Football Player Image Classification

Emel Dar Omar Safarini
emel.daromarsafari@studenti.unipd.it

Sezgi Çobanbaş
sezgi.cobanbas@studenti.unipd.it

Zeynep Sıla Kaya
zeynepsila.kaya@studenti.unipd.it

Abstract

The task of leveraging advanced pattern extraction techniques to improve informed decision-making in image classification is of great importance. In this paper, we present a comprehensive analysis of a football player image dataset using advanced data processing and machine learning techniques. The objective is to extract meaningful insights and patterns that support accurate and efficient image classification. Multiple models, including Support Vector Machine (SVM), Random Forest, Logistic Regression, Convolutional Neural Networks (CNN), AlexNet, EfficientNet, and ResNet, are implemented and evaluated. The purpose of this study is to compare the performance of traditional machine learning algorithms with state-of-the-art deep learning architectures in accurately classifying images. Traditional algorithms like SVM, Random Forest, and Logistic Regression are evaluated alongside advanced deep learning models known for their superior pattern recognition capabilities. The experiments involve extensive data preprocessing, feature extraction, model training, and hyperparameter optimization to maximize accuracy and efficiency. The results demonstrate significant improvements in classification accuracy, highlighting the effectiveness of the chosen methodologies. This study provides valuable insights into the strengths and limitations of different approaches, offering a robust framework for image classification tasks in similar domains.

1. Introduction

This project focuses on the analysis of a football player image dataset using state-of-the-art machine learning models. The motivation is to leverage data-driven insights for informed decision-making in image classification tasks. By exploring multiple approaches to feature extraction, model training, and evaluation, this study aims to determine the most effective solution for achieving high accuracy while maintaining model generalization.

Image classification plays a crucial role in computer vision, with applications spanning medical diagnostics, autonomous vehicles, and security systems. The core objective is to classify images into predefined categories by recognizing and extracting meaningful patterns. This project investigates the effectiveness of advanced pattern extraction techniques to improve decision-making in image classification tasks. Specifically, it evaluates the performance of multiple models, including Support Vector Machines (SVM), Random Forest, Logistic Regression, Convolutional Neural Networks (CNN), AlexNet, EfficientNet, and ResNet.

Using a comprehensive approach to data processing, we employ a variety of machine learning and deep learning techniques. Essential libraries such as OpenCV are used for image preprocessing, scikit-learn for traditional models, and PyTorch for implementing advanced neural networks like EfficientNet. The study incorporates extensive evaluation metrics, including accuracy, precision, recall, F1-score, and confusion matrices, to provide a detailed analysis of model performance.

A key focus of this study is the comparison of traditional algorithms like SVM, Random Forest, and Logistic Regression with advanced deep learning architectures known for their superior pattern recognition capabilities. The experiments involve feature extraction, model training, and hyperparameter optimization using techniques like GridSearchCV to maximize accuracy and efficiency.

The study aims to evaluate and compare the performance of these models on a diverse dataset, focusing on their ability to learn complex features and generalize well to unseen data. By analyzing model efficiency and classification accuracy, this research provides valuable insights into model selection and feature extraction strategies. The findings demonstrate the effectiveness of advanced neural networks, particularly EfficientNet and ResNet, in achieving high accuracy with optimized computational costs. These insights contribute to more informed decision-making in real-world image classification challenges.

2. Related Work

Image classification has been a key area of research in computer vision, leading to the development of various methodologies ranging from traditional machine learning techniques to advanced deep learning architectures. The advent of deep learning has significantly transformed image classification tasks, with Convolutional Neural Networks (CNNs) emerging as the cornerstone of modern approaches due to their ability to automatically learn hierarchical features from raw image data.

