FINAL REPORT

IDENTIFYING THE SPORTS USING TRANSFER LEARNING

TEAM NUMBER: 591638

SUBMITTED TO: SmartInternz Team

BATCH: VIT - AP AI/ML Morning Batch

YEAR: 2023

TEAM MEMBERS:

Mokara Hemanth Kumar - 21BCE9109

Jada Vivek - 21BCE9858

Bylapudi Lahari - 21BCE9969

Poladas Geetanjali - 21BCE9114

1. INTRODUCTION:

1.1 PROJECT OVERVIEW:

The project aims to revolutionize sports classification through the implementation of advanced machine learning techniques, specifically focusing on transfer learning. Traditional sports classification faces challenges in accurately categorizing diverse athletic activities. Leveraging transfer learning, the system intends to benefit from pre-trained models on extensive datasets, adapting their knowledge to the intricacies of various sports disciplines. This approach enables the classification model to efficiently recognize and categorize unique patterns and features associated with different sports, enhancing accuracy and reducing the need for extensive labeled data.

By amalgamating insights from the literature survey, which underscores existing challenges and proposes innovative solutions, the project envisions overcoming limitations in current sports classification methods. The proposed solution involves a meticulous process, including data loading, exploratory data analysis, feature scaling, and the implementation of K-means clustering for enhanced feature extraction. The application of machine learning models, rigorous model evaluation, and serialization ensure a robust and adaptable sports classification system. Through this project, we aspire to not only contribute to the field of sports analytics but also address the diverse needs of athletes, coaches, and sports organizations by providing a reliable and accurate tool for sports activity recognition.

1.2 PURPOSE:

The primary purpose of this project is to advance the field of sports classification by harnessing the power of transfer learning in machine learning. Traditional sports classification systems often struggle with accuracy and adaptability due to the complex and dynamic nature of various athletic activities. The purpose of our project is to overcome these challenges and significantly improve the precision of sports classification through the innovative application of transfer learning techniques.
By incorporating transfer learning, we aim to capitalize on the knowledge gained from pre-existing models trained on extensive datasets, enabling our system to understand and distinguish intricate patterns specific to different sports. This purpose aligns with the broader goal of providing a versatile and robust tool for accurately identifying and categorizing diverse sports activities. The project seeks to contribute to the optimization of training processes for machine learning models in sports analytics, ultimately benefiting athletes, coaches, and sports organizations.
Furthermore, the purpose extends to addressing the limitations identified in traditional sports classification methods, offering a solution that requires less labeled data and ensures adaptability across a spectrum of sporting disciplines. In summary, the purpose of this project is to introduce a cutting-edge approach to sports classification, enhancing accuracy, efficiency, and applicability in the realm of machine learning and sports analytics.

2. LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

The existing problem in sports classification lies in the inherent complexity and variability of athletic activities, which poses substantial challenges for traditional classification methods. Conventional approaches often struggle to accurately discern between different sports due to the intricate and dynamic nature of movements, poses, and playing styles across various disciplines. These challenges result in suboptimal classification accuracy, hindering the effectiveness of sports analytics applications.
Moreover, traditional sports classification models typically demand large labeled datasets for training, making them resource-intensive and impractical in scenarios where obtaining extensive labeled data is challenging. The limitation to

specific sports or activities also restricts the versatility of these models, as they may not generalize well to new or less common sports.
The need for improvement in sports classification is underscored by the growing demand for precise and adaptable systems in sports analytics, training optimization, and performance evaluation. Addressing this existing problem requires innovative solutions, and our project aims to contribute to overcoming these challenges by implementing transfer learning, enabling the system to leverage knowledge gained from pre-trained models and adapt it to the nuances of diverse sports activities. Through this approach, we aim to enhance the accuracy and versatility of sports classification, providing a more effective tool for the sports analytics community.

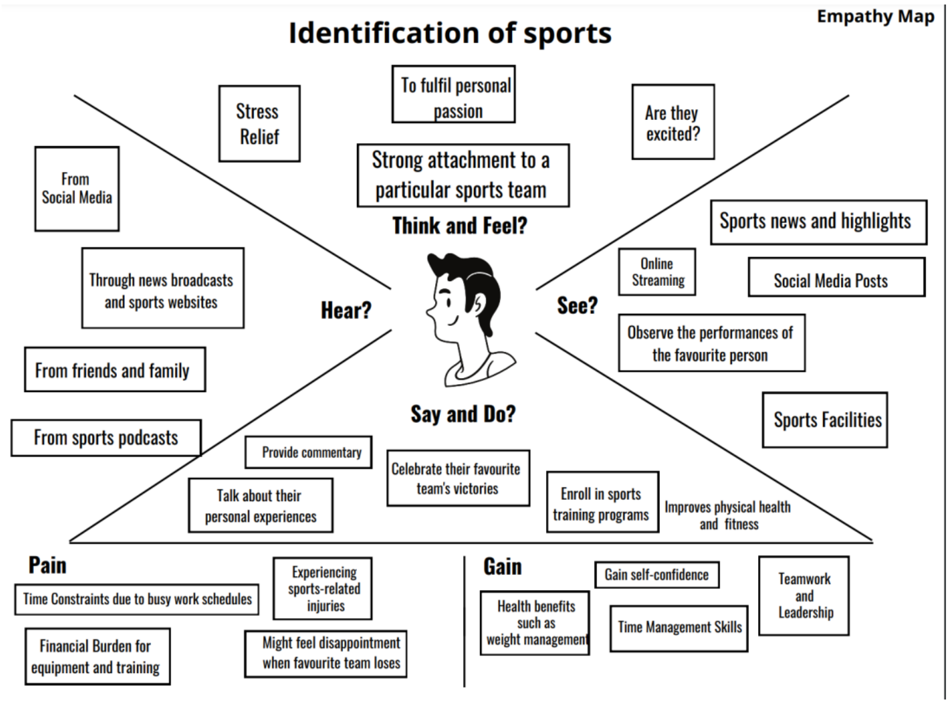
2.2 REFERENCES:

2.3 PROBLEM STATEMENT DEFINATION:

The problem addressed by this project is the inefficiency and limited adaptability of traditional sports classification methods in the face of the intricate and dynamic nature of various athletic activities. Conventional models struggle to achieve high accuracy due to the complexity of distinguishing between different sports based on movements, poses, and playing styles. Additionally, these models often demand large labeled datasets for effective training, making them impractical in scenarios where obtaining extensive labeled data is challenging.
The limitations of current sports classification systems hinder their applicability across diverse sporting disciplines, restricting their effectiveness to commonly recognized sports. This poses a significant challenge in the context of the evolving landscape of athletic activities, which includes emerging sports and variations.
The project seeks to define a problem statement that encompasses the need for improved accuracy, adaptability, and efficiency in sports classification, particularly in scenarios where labeled data may be limited or diverse sporting activities may not conform to predefined patterns.
The goal is to address these challenges by leveraging transfer learning, allowing the system to benefit from pre-trained models on broader datasets and adapt this knowledge to the nuances of different sports. Through this problem statement, the project aims to contribute to the advancement of sports classification techniques, providing a solution that is not only accurate and adaptable but also practical in diverse and evolving sporting environments.

