**VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY**

**THE INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**THESIS REPORT**

**Building a Sport Classification Website Using Images**

**By**

**Student Name: NGUYEN VIET ANH**

**ID: ITDSIU18027**

**Advisor: Dr MAI HOANG BAO AN**

**A thesis submitted to the School of Computer Science and Engineering**

**In partial fulfilment of the requirements for the degree of Bachelor of Data Science**

**Ho Chi Minh City, Vietnam**

**2023**

**Building a Sport Classifier Website Using Images**

APPROVED BY: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **ACKNOWLEDGEMENTS**

I am deeply thankful to my advisor, Dr Mai Hoang Bao An, for his invaluable guidance. I also extend my gratitude to the Vietnam National University, the International University, School of Computer Science and Engineering, for their support.

I appreciate the encouragement and understanding of my family and friends throughout this journey. Special thanks to the participants and volunteers who contributed to this research.

This thesis would not have been possible without your support.

# **TABLE OF CONTENTS**

[**ACKNOWLEDGEMENTS** 3](#_Toc157588927)

[**TABLE OF CONTENTS** 4](#_Toc157588928)

[**LIST OF FIGURES** 8](#_Toc157588929)

[**LIST OF TABLES** 9](#_Toc157588930)

[**ABSTRACT** 10](#_Toc157588931)

[**CHAPTER 1: INTRODUCTION** 11](#_Toc157588932)

[**1.** **The Background of the Project:** 11](#_Toc157588933)

[**2.** **Problem Statement:** 11](#_Toc157588934)

[**3.** **Scope and Objectives:** 11](#_Toc157588935)

[***3.1 Scope:*** 11](#_Toc157588936)

[***3.2 Objectives:*** 12](#_Toc157588937)

[**4.** **Assumption and Solution:** 12](#_Toc157588938)

[**5.** **Resource Allocation:** 13](#_Toc157588939)

[***5.1 Budget:*** 13](#_Toc157588940)

[***5.2 Software Resources:*** 13](#_Toc157588941)

[*5.2.1 Development Software:* 13](#_Toc157588942)

[*5.2.2 Testing Software:* 13](#_Toc157588943)

[*5.2.3 Deployment Software:* 13](#_Toc157588944)

[***5.3 Hardware Resources:*** 13](#_Toc157588945)

[*5.3.1 Development Hardware:* 13](#_Toc157588946)

[*5.3.2 Testing Hardware:* 13](#_Toc157588947)

[**6.** **Thesis Structure Overview:** 14](#_Toc157588948)

[**CHAPTER 2: LITERATURE REVIEW** 15](#_Toc157588949)

[**1.** **Introduction to Image Classification in Sports:** 15](#_Toc157588950)

[**2.** **Foundations of Image Classification:** 15](#_Toc157588951)

[***2.1 Concept of Image Classification:*** 15](#_Toc157588952)

[***2.2 Steps of Image Classification:*** 16](#_Toc157588953)

[***2.3 Techniques:*** 17](#_Toc157588954)

[*2.3.1 Supervised Classification:* 17](#_Toc157588955)

[a. Concept: 18](#_Toc157588956)

[b. Steps: 18](#_Toc157588957)

[c. Principles: 19](#_Toc157588958)

[d. Algorithms: 19](#_Toc157588959)

[e. Advantages: 20](#_Toc157588960)

[f. Disadvantages: 20](#_Toc157588961)

[*2.3.1 Unsupervised Classification:* 21](#_Toc157588962)

[a. Concept: 21](#_Toc157588963)

[b. Steps: 21](#_Toc157588964)

[c. Principles: 21](#_Toc157588965)

[d. Algorithms: 21](#_Toc157588966)

[e. Advantages: 22](#_Toc157588967)

[f. Disadvantages: 22](#_Toc157588968)

[**3.** **Previous Applications in Image Classification in Sports:** 23](#_Toc157588969)

[***3.1 Hudl:*** 23](#_Toc157588970)

[***3.2 Catapult:*** 23](#_Toc157588971)

[**4.** **Existing Challenges and Limitations:** 24](#_Toc157588972)

[***4.1 Data Variability:*** 24](#_Toc157588973)

[***4.2 Real-Time Processing:*** 24](#_Toc157588974)

[***4.3 Ambiguity in Player Actions:*** 24](#_Toc157588975)

[***4.4 Wearable Technology Integration:*** 24](#_Toc157588976)

[***4.5 Data Privacy and Security:*** 25](#_Toc157588977)

[***4.6 Catapult’s Local Positioning System:*** 25](#_Toc157588978)

[***4.7 Cost and Accessibility:*** 25](#_Toc157588979)

[**5.** **Relevance to the thesis:** 25](#_Toc157588980)

[**CHAPTER 3: METHODOLOGY** 27](#_Toc157588981)

[**1.** **Dataset**: 27](#_Toc157588982)

[**2.** **Pre-processing:** 27](#_Toc157588983)

[***2.1 Setting Batch and Image Size:*** 27](#_Toc157588984)

[*2.1.1 Purposes:* 27](#_Toc157588985)

[a. Batch Size Configuration: 27](#_Toc157588986)

[b. Image Size Configuration: 28](#_Toc157588987)

[2.1.2 Explanation: 28](#_Toc157588988)

[***2.2 ImageDataGenerator:*** 28](#_Toc157588989)

[*2.2.1 Purposes:* 28](#_Toc157588990)

[*2.2.2 Explanation:* 28](#_Toc157588991)

[***2.3 Creating Data Generators for Different Sets:*** 29](#_Toc157588992)

[*2.3.1 Purposes:* 29](#_Toc157588993)

[a. Automated Data Handling: 29](#_Toc157588994)

[b. Memory Management: 29](#_Toc157588995)

[c. Consistent Pre-processing: 29](#_Toc157588996)

[d. Label Handling and Encoding: 29](#_Toc157588997)

[e. Setup: 29](#_Toc157588998)

[*2.3.2 Explanation:* 29](#_Toc157588999)

[**CHAPTER 4: IMPLEMENTATION AND RESULTS** 31](#_Toc157589000)

[**1.** **Architectures of CNN:** 31](#_Toc157589001)

[***1.1 EfficientNet B0*** 31](#_Toc157589002)

[***1.2 VGG16*** 36](#_Toc157589003)

[***1.3 ResNet50*** 38](#_Toc157589004)

[***1.4 MobileNetV3 Large*** 39](#_Toc157589005)

[***1.5 Inception V3*** 40](#_Toc157589006)

[2. **Results**: 41](#_Toc157589007)

[***2.1 From the "Original Dataset Results" table:*** 41](#_Toc157589008)

[***2.2 From the "Post-Optimisation Dataset Results" table:*** 41](#_Toc157589009)

[***2.3 Overall Analysis:*** 42](#_Toc157589010)

[**3.** **Web Server:** 42](#_Toc157589011)

[4. **Features**: 43](#_Toc157589012)

[***4.1 Upload Functionality:*** 43](#_Toc157589013)

[***4.2 Drag and Drop:*** 43](#_Toc157589014)

[***4.3 Sample Images:*** 44](#_Toc157589015)

[***4.4 Sample Image Display:*** 44](#_Toc157589016)

[***4.5 Design and Layout:*** 44](#_Toc157589017)

[**CHAPTER 5: EVALUATION AND DISCUSSION** 45](#_Toc157589018)

[1. **Evaluation**: 45](#_Toc157589019)

[***1.1 Accuracy of Image Classification:*** 45](#_Toc157589020)

[***1.2 UI and UX:*** 45](#_Toc157589021)

[***1.3 Training Data and Model Performance:*** 45](#_Toc157589022)

[***1.4 Cost-effectiveness:*** 45](#_Toc157589023)

[***1.5 Comparison with Existing Models:*** 46](#_Toc157589024)

[*1.5.1 Original Dataset Results:* 46](#_Toc157589025)

[*1.5.2 Post-Optimization Results:* 46](#_Toc157589026)

[***1.6 Future Potential:*** 46](#_Toc157589027)

[***1.7 Performance Metrics:*** 47](#_Toc157589028)

[***1.8 Technical Challenges:*** 47](#_Toc157589029)

[2. **Discussion**: 48](#_Toc157589030)

[**CHAPTER 6: CONCLUSION AND FUTURE WORK** 49](#_Toc157589031)

[1. **Conclusion**: 49](#_Toc157589032)

[**2.** **Future Work:** 49](#_Toc157589033)

[**REFERENCES** 50](#_Toc157589034)

[**APPENDICES** 52](#_Toc157589035)

# **LIST OF FIGURES**

[Figure 1: Digital Image Processing and Result Assessment 15](#_Toc157588236)

[Figure 2: Image Classification Flow Chart 16](#_Toc157588237)

[Figure 3: Steps of Image Classification 16](#_Toc157588238)

[Figure 4: Supervised Classification 17](#_Toc157588239)

[Figure 5: Steps of Supervised Classification 18](#_Toc157588240)

[Figure 6: Supervised Classification Stages 18](#_Toc157588241)

[Figure 7: Minimum Distance Classifier 19](#_Toc157588242)

[Figure 8: Unsupervised Classification 21](#_Toc157588243)

[Figure 9: K-means Clustering 22](#_Toc157588244)

[Figure 10: Hudl Website 23](#_Toc157588245)

[Figure 11: Catapult Website 23](#_Toc157588246)

[Figure 12: Original Data 27](#_Toc157588247)

[Figure 13: New Data 27](#_Toc157588248)

[Figure 14: Data Pre-processing 30](#_Toc157588249)

[Figure 15: EfficientNetB0 Network Architecture 31](#_Toc157588250)

[Figure 16: Data Preparation 32](#_Toc157588251)

[Figure 17: Model Loading and Layer Freezing 33](#_Toc157588252)

[Figure 18: Data Augmentation 33](#_Toc157588253)

[Figure 19: Model Training 34](#_Toc157588254)

[Figure 20: Visualisation 35](#_Toc157588255)

[Figure 21: Testing the Model 36](#_Toc157588256)

[Figure 22: F1 Calculation and Classification Report 36](#_Toc157588257)

[Figure 23: VGG16 Architecture 36](#_Toc157588258)

[Figure 24: VGG16 Model Loading and Freezing 37](#_Toc157588259)

[Figure 25: VGG16 Data Augmentation and Model Modification 37](#_Toc157588260)

[Figure 26: ResNet50 Architecture 38](#_Toc157588261)

[Figure 27: Mobilenet V3 Architecture 39](#_Toc157588262)

[Figure 28: Inception V3 Architecture 40](#_Toc157588263)

[Figure 29: Original Dataset Results 41](#_Toc157588264)

[Figure 30: Post-Optimisation Dataset Results 41](#_Toc157588265)

[Figure 31: Website's User Interface 42](#_Toc157588266)

[Figure 32: Website's Key Components and Features 43](#_Toc157588267)

[Figure 33: F1 Score Formula 47](#_Toc157588268)

# **LIST OF TABLES**

[Table 1: Budget Allocation 13](#_Toc157588269)

[Table 2: Terms of Glossary 52](#_Toc157588270)

# **ABSTRACT**

As technology progresses, the use of artificial intelligence (AI) in various applications has become increasingly prominent. This thesis examined the development and implementation of a web-based platform that employs AI to recognise and classify sports images. By utilising a Convolutional Neural Network (CNN) architecture, the system enables users to upload image files seamlessly from their devices or predefined datasets, providing a versatile and user-friendly experience.

The project implements an Agile approach in designing the system, which prioritises adaptability and iterative development. This methodology results in a website with an interactive interface that enables users to dynamically interact with a sports classification AI model. The primary objectives of this thesis are to develop a user-friendly platform, train a CNN model for precise sports classification, and assess performance metrics to evaluate the accuracy and efficiency of the model.

This study examines the hurdles faced while developing the system and presents effective solutions to improve its functionality. Moreover, it looks into potential avenues for future work, highlighting ways to broaden the use of AI in sports classification. Ultimately, this thesis makes a meaningful contribution to the expanding field of AI applications and offers a useful tool for users who seek automated sports recognition through image analysis.

# **CHAPTER 1: INTRODUCTION**

## **The Background of the Project:**

The landscape of sports classification has witnessed a transformative shift in recent years, driven by advancements in AI and information technologies. In the context of image recognition and further, the application of the CNN architecture has demonstrated remarkable potential, particularly in the automated identification of sports within images. Traditional methods for sports recognition are often time-consuming and subjective, highlighting the need for an innovative approach that leverages the capabilities of AI.

With the increasing ubiquity of AI technologies, particularly CNNs, there is a unique opportunity to develop a web-based platform that empowers users to classify sports effortlessly in images. This thesis seeks to address this opportunity by designing and implementing a user-friendly website. Users can seamlessly upload this information in .jpg files from their devices or utilise predefined data images for sports classification, thereby contributing to a more streamlined and efficient process.

