion: Combining video-derived features with tracking data was challenging due to differences in structure, timing, and resolution. Careful design was needed to avoid redundancy, information loss, or temporal drift.

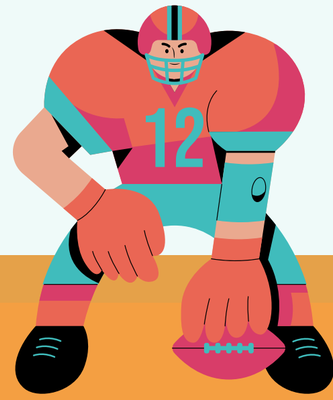
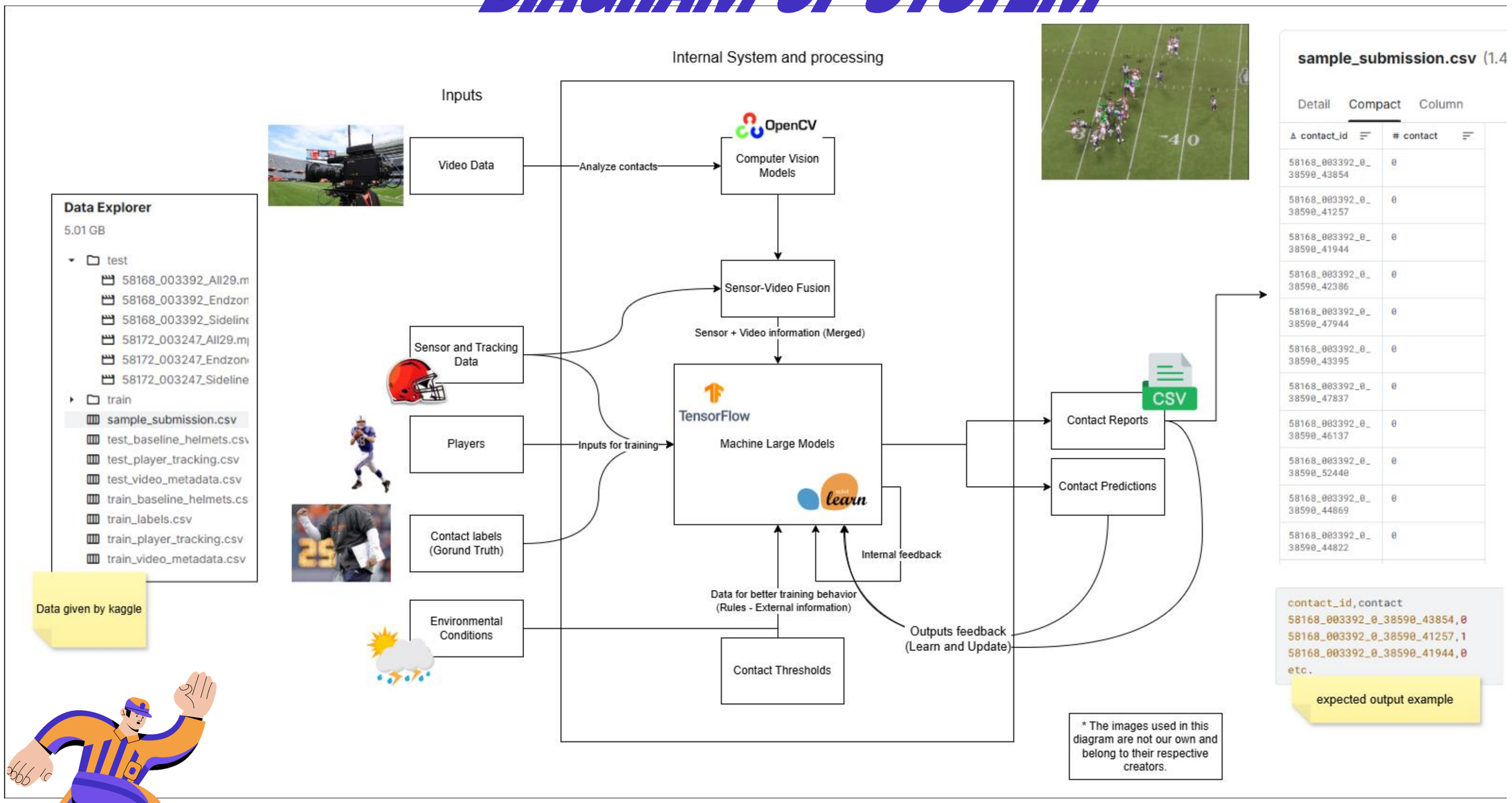