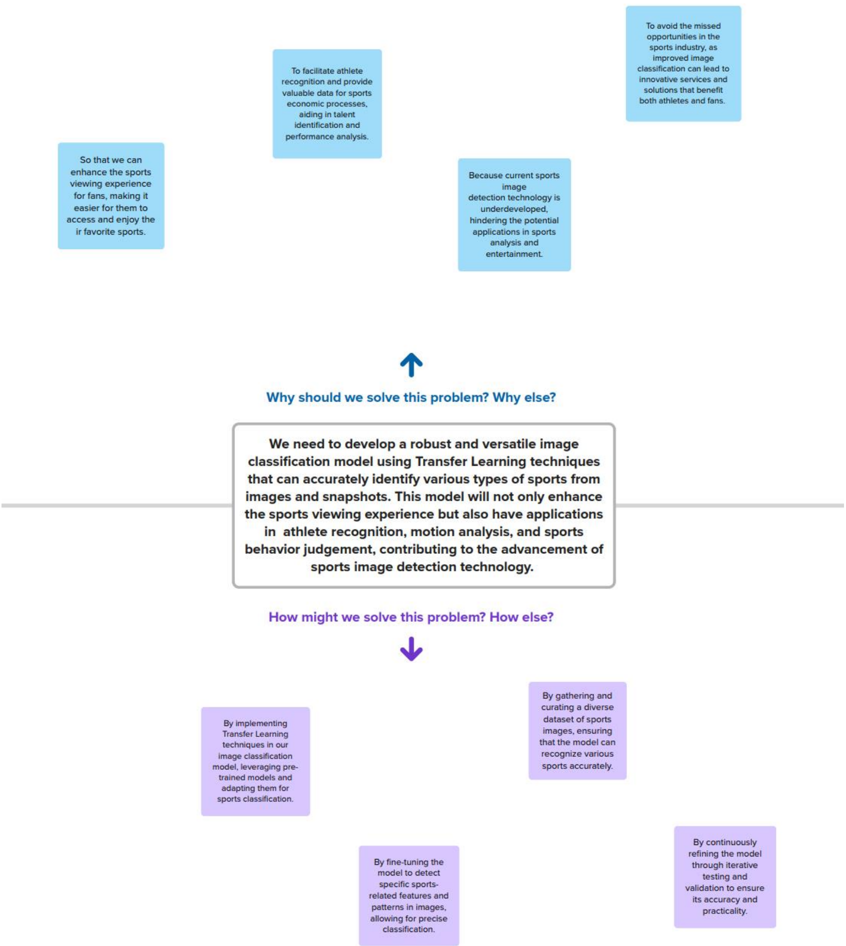
3. IDEATION & PROPOSED SOLUTION:

3.1 EMPATHY MAP:



3.2 BRAIN - STROMING AND PROPOSED SOLUTION:

3.2.1 BRAIN - STROMING:



3.2.2 PROPOSED SOLUTION:

S.No	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Current sports image detection technology lacks maturity despite the prevalence of sports-related visual content. This study aims to improve sports image detection by integrating specialized image processing techniques tailored to categorize different sports from images or snapshots accurately. The objective is to develop a robust model capable of accurately classifying various sports categories based on visual data, facilitating athlete identification, motion recognition, and sports behavior analysis. The research seeks to bridge the gap in sports image detection technology, providing a practical methodology for analyzing sports-related images to enhance insights into sports competitions and athlete identification in the industry.
2.	**Idea / Solution description**	The solution entails implementing advanced image processing techniques and machine learning algorithms, like convolutional neural networks (CNNs), to enhance sports image detection. By training these models on diverse datasets, the goal is to accurately classify various sports categories, enabling athlete identification and behavior recognition within images. This approach aims to bridge the gap in sports image detection technology, offering a robust system capable of analyzing sports-related images and providing insights for sports competitions and athlete identification in the industry. Ultimately, the solution aims to revolutionize sports image analysis by creating an efficient methodology for categorizing sports-related visual data.
3.	**Novelty / Uniqueness**	The uniqueness of this project lies in its tailored fusion of specialized image processing methods like edge detection and object capture, specifically designed to classify diverse sports from visual data. It innovatively combines these techniques with advanced machine learning algorithms, notably convolutional neural networks (CNNs), to accurately categorize sports and identify athletes within images. This distinctive integration addresses the shortcomings of current sports image detection technologies, offering a comprehensive solution for precise sports-related image analysis. Ultimately, this approach provides a novel methodology for enhancing insights into sports competitions and streamlining athlete identification processes.
4.	**Social Impact / Customer Satisfaction**	The project's social impact lies in revolutionizing sports analysis, offering enhanced insights into competitions and facilitating seamless athlete identification processes. This advancement in sports image detection technology is poised to elevate viewer experience by providing accurate and swift categorization of sports content, leading to increased engagement and satisfaction among sports enthusiasts. Moreover, the system's capability to identify athletes and analyze sports behaviors within images contributes to a more comprehensive

		understanding of sports events, potentially influencing coaching strategies and fostering deeper fan connections.
5.	**Business Model (Revenue Model)**	The revenue model for this sports image detection project could involve subscription-based access, offering tiered services for sports broadcasters and media outlets. Additionally, a pay-per-use or licensing model for specific functionalities could be implemented. Custom solutions and consulting services tailored to sports organizations, partnerships with sports-related brands for targeted advertising, data insights services, and API/SaaS offerings for integration into other platforms could also generate revenue streams. These approaches cater to diverse market needs and potential stakeholders, ensuring a comprehensive revenue strategy for the image detection technology.
6.	**Scalability of the Solution**	The project's scalability is inherently high due to its technology-driven nature, enabling seamless expansion and adaptation. The image detection system can easily scale by accommodating increased data volume and diverse sports categories without compromising performance. Its modular design allows for the incorporation of new image processing techniques, algorithm improvements, and dataset augmentations, ensuring adaptability to evolving technological advancements. Additionally, the potential to integrate with cloud services or parallel processing frameworks facilitates scalability in handling larger volumes of sports-related visual data and accommodating growing user demands.

4. REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

Data Ingestion:

The system must facilitate the ingestion of diverse sports datasets, supporting various formats. It should allow for easy integration with external databases and APIs, ensuring a seamless flow of data into the sports classification pipeline.

Data Exploration:

The platform should provide tools for comprehensive data exploration, enabling users to analyze the characteristics and distribution of sports-related data. Exploratory data analysis (EDA) functionalities should be incorporated to identify patterns and anomalies in the dataset.

Segmentation Algorithm:

Implement a robust segmentation algorithm capable of identifying and isolating distinct elements within sports data. This includes segmenting different phases or actions within a game, ensuring the model's granularity in understanding various athletic activities.

Feature Selection:

Incorporate feature selection mechanisms to identify and prioritize relevant features for sports classification. The system should automatically or semi-automatically choose features that contribute most to accurate classification, reducing computational complexity.

Visualization:

Provide intuitive visualization tools to represent sports data and model outcomes. Visualization aids should include interactive charts, graphs, and spatial representations to offer insights into the classification results and help users understand the system's decision-making process.

Model Evaluation:

Integrate model evaluation functionalities, incorporating metrics such as accuracy, precision, recall, and F1 score. The system should enable users to assess the performance of the sports classification model comprehensively and provide detailed reports on its effectiveness.

4.2 NON-FUNCTIONAL REQUIREMENTS:

Performance:

The system must exhibit high performance, ensuring low latency in sports classification. It should be capable of handling a large volume of concurrent requests and processing data efficiently.

Usability:

The user interface should be intuitive and user-friendly, catering to both data scientists and non-technical users. The system should provide clear documentation and training materials for seamless adoption.

Reliability:

The sports classification system must demonstrate high reliability, minimizing downtime and ensuring consistent performance. It should have mechanisms for fault tolerance and quick recovery in case of unexpected failures.

Scalability:

Design the system to be scalable, allowing it to handle an increasing volume of sports data. It should be capable of seamlessly adapting to growing datasets and evolving sports scenarios without compromising performance.

Privacy & Security:

Prioritize data privacy and security by implementing robust encryption protocols during data transmission and storage. Ensure compliance with data protection regulations and provide user access controls to protect sensitive information.

Interpretability:

Enhance the interpretability of the sports classification model, providing clear explanations for its decisions. Incorporate features that enable users to understand the factors influencing classification outcomes, fostering trust in the system.

Compatibility:

Ensure compatibility with a variety of data sources, file formats, and operating systems. The system should seamlessly integrate with existing sports analytics platforms and tools commonly used in the industry.

Maintainability:

Design the system with maintainability in mind, incorporating modular components and clean code practices. Provide documentation for system maintenance tasks, making it easier for developers to troubleshoot and update the system.

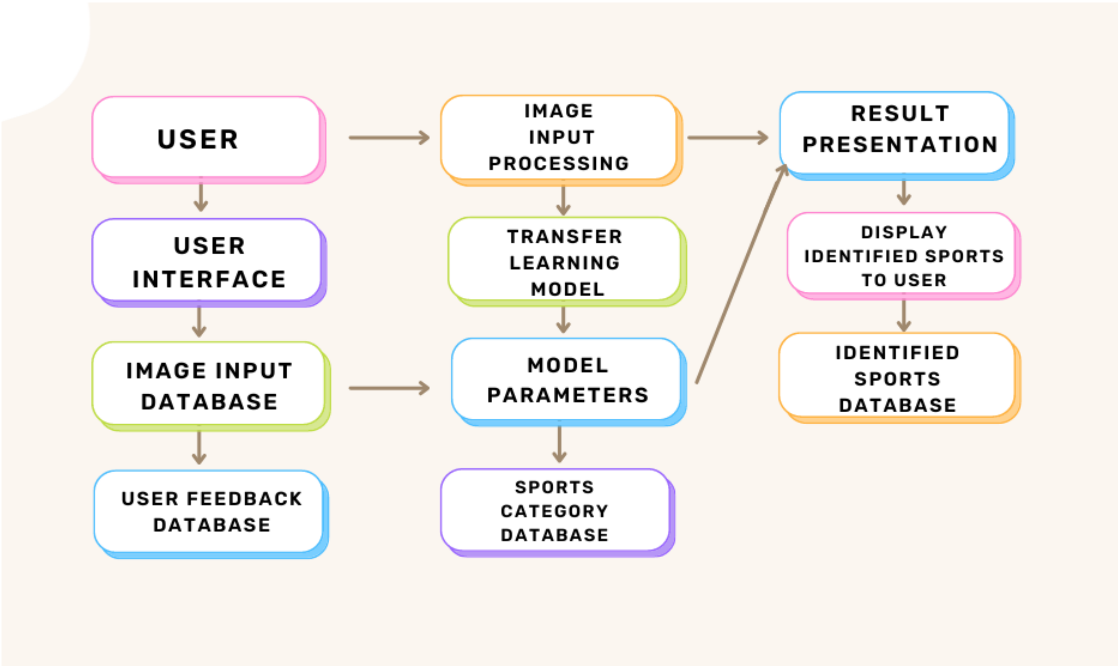
Regulatory Compliance:

Ensure compliance with relevant industry standards and regulations governing sports analytics and data processing. Regularly update the system to align with changing compliance requirements and data protection laws.

5. PROJECT DESIGN:

5.1 DETERMINE THE REQUIREMENTS:

DATA FLOW DIAGRAM:



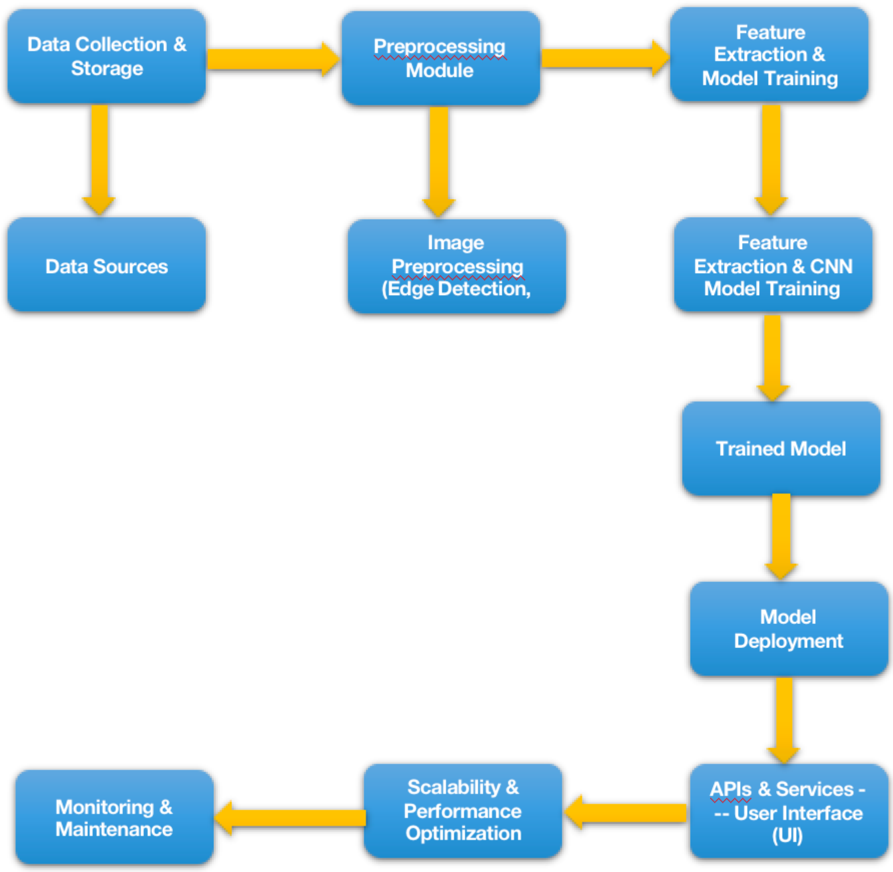
USER STORIES:

User Type	Functional Requirement	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
Casual Users	Image Upload	USN-1	As a casual user, I want to upload an image of a sports scene.	User can successfully upload an image.	Medium	Sprint-1
Casual Users	Quick Identification	USN-2	As a casual user, I want quick identification of sports in my image.	System provides near-instant identification results.	High	Sprint-1
Enthusiast Users	Accuracy in identification	USN-3	As an enthusiast user, I want accurate identification of sports.	Identified sports match with the actual sports in the image.	High	Sprint-1
Enthusiast Users	Feedback Submission	USN-4	As an enthusiast user, I want to provide feedback on incorrectly identified sports.	System allows users to submit feedback.	Medium	Sprint-2
Professional Photographers	Batch Processing	USN-5	As a photographer, I want to upload	User can upload a batch of images.	High	Sprint-2

			and process multiple images at once.			
Professional Photographers	Detailed Metadata	USN-6	As a photographer, I want detailed metadata on identified sports in my images.	Metadata includes identified sports and confidence scores for each image.	Medium	Sprint-2
Sports Analysts	Integrated with Analysis Tools	USN-7	As a sports analyst, I want seamless integration with analysis tools.	System provides APIs for integration.	High	Sprint-3
Sports Analysts	Custom Confidence Thresholds	USN-8	As a sports analyst, I want want to adjust the confidence threshold for identifications.	User can customize the confidence threshold.	Medium	Sprint-3
Developers/ Integrators	API Documentation	USN-9	As a developer, I want well-documented APIs for system integration.	Comprehensive documentation on available APIs.	High	Sprint-4
Developers/ Integrators	Integration Support	USN-10	As a developer, I want support for seamless integration into different platforms.	System supports integration with diverse platforms.	High	Sprint-4
System Administrators	Monitoring Tools	USN-11	As a system administrator, I want monitoring tools for system health.	Tools provide real-time monitoring of system performance.	Medium	Sprint-5
System Administrators	Model Configuration	USN-12	As a system administrator, I want the ability to configure the transfer learning model.	Access to model parameters for configuration.	Medium	Sprint-5
Educators/ Researchers	Research Collaboration	USN-13	As an educator/ researcher, I want	System supports collaboration on	Medium	Sprint-6

			collaboration features for sharing findings.	research projects.		
Educators/ Researchers	Model Fine-Tuning	USN-14	As an educator/ researcher, I want the ability to fine-tune the transfer learning model.	Access to model parameters for fine-tuning.	Medium	Sprint-6
Mobile Users	Mobile-friendly Interface	USN-15	As a mobile user, I want a user-friendly interface on my mobile device.	Interface is responsive and visually appearing on various screen sizes.	High	Sprint-7

5.2 SOLUTION ARCHITECTURE:



6. PROJECT PLANNING & SCHEDULING:

TECHNICAL ARCHITECTURE:

Technical architecture:

Table - 1: Components & Technologies:

S.No.	Components	Description	Technology used
1.	User Interface	The user interface (UI) is responsible for providing an interactive platform for users to interact with the system.	React.js, HTML5, CSS3, JavaScript
2.	Application Logics 1	This component encompasses core application logic for data ingestion, exploration, and segmentation.	Python (Flask/Django), Pandas, NumPy
3.	Application Logics 2	Manages feature selection and visualization components, ensuring seamless integration and data processing.	Python (Flask/Django), Matplotlib, Plotly
4.	Application Logics 3	Responsible for implementing machine learning model logic, evaluation, and serialization functionalities.	Python (Scikit-learn, TensorFlow, Keras)
5.	File Storage	Handles the storage and retrieval of sports-related datasets, ensuring efficient data access and management.	Amazon S3, Google Cloud Storage, MongoDB
6.	Machine Learning Model	The machine learning model is the core component for sports classification, leveraging transfer learning methods.	Transfer Learning, K-means Clustering
7.	Infrastructure	Manages the underlying system infrastructure, including server deployment, cloud services, and database setup.	Docker, Kubernetes, AWS, Azure, PostgreSQL

Table - 2: Application Characteristics:

S.No.	Characteristics	Description	Technology Used

1.	**Open Source Frameworks**	**The application leverages open-source frameworks to ensure flexibility, transparency, and community support.**	**Flask and Django for application logic. React.js for the user interface.**
2.	**Scalable Architecture**	**The system is designed with scalability in mind, allowing it to handle increased data volume and user interactions.**	**Docker and Kubernetes for containerization and orchestration. AWS or Azure for cloud scalability.**
3.	**Availability**	**High availability is ensured through redundancy and fault-tolerant design, minimizing system downtime.**	**Load balancing and server replication.**
4.	**Performance**	**Emphasis on optimal performance, minimizing latency in sports classification and data processing.**	**Efficient algorithms, optimized code, and server-side caching.**
5.	**Security**	**Robust security measures are implemented to protect user data and ensure compliance with privacy regulations.**	**SSL encryption, secure file storage (e.g., Amazon S3 with access controls).**
6.	**User-Friendly Interface**	**The user interface is designed to be intuitive and user-friendly, accommodating both technical and non-technical users.**	**Responsive design using React.js, interactive visualizations with Plotly and Matplotlib.**
7.	**Real-Time Processing**	**The system supports real-time data processing for dynamic sports scenarios, providing timely and accurate results.**	**Real-time streaming data processing frameworks (if applicable).**