In tandem with the technical aspects, the project adopted an Agile methodology in system design, embracing adaptability, and iterative development. This approach ensures responsiveness to evolving user needs and emerging challenges, aligned with the dynamic nature of AI development.

The overarching objective of this project is to not only provide a practical solution for sports enthusiasts and analysts, but also to contribute to the broader discourse on the intersection of AI and sports recognition. By delving into the challenges and opportunities presented by this technological landscape, this study aims to establish a foundation for the future of AI-based sports classification within the context of web-based applications.

## **Problem Statement:**

In contemporary society, the landscape of sports is faced with persistent challenges in awareness and analytics, hindering inclusive engagement and personalised performance enhancement. While technology has advanced, there remains a significant gap in recognizing and promoting lesser-known sports, limiting the accessibility of advanced analytics for athletes seeking tailored improvement. These challenges underscore the imperative for innovative solutions to bridge gaps in sports education, awareness, and performance optimization.

The proposed project addresses these societal challenges by developing a web-based platform leveraging CNN architecture for the automated classification of sports images. The platform aims to serve a dual purpose: enhancing awareness and education about a diverse range of sports and providing athletes with accessible advanced analytics for personalised performance improvement.

## **Scope and Objectives:**

### ***3.1 Scope:***

The scope of this endeavour encompasses the creation of a user-friendly website where users, spanning from athletes and enthusiasts to coaches, can effortlessly contribute images for comprehensive sports analysis. The project not only focuses on accurate image classification, but also adopts the Agile methodology in system design to ensure adaptability and responsiveness, fostering dynamic user interactions with the AI model.

### ***3.2 Objectives:***

The project aims to design and implement a user-friendly website dedicated to sports image classification, ensuring a seamless and intuitive experience for users. The development and training of the CNN model will be carried out with a sharp focus on achieving superior accuracy in recognizing a diverse range of sports. The adoption of Agile methodology in system design is integral to creating a flexible and responsive platform, allowing for adaptability and iterative development. The implementation of an interactive user interface is a key objective, to enhance user engagement and overall experience. The project will rigorously evaluate the AI model's performance through metrics, with a specific emphasis on accuracy and efficiency. Addressing potential development challenges will be an ongoing effort, with a commitment to continually improving system functionality. Furthermore, the project will explore future enhancements, placing a particular focus on contributing to sports education and analytics, thereby aiming to advance the broader landscape of sports technology and knowledge.

## **Assumption and Solution:**

In the contemporary landscape, a prevailing challenge emerges from a significant lack of awareness and education regarding lesser-known sports, coupled with a scarcity of platforms dedicated to promoting diverse athletic activities. This dearth of knowledge is notably pronounced among the younger demographics, leading to limited awareness about a wide array of sports. The consequence is twofold: firstly, it inhibits youngsters from making informed choices about suitable sports, potentially hindering their engagement in physical activities; and secondly, the absence of diverse sports knowledge poses a barrier to fostering a healthy lifestyle and promoting sports among young athletes. This assumption underscores the critical need to address these awareness gaps, not only for the individual well-being of young enthusiasts but also for the broader promotion of sports within the younger generation.

Compounding the challenges in the sports domain, there exists a significant impediment characterised by limited access to advanced sports analytics for personalised improvement, coupled with the inadequacy of tools available to athletes and coaches for effective performance analysis. This scarcity hampers the ability of athletes and coaching professionals to access comprehensive insights tailored to individual performance metrics. The lack of sophisticated tools further exacerbates the situation, hindering their capacity to conduct in-depth analyses crucial for refining training strategies and optimising athletic performance. Consequently, athletes are deprived of the personalised feedback necessary for targeted improvement, while coaches face limitations in crafting tailored training programs. This dual challenge accentuates the pressing need for innovative solutions that democratise access to advanced sports analytics and equip athletes and coaches with effective tools, fostering a paradigm shift towards more informed and data-driven training methodologies.

In addressing the identified challenges, this thesis proposes a transformative solution harnessing the capabilities of AI. With the ongoing advancement and meticulous training of the AI model, it holds the potential to revolutionise the landscape of sports education and analytics. Through AI-driven educational content integration, the platform could dynamically curate and recommend various sports, specifically tailored to cater to the limited awareness among youngsters. This innovative approach not only broadens their sports knowledge but also empowers them to make informed choices about suitable physical activities, thereby promoting a healthier lifestyle. Additionally, the envisioned AI model can pave the way for advanced analytics features, offering athletes and coaches unparalleled access to personalised insights, performance trends, and targeted suggestions for improvement. By embracing AI, this thesis envisions a future where sports education becomes more inclusive, personalised, and technologically advanced, fostering a holistic approach to athletic development.

## **Resource Allocation:**

The thesis project prioritises efficient resource allocation, balancing budget considerations with the need for cutting-edge tools and software. Strategic investment in hardware, including high-performance servers and GPUs, aligns with scalability requirements for real-time sports data processing. This approach aims to optimise project outcomes while ensuring fiscal responsibility.

### ***5.1 Budget:***

|  |  |  |
| --- | --- | --- |
| **Categories** | **Quantity** | **Allocation** |
| Google Colab Pro (GPU) | 2 | $9.99 |
| Total |  | $19.98 |

Table 1: Budget Allocation

### ***5.2 Software Resources:***

#### *5.2.1 Development Software:*

* Programming Language: Jupyter Notebook.
* Machine Learning Frameworks: TensorFlow, Keras.
* Library: OS, TensorFlow, Keras, PIL (Python Image Library), Streamlit.
* Version Control and Project Management: GitHub.
* IDE: Google Colab, PyCharm, Jupyter Notebook.
* Web Technologies: HTML, CSS.

#### *5.2.2 Testing Software:*

* Google Colab.
* Streamlit.

#### *5.2.3 Deployment Software:*

* Streamlit Sharing.

### ***5.3 Hardware Resources:***

#### *5.3.1 Development Hardware:*

* Computers or laptops with sufficient processing power and memory.
* High-resolution displays for efficient coding.

#### *5.3.2 Testing Hardware:*

Devices for cross-browser and cross-platform testing, such as smartphones, tablets, and various operating systems.

## **Thesis Structure Overview:**

The thesis is organised into six chapters, each representing a distinct section of the document. The structure of the chapters is as follows:

* **Chapter 1: Introduction:** Introducing the background, the problem statement, scope, objectives, assumptions, and the proposed solution, this section furnishes an overview of the thesis topic, outlines its key points, and briefly presents underlying assumptions.
* **Chapter 2: Literature Review:** Describing the techniques employed by the website.
* **Chapter 3: Methodology:** Detailing the systematic approach used in developing the AI-driven sports image classification platform, incorporating the Convolutional Neural Network (CNN) architecture and Agile methodology for adaptability.
* **Chapter 4: Implementation and Results:** Detailing the execution of the website and the outcomes of the entire endeavour.
* **Chapter 5: Discussion and Evaluation:** Delineating the constraints associated with the results.
* **Chapter 6: Conclusion and Future Application:** Providing an overarching view of the thesis.

# **CHAPTER 2: LITERATURE REVIEW**

## **Introduction to Image Classification in Sports:**

In the realm of modern sports, the fusion of cutting-edge technology and athletic prowess has ushered in a new era of comprehensive analysis and performance enhancement. At the forefront of this intersection lies the compelling application of image classification, a transformative technique that lends unprecedented depth to the understanding of sporting events. Image classification, with its capacity to discern and interpret visual data, emerges as a powerful tool for redefining the landscape of sports analytics. This introduction marks the commencement of our exploration into the multifaceted role of image classification within the sporting domain, delving into its instrumental contributions to player performance evaluation, referee decision-making processes, biomechanics scrutiny, and beyond. As we embark on this journey, the pivotal role of image classification becomes apparent, propelling sports analysis into an era where technology harmonises seamlessly with the intricacies of athletic prowess.

## **Foundations of Image Classification:**

### ***2.1 Concept of Image Classification:***

Image classification in computer vision is a comprehensive process that involves analysing and categorising images based on their visual content. It includes the use of both classic and deep learning techniques to interpret complex visual data. Essential steps in this process include image pre-processing, feature extraction, and the final classification of images into predefined classes using algorithms like SVMs, KNN, and CNNs. The field also extends to specific applications like land cover classification in remote sensing, utilising spectral and spatial pattern recognition. This process can be either supervised, where the model is trained with labelled data, or unsupervised, where the model discerns patterns without pre-labelled data. Overall, image classification is pivotal in various applications, from identifying objects in photographs to analysing terrain in satellite imagery.

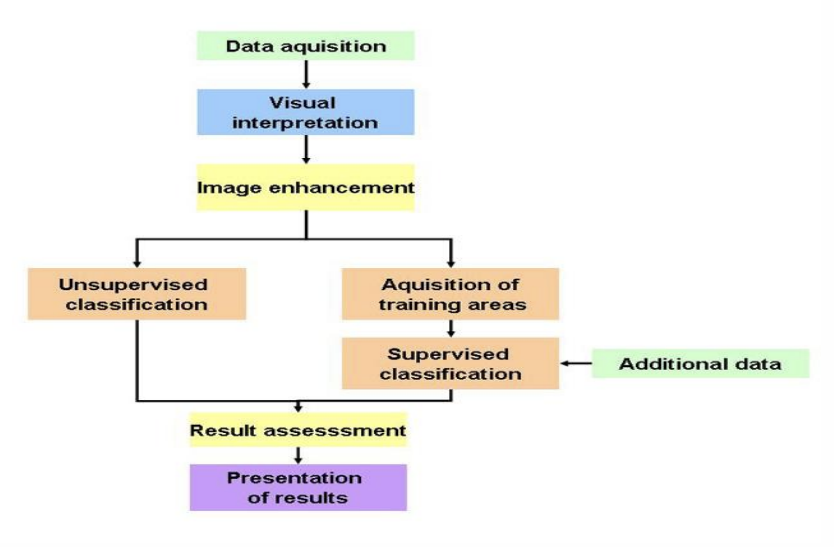


Figure 1: Digital Image Processing and Result Assessment

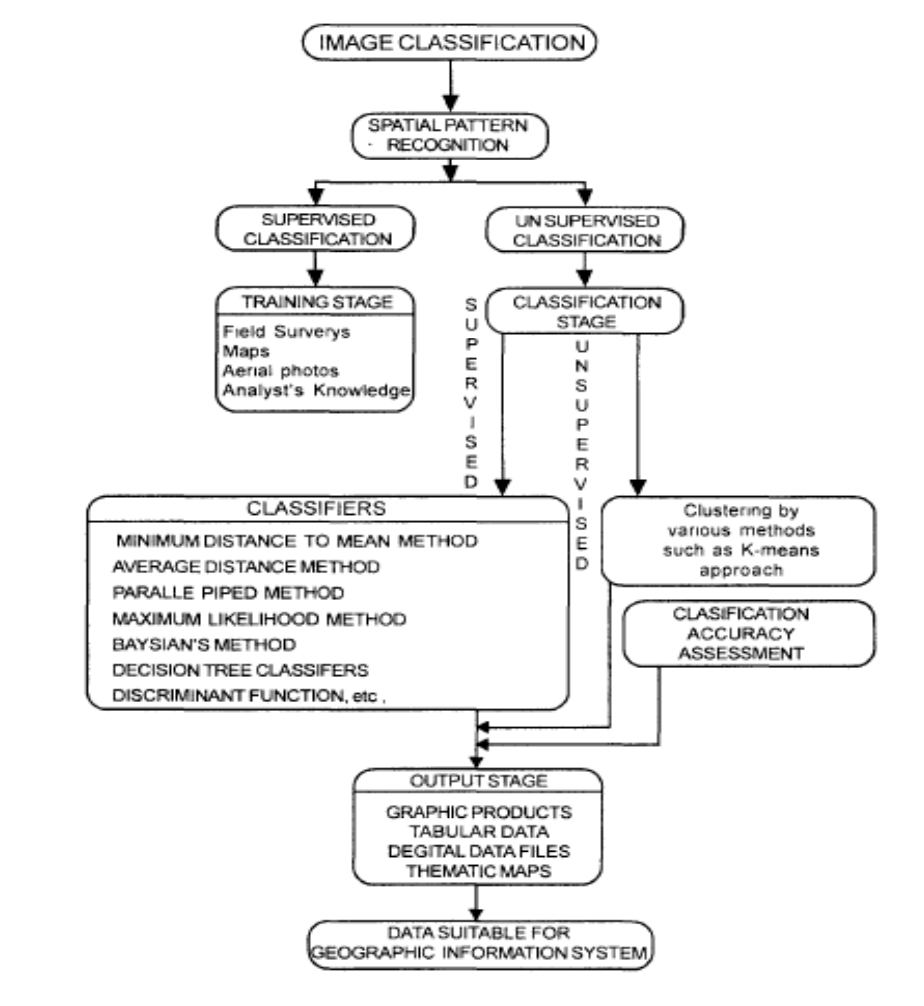


Figure 2: Image Classification Flow Chart

### ***2.2 Steps of Image Classification:***

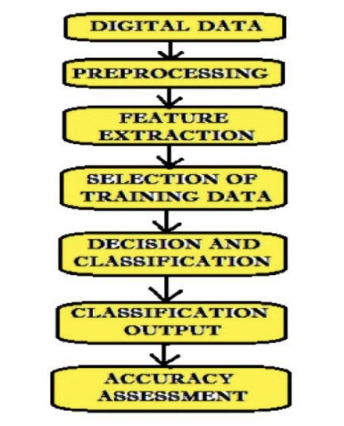


Figure 3: Steps of Image Classification

* Step 1: Class Definition: Clearly define the classes for classification based on the image data's objective and characteristics.
* Step 2: Feature Selection: Identify relevant attributes to differentiate between the classes, encompassing factors such as spectral attributes, colour information, and textural characteristics.
* Step 3: Training Data Sampling: Collect training data to determine suitable decision rules and choose between supervised or unsupervised learning techniques.
* Step 4: Decision Rule Identification: Compare various classification techniques with training data to select an appropriate decision rule.
* Step 5: Classification Process: Classify all pixels into a single class based on the decision rule, using either pixel-by-pixel or per-field methods.
* Step 6: Result Verification: Validate the classified results for accuracy and reliability.