DIAGRAM OF SYSTEM



Input

Videos

Video	
Duración	00:00:11
Ancho fotograma	1920
Alto fotograma	1080
Velocidad de datos	2281kbps
Velocidad de bits total	2376kbps
Velocidad fotograma	59.94 fotogramas/segu...
Audio	

Input

CSV - DATA

- baselline-helmets

	A
1	game_play,game_key,play_id,view,video,frame,nfl_player_id,player_label,left,width,top,height
2	58168_003392,58168,3392,Endzone,58168_003392_Endzone.mp4,290,39947,H72,946,25,293,34
3	58168_003392,58168,3392,Endzone,58168_003392_Endzone.mp4,290,37211,H42,151,25,267,33

- player-tracking

	A
1	game_play,game_key,play_id,nfl_player_id,datetime,step,team,position,jersey_number,x_position,y_position,speed,distance,direction,orientation,acceleration,sa
2	58172_003247,58172,3247,41937,2020-09-13T19:30:20.200Z,-272,home,MLB,57,64.28,11.29,4.82,0.46,20.74,12.43,3.13,3
3	58172_003247,58172,3247,45345,2020-09-13T19:30:20.200Z,-272,away,ILB,50,76.03,36.74,3.15,0.32,179.52,184.91,1.87,-1.81
4	58172_003247,58172,3247,46205,2020-09-13T19:30:20.200Z,-272,home,DE,98,65.64,15.74,1.8,0.2,6.73,339.85,0.69,-0.68

- video-metada

A
game_play,game_key,play_id,view,start_time,end_time,snap_time
58168_003392,58168,3392,Endzone,2020-09-11T03:01:43.134Z,2020-09-11T03:01:54.971Z,2020-09-11T03:01:48.134Z
58168_003392,58168,3392,Sideline,2020-09-11T03:01:43.134Z,2020-09-11T03:01:54.971Z,2020-09-11T03:01:48.134Z