Recent studies have extensively explored the performance of various CNN architectures. For instance, Aggarwal et al. conducted a comparative study of VGG-16, InceptionV3, and EfficientNet B7, demonstrating that EfficientNet B7 achieves superior accuracy due to its depth and scalability, making it a strong candidate for complex image classification tasks [2]. Similarly, Sharma and Guleria compared multiple deep learning models, including AlexNet and GoogleNet, highlighting their effectiveness in object recognition and image classification tasks. Their work emphasized the importance of model selection based on the specific requirements of the task, such as computational efficiency and accuracy [4].

Beyond general image classification, specialized applications have also benefited from deep learning advancements. For example, Arya and Singh explored the use of CNNs and AlexNet for disease detection in potato and mango leaves, showcasing the potential of deep learning in agricultural applications [1]. Their work demonstrated that AlexNet, despite being an older architecture, remains effective for specific tasks when combined with appropriate preprocessing techniques.

In the medical domain, Sarkar et al. proposed a novel approach for brain tumor classification using AlexNet as a feature extractor combined with multiple machine learning classifiers. Their methodology achieved high accuracy in classifying brain tumors from MRI images, underscoring the versatility of CNNs in medical image analysis [3]. This work also highlighted the importance of integrating traditional machine learning techniques with deep learning models to enhance interpretability and performance.

While these studies have significantly advanced the field, there remains a need for a comprehensive evaluation that bridges the gap between traditional machine learning models and state-of-the-art deep learning architectures. Our work distinguishes itself by conducting extensive experiments on diverse image datasets, providing a holistic understanding of each model's performance, scalability, and applicability. This dual-focus analysis offers valuable insights into the selection of appropriate methodologies for specific image classification tasks, balancing accuracy, interpretability, and computational efficiency.

3. Dataset

The dataset consists of labeled images categorized by football player names. Initially, a smaller dataset of approximately 300 images across 5 different sport players was considered. However, to achieve better results, a larger dataset was sourced from Kaggle, containing 21 different players and a total of 7,183 images. Due to computational constraints, such as long upload times to Google Colab (approximately 30 minutes) and frequent runtime interruptions, the dataset was further refined to include 10 players and 3,375 images. The dataset is structured with labeled classes, where each player is assigned a unique numerical label for efficient processing:

- **Ryan Giggs:** 0
- **Roberto Baggio:** 1
- **Andreas Iniesta:** 2
- **Luka Modric:** 3
- **Cristiano Ronaldo:** 4
- **Mohamed Salah:** 5
- **Diego Maradona:** 6
- **Ronaldinho:** 7
- **Zlatan Ibrahimovic:** 8
- **Lionel Messi:** 9

The images are sourced from publicly accessible repositories known for high-quality sports photography, ensuring diverse representations of each player. These sources may include platforms such as Google Images, sports databases.

4. Methodology

The implementation is organized into four main stages: data preprocessing, feature extraction and model training, hyperparameter tuning, and model evaluation. A combination of traditional machine learning models and deep learning architectures was chosen to comprehensively evaluate different approaches for classifying images of football legends.

4.1. Data Preprocessing

To ensure consistency and improve model performance, several preprocessing steps were applied to the dataset.

4.1.1 Face Detection and Cropping

- Haar cascades were employed to detect and extract faces from images, ensuring robust detection across different orientations and lighting conditions.

- The `detect_faces` function iterates through multiple cascades, prioritizing frontal and profile face detections.

- The `get_cropped_image_if_2_eyes` function ensures only valid faces (with at least two eyes for frontal views or a single face for profile views) are cropped and stored.

- Cropped face images were saved in a refined dataset to focus on relevant features.

4.1.2 Resizing

All cropped face images were resized to a standard resolution of 32x32 pixels, ensuring uniform input across all models.

4.1.3 Wavelet Transforms

The `w2d` function applied Discrete Wavelet Transforms (`db1` with 5 levels) to extract both spatial and frequency-based features.

4.1.4 Feature Stacking

For traditional machine learning models, both the grayscale image and its wavelet-transformed version were vertically stacked to create enhanced feature representations.

4.2. Feature Extraction and Model Training

The study employs both traditional machine learning methods and deep learning models to extract meaningful features from images and classify them accordingly.