### ***2.3 Techniques:***

Image classification is a pivotal process in the interpretation and analysis of remotely sensed imagery, commonly used in geographic information systems (GIS). It involves categorising pixels within an image into various land cover classes or themes. There are two primary techniques employed in remote sensing: supervised and unsupervised classification.

Supervised classification (human-guided) is a method that relies on prior knowledge of the classes. An analyst uses known labelled data, or 'training sites,' to guide the algorithm in recognizing the classes within the rest of the image. This technique incorporates a variety of algorithms, each with its advantages and principles that influence their application.

Unsupervised classification (calculated by software), on the other hand, does not utilise pre-labelled data but instead identifies natural groupings or clusters within the image data based on statistical processes.

Both techniques aim to transform raw spectral data into meaningful thematic information, with applications ranging from environmental monitoring to urban planning. Each approach, supervised and unsupervised, has distinct methodologies, benefits, and challenges, making them suitable for different scenarios in the field of remote sensing.

#### *2.3.1 Supervised Classification:*

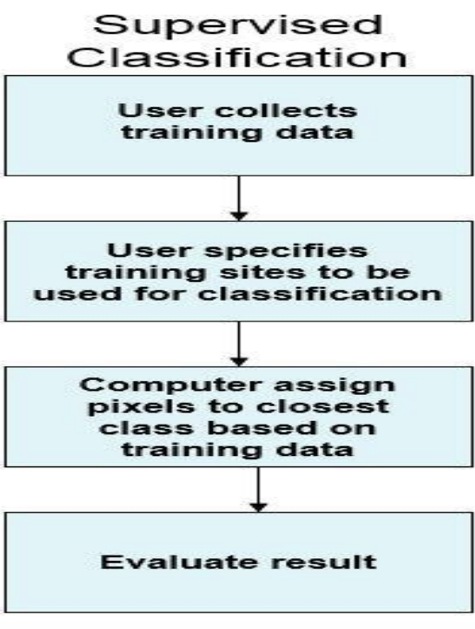


Figure 4: Supervised Classification

##### Concept:

Supervised classification involves using known, labelled classes to guide the classification of pixels in an image. The analyst predetermines and conceives the classes, using training data labelled with these classifications to instruct the classification algorithm.

##### Steps:

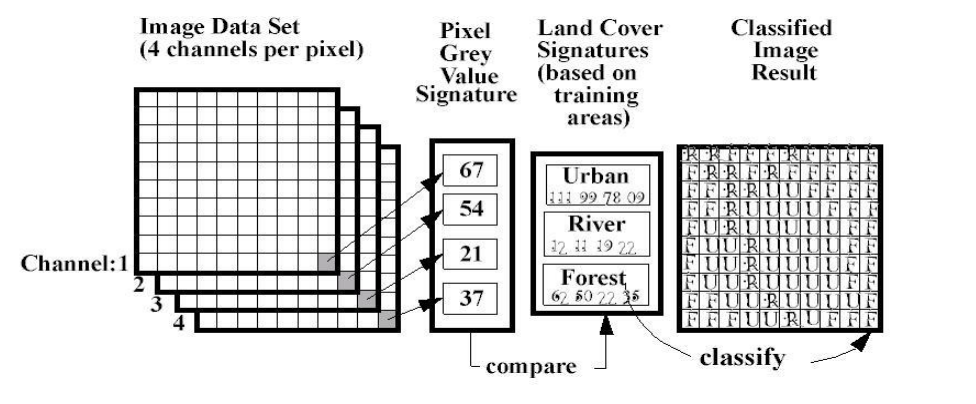


Figure 5: Steps of Supervised Classification

* Step 1: Training Stage: Analysts select training areas representative of land cover types and develop numerical descriptions of spectral signatures.
* Step 2: Decision Rule (Signature File Generation): The classifier employs spectral patterns obtained from the training regions to classify the entire image.
* Step 3: Output Stage (Classification): The algorithm classifies the image into thematic classes using spectral signatures as a reference.



Figure 6: Supervised Classification Stages

##### Principles:

* The classifier gains an understanding of the unique attributes linked to different thematic classes through the utilisation of a set of pixels selected by the analyst.
* Analysts, drawing upon their expertise, carefully select prototype pixels for each class. This process guides the classifier in establishing a connection between the data and the various classes.

##### Algorithms:

* Maximum likelihood: This algorithm classifies pixels by calculating the probability that a pixel pertains to a particular class based on statistical characteristics, with the pixel being assigned to the class for which it has the highest likelihood.
* Minimum distance: This classifier measures the spectral distance from each pixel to the mean vector of each class and assigns pixels to the class that is closest in terms of this Euclidean distance.

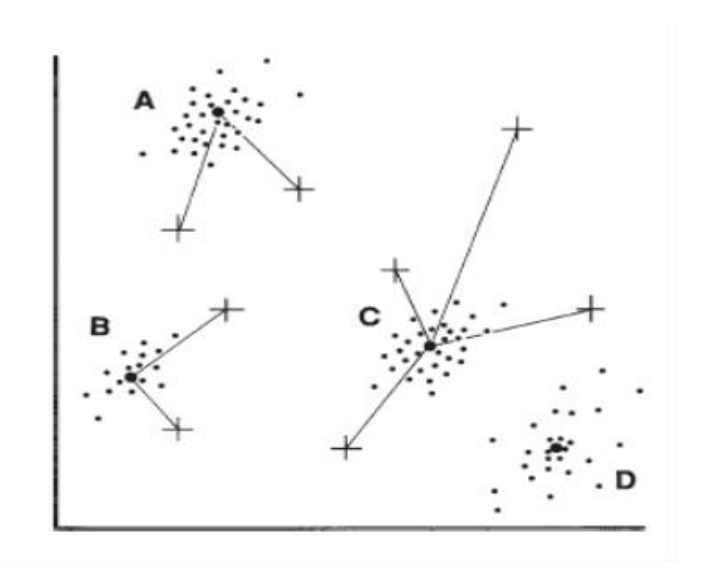


Figure 7: Minimum Distance Classifier

* Principal components: Often used for dimensionality reduction, this technique involves reorienting the data along the axes of the greatest variance and can be used to enhance classification by reducing noise and highlighting important features.
* Support Vector Machine (SVM): SVM is a robust algorithm that finds the optimal hyperplane to separate different classes in a multi-dimensional space, ensuring the greatest possible margin between classes.
* Iso Cluster: It is an iterative, unsupervised method that also finds utility in supervised classification, typically by helping to determine the initial class means in a multidimensional space before further refinement.
* Artificial Neural Networks (ANN): ANNs draw inspiration from the neuronal structure of the human brain and acquire the ability to classify pixels based on the input training data by utilising a network of interconnected nodes or neurons.
* Parallelepiped: This method classifies pixels based on whether they fall within a multidimensional box defined by the range of values for each class in the training datasets.
* Mahalanobis Distance: This method takes into account the variance of each class, providing a direction-sensitive distance classification that is more refined than Euclidean distance alone.

##### Advantages:

* When quality training data and analyst expertise are available, supervised classification can be very accurate.
* It allows for the distinct identification of classes based on spectral signatures.
* The process benefits from the analyst’s domain knowledge, which guides the learning algorithm.

##### Disadvantages:

* Supervised classification can be time-consuming and costly due to the need to collect extensive training data.
* It requires prior knowledge of the area being classified, which may not always be available.
* The accuracy of classification relies significantly on both the quality of the training data and the skills of the individual processing the image.
* If classes have similar spectral reflectance, it can result in high misclassification rates.
* Close attention to the development of training data is essential in this process; inadequate representation of the variability within a specific land cover type could compromise classification accuracy.

#### *2.3.1 Unsupervised Classification:*

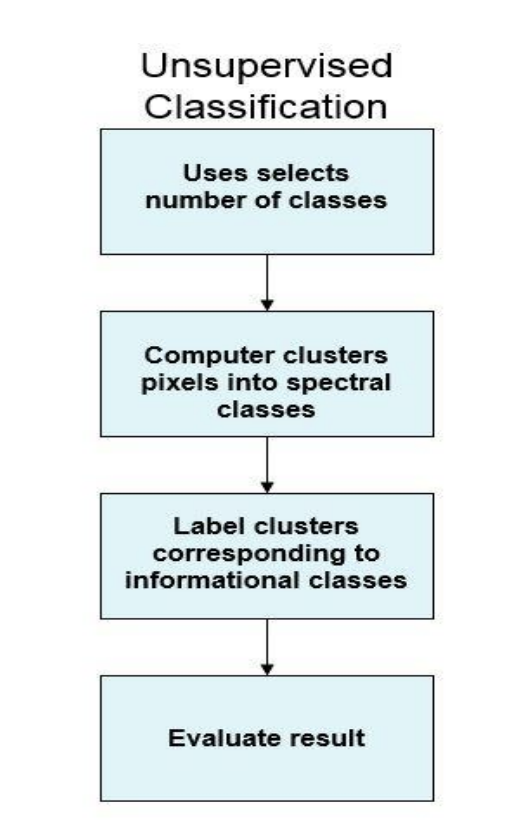


Figure 8: Unsupervised Classification

##### Concept:

Unsupervised classification, also known as clustering, is a method that analyses image data without pre-labelled classes. It identifies natural groupings or clusters in the data based on statistical similarity, to discover patterns or structures inherent in the data.

##### Steps:

* Step 1: Generate Clusters: The algorithm organises pixels into a specified number of classes based on their spectral similarity without prior knowledge of the classes.
* Step 2: Assign Classes: After clustering, the analyst interprets and labels the output clusters to correspond with meaningful real-world categories.

##### Principles:

* Unsupervised classification operates on the principle that spectrally similar pixels will naturally form statistically distinct clusters.
* The analyst does not initially know what these clusters represent and must use their understanding of the area to assign real-world labels to the clusters post-classification.

##### Algorithms:

* K-means Clustering: An iterative method that divides the dataset into separate and non-overlapping subgroups (clusters), assigning each data point to the cluster whose mean is closest to it.
* PCA-based K-means Clustering: A variant of K-means that first reduces dimensionality using Principal Component Analysis (PCA) before clustering, which can enhance classification accuracy.