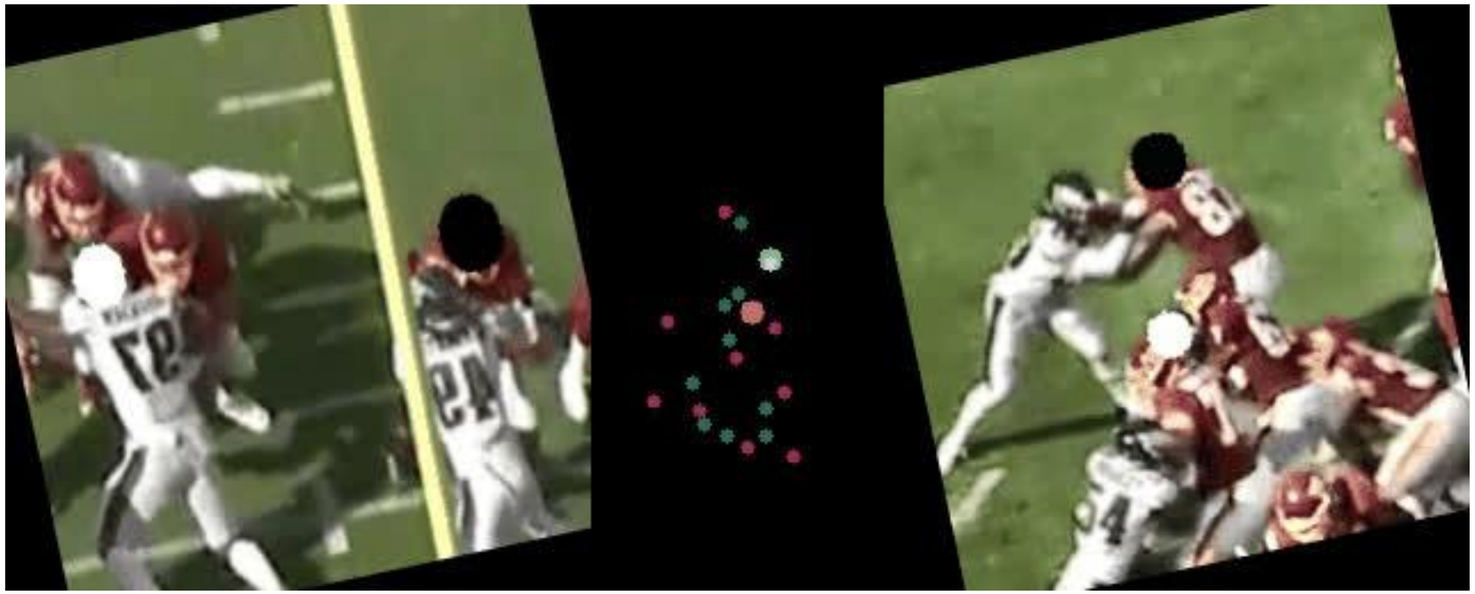
- train-label

A
contact_id,game_play,datetime,step,nfl_player_id_1,nfl_player_id_2,contact
58168_003392_0_38590_43854,58168_003392,2020-09-11T03:01:48.100Z,0,38590,43854,0
58168_003392_0_38590_41257,58168_003392,2020-09-11T03:01:48.100Z,0,38590,41257,0
58168_003392_0_38590_41944,58168_003392,2020-09-11T03:01:48.100Z,0,38590,41944,0

Internal system

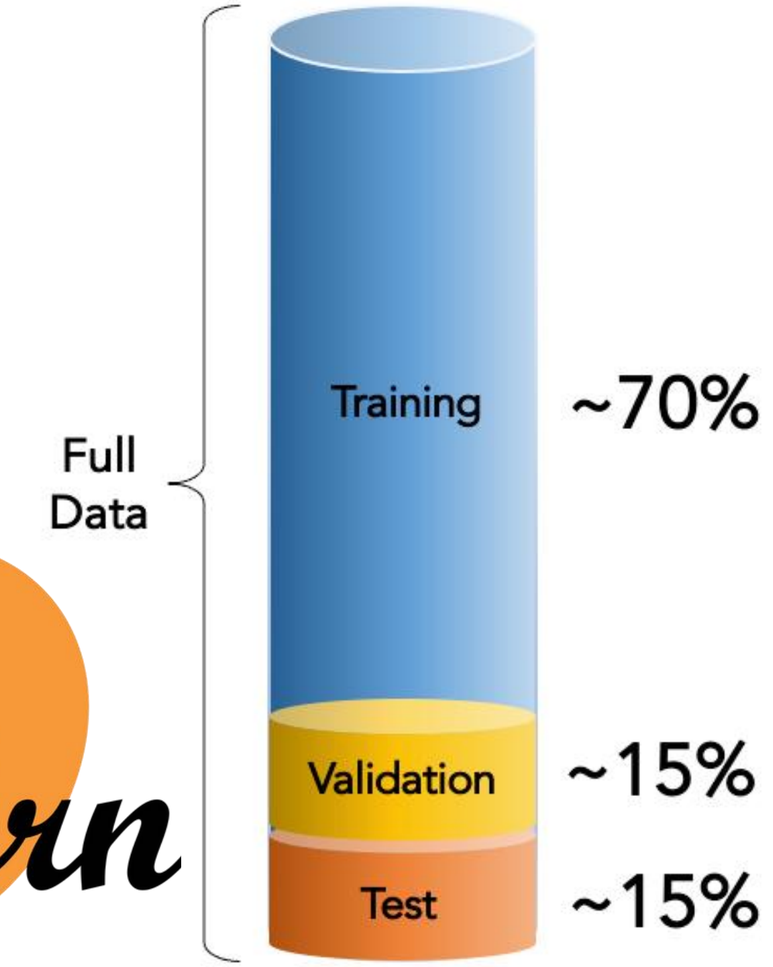
OpenCV

Video - Data processing



Internal system

A RandomForestClassifier from Scikit-learn was chosen for its robustness and efficiency. This ensemble of decision trees can handle diverse features and im balanced classes via class weights.




```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score
```