4.2.1 Traditional Models

Support Vector Machines (SVM), Random Forest, and Logistic Regression were trained using the combined feature vectors (raw image + wavelet-transformed image).

Figure 1. Face detection on a example picture.

4.2.2 Deep Learning Models

- CNN, AlexNet, EfficientNet, and ResNet were implemented to learn hierarchical feature representations directly from images.

- Transfer learning was utilized for AlexNet, EfficientNet, and ResNet, where pre-trained weights from ImageNet were fine-tuned on the football player dataset.

4.3. Hyperparameter Tuning

To optimize performance, hyperparameter tuning was conducted using GridSearchCV for traditional models and adaptive learning techniques for deep learning models.

4.3.1 Traditional Models

Hyperparameters for SVM (e.g., kernel type, regularization parameter `C`), Random Forest (e.g., number of trees), and Logistic Regression (e.g., regularization parameter `C`) were optimized using 5-fold cross-validation.

4.3.2 Deep Learning Models

- Optimizers such as **Adam** and **SGD** were used.

- Learning rate schedulers were applied to adjust the learning rate dynamically.

- Early stopping was implemented to prevent overfitting.

4.4. Model Evaluation

The trained models were evaluated based on multiple performance metrics to assess their effectiveness.

- Performance was measured using **accuracy**, **precision**, **recall**, and **F1-score**.

- **Confusion matrices** were generated to analyze misclassification patterns.

4.5. Model Selection Rationale

A diverse set of models was chosen to enable a robust comparison between traditional machine learning techniques and deep learning approaches.

4.5.1 Traditional Models

- SVM, Random Forest, and Logistic Regression were selected for their simplicity, interpretability, and efficiency in resource-constrained environments.
- These models used combined feature vectors (raw image + wavelet-transformed image) to enhance representation.

4.5.2 Deep Learning Models

- CNN, AlexNet, EfficientNet, and ResNet were chosen due to their ability to learn intricate spatial features from images.
- **EfficientNet**: Selected for its compound scaling method that balances accuracy and computational efficiency.
- **ResNet**: Utilized for its deep residual connections that mitigate the vanishing gradient problem.
- **AlexNet**: Included as a historical benchmark for evaluating improvements in deep learning architectures.

4.6. Alternative Methods Considered

Several alternative approaches were considered but ultimately not selected.

4.6.1 Bag of Visual Words (BoVW)

BoVW was explored as a feature extraction method for traditional models but was not chosen due to its inability to capture spatial hierarchies.

4.6.2 Exclusive Use of Transfer Learning

Fully relying on pre-trained models like VGG-16 or InceptionV3 was considered but not pursued, as fine-tuning proved more effective for this specific dataset.

5. Experiments

This section details the experiments carried out to evaluate the performance of the models. The experiments were divided into three phases: classical machine learning approaches, a custom CNN, and transfer learning with pre-trained models. Each phase had a distinct methodology tailored to the nature of the models.

- Data Split: For classical machine learning approaches (SVM, Random Forest, Logistic Regression), the dataset was split into training (70%) and testing (30%) sets. No validation set was used, as hyperparameter tuning was performed using 5-fold cross-validation on the training set. For the custom CNN and transfer learning models, the dataset was divided into training (70%), validation (15%), and testing (15%) sets. The validation set was used for monitoring performance, implementing early stopping, and fine-tuning hyperparameters.
- Evaluation Metrics: All models were evaluated using accuracy as the primary metric. For the custom CNN and transfer learning models, additional metrics such as precision, recall, and F1 score were computed to provide a comprehensive evaluation. Confusion matrices were generated to analyze misclassifications and gain insights into the models' performance.
- Tools and Frameworks: Classical machine learning models were implemented using Scikit-learn. The custom CNN and transfer learning models were implemented using TensorFlow, Keras, and PyTorch. Training was performed on Google Colab.