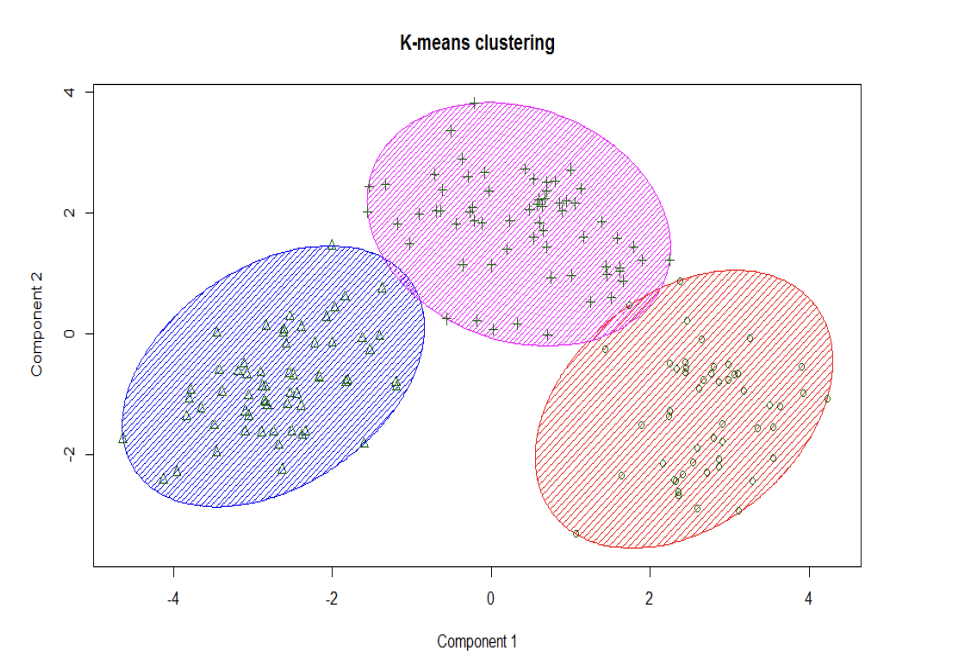


Figure 9: K-means Clustering

* Fuzzy C-Means Clustering (FCM): Similar to K-means but allows for data points to belong to multiple clusters with varying degrees of membership, reflecting the inherent fuzziness in real-world data.

##### Advantages:

* Unsupervised classification is quick and does not require extensive prior knowledge of the area or manual labelling of training data, making it a good preliminary analysis tool.
* It is objective to create spectral classes based purely on numerical information from the data, which might reveal unexpected patterns.

##### Disadvantages:

* The generated spectral classes may not align directly with informative classes, necessitating substantial post-classification interpretation.
* Without the guidance of training data, the unsupervised classification may be less accurate in distinguishing classes that have subtle spectral differences.

## **Previous Applications in Image Classification in Sports:**

### ***3.1 Hudl:***

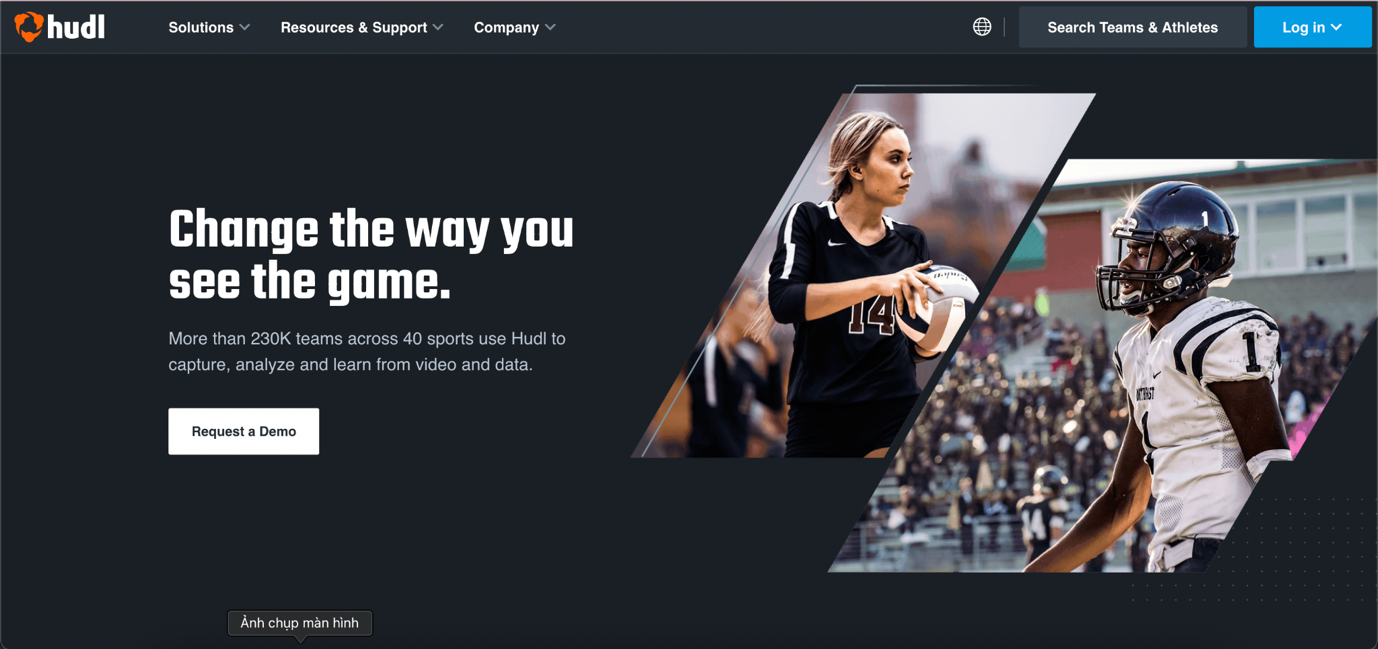
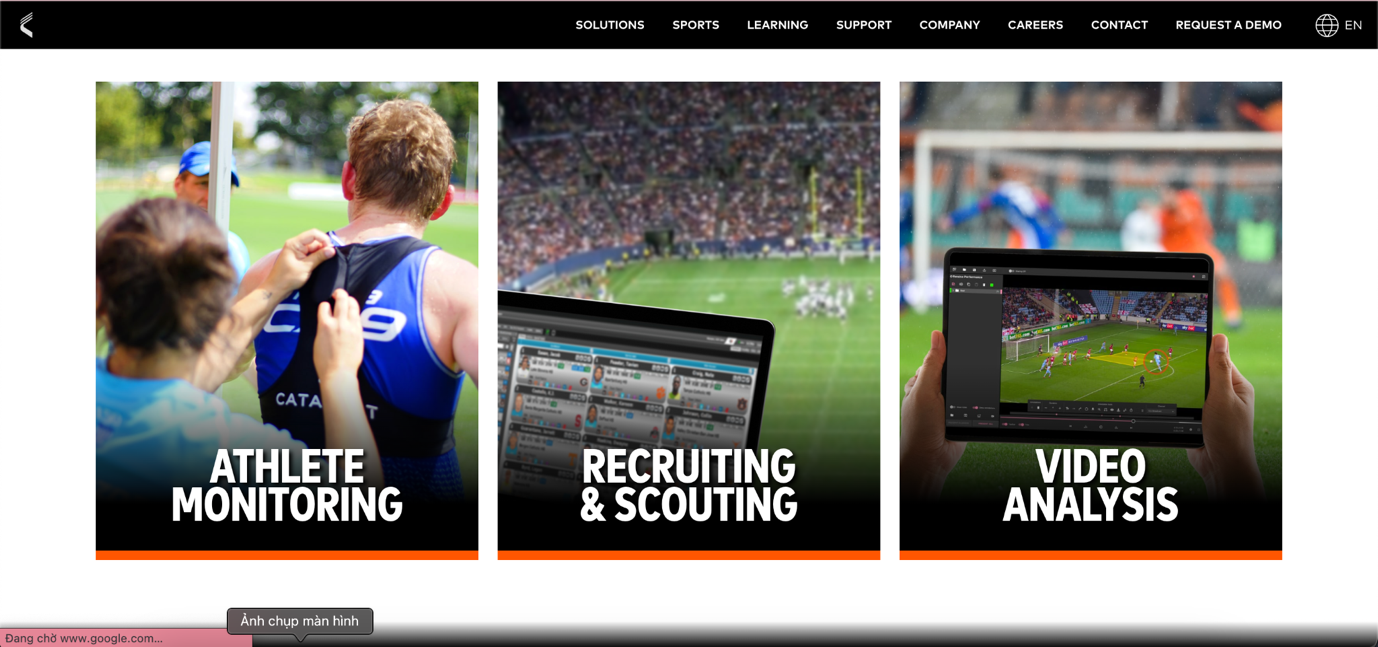


Figure 10: Hudl Website

Enter the realm of sports innovation with Hudl, a pioneering force reshaping the landscape of performance analysis. Hudl stands at the forefront of sports technology, providing a robust platform for video analysis and coaching tools. At the core of its capabilities is advanced image classification, a dynamic feature that propels the platform beyond conventional boundaries. With an intuitive interface, Hudl enables coaches to dissect game footage, employing image classification to identify pivotal player actions, refine scouting strategies, and delve into nuanced aspects crucial for team success. Hudl's dedication to technological advancement, particularly in image classification, cements its role as a transformative force, empowering athletes and coaches with unparalleled insights and strategic precision.

### ***3.2 Catapult:***

Figure 11: Catapult Website



Embark on a journey of sports analytics excellence with Catapult, an industry leader in wearable technology and video analysis. At the forefront of its capabilities is a seamless integration of image classification processes, elevating the platform's significance in the realm of performance optimization. Catapult's cutting-edge technology employs image classification to meticulously track and interpret player movements, facilitating unparalleled insights into biomechanics and game dynamics. This introduction sets the stage for exploring Catapult's pivotal role in merging sophisticated image classification with wearable devices, offering coaches and athletes a comprehensive set of tools to enhance training regimens, prevent injuries, and ultimately elevate athletic performance to unprecedented heights.

## **Existing Challenges and Limitations:**

### ***4.1 Data Variability:***

In sports environments, the dynamic nature introduces constant changes in lighting conditions, diverse player positions, and varied backgrounds. These fluctuations present a formidable challenge for image classification models, as the ability to generalise effectively across such diverse and evolving conditions becomes inherently complex. The dynamic nature of sports settings necessitates robust and adaptive image classification algorithms to accurately interpret visual data amidst the ever-changing dynamics of the field.

### ***4.2 Real-Time Processing:***

In the context of live sports scenarios, the imperative for real-time image classification imposes a significant demand on platforms to rapidly process and analyse substantial volumes of data. This necessity underscores the essential requirement for robust computational resources capable of swiftly executing the intricate tasks associated with image classification. The challenge lies in achieving seamless, instantaneous analysis, ensuring that the insights derived from the visual data align with the pace and dynamism inherent in live sports events.

### ***4.3 Ambiguity in Player Actions:***

The task of identifying and classifying intricate player actions, particularly in team sports characterised by intricate interactions, presents a notable challenge. This complexity arises from the inherent ambiguity and variability in the movements of players during dynamic, collaborative gameplay. Deciphering nuanced actions within the context of team dynamics requires image classification models to navigate through a spectrum of potential movements, adding a layer of intricacy to the process and emphasising the need for advanced algorithms capable of discerning subtle variations in player actions.

### ***4.4 Wearable Technology Integration:***

In the dynamic landscape of sports technology, wearable devices play a pivotal role, offering athletes and coaches an array of data from heart rate to acceleration. This wealth of information aids in crafting enhanced training plans, pinpointing performance weaknesses, and tracking progress towards fitness goals. Among the myriad applications, heart rate monitors meticulously capture breathing and pulse patterns, gyroscopes and accelerometers provide vital parameters for various sports positions, GPS tracks distance and velocity for cyclists and runners, and sleep monitors contribute to tailored recovery strategies. Moreover, the integration of wearable devices with advanced image classification, as exemplified in platforms like Catapult, introduces a new frontier. This fusion, however, necessitates seamless synchronisation to ensure accurate and timely data collection, presenting technical challenges that underscore the intricacies of merging wearable technology with image analysis in the pursuit of optimising athletic performance. As a testament to innovation, athletes can even engage in VR training sessions, leveraging sensors strategically placed in their rooms and on their feet.

### ***4.5 Data Privacy and Security:***

Protecting sensitive player data captured through image classification is of paramount importance. The continuous challenges lie in guaranteeing compliance with privacy regulations and establishing robust security measures to safeguard this valuable information. As technology evolves, it becomes imperative to stay vigilant and proactive in addressing the intricate landscape of data privacy and security.

### ***4.6 Catapult’s Local Positioning System:***

The accuracy of Catapult's Local Positioning Systems (LPS) used in indoor sports is impacted by several factors. These include the type of signal used, environmental conditions like obstructions and materials surrounding the field, and the space between the signal anchor nodes and athlete units. Indoor venues show greater errors due to increased signal interference compared to outdoor conditions. Additionally, the data processing methods used in these systems can vary, affecting the validity of the derived metrics. The setup of the anchor nodes and their placement relative to building walls also significantly impacts the system's accuracy, especially in indoor environments. This complex interplay of factors makes it challenging to ensure consistent and accurate tracking in different settings. For a detailed understanding, you can refer to the study on Frontiers in Sports and Active Living.