Output

To prepare the submission:

1. Load test_player_tracking.csv and apply the same preprocessing and feature extraction as used for training.
2. Use the trained RandomForest to predict contact probabilities for each test contact_id.
3. Format the predictions into a submission CSV with columns contact_id and contact_probability.
4. **Save the submission file and upload to the Kaggle competition platform.**

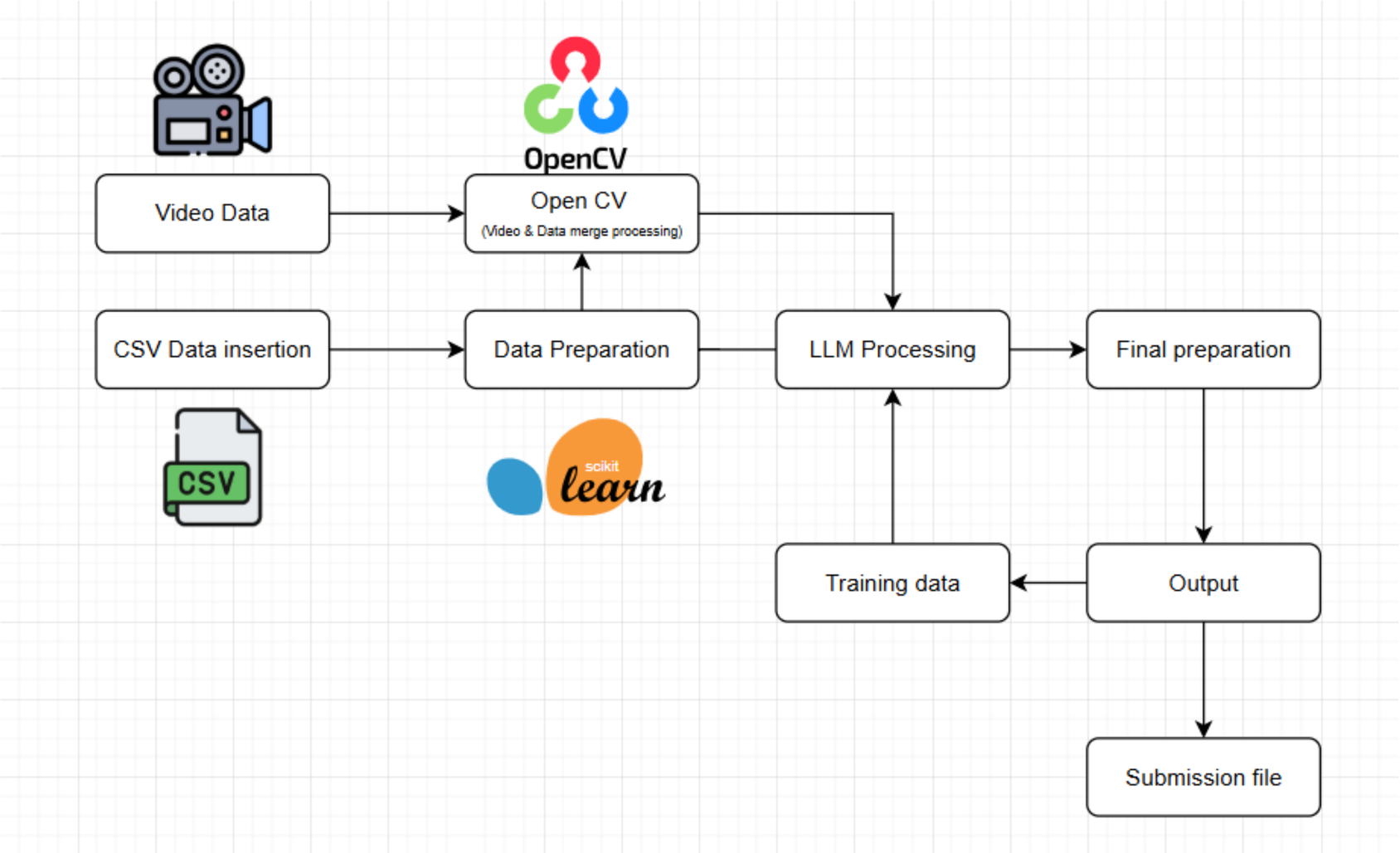
This process follows the standard Kaggle guidelines. The RandomForest pipeline was efficient enough to process the entire test set within the competition's time limits.