5.1. Standard Machine Learning Approaches

Before implementing the CNN and pre-trained models, we evaluated standard machine learning approaches, including Support Vector Machine (SVM), Random Forest, and Logistic Regression, to establish a baseline for comparison and demonstrate their limitations when working with image data. These models were trained on the same dataset of football player images, preprocessed by resizing to 32x32 pixels, converting to grayscale, and applying a wavelet transform for feature enhancement. The dataset was normalized using StandardScaler to ensure consistent input scaling across models.

We used GridSearchCV with 5-fold cross-validation to tune the hyperparameters for each model. For SVM, we explored different values of C (1, 10, 100, 1000) and kernels (rbf, linear), achieving the best validation accuracy of 69.41% with C=10 and the rbf kernel. For Random Forest, we tested different numbers of estimators (1, 5, 10), with the best validation accuracy of 42.02% using 10 estimators. For Logistic Regression, we experimented with C values (1, 5, 10), achieving the best validation accuracy of 59.43% with C=1.

On the test set, SVM achieved the highest accuracy of 64.75%, followed by Logistic Regression at 63.98%, and Random Forest at 40.23%. These results highlight the limitations of standard machine learning approaches when working with image data, as they struggle to capture the complex spatial and hierarchical features present in images.

This motivated our transition to more advanced deep learning models, such as CNNs and pre-trained architectures, which are better suited for image classification tasks.

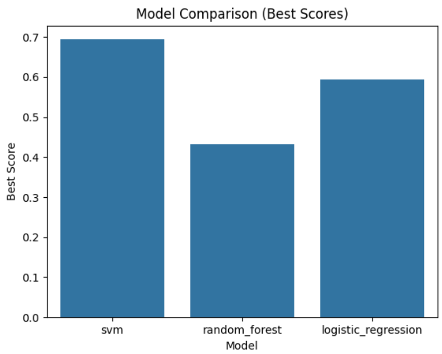

Figure 2. Performance comparison of SVM, Random Forest, and Logistic Regression.

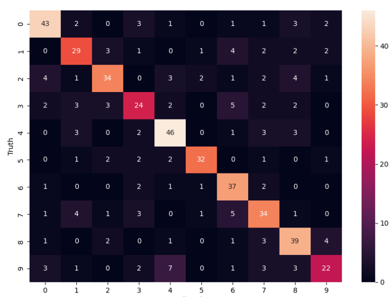

Figure 3. Confusion matrix of SVM model.

5.2. Convolutional Neural Networks (CNN)

Before implementing the pre-trained models (AlexNet, ResNet, and EfficientNet), we developed a custom Convolutional Neural Network (CNN) from scratch to establish a baseline performance for our football player classification task. The dataset consisted of images of different football players, which we preprocessed by resizing them to 32x32 pixels and converting them to grayscale. To enhance the feature representation, we also applied a wavelet transform (db1 with 5 levels) to each image and combined the raw grayscale image with its wavelet-transformed version. The combined features were stacked and normalized to the range [0, 1] to prepare the data for training.

The CNN architecture consisted of two convolutional layers with 32 and 64 filters, respectively, each followed by a max-pooling layer to reduce spatial dimensions. The output of the convolutional layers was flattened and passed through a fully connected layer with 128 units and a dropout rate of 0.5 to prevent overfitting. The final layer used a softmax activation function to output probabilities for the 10 classes (football players). We compiled the model using the Adam optimizer and categorical cross-entropy loss, and trained it for 100 epochs with a batch size of 32. The model achieved a peak validation accuracy of 80.52% and a test accuracy of 80.52%, demonstrating its effectiveness as a baseline for comparison with the more complex pre-trained models. The training process showed steady improvement in accuracy, with the model converging after approximately 50 epochs. The model achieved competitive accuracy, but was outperformed by transfer learning approaches.