### ***4.7 Cost and Accessibility:***

The substantial cost of implementing Catapult's elite sports performance analysis systems, particularly for professional teams, is a prime example of the financial burden of advanced sports technology. The initial investment includes not only the purchase of wearable devices like the OptimEye S5 for tracking athlete performance metrics but also additional hardware like charging stations and GPS units. On top of these are software subscription fees for accessing Catapult's analytical platforms, which require regular updates. Implementing these systems also involves costs for staff training and potential integration with existing systems. Moreover, ongoing expenses include continuous technical support, maintenance, and upgrades for the hardware, and secure data storage solutions for the vast amounts of data generated. For a professional sports team, the total annual expenditure for utilising a system like Catapult's can easily reach tens of thousands of dollars, a figure that is often prohibitive for smaller or amateur organisations. This case clearly illustrates the significant financial commitment required for cutting-edge sports performance analysis technology.

## **Relevance to the thesis:**

The literature review enhances the relevance of our thesis on building a Sports Classification Website with AI-based image classification.

* Foundational Knowledge: It provides essential knowledge for our AI model's development, ensuring our approach is technically sound.
* Contextualization: The review places our project within the sports technology landscape, showing its relevance in practical applications.
* Problem Identification: By highlighting challenges in sports image classification, it emphasises the real-world issues our thesis addresses.
* Solution Exploration: It offers solutions and best practices from sports technology, making our project more relevant and effective.
* Field Alignment: The review confirms that our thesis aligns with existing sports technology, underscoring its practical relevance and potential impact.

In essence, the literature review's relevance ensures our thesis is well-informed, problem-focused, and equipped to contribute effectively to sports image classification.

# **CHAPTER 3: METHODOLOGY**

## **Dataset**:

I acquired the dataset from Kaggle, which initially comprised 13,492 training images, 500 test images, and 500 validation images, each in the format of 224x224x3 JPEG and spanning across 100 different classes. I have since expanded and updated this dataset to include 14,492 training images, 510 test images, and 510 validation images, now covering 110 distinct classes. The additional sports categories incorporated into the new dataset are parkour, soccer, chess, esports, kayaking, sandboarding, scuba diving, track and field, dance sports, and paintball.

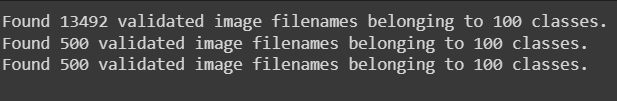


Figure 12: Original Data

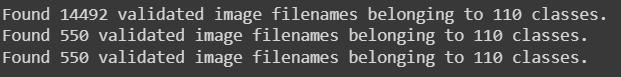


Figure 13: New Data

## **Pre-processing:**

In the pre-processing phase of the thesis project, I focus on preparing the images for efficient training and evaluation within the neural network models. This phase involves setting crucial parameters and utilising TensorFlow's ImageDataGenerator for efficient data handling and augmentation. This procedure is applied similarly for VGG16, ResNet50, MobileNetV3 Large and Inception V3.

### ***2.1 Setting Batch and Image Size:***

#### *2.1.1 Purposes:*

This part plays a specific and critical role in preparing and processing the image data for efficient classification.

##### Batch Size Configuration:

* Handling Large Datasets: The project involves 14,429 images. Processing these images in 10 batches makes it feasible to train the model even with limited computational resources like memory and processing power.
* Training Efficiency: Training on batches makes each iteration of the model's training quicker than processing the entire dataset instantly. This feature speeds up the learning process, making the project more time efficient.
* Improved Learning Dynamics: Training in batches introduces a level of randomness in the optimisation process, which can help the model escape local minima and potentially lead to better generalisation on unseen data.

##### Image Size Configuration:

* Consistency Across Data: In image classification, all input images must have the same size so the neural network can learn from them effectively. The process ensures uniformity across the dataset by resizing all images to 224x224 pixels.
* Compatibility with Pre-trained Models: The pre-trained models require a specific input size. The chosen size of 224x224 is a common requirement for many such models, making the dataset compatible.
* Balancing Detail and Performance: The dimension 224x224 is large enough to retain critical visual details in the images for the classification, yet small enough to keep computational requirements (e.g., processing power and memory) manageable.

#### 2.1.2 Explanation:

* BATCH\_SIZE = 10: Sets the number of images to process in a batch. The purpose is to define the number of images processed simultaneously during the training phase.
* IMAGE\_SIZE = (224, 224): Defines the target size for each image. This action resizes the images to 224x224 pixels, a popular size for CNN inputs.

### ***2.2 ImageDataGenerator:***

#### *2.2.1 Purposes:*

The purpose of using “ImageDataGenerator” with a specific pre-processing function for EfficientNet is to automate and standardise the process of preparing the image data for the model. It includes resizing images, applying necessary transformations, and normalising pixel values. The “preprocessing\_function” can adapt the images to the specific requirements of each model, ensuring that the input data is in the optimal format for the network to process.

#### *2.2.2 Explanation:*

* generator = ImageDataGenerator(preprocessing\_function = tf.keras.applications.efficientnet.preprocess\_input): An ImageDataGenerator is created with a pre-processing function specific to EfficientNet and will vary across the application of VGG16, ResNet50, MobileNetV3 Large and Inception V3. This function will pre-process the images to be suitable for EfficientNet, typically normalising the pixel values as the model expects.

### ***2.3 Creating Data Generators for Different Sets:***

#### *2.3.1 Purposes:*

##### Automated Data Handling:

* Streamlining Data Input: The data generators automate the process of loading and processing images from the respective data frames. It eliminates the manual handling needs of each image file, significantly streamlining the workflow.
* Dynamic Image Augmentation: For the training generator, I can easily integrate image augmentation techniques (e.g., rotation, scaling, etc.) if needed. It helps create a more robust model by exposing it to more data scenarios.

##### Memory Management:

* On-the-fly Image Processing: Data generators load and process images in real-time during the training or evaluation phases. This approach is memory efficient as it does not require loading the entire dataset into memory at a time.

##### Consistent Pre-processing:

* Uniform Image Formatting: All images are automatically resized to the specified size and converted to the appropriate colour mode (e.g., RGB). It ensures consistency across all images fed into the model.
* Applying Model-Specific Pre-processing: The “preprocessing\_function” in “ImageDataGenerator” allows for model-specific image pre-processing (e.g., pixel normalisation specific to EfficientNet).

##### Label Handling and Encoding:

* Automatic Label Extraction: The generators extract labels directly from the data frames, associating each image with its corresponding label efficiently.
* Categorical Conversion: The generators can automatically convert labels into categorical form for classification tasks, which is necessary for training models on multi-class classification problems.

##### Setup:

* Training Data Generator: Generates batches of training data, shuffling the data to introduce randomness, which is beneficial for model training.
* Validation Data Generator: Produces batches of validation data, allowing the model evaluation on a separate set of data during the training process.
* Test Data Generator: Used for final model evaluation, generating batches of test data without shuffling to ensure consistent evaluation metrics.

#### *2.3.2 Explanation:*

* train\_images = generator.flow\_from\_dataframe(...): This line creates a data generator for the training set. It loads the data from a data frame (train\_df), specifying the paths to the images (imgpath) and their corresponding labels (labels). Resize the images to 224x224 pixels and set them to RGB colour mode. Treat the labels as categorical, which is suitable for multi-class classification. Set the batch size to 10 and shuffle the data.
* val\_images = generator.flow\_from\_dataframe(...): Similar to the training set, this creates a data generator for the validation set using valid\_df.
* test\_images = generator.flow\_from\_dataframe(...): This creates a data generator for the test set using test\_df. The data is not shuffled, which is standard for test set evaluation.

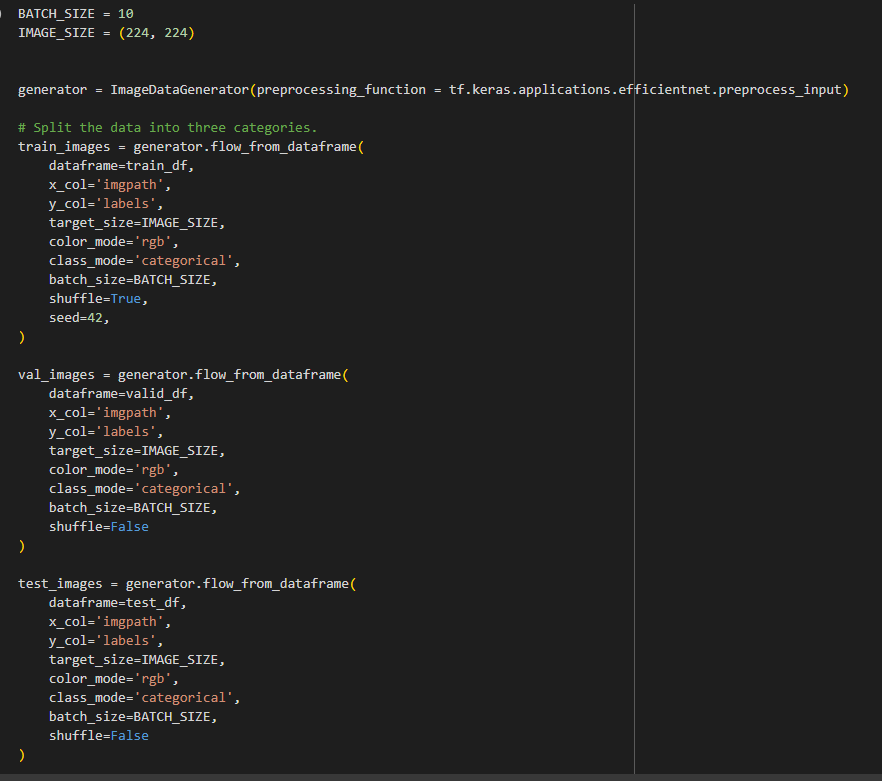


Figure 14: Data Pre-processing

# **CHAPTER 4: IMPLEMENTATION AND RESULTS**

## **Architectures of CNN:**

### ***1.1 EfficientNet B0***

Figure 15: EfficientNetB0 Network Architecture

The image above is the EfficientNet model's design, which processes a 224x224 pixel image through various layers, including:

An initial convolution layer that increases the number of channels to 32.

Multiple MBConv layers, which are efficient and inverted residual blocks with different kernel sizes, process and transform the input while changing its dimensions and channel depth.

The architecture employs downsampling to decrease spatial dimensions and increase channel depth at different points, a strategy known as compound scaling.

The final layers consist of a 1x1 convolution that compacts features into 1280 channels, a global average pooling to condense features to a single vector, a fully connected layer that outputs two values, and a softmax function to generate the final probabilities for the two classes.

EfficientNet is recognized for its effectiveness and accuracy in image classification, capable of capturing detailed features.



Figure 16: Data Preparation

* Data Preparation: I set BATCH\_SIZE and IMAGE\_SIZE for my image data generator. I use TensorFlow's ImageDataGenerator class with a preprocessing function designed for EfficientNet to ensure the images I feed into the model are processed in the same way the original EfficientNet was trained.
* Data Loading: I load my training, validation, and test datasets using the flow\_from\_dataframe method. This method takes care of reading images from their paths and assigning the corresponding labels. I shuffle only my training data to introduce randomness which is a good practice to help the model generalize better.
* 

Figure 17: Model Loading and Layer Freezing

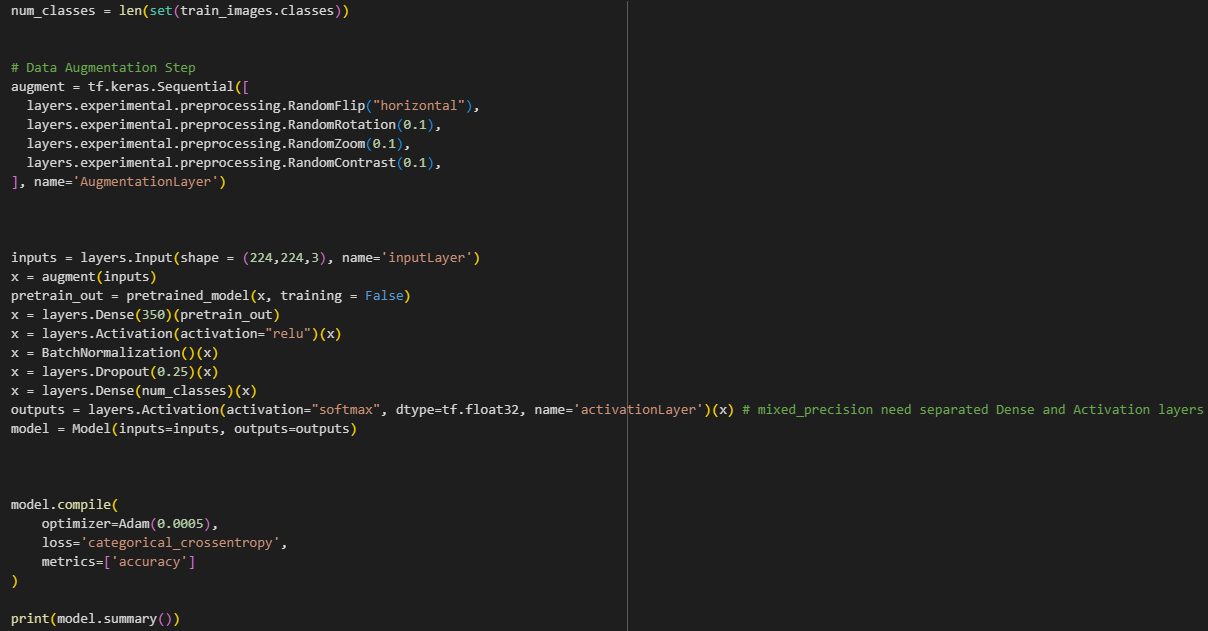
* Model Loading: I load the EfficientNetB0 model pre-trained on the ImageNet dataset. I'm interested in using the learned weights but not the top layers, as I'll be adding layers suited to my specific classification task.
* Freezing Layers: I freeze all the layers of the EfficientNetB0 model by iterating through them and setting their trainable attribute to False. This means I intend to keep the learned features from ImageNet fixed while I train only the layers I'll add on top.