sample_submission.csv (1.47 MB)

Detail | Compact | Column

contact_id	contact
58168_003392_0_38590_43854	0
58168_003392_0_38590_41257	0
58168_003392_0_38590_41944	0
58168_003392_0_38590_42386	0
58168_003392_0_38590_47944	0
58168_003392_0_38590_43395	0
58168_003392_0_38590_47837	0
58168_003392_0_38590_46137	0
58168_003392_0_38590_52440	0
58168_003392_0_38590_44869	0
58168_003392_0_38590_44822	0

Methodology



Discussion

• **Strengths: The model trained quickly and was easy to interpret. Feature importance scores from RandomForest helped confirm that intuitive fea tures (e.g., distance between players) were indeed relevant. The approach was computationally efficient compared to the previously attempted LLM method, confirming the choice for numeric models.**

• **Limitations: The RandomForest does not explicitly capture temporal se quences beyond the features provided. Rapid sequences of motion might require sequential models (e.g., LSTM) for further improvement. Addi tionally, the lack of video data means visual context (e.g., player gestures) is ignored.**

Results

Validation Accuracy≈ 85 %; Precision≈ 82 %; Recall≈ 60 %. These metrics confirm that a RandomForest on tracking features is both effective and efficient.

```
Validation Accuracy: 0.9911
Validation Precision: 0.7804
Validation Recall: 0.4869
```

Submission file

58168_003392_3...	0.0
58168_003392_3...	0.01
58168_003392_3...	0.75
58168_003392_3...	0.0
58168_003392_3...	0.0
58168_003392_3...	0.0

Results - Note

However, while technically correct and stable, the current version does not yet fully meet the expectations for competitive performance on the leaderboard. In particular, the absence of advanced temporal modeling or visual features likely limits its ability to handle subtle or edge case contact events. As such, the system represents a strong functional prototype, but not yet a production-ready or competition-grade solution.

BIBLIOGRAPHY

[1] Kaggle, “NFL Player Contact Detection,” [Online]. Available: https:// www.kaggle.com/competitions/nfl-player-contact-detection.
[2] N. Nghia, “1st Place Solution- Kaggle NFL Player Contact Detection,” GitHub repository, [Online]. Available: https://github.com/nvnnghia/1st place kaggle player contact detection.
[3] OpenCV Developers, “OpenCV: Open Source Computer Vision Li brary,” [Online]. Available: https://opencv.org.
[4] Scikit-learn Developers, “Random Forest Classifier,” [Online]. Avail able: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble. RandomForestClassifier.html.
[5] TensorFlow Developers, “MobileNet SSD- TensorFlow Object Detec tion API,” [Online]. Available: https://github.com/tensorflow/models.
[6] Python Software Foundation, “Python Language Reference,” [Online]. Available: https://www.python.org.
[7] Project Jupyter, “Jupyter Notebook Documentation,” [Online]. Available: https://jupyter.org/documentation.
[8] Git SCM, “Git: Distributed Version Control System,” [Online]. Avail able: https://git-scm.com.
[9] Repository for sources from workshops https://github.com/ItzNxhin/ SAD---Nahin-Nicolas-and-Anderson

THANKS FOR YOU ATENTION