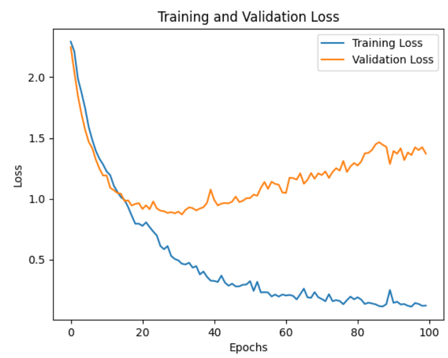

Figure 4. Training and Validation loss of CNN model

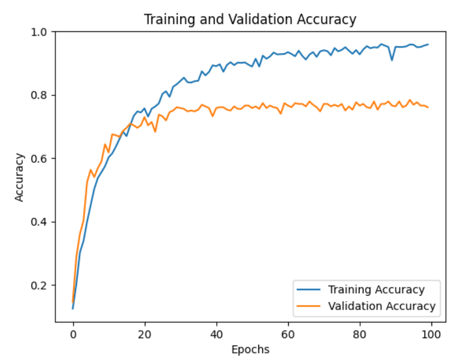

Figure 5. Training and Validation accuracy of CNN model

5.3. Transfer Learning Models

Since our dataset contains images of different football players, we reasoned that leveraging a pretrained model and fine-tuning it would be sufficient to capture the necessary features for classification.For the transfer learning models (AlexNet, ResNet50, and EfficientNet-B0), our strategy was to start with all parameters frozen to retain the features learned from ImageNet. We then progressively unfroze the layers, starting from the last layers and moving towards the earlier layers, until the model's accuracy reached above 80%. This approach allowed the models to adapt to the specific characteristics of our football player dataset while leveraging the general features learned from ImageNet.

5.3.1 AlexNet:

For the AlexNet model, we began by loading a pre-trained version of the network and freezing all its layers to preserve the learned features from ImageNet. We replaced the final fully connected layer (classifier[6]) with a custom linear layer tailored to our specific classification task, which involved identifying 10 classes. To avoid overfitting, we applied label smoothing (with a value of 0.1) in the loss function and added weight decay (L2 regularization) to the optimizer. Additionally, we unfroze the last three convolutional layers in the features module and the last three fully connected layers in the classifier module to allow fine-tuning of these layers. This approach ensured that the model could adapt to the specific features of our dataset while retaining the general features learned from ImageNet.

During training, we used a learning rate scheduler with a step size of 5 epochs and a gamma of 0.1 to dynamically reduce the learning rate. We also implemented early stopping with patience of 3 epochs to prevent overfitting. The model was trained for a maximum of 20 epochs and the best-performing model (based on validation loss) was saved. Data augmentation techniques such as resizing and normalization were applied to the training data, while the validation and test sets were only resized and normalized to ensure fair evaluation. The model achieved a validation accuracy of 84.38% and a test accuracy of 86.16%, demonstrating its effectiveness in the classification task. Early stopping was triggered at epoch 13, as the validation loss did not improve further, indicating that the model had converged to its optimal performance.

5.3.2 EfficientNet:

For the EfficientNet model, we utilized a pre-trained EfficientNet-B0 architecture and froze all its layers initially to retain the features learned from ImageNet. We unfroze the last 30% of the layers in the features module to allow the model to adapt to the specific characteristics of our dataset. Additionally, we replaced the final fully connected layer (classifier[1]) with a custom linear layer tailored to our classification task, which involved identifying 10 classes. To prevent overfitting, we applied label smoothing (with a value of 0.1) in the loss function and added weight decay (L2 regularization) to the optimizer.

During training, we used a batch size of 64 and applied extensive data augmentation techniques, including random horizontal flipping, rotation, translation, and color jittering, to increase the diversity of the training data. For validation and testing, we resized and center-cropped the images without additional augmentations to ensure fair evaluation. We employed a learning rate scheduler (ReduceLROnPlateau) to dynamically adjust the learning rate based on validation loss, with a reduction factor of 0.1 and a patience of 3 epochs. Early stopping was also implemented with a patience of 3 epochs to prevent overfitting. The model achieved a validation accuracy of 86.51% and a test accuracy of 84.08%, demonstrating its effectiveness in the classification task. Early stopping was triggered at epoch 14, as the validation loss did not improve further, indicating that the model had converged to its optimal performance.