Figure 18: Data Augmentation

* Data Augmentation: I create an augmentation layer that randomly flips, rotates, zooms, and adjusts the contrast of my images. This will help me create a more robust model by simulating a variety of conditions it might encounter in the real world.
* Model Extension: I build my custom model by adding new layers on top of the EfficientNetB0 base. I added a Dense layer with 350 units followed by a ReLU activation, batch normalization, and dropout to reduce overfitting. Then, I add a final Dense layer with a number of units corresponding to the number of classes I have in my dataset.
* Model Compilation: I compile my model using the Adam optimizer with a learning rate of 0.0005, and since this is a classification problem, I use the categorical\_crossentropy loss function. I also track accuracy as my performance metric.

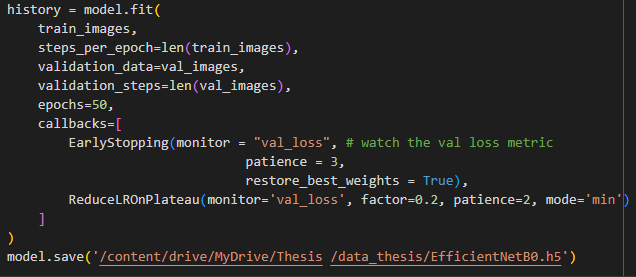


Figure 19: Model Training

* Model Training: I train my model with the datasets I prepared, setting steps per epoch and validation steps to the length of the train and validation image iterators, respectively. I use a patience of 3 epochs for early stopping and a reduction factor of 0.2 for the learning rate if the validation loss stops improving.
* Saving the Model: After training, I save my model to a specific path on my Google Drive. This will allow me to reload the model later without needing to retrain it.

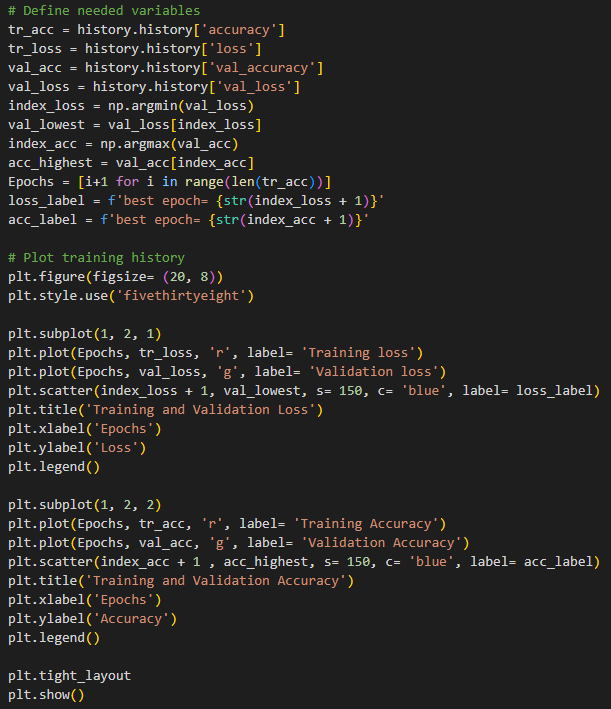


Figure 20: Visualisation

* Performance Visualisation: I plot the training and validation loss and accuracy to visually inspect the training process. I mark the best epochs for both loss and accuracy on the plots, which helps me quickly assess how well the training went and at which epoch the model performed best.

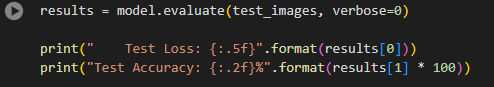


Figure 21: Testing the Model

* Testing the Model: I evaluate my model on the test set without shuffling to keep the order of the data consistent and print out the test loss and accuracy. This gives me an unbiased estimate of my model's performance on unseen data.

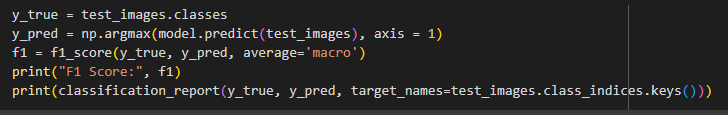


Figure 22: F1 Calculation and Classification Report

* Calculating F1 Score and Classification Report: I predict the classes of my test images and calculate the F1 score, which is a better measure than accuracy when dealing with imbalanced datasets. I also print out a detailed classification report that gives me insight into the precision, recall, and F1 score for each class.

### ***1.2 VGG16***

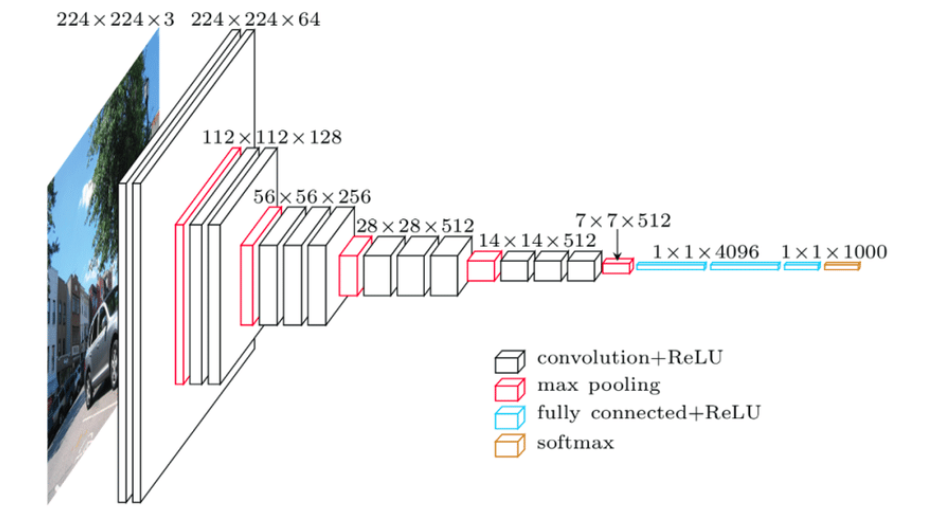


Figure 23: VGG16 Architecture

The image illustrates the architecture of the VGG16 model, a type of Convolutional Neural Network (CNN) commonly used in image recognition. The VGG16 model processes input images of size 224x224 pixels with three colour channels (RGB) through a sequence of layers:

* Convolutional layers with ReLU activations: These extract spatial features from the image through filters and introduce non-linearities into the network.
* Max pooling layers: These reduce the spatial dimensions of the feature maps to decrease the number of parameters and computation, thus helping to prevent overfitting.
* Fully connected layers with ReLU activations: These layers interpret the features extracted by the convolutional layers and transform them into a form suitable for classification.
* Softmax layer: The final layer that outputs a probability distribution across multiple classes, assigning the likelihood that the input image belongs to each class.

In the context of transfer learning, the convolutional base of VGG16 is often reused while the fully connected layers are replaced to suit the specific number of classes for a new task. The pre-trained convolutional base captures generic features that are useful across different image recognition tasks, while the new fully connected layers make the network specific to the new classification problem at hand.

Data Preparation, Data Loading, Model Training, Saving Model, Performance Visualization, Testing the Model, Calculating F1 score and Classification Report are like EfficientNet B0.

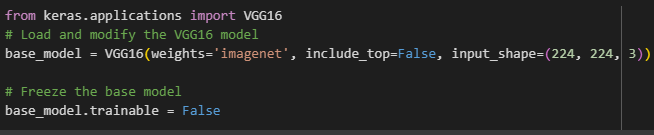


Figure 24: VGG16 Model Loading and Freezing

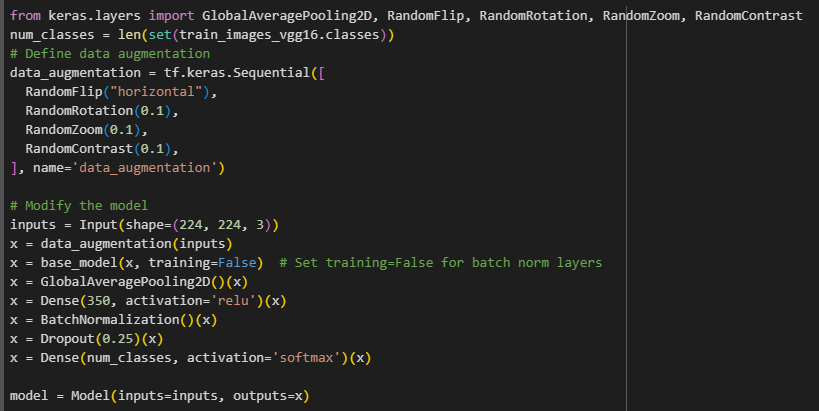


Figure 25: VGG16 Data Augmentation and Model Modification

I have loaded the VGG16 model with pre-trained ImageNet weights and set include\_top=False to remove the top layer, adjusting the input shape to (224, 224, 3), which is standard for VGG16. I then freeze the layers of the VGG16 model to retain the pre-trained features, as they've proven to be effective for a wide range of image classification tasks.

To augment my training data, I've set up a sequential model that includes random flips, rotations, zooms, and contrast adjustments. This will help my model generalize better by simulating various viewing conditions and image variations it might encounter in the real world.

Building on the VGG16 base, I apply my data augmentation strategies to the input, pass the augmented data through the VGG16 base (ensuring the model is in inference mode with training=False), and then use a GlobalAveragePooling2D layer to condense the feature maps to a single vector per image. I added a dense layer with 350 units followed by batch normalization and a 25% dropout to reduce overfitting. Finally, I added a softmax layer to handle the multi-class classification for the number of classes in my dataset.

I compile the model with the Adam optimizer, using a learning rate of 0.0005, and choose categorical\_crossentropy for my loss function because it's well-suited for multi-class classification problems. I use accuracy as my metric to track how well the model performs both in training and validation.

### ***1.3 ResNet50***

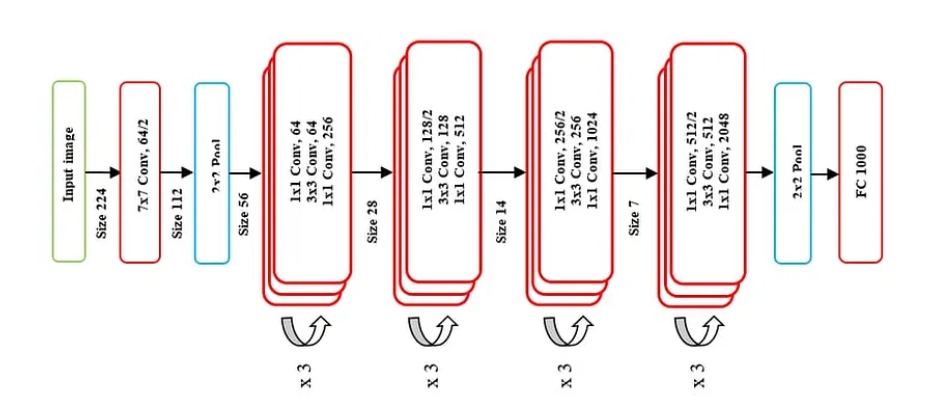


Figure 26: ResNet50 Architecture

ResNet, which stands for Residual Network, is known for its deep architecture and the use of skip connections or residual connections.