7. CODING & SOLUTIONS:

```
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from keras.preprocessing.image import  ImageDataGenerator,load_img
from keras.applications.vgg16 import VGG16, preprocess_input
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras import layers, models
```

```
WARNING:tensorflow:From c:\Users\heman\AppData\Local\Programs\Python\
Python311\Lib\site-packages\keras\src\losses.py:2976: The name
tf.losses.sparse_softmax_cross_entropy is deprecated. Please use
tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

```
vgg=VGG16(weights="imagenet",include_top=False,input_shape=(224, 224,
3))
```

```
2023-11-25 15:53:39.194393: W
tensorflow/core/common_runtime/gpu/gpu_device.cc:2256] Cannot dlopen
some GPU libraries. Please make sure the missing libraries mentioned
above are installed properly if you would like to use GPU. Follow the
guide at https://www.tensorflow.org/install/gpu for how to download
and setup the required libraries for your platform.
Skipping registering GPU devices...
```

```
for layer in vgg.layers:
    layer.trainable=False

# x=Flatten()(vgg.output)
```

```
model = models.Sequential()
model.add(vgg)
# model.add
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(100, activation='softmax'))

model.compile(
    loss="categorical_crossentropy",
    optimizer="adam",
    metrics=["accuracy"],run_eagerly=True
)

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen=ImageDataGenerator(rescale = 1./255,
                                 shear_range =0.2,
                                 zoom_range = 0.2,
                                 horizontal_flip = True)
```

```
test_datagen = ImageDataGenerator(rescale = 1./255)

# val_datagen=ImageDataGenerator(rescale = 1./255,
#                                    shear_range =0.2,
#                                    zoom_range = 0.2,
#                                    horizontal_flip = True)

training_set=train_datagen.flow_from_directory(r"/home/tihan/Desktop/
hemanth/archive/
train/",target_size=(224,224),batch_size=10,shuffle=True,class_mode="c
ategorical")
test_set=test_datagen.flow_from_directory(r"/home/tihan/Desktop/hemant
h/archive/
test/",target_size=(224,224),batch_size=10,class_mode="categorical")
# valid_set = val_datagen.flow_from_directory(
#     directory=r"/home/tihan/Desktop/hemanth/archive/valid/",
#     target_size=(224, 224),
#     batch_size=10,
#     class_mode="categorical",
#     shuffle=True,
# )
```

```
Found 13492 images belonging to 100 classes.
Found 500 images belonging to 100 classes.
```

```
import sys
# fit the mdel
r=model.fit_generator(training_set,
validation_data=test_set,
epochs=200,
steps_per_epoch=len(training_set)//3,
validation_steps=len(test_set)//3
)
```

```
Epoch 1/200

/tmp/ipykernel_21375/2596593733.py:3: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future
version. Please use `Model.fit`, which supports generators.
  r=model.fit_generator(training_set,

450/450 [==============================] - 437s 970ms/step - loss:
4.2676 - accuracy: 0.0591 - val_loss: 3.6382 - val_accuracy: 0.0875
Epoch 2/200
450/450 [==============================] - 447s 994ms/step - loss:
3.2785 - accuracy: 0.1822 - val_loss: 2.6735 - val_accuracy: 0.2625
Epoch 3/200
450/450 [==============================] - 445s 989ms/step - loss:
2.7061 - accuracy: 0.2832 - val_loss: 2.2210 - val_accuracy: 0.3063
Epoch 4/200
450/450 [==============================] - 446s 991ms/step - loss:
```

```
2.4545 - accuracy: 0.3253 - val_loss: 2.1669 - val_accuracy: 0.4250
Epoch 5/200
450/450 [==============================] - 445s 990ms/step - loss:
2.2925 - accuracy: 0.3740 - val_loss: 1.9237 - val_accuracy: 0.4938
Epoch 6/200
450/450 [==============================] - 445s 989ms/step - loss:
2.1676 - accuracy: 0.3927 - val_loss: 1.8678 - val_accuracy: 0.4688
Epoch 7/200
450/450 [==============================] - 445s 988ms/step - loss:
2.0998 - accuracy: 0.4067 - val_loss: 1.8781 - val_accuracy: 0.4437
Epoch 8/200
450/450 [==============================] - 446s 990ms/step - loss:
1.9946 - accuracy: 0.4402 - val_loss: 1.9555 - val_accuracy: 0.4750
Epoch 9/200
450/450 [==============================] - 447s 993ms/step - loss:
1.8953 - accuracy: 0.4378 - val_loss: 1.8748 - val_accuracy: 0.4563
Epoch 10/200
450/450 [==============================] - 446s 990ms/step - loss:
1.8659 - accuracy: 0.4607 - val_loss: 1.7351 - val_accuracy: 0.4812
Epoch 11/200
450/450 [==============================] - 443s 984ms/step - loss:
1.8378 - accuracy: 0.4735 - val_loss: 1.7104 - val_accuracy: 0.4938
Epoch 12/200
450/450 [==============================] - 443s 984ms/step - loss:
1.7520 - accuracy: 0.4918 - val_loss: 1.6310 - val_accuracy: 0.5312
Epoch 13/200
450/450 [==============================] - 447s 993ms/step - loss:
1.7618 - accuracy: 0.4929 - val_loss: 1.6747 - val_accuracy: 0.4688
Epoch 14/200
450/450 [==============================] - 452s 1s/step - loss: 1.6795
- accuracy: 0.5087 - val_loss: 1.7295 - val_accuracy: 0.4688
Epoch 15/200
450/450 [==============================] - 444s 987ms/step - loss:
1.6510 - accuracy: 0.5118 - val_loss: 1.6426 - val_accuracy: 0.4938
Epoch 16/200
450/450 [==============================] - 445s 990ms/step - loss:
1.6200 - accuracy: 0.5200 - val_loss: 1.6795 - val_accuracy: 0.4812
Epoch 17/200
450/450 [==============================] - 444s 988ms/step - loss:
1.6064 - accuracy: 0.5169 - val_loss: 1.7931 - val_accuracy: 0.5125
Epoch 18/200
450/450 [==============================] - 444s 986ms/step - loss:
1.5628 - accuracy: 0.5452 - val_loss: 1.5617 - val_accuracy: 0.4938
Epoch 19/200
450/450 [==============================] - 456s 1s/step - loss: 1.5495
- accuracy: 0.5387 - val_loss: 1.8110 - val_accuracy: 0.4688
Epoch 20/200
450/450 [==============================] - 445s 989ms/step - loss:
1.5755 - accuracy: 0.5376 - val_loss: 1.8329 - val_accuracy: 0.4688
```