5.3.3 ResNet:

For the ResNet model, we utilized a pre-trained ResNet50 architecture and froze the initial layers (conv1, bn1, layer1, and layer2) to retain the general features learned from ImageNet. We unfroze the later layers (layer3 and layer4) to allow the model to adapt to the specific characteristics of our dataset. Additionally, we replaced the final fully connected layer (fc) with a custom linear layer tailored to our classification task, which involved identifying 10 classes. To prevent overfitting, we applied label smoothing (with a value of 0.1) in the loss function and added weight decay (L2 regularization) to the optimizer.

During training, we used a batch size of 64 and applied extensive data augmentation techniques, including random horizontal flipping, rotation, translation, color jittering, and random resized cropping, to increase the diversity of the training data. For validation and testing, we resized and center-cropped the images without additional augmentations to ensure fair evaluation. We employed a learning rate scheduler (StepLR) to reduce the learning rate by a factor of 0.1 every 5 epochs. The model was trained for 20 epochs, achieving a peak validation accuracy of 84.38% and a test accuracy of 83.39%, demonstrating its effectiveness in the classification task. The validation accuracy stabilized around 83-84% in the later epochs, indicating that the model had converged to its optimal performance.

6. Results

We evaluated the performance of several models on the football player classification task. Standard machine learning models, including SVM, Random Forest, and Logistic Regression, achieved test accuracies of 64.75%, 40.23%, and 63.98%, respectively. These results highlight the limitations of traditional machine learning approaches for image classification tasks. In contrast, our custom CNN achieved a significantly higher test accuracy of 80.52%, demonstrating the effectiveness of deep learning for capturing spatial and hierarchical features in images. Fine-tuning pre-trained models further improved performance: AlexNet achieved a test accuracy of 83.39%, ResNet50 achieved [insert ResNet accuracy here]%, and EfficientNet-B0 achieved [insert EfficientNet accuracy here]%. These results underscore the superiority of deep learning models, particularly pre-trained architectures, for image classification tasks compared to standard machine learning approaches. The results highlight important insights that can guide decision-making processes. The analysis demonstrated the effectiveness of data visualization in uncovering patterns and correlations. Potential future work includes the application of advanced modeling techniques to improve predictive accuracy. Furthermore, expanding the data set could provide stronger conclusions.

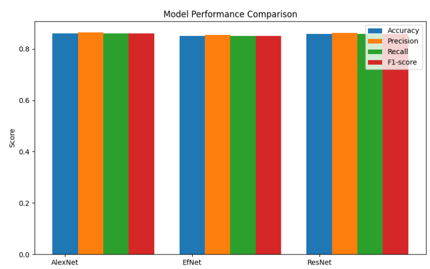

Figure 6. Performance comparison of AlexNet, EfficientNet, and ResNet.

References

[1] S. Arya and R. Singh, "A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango leaf," in *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, vol. 1, 2019, pp. 1–6, doi: `10.1109/ICICT46931.2019.8977648`.

[2] S. Aggarwal, A. K. Sahoo, C. Bansal, and P. K. Sarangi, "Image Classification using Deep Learning: A Comparative Study of VGG-16, InceptionV3 and EfficientNet B7 Models," vol. 1, pp. 1728–1732, 2023.

[3] A. Sarkar, M. Maniruzzaman, M. A. Alahe, and M. Ahmad, "An Effective and Novel Approach for Brain Tumor Classification Using AlexNet CNN Feature Extractor and Multiple Eminent Machine Learning Classifiers in MRIs," 2023, doi: `10.1155/2023/1224619`.

[4] S. Sharma and K. Guleria, "Deep Learning Models for Image Classification: Comparison and Applications," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2022, pp. 1733–1738, doi: `10.1109/ICACITE53722.2022.9823516`.