* Input Layer: The network takes an input image of size 224x224x3.
* Initial Convolution and Pooling: The image is first processed by a 7x7 convolutional layer with 64 filters, followed by a max-pooling layer.
* Convolutional Blocks: The core of the network consists of a series of convolutional blocks. Each block has three convolutional layers with varying numbers of filters (64, 128, 256, and 512). These blocks use 1x1 convolutions to reduce and then increase dimensions, with a 3x3 convolution in between. Each block has a skip connection that bypasses the layers to help mitigate the vanishing gradient problem in deep networks.
* Identity Blocks: Between the convolutional blocks are identity blocks, which are similar to convolutional blocks but do not alter the dimensionality. They have the same number of filters throughout.
* Global Average Pooling: After the last convolutional block, a global average pooling layer is used to reduce each feature map to a single value.
* Fully Connected Layer: Following the pooling layer, a fully connected layer with 1000 units is used, corresponding to the number of classes in the classification task.
* Softmax Activation: Finally, a softmax activation function is applied to output a probability distribution over the 1000 classes.

The code process and parameters are like EfficientNet B0.

### ***1.4 MobileNetV3 Large***

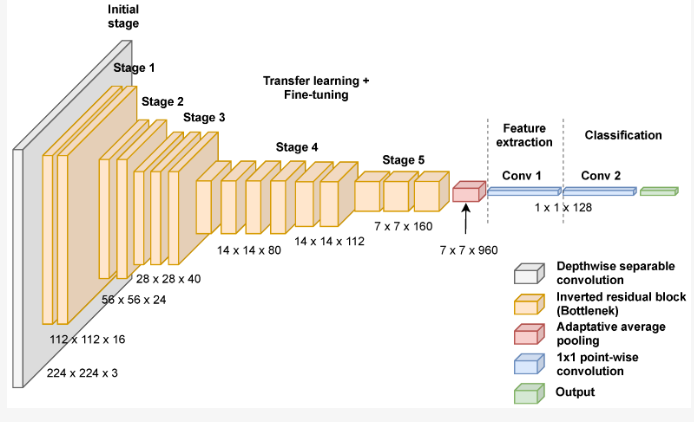


Figure 27: Mobilenet V3 Architecture

The image appears to be a diagram of the MobileNet architecture, highlighting its structure optimized for mobile and edge devices. It includes several stages of depthwise separable convolutions, which is a hallmark of MobileNet, as well as inverted residual blocks with linear bottlenecks. These architectural choices are designed to provide high efficiency in terms of computational cost and model size, making them suitable for environments with limited resources.

The diagram also shows that the network is divided into stages, with each stage likely representing a combination of convolutional layers, batch normalization, and non-linear activation functions. The final stages are dedicated to feature extraction and classification. This is typical for a deep learning model, where the initial layers are responsible for extracting low-level features and subsequent layers combine these into more abstract representations.

In the context of transfer learning and fine-tuning:

* Transfer Learning: You would typically use the pre-trained weights as they are for the initial layers and only train the latter layers of the model, where the more task-specific features are learned.
* Fine-tuning: Once the upper layers have been trained, or if you have a sufficiently large and representative dataset of your own, you might opt to fine-tune the entire network by continuing the training process, allowing earlier layers to adjust their weights as well.

The code processing of MobileNet V3 Large is akin to EfficientNet B0.

### ***1.5 Inception V3***

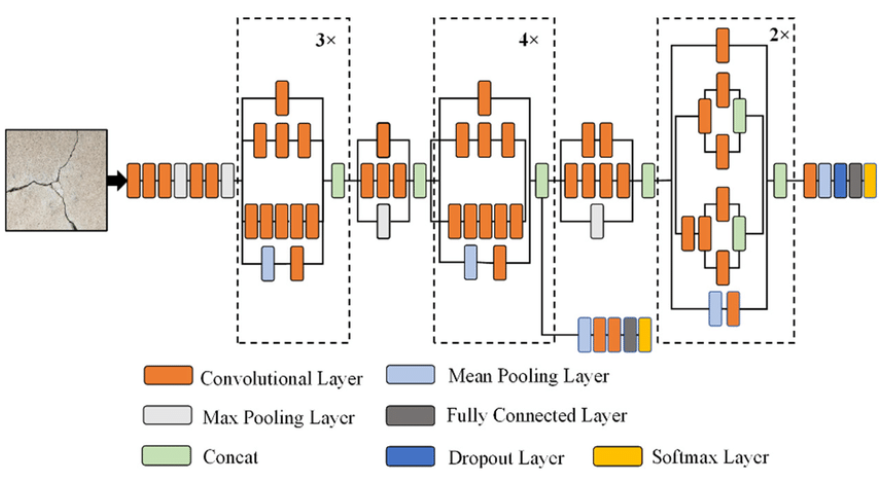


Figure 28: Inception V3 Architecture

The image representation of an architecture Inception V3, which is a type of Convolutional Neural Network (CNN) known for its complex structure that includes multiple parallel convolutional and pooling layers at each stage of the network, known as inception modules. The Inception V3 architecture is designed to efficiently learn representations from images with a large variation in the size of the objects depicted within them.

The key components of the Inception V3 architecture reflected in the image are:

* Convolutional Layers: These layers apply various filters to the input image to extract features. In Inception modules, convolutional layers with different kernel sizes are applied in parallel to capture patterns over varying spatial extents.
* Pooling Layers: Both max pooling (light blue) and average pooling (dark blue) layers are used in Inception models. Pooling layers reduce the spatial dimensions of the feature maps, which decreases the computational load and helps to make the representation more robust to the location of features in the input image.
* Concatenation: The outputs of the parallel convolutional and pooling layers are concatenated along the channel dimension. This concatenation combines the features extracted by different filters, allowing the network to decide which features are most useful for a given task.
* Fully Connected Layers: These layers, typically found near the end of the network, are used to perform classification based on the high-level features extracted by the convolutional and pooling layers.
* Dropout Layer: Dropout is a regularization technique used to prevent overfitting. By randomly dropping units (along with their connections) during training, dropout forces the network to learn more robust features that are not reliant on any small set of neurons.
* Softmax Layer: This layer is used for multi-class classification tasks. It converts the final features into probabilities for each class.

The numbers (3x, 4x, 2x) likely indicate the repetition of certain inception modules or blocks within the architecture, meaning that similar sets of layers are stacked multiple times in sequence.

The Inception V3 architecture is quite complex and includes many more layers and connections than what is typically shown in such a simplified diagram. The real Inception V3 model also includes many 1x1 convolutions used to reduce dimensionality, as well as more complex arrangements of convolutions and pooling operations within each inception module.

The code process step is similar to EfficientNet B0.

## **Results**:

Figure 29: Original Dataset Results

Figure 30: Post-Optimisation Dataset Results

### ***2.1 From the "Original Dataset Results" table:***

* EfficientNetB0 performed the best with the original dataset, achieving 96% accuracy and an F1 score of 0.95.
* MobileNetV3 Large and ResNet50 also performed well, with over 90% accuracy and F1 scores of 0.93 and 0.89, respectively.
* VGG16 and InceptionV3 had lower performance metrics compared to the others, with VGG16 at 81.4% accuracy and InceptionV3 at 80% accuracy.

### ***2.2 From the "Post-Optimisation Dataset Results" table:***

* After dataset optimisation, EfficientNetB0 still has the highest accuracy, slightly reduced to 96.55%, but with a slightly improved F1 score of 0.96.
* MobileNetV3 Large showed a slight decrease in accuracy to 92.36% and a decrease in the F1 score to 0.92.
* ResNet50's accuracy decreased to 89.64%, with the F1 score remaining the same.
* VGG16 showed an improvement after dataset optimization, with accuracy increasing to 84% and the F1 score improving to 0.83.
* InceptionV3's accuracy also improved to 78.73%, but still remained the lowest among the models, with a slight decrease in the F1 score to 0.77.

### ***2.3 Overall Analysis:***

* Dataset optimization generally maintained or slightly improved the performance of the models, with EfficientNetB0 retaining the top position in both accuracy and F1 score.
* VGG16 and InceptionV3 benefited from the optimization with improvements in accuracy, suggesting that the changes in the dataset positively impacted their ability to generalize.
* The changes in the F1 score for InceptionV3 suggest that while accuracy improved, the balance between precision and recall might have been affected.
* The dataset optimization seems to have been particularly effective for VGG16, which showed the most significant relative improvement among the models.
* The slight decrease in performance for MobileNetV3 Large and ResNet50 suggests that these models may have been slightly overfitting to the original dataset, or that the optimizations made were more favorable to the other architectures.

## **Web Server:**

Figure 31: Website's User Interface

The image above is a user interface for a "Sports Image Classifier" web application. It invites users to upload an image of a sport, and the AI will predict which sport is shown in the image. The interface offers an option to drag and drop a file or browse files from the local storage, with a limit of 200MB per file, and it specifies that the image should be in JPG or JPEG format.

Additionally, there's an option to try the classifier with sample images, with a dropdown list to select from, and a sample image shown presumably for the sport of kayaking. The visible image features two people, with the person in the foreground holding a paddle and wearing sunglasses, indicative of a sunny day on the water, likely engaging in the sport of kayaking. The interface is user-friendly and seems straightforward to use, clearly guiding the user through the process of image selection for the classification task.

Link to the web: https://sportsimageclassifier.streamlit.app/

## **Features**:

Figure 32: Website's Key Components and Features

### ***4.1 Upload Functionality:***

The application provides an option to upload an image file. It supports JPG or JPEG formats, and there's a file size limit of 200MB, which is quite generous and should accommodate high-resolution images.

### ***4.2 Drag and Drop:***

Users can either drag and drop an image file directly into a designated area or use the "Browse files" button to select a file from their device.

### ***4.3 Sample Images:***

For convenience and to showcase the application's capabilities, there is a section where users can try the classifier with pre-loaded sample images. This helps users test the functionality without needing to upload their own image first.

### ***4.4 Sample Image Display:***

A sample image is displayed, named "test\_esport.jpg," indicating that this image is related to e-sports. This implies that the classifier is likely capable of recognizing a wide range of sports activities, including competitive gaming.

### ***4.5 Design and Layout:***

The interface has a clean and modern design, with a dark theme and high-contrast text, making it visually accessible and easy to navigate.

# **CHAPTER 5: EVALUATION AND DISCUSSION**

## **Evaluation**:

### ***1.1 Accuracy of Image Classification:***

The accuracy metrics across different models reveal a comprehensive picture of their performance in image classification. Initially, EfficientNetB0 led the pack with a high accuracy of 96%. Even after dataset optimization, this model maintained its superiority, albeit with a slight increase to 96.55%. MobileNetV3 Large and ResNet50 also showed commendable accuracy in the original dataset, surpassing the 90% mark. However, following optimization, MobileNetV3 Large experienced a slight drop in accuracy to 92.36%, and ResNet50 saw a more noticeable decrease to 89.64%.

In contrast, VGG16 and InceptionV3, which initially lagged behind with accuracies of 81.4% and 80% respectively, displayed different trajectories post-optimization. VGG16's accuracy improved significantly to 84%, indicating a positive response to the dataset changes. On the other hand, InceptionV3, despite an improvement, continued to have the lowest accuracy among the models at 78.73%. This varied response to optimization underscores the distinct characteristics and capabilities of each model architecture in handling image classification tasks.

### ***1.2 UI and UX:***

In the realm of UI and UX, the website shines with its emphasis on creating a highly user-friendly environment. The interface is thoughtfully designed, featuring dynamic buttons and intuitive features that empower users to navigate and interact with the website effectively and effortlessly. This user-centric approach not only enhances the overall experience but also ensures that users can make the most of the platform's capabilities without encountering unnecessary complexities.

### ***1.3 Training Data and Model Performance:***

The dataset optimization process generally maintained or slightly enhanced the performance of most models, with EfficientNetB0 retaining its position as the top performer in both accuracy and F1 score. The most significant relative improvement was observed in VGG16, indicating that the dataset changes positively impacted its ability to generalize. Although InceptionV3’s accuracy improved, the decrease in its F1 score suggests a potential shift in the balance between precision and recall. The slight drop in performance metrics for MobileNetV3 Large and ResNet50 post-optimization may indicate an initial overfitting to the original dataset or that the optimizations were more conducive to the other models' architectures.

### ***1.4 Cost-effectiveness:***

In terms of cost-effectiveness, the website and model prove to be economically efficient as they require minimal data sets and do not demand extensive GPU resources. Furthermore, the website's commitment to a simple design not only enhances usability but also carries a cost-effective advantage. By prioritising simplicity in the design process, the project has effectively minimised the expenditure associated with elaborate and intricate design elements. This prudent approach not only benefits the project financially but also aligns to deliver a streamlined and accessible platform to a wide range of users.