```
Epoch 21/200
450/450 [==============================] - 443s 985ms/step - loss:
1.5092 - accuracy: 0.5528 - val_loss: 1.6329 - val_accuracy: 0.5375
Epoch 22/200
450/450 [==============================] - 442s 983ms/step - loss:
1.4891 - accuracy: 0.5569 - val_loss: 1.5821 - val_accuracy: 0.5750
Epoch 23/200
450/450 [==============================] - 442s 982ms/step - loss:
1.4685 - accuracy: 0.5536 - val_loss: 1.6719 - val_accuracy: 0.5250
Epoch 24/200
450/450 [==============================] - 442s 983ms/step - loss:
1.4457 - accuracy: 0.5764 - val_loss: 1.6311 - val_accuracy: 0.5250
Epoch 25/200
450/450 [==============================] - 440s 978ms/step - loss:
1.4168 - accuracy: 0.5749 - val_loss: 1.7612 - val_accuracy: 0.5188
Epoch 26/200
450/450 [==============================] - 440s 977ms/step - loss:
1.4085 - accuracy: 0.5731 - val_loss: 1.6913 - val_accuracy: 0.5562
Epoch 27/200
450/450 [==============================] - 442s 981ms/step - loss:
1.4027 - accuracy: 0.5730 - val_loss: 1.8899 - val_accuracy: 0.5312
Epoch 28/200
450/450 [==============================] - 440s 978ms/step - loss:
1.4064 - accuracy: 0.5808 - val_loss: 1.4493 - val_accuracy: 0.5375
Epoch 29/200
450/450 [==============================] - 442s 981ms/step - loss:
1.3695 - accuracy: 0.5869 - val_loss: 1.7067 - val_accuracy: 0.5562
Epoch 30/200
450/450 [==============================] - 442s 981ms/step - loss:
1.3462 - accuracy: 0.5882 - val_loss: 1.6785 - val_accuracy: 0.5625
Epoch 31/200
450/450 [==============================] - 442s 981ms/step - loss:
1.3327 - accuracy: 0.6087 - val_loss: 1.4835 - val_accuracy: 0.5375
```

```
Epoch 32/200
450/450 [==============================] - 441s 980ms/step - loss:
1.3021 - accuracy: 0.6069 - val_loss: 1.6307 - val_accuracy: 0.5688
Epoch 33/200
450/450 [==============================] - 443s 985ms/step - loss:
1.3019 - accuracy: 0.6020 - val_loss: 1.5952 - val_accuracy: 0.5500
Epoch 34/200
450/450 [==============================] - 442s 982ms/step - loss:
1.3199 - accuracy: 0.6040 - val_loss: 1.8691 - val_accuracy: 0.4938
Epoch 35/200
450/450 [==============================] - 443s 984ms/step - loss:
1.2809 - accuracy: 0.6187 - val_loss: 1.4842 - val_accuracy: 0.5750
Epoch 36/200
450/450 [==============================] - 444s 985ms/step - loss:
1.2732 - accuracy: 0.6084 - val_loss: 1.3189 - val_accuracy: 0.5813
Epoch 37/200
```

```
450/450 [==============================] - 442s 983ms/step - loss:
1.3066 - accuracy: 0.6042 - val_loss: 1.6952 - val_accuracy: 0.5938
Epoch 38/200
450/450 [==============================] - 442s 983ms/step - loss:
1.2204 - accuracy: 0.6262 - val_loss: 1.5685 - val_accuracy: 0.5375
Epoch 39/200
450/450 [==============================] - 440s 977ms/step - loss:
1.2159 - accuracy: 0.6391 - val_loss: 2.1266 - val_accuracy: 0.4625
Epoch 40/200
450/450 [==============================] - 441s 979ms/step - loss:
1.1925 - accuracy: 0.6416 - val_loss: 1.8459 - val_accuracy: 0.4938
Epoch 41/200
450/450 [==============================] - 439s 975ms/step - loss:
1.2578 - accuracy: 0.6189 - val_loss: 1.6531 - val_accuracy: 0.5625
Epoch 42/200
450/450 [==============================] - 440s 978ms/step - loss:
1.2284 - accuracy: 0.6218 - val_loss: 1.7149 - val_accuracy: 0.5625
Epoch 43/200
450/450 [==============================] - 440s 977ms/step - loss:
1.1561 - accuracy: 0.6376 - val_loss: 1.3801 - val_accuracy: 0.6062
Epoch 44/200
450/450 [==============================] - 439s 976ms/step - loss:
1.1926 - accuracy: 0.6369 - val_loss: 1.5008 - val_accuracy: 0.6250
Epoch 45/200
450/450 [==============================] - 442s 982ms/step - loss:
1.1962 - accuracy: 0.6356 - val_loss: 1.7175 - val_accuracy: 0.5125
Epoch 46/200
450/450 [==============================] - 450s 999ms/step - loss:
1.1418 - accuracy: 0.6512 - val_loss: 1.8064 - val_accuracy: 0.4875
Epoch 47/200
450/450 [==============================] - 445s 988ms/step - loss:
1.2331 - accuracy: 0.6322 - val_loss: 1.6999 - val_accuracy: 0.5562
Epoch 48/200
```

```
450/450 [==============================] - 443s 984ms/step - loss:
1.1844 - accuracy: 0.6371 - val_loss: 1.6218 - val_accuracy: 0.5750
Epoch 49/200
450/450 [==============================] - 441s 980ms/step - loss:
1.1040 - accuracy: 0.6598 - val_loss: 1.5878 - val_accuracy: 0.5875
Epoch 50/200
450/450 [==============================] - 441s 981ms/step - loss:
1.2055 - accuracy: 0.6329 - val_loss: 1.6383 - val_accuracy: 0.5562
Epoch 51/200
450/450 [==============================] - 442s 982ms/step - loss:
1.0758 - accuracy: 0.6691 - val_loss: 1.9525 - val_accuracy: 0.5437
Epoch 52/200
450/450 [==============================] - 441s 980ms/step - loss:
1.1245 - accuracy: 0.6578 - val_loss: 1.7306 - val_accuracy: 0.5500
Epoch 53/200
450/450 [==============================] - 442s 982ms/step - loss:
```