### ***1.5 Comparison with Existing Models:***

#### *1.5.1 Original Dataset Results:*

EfficientNetB0 emerged as the top performer when using the original dataset, achieving an impressive 96% accuracy and an F1 score of 0.95. Close behind were MobileNetV3 Large and ResNet50, both demonstrating strong performance with accuracies surpassing 90% and F1 scores of 0.93 and 0.89, respectively. On the other hand, VGG16 and InceptionV3 lagged in performance compared to their counterparts, with VGG16 recording an accuracy of 81.4% and InceptionV3 at 80%.

#### *1.5.2 Post-Optimization Results:*

After the implementation of dataset optimization, EfficientNetB0 continued to lead in performance, albeit with a slight increase in accuracy to 96.55% and an improvement in the F1 score to 0.96. In contrast, MobileNetV3 Large and ResNet50 showed a marginal decline in their metrics, with MobileNetV3 Large experiencing a slight dip in accuracy to 92.36% and a decrease in its F1 score to 0.92, and ResNet50's accuracy decreasing to 89.64% but maintaining the same F1 score. Notably, both VGG16 and InceptionV3 saw improvements in their accuracy figures post-optimization, with VGG16’s accuracy climbing to 84% and its F1 score to 0.83, and InceptionV3’s accuracy improving to 78.73%. However, InceptionV3 witnessed a slight decrease in its F1 score to 0.77.

### ***1.6 Future Potential:***

In terms of future potential, when the AI model is developed with proper professionalism and adequate budget allocation, it holds the capability to significantly enhance the sports classification process. With further refinement and investment, this AI model has the potential to revolutionise the way sports are classified.

One of its most compelling prospects lies in streamlining the development process for sports-related applications or websites. Rather than creating separate, specialised apps or websites for each sport, organisations can simply leverage this versatile AI model. By using it to classify the specific sport depicted in an image, the AI can then direct users to the corresponding version of the app or website tailored to that sport. This streamlined approach not only reduces development time and costs but also offers users a more efficient and customised experience.

Beyond its applications in sports technology, this AI model has the potential to contribute to sports education and awareness. Through educational programs and initiatives, it can play a pivotal role in enhancing people's knowledge about a wide range of sports. By dynamically curating and recommending various sports based on individual preferences and awareness gaps, AI can encourage engagement in physical activities and promote a healthier lifestyle. This educational aspect adds another layer of value to the model's future potential, impacting not only the world of technology but also contributing to the broader landscape of sports and well-being.

### ***1.7 Performance Metrics:***

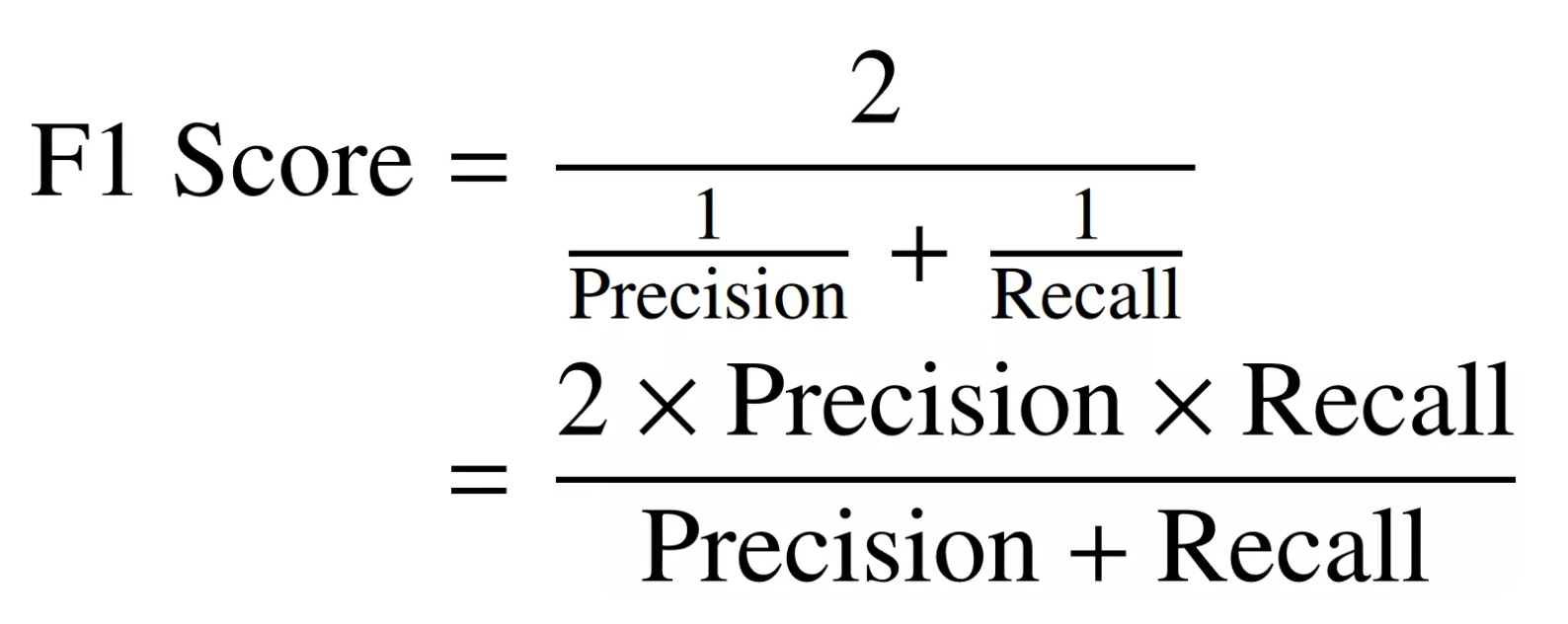


Figure 33: F1 Score Formula

The F1 score, which harmonizes the balance between precision and recall, shows EfficientNetB0 as the standout model. Originally, it achieved an F1 score of 0.95, which post-optimization, saw a marginal improvement to 0.96. This increment underscores EfficientNetB0's robustness and its enhanced capability in maintaining a balanced precision-recall trade-off after dataset optimization. In contrast, MobileNetV3 Large and ResNet50 exhibited a slight shift in their F1 scores. MobileNetV3 Large decreased from 0.93 to 0.92, and ResNet50 maintained a steady F1 score, despite a drop in accuracy. This stability in ResNet50's F1 score suggests its consistent performance in terms of precision and recall, despite varying accuracy rates. VGG16's F1 score improvement from 0.81 to 0.83 post-optimization indicates a notable enhancement in its classification balance, aligning with its accuracy gains. However, InceptionV3 presents an intriguing case; despite an improvement in accuracy, its F1 score slightly decreased from the original dataset, moving from 0.77 to 0.76. This suggests a potential trade-off situation where the model's increased accuracy might be coming at the cost of precision and recall balance. The F1 score thus serves as a vital metric, providing a more nuanced view of each model's performance beyond mere accuracy.

### ***1.8 Technical Challenges:***

* GPU Limitations: One of the foremost technical challenges revolves around GPU resources. The availability and capacity of GPUs play a crucial role in the training and execution of complex AI models, such as the one employed in this project. Insufficient GPU resources can result in slower model training times and may necessitate compromises in model complexity.
* Budget Constraints: Financial limitations pose another significant hurdle. The development and fine-tuning of AI models often require substantial financial resources, encompassing expenses for hardware, software, data acquisition, and personnel. Budget constraints can limit the scope of the project and impact its overall efficiency and effectiveness.
* Hardware Quality: The quality and capabilities of the hardware utilised in the AI development process are crucial factors. Suboptimal hardware can result in performance bottlenecks, hindering the model's ability to process data efficiently and deliver accurate results. Ensuring access to high-quality hardware is essential for overcoming this challenge.
* Data Quantity: The quality and quantity of available data are paramount in training accurate and robust AI models. Inadequate or limited datasets can impede the ability of the AI model to generalise effectively across diverse sports and scenarios. A scarcity of data can hinder the achievement of high classification accuracy.

## **Discussion**:

The AI-powered Sports Classification Website demonstrated favourable outcomes in the realm of sports image classification. The anticipated attributes extracted from this classification hold significant potential for offering valuable insights and facilitating the implementation of necessary corrective measures in the sports domain.

# **CHAPTER 6: CONCLUSION AND FUTURE WORK**

## **Conclusion**:

In conclusion, this thesis has introduced a Sports Classification Website utilising advanced AI-powered image classification. We tackled the challenges and opportunities in sports classification, aiming to make a significant contribution to sports technology.

The project employed Convolutional Neural Networks (CNNs) and an Agile approach to create a user-friendly platform for seamless sports image classification. Our objectives encompassed simplifying sports classification, training a precise CNN model, and rigorous performance evaluation. Despite facing hurdles such as GPU limitations, budget constraints, and data availability, we remain committed to improving the system's functionality. Looking forward, we anticipate a promising future for image classification in sports and education. This technology can motivate people to explore sports, discover suitable activities, and enhance athlete protection.

In essence, this thesis represents not only the culmination of our development journey but also the beginning of a bright future in sports technology. Our commitment to innovation and improvement will continue to make a lasting impact in the world of sports and beyond.

## **Future Work:**

For future work, several exciting enhancements are envisioned to further elevate the Sport Recognition Website and its AI model. One key aspect of this future development plan is the incorporation of user-generated data. To enhance the dataset and improve the AI's accuracy, a user-friendly feature will be implemented. Users will have the option to contribute their sports-related images for classification. However, strict privacy measures will be in place to ensure the confidentiality of user-contributed images. Users must consent to the terms of data usage, and their images will be stored securely and used exclusively for machine learning purposes. This approach not only enriches the dataset but also fosters a sense of user engagement and collaboration in the improvement of the AI model.

In addition to user-generated data, active machine-learning techniques will be integrated into the AI model. This enhancement represents a significant step forward in the AI's capabilities. Instead of solely relying on the existing dataset, the AI will actively learn and adapt from user-contributed data. By continuously evolving and self-improving, the AI will become more proficient at recognizing a wider range of sports, ultimately leading to enhanced classification accuracy. These planned upgrades signify a commitment to creating a more dynamic, user-driven, and self-improving system. Through the combination of user contributions and active machine learning, the Sports Classification Website will continue to evolve, providing users with even more accurate and versatile sports image classification capabilities.

# **REFERENCES**

1. *About Catapult - Catapult*. (2024, January 5). Catapult. https://www.catapult.com/company/about-catapult
2. Catapult. (2023, November 10). Catapult Launches Advanced Integrated Heart Rate Solution for their Vector Athlete Monitoring System. *GlobeNewswire News Room*. https://www.globenewswire.com/news-release/2023/11/10/2778494/0/en/Catapult-Launches-Advanced-Integrated-Heart-Rate-Solution-for-their-Vector-Athlete-Monitoring-System.html
3. *How Wearable Technology Is Changing the Sports Industry*. (n.d.). https://www.cogniteq.com/blog/how-wearable-technology-changing-sports-industry
4. *Introduction to Remote Sensing*. (n.d.). https://seos-project.eu/remotesensing/remotesensing-c06-p01.html
5. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *ResearchGate*. https://www.researchgate.net/publication/319770183\_Imagenet\_classification\_with\_deep\_convolutional\_neural\_networks
6. Lorente, Ò., Riera, I., & Rana, A. (2021). Image Classification with Classic and Deep Learning Techniques. *ResearchGate*. https://www.researchgate.net/publication/351511208\_Image\_Classification\_with\_Classic\_and\_Deep\_Learning\_Techniques
7. Luteberget, L. S., Spencer, M., & Gilgien, M. (2018). Validity of the Catapult ClearSky T6 Local Positioning System for Team Sports Specific Drills, in Indoor Conditions. *Frontiers in Physiology*, *9*. https://doi.org/10.3389/fphys.2018.00115
8. *One platform to help the whole team improve | Hudl*. (n.d.). Hudl. <https://www.hudl.com/products/hudl>
9. Kokhan, S. (2007). Classification of remotely sensed data. In *NATO science for peace and security series* (pp. 239–247). https://doi.org/10.1007/978-1-4020-6438-8\_14
10. Machine Learning Advancements in Organic Synthesis: A focused exploration of artificial intelligence applications in chemistry. (2024). *ScienceDirect*. https://doi.org/10.1016/j.aichem.2024.100049

# **APPENDICES**

|  |  |
| --- | --- |
| **Terms** | **Definitions** |
| AI | Artificial intelligence. |
| ANN | Artificial Neural Network |
| CNN | Convolutional Neural Network |
| FCM | Fuzzy C-Means |
| GPU | Graphics Processing Unit |
| IDE | Integrated Development Environment |
| KNN | K-Nearest Neighbors |
| LPS | Local Positioning System |
| PCA | Principal Component Analysis |
| PIL | Python Image Library |
| SVMs | Support Vector Machines |
| UI | User Interface |
| UX | User Experience |

Table 2: Terms of Glossary