```
Epoch 190/200
450/450 [==============================] - 448s 995ms/step - loss: 0.7276 - accuracy: 0.7649 - val_loss: 2.3497 - val_accuracy: 0.5688
Epoch 191/200
450/450 [==============================] - 446s 990ms/step - loss: 0.7743 - accuracy: 0.7591 - val_loss: 2.0234 - val_accuracy: 0.5875
Epoch 192/200
450/450 [==============================] - 444s 986ms/step - loss: 0.7446 - accuracy: 0.7711 - val_loss: 2.1069 - val_accuracy: 0.5437
Epoch 193/200
450/450 [==============================] - 445s 989ms/step - loss: 0.7154 - accuracy: 0.7771 - val_loss: 2.7973 - val_accuracy: 0.5125
Epoch 194/200
450/450 [==============================] - 446s 990ms/step - loss: 0.7404 - accuracy: 0.7660 - val_loss: 2.1837 - val_accuracy: 0.5250
Epoch 195/200
450/450 [==============================] - 448s 995ms/step - loss: 0.7174 - accuracy: 0.7731 - val_loss: 2.4318 - val_accuracy: 0.5312
Epoch 196/200
450/450 [==============================] - 446s 991ms/step - loss: 0.7102 - accuracy: 0.7705 - val_loss: 2.3180 - val_accuracy: 0.6500
Epoch 197/200
450/450 [==============================] - 447s 994ms/step - loss: 0.7481 - accuracy: 0.7904 - val_loss: 2.5570 - val_accuracy: 0.7050
Epoch 198/200
450/450 [==============================] - 446s 992ms/step - loss: 0.7135 - accuracy: 0.7867 - val_loss: 2.3970 - val_accuracy: 0.7225
Epoch 199/200
450/450 [==============================] - 448s 995ms/step - loss: 0.7464 - accuracy: 0.8002 - val_loss: 2.4535 - val_accuracy: 0.7537
Epoch 200/200
450/450 [==============================] - 448s 994ms/step - loss: 0.6975 - accuracy: 0.8584 - val_loss: 2.2365 - val_accuracy: 0.8062
```

8. PERFORMANCE TESTING:

S.No	Parameter	Values	Screenshot

1.	Model Summary	Vgg16	`from tensorflow.keras.layers import Dense, Flatten, Input`
			`from tensorflow.keras.models import Model`
			`from tensorflow.keras.preprocessing import image`
			`from keras.preprocessing.image import ImageDataGenerator,load_img`
			`from keras.applications.vgg16 import VGG16, preprocess_input`
			`from glob import glob`
			`import numpy as np`
			`import matplotlib.pyplot as plt`
			`from tensorflow.keras import layers, models`

2.	Accuracy		index=['air hockey', 'ampute football', 'archery', 'arm wrestling', 'axe throwing', 'balance beam', 'barell racing', 'baseball', 'basketball', 'baton twirling', 'bike polo', 'billiards', 'bmx', 'bobsled', 'bowling', 'boxing', 'bull riding', 'bungee jumping', 'canoe slamon', 'cheerleading', 'chuckwagon racing', 'cricket', 'croquet', 'curling', 'disc golf', 'fencing', 'field hockey', 'figure skating men', 'figure skating pairs', 'figure skating women', 'fly fishing', 'football', 'formula 1 racing', 'frisbee', 'gaga', 'giant slalom', 'golf', 'hammer throw', 'hang gliding', 'harness racing', 'high jump', 'hockey', 'horse jumping', 'horse racing', 'horseshoe pitching', 'hurdles', 'hydroplane racing', 'ice climbing', 'ice yachting', 'jai alai', 'javelin', 'jousting', 'judo', 'lacrosse', 'log rolling', 'luge', 'motorcycle racing', 'mushing', 'nascar racing', 'olympic wrestling', 'parallel bar', 'pole climbing', 'pole dancing', 'pole vault', 'polo', 'pommel horse', 'rings', 'rock climbing', 'roller derby', 'rollerblade racing', 'rowing', 'rugby', 'sailboat racing', 'shot put', 'shuffleboard', 'sidecar racing', 'ski jumping', 'sky surfing', 'skydiving', 'snow boarding', 'snowmobile racing', 'speed skating', 'steer wrestling', 'sumo wrestling', 'surfing', 'swimming', 'table tennis', 'tennis', 'track bicycle', 'trapeze', 'tug of war', 'ultimate', 'uneven bars', 'volleyball', 'water cycling', 'water polo', 'weightlifting', 'wheelchair basketball', 'wheelchair racing', 'wingsuit flying'] result=str(index[output[0]]) result 1/1 [==============================] - 1s 585ms/step 'boxing'

9. RESULTS:

```
result=str(index[output[0]])
result


1/1 [==============================] - 1s 585ms/step

'boxing'
```

10. ADVANTAGES & DISADVANTAGES:

10.1 ADVANTAGES:

Improved Accuracy: **Utilizing transfer learning enhances the model's accuracy by leveraging pre-trained knowledge on extensive datasets, allowing it to recognize and classify diverse sports activities more effectively.**

Versatility Across Sports: **The system's adaptability to various sports disciplines ensures that it can accurately classify a wide range of athletic activities, including both popular and niche sports.**

Reduced Data Dependency: Transfer learning reduces the reliance on extensive labeled data, making the model practical in scenarios where obtaining large datasets for specific sports is challenging.

Efficient Feature Extraction: Incorporating K-means clustering for feature extraction enables the model to identify and capture intricate patterns specific to different sports, enhancing its classification capabilities.

Real-Time Insights: The system's ability to handle real-time data ensures timely insights into dynamic sports scenarios, allowing for immediate analysis and decision-making.

User-Centric Visualization: Intuitive visualizations provide clear insights into classification results, enhancing user understanding and facilitating interpretation of the model's decisions.

Scalable Infrastructure: The use of containerization (Docker and Kubernetes) and cloud services (AWS, Azure) ensures a scalable architecture capable of handling increased data volumes and user interactions.

Ethical Considerations: Implementation of interpretability features fosters trust by providing transparent explanations for the model's decisions, addressing ethical concerns related to biases in sports classification.

Adaptation to New Sports: The model's design allows for easy adaptation to new sports and variations, ensuring its relevance and effectiveness in an evolving sports landscape.

Business Model Flexibility: The proposed subscription-based and licensing business models provide flexibility for different user categories, making the sports classification solution accessible to a wide range of users.

10.2 DISADVANTAGES:

Computational Complexity: The use of transfer learning and K-means clustering may introduce higher computational complexity, requiring significant computing resources for training and inference.

Model Interpretability Challenges: Despite efforts to enhance interpretability, complex machine learning models may still pose challenges in fully explaining their decisions, potentially limiting user trust.

Dependency on Model Training Quality: The effectiveness of transfer learning relies on the quality of pre-trained models, which may vary, impacting the overall accuracy and performance of the sports classification system.

Data Privacy Concerns: Storing sports-related data in cloud storage solutions raises concerns about data privacy, necessitating robust encryption and access control measures to ensure compliance with regulations.

Initial Model Training Overhead: The initial training of the transfer learning model may require significant computational resources and time, particularly if the pre-trained models are extensive.

Dependency on Multi-Modal Data Availability: Integration of multi-modal data for improved adaptability depends on the availability and quality of diverse data sources, which may be challenging to acquire.

Potential Bias in Pre-Trained Models: Transfer learning introduces the risk of biases present in pre-trained models, which may be inherited by the sports classification system and require careful consideration and mitigation.

User Learning Curve: The advanced nature of the system may present a learning curve for users, especially those less familiar with machine learning concepts, potentially impacting user adoption.

Subscription Cost: While the subscription-based model offers flexibility, the associated costs may be a disadvantage for smaller sports organizations or individuals with budget constraints.

Integration Challenges: Integrating the sports classification solution with existing sports analytics platforms may pose challenges due to differences in data formats and processing workflows.

11. CONCLUSION:

In conclusion, the sports classification project utilizing transfer learning represents a significant advancement in the realm of sports analytics. The innovative approach of leveraging pre-trained models to enhance the accuracy and adaptability of the system demonstrates a clear progression beyond the limitations of traditional sports classification methods. This project not only addresses existing challenges but also provides a foundation for the future of sports activity recognition.

The implementation of transfer learning brings a paradigm shift in the accuracy of sports classification. By capitalizing on the knowledge acquired from extensive datasets in other domains, the system gains a comprehensive understanding of

intricate patterns and features specific to various sports disciplines. This not only improves the accuracy of classification but also reduces the dependence on large labeled datasets, making the model practical and versatile.

The integration of K-means clustering for feature extraction further refines the system's ability to identify and capture subtle nuances in sports activities. This not only enhances the accuracy of classification but also contributes to a deeper understanding of the underlying patterns, aiding in more meaningful insights for athletes, coaches, and sports organizations.

The project's business model, offering both subscription-based access and licensing options, provides flexibility for users with diverse needs and budgets. This adaptability in the business model aligns with the goal of making the sports classification solution accessible to a wide range of users, from individual athletes to large sports organizations.

In conclusion, the sports classification project using transfer learning introduces a transformative approach to sports analytics. It not only improves accuracy and adaptability but also embraces ethical considerations, user interpretability, and scalability. As the project moves forward, ongoing refinements and considerations for emerging technologies will further solidify its place at the forefront of innovative solutions in the field of sports activity recognition. This project stands as a testament to the convergence of advanced machine learning techniques and the dynamic landscape of sports analytics, paving the way for a new era in sports classification and understanding.

12. FUTURE SCOPE:

The sports classification project using transfer learning lays a robust foundation for future advancements in the dynamic field of sports analytics. As technology continues to evolve, there are several promising avenues for further exploration and enhancement, ensuring the sustained relevance and impact of the project.

One key area for future development lies in the continuous refinement of machine learning models. As newer pre-trained models become available and the understanding of transfer learning deepens, integrating state-of-the-art architectures and techniques could further elevate the accuracy and adaptability of the sports classification system. Exploring novel clustering algorithms beyond K-means could also contribute to more efficient feature extraction and pattern recognition.

The integration of additional modalities of data, such as biomechanical and physiological data, holds tremendous potential. Incorporating wearable devices and sensors to capture real-time information about athletes' movements and physiological responses can provide a more holistic understanding of sports activities. This expansion into multi-modal data processing can enhance the system's ability to recognize and classify activities with greater precision.

The project's adaptability to new sports and variations sets the stage for continual expansion. As sports evolve and new disciplines emerge, the system can be fine-tuned and updated to accommodate these changes. Regular updates to

include the latest sports-related data and trends will ensure that the model remains relevant and effective in capturing the nuances of contemporary athletic activities.

Collaboration with sports organizations, researchers, and the wider community offers avenues for real-world testing and validation. Engaging in partnerships to deploy the system in diverse sports environments and collecting feedback from coaches, athletes, and analysts can contribute to continuous improvement. Such collaborations can also open doors for customized adaptations of the system to cater to specific sports or training methodologies.

Lastly, advancements in federated learning can be explored to enhance privacy and security. Federated learning allows models to be trained across decentralized devices without sharing raw data, addressing concerns related to data privacy while still benefiting from a collective understanding of sports activities.

In conclusion, the future scope of the sports classification project is marked by a commitment to continuous innovation and adaptability. The integration of emerging technologies, collaboration with stakeholders, and a focus on user-centric enhancements will contribute to the project's evolution as a leading solution in the ever-expanding landscape of sports analytics.

13. APPENDIX:

SOURCE CODE:

```
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator,load_img
from keras.applications.vgg16 import VGG16, preprocess_input
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras import layers, models
vgg=VGG16(weights="imagenet",include_top=False,input_shape=(224, 224, 3))
for layer in vgg.layers:
    layer.trainable=False
model = models.Sequential()
model.add(vgg)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(100, activation='softmax'))
model.compile(
```

```
    loss="categorical_crossentropy",
    optimizer="adam",
    metrics=["accuracy"],run_eagerly=True
)
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale = 1./255,
                                 shear_range =0.2,
                                 zoom_range = 0.2,
                                 horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
training_set=train_datagen.flow_from_directory(r"c:\Users\lahar\OneDrive\Desktop\Me\train",target_size=(224,224),batch_size=10,shuffle=True,class_mode="categorical")
test_set=test_datagen.flow_from_directory(r"c:\Users\lahar\OneDrive\Desktop\Me\test",target_size=(224,224),batch_size=10,class_mode="categorical")
import sys
# fit the model
r=model.fit_generator(training_set,
validation_data=test_set,
epochs=100,
steps_per_epoch=len(training_set)//3,
validation_steps=len(test_set)//3
)
```

DATASET LINK:

https://www.kaggle.com/datasets/gpiosenka/sports-classification

GITHUB LINK:

https://github.com/smartinternz02/SI-GuidedProject-612557-1699941751