Sports Classification

Course: Advanced Machine Learning

Group 3 – Álvaro Sáenz-Torre, Mariona Pla, and Martina Carretta

Table of contents:

[**1. Introduction 2**](#_wmzl60lq4k6)

[**2. Data 2**](#_612khk3rbhd5)

[2.1. Data splits 3](#_vky7a3roa9g6)

[**3. Methodology 3**](#_2yo58t1kz1rh)

[3.1. Problem formulation 3](#_oc9cp64l9m4s)

[**4. Model architecture 3**](#_rk1fbhp95hj)

[4.1. Routing 3](#_3kkklbysf1rr)

[**5. Training Strategy 3**](#_xgcctdum97f4)

[5.1. Warmup phase 3](#_5n00q9hd804o)

[5.2. Semi-supervised Training phase 4](#_k2y0zh12vrnk)

[5.3. Penalization term 4](#_ubaoevcm21cr)

[**6. Experimental results 5**](#_12mt2jzenrjl)

[6.1. Baseline 5](#_fsvtxr51kcrj)

[6.2. Training from scratch + warmup and penalization 5](#_czbb015qiib7)

[6.3. Pretrained Backbone 5](#_vin3ih1d8wfe)

[6.4. Effect of unlabelled percentage 6](#_xxz7fpdzst95)

[6.5. Effect of penalization 6](#_st2wkdweqanf)

[6.6. Effect of warmup when using pretrained weights 6](#_cx53x1uahsbs)

[**7. Results analysis 7**](#_qzmfcai6ijht)

[**8. Limitations and future extensions of our work 7**](#_jjiotrmqow)

[**9. Conclusions 7**](#_rex1kicg3lp0)

[**10. References 9**](#_4kfzv6bv8rtt)

[**11. Appendices 10**](#_n8tsz33y5ff2)

# Introduction

Determining the sport depicted in an image is typically straightforward for humans, largely because contextual cues are interpreted automatically. The presence of water suggests aquatic sports such as swimming or rowing; snow indicates winter sports such as skiing or snowboarding; and a court environment naturally narrows the possibilities to sports such as tennis, volleyball, or basketball.

This project explores whether a single vision system can classify 100 fine-grained sports categories by first grounding its understanding in 11 high-level superclasses such as water sports, aerial sports, precision and target sports, and motor and wheel racing sports. These broad contextual classes provide a useful structural prior, yet the real challenge lies in distinguishing the much subtler variations within them, such as wingsuit flying vs. skydiving or parallel bars vs. uneven bars.

Complicating the task further, many subclass labels are absent, transforming the problem into a multi-task semi-supervised learning scenario. To leverage this partially labeled data, Unsupervised Data Augmentation (UDA) has been employed, enabling the model to extract meaningful supervisory signals from unlabelled images through consistency regularization.

# Data

This project makes use of two datasets comprising a combined total of approximately 24,000 images spanning 100 sports categories. For the purposes of multi-task, these 100 subclasses were aggregated into 11 broader superclasses, reflecting high-level contextual groupings.

| Superclass | Examples of subclasses |
| --- | --- |
| Precision & target sports | Archery, Billiards, Golf… |
| Water sports | Swimming, Rowing, Surfing… |
| Field & team call sports | Baseball, Lacrosse, Rugby… |
| Court ball sports | Basketball, Tennis, Table Tennis… |
| Combat & strength sports | Boxing, Judo, Rock Climbing… |
| Equestrian & animal sports | Barrel Racing, Bull Riding, Polo… |
| Ice & snow sports | Hockey, Ice Climbing, Snowboarding… |
| Motor & wheel racing | BMX, Bike Polo, Sidecar Racing… |
| Gymnastics | Balance Beam, Rings, Trapeze… |
| Aerial sports | Bungee Jumping, Sky Surfing, Skydiving… |
| Track & Field athletics | Hammer Throw, Javelin, Pole Vault… |

## Data splits

The data were partitioned into training (70%), validation (15%), and test (15%) splits. The division was performed in a stratified manner to preserve class balance across both the superclass and subclass levels, ensuring that performance evaluation is not biased by class imbalance.

# Methodology

## Problem formulation

Given an input image , we have a superclass label , and its subclass label (when available) The objective is to train a model that predicts a coarse superclass and a fine-grained subclass while using unlabelled subclass data via semi-supervised learning

# Model architecture

The model has a hierarchical multi-task design in which superclass and subclass classification are handled jointly but with different heads.

1. Backbone: A ResNet-18 convolutional neural network extracts visual features from the input.
2. A projection head reduces the backbone output to a lower-dimensional feature vector.
3. A superclass classifier predicts one of the 11 superclasses.
4. Eleven specialized subclass heads, one per superclass, predict the subclass within the corresponding superclass using the shared projection features rather than the superclass logits.

The superclass head is slightly more complex and uses dropout to make general predictions across all data, while the subclass heads are simple linear layers that specialize in distinguishing classes within a single superclass.

## Routing

During training, the features extracted for each sample are routed to the appropriate subclass head based on its ground-truth superclass (for labelled and unlabelled data). During validation, routing is performed based on the predicted superclass, rather than the ground-truth label, making it depend solely on itself’s prediction thus replicating the conditions of inference.

# Training Strategy

## Warmup phase

Training begins with a warmup stage, during which only the superclass classifier is optimized. This stabilises the feature extractor and ensures the network learns contextual structure before tackling subclass prediction or unlabelled samples.

This phase stabilises the backbone and projection head and avoids noisy subclass optimisation early in training.

## Semi-supervised Training phase

After the warmup, the full semi-supervised training begins. Each training iteration receives a batch of labelled images and a batch of unlabelled images.

* **Labelled samples**: the model is supervised at both levels
  + Superclass loss
  + Subclass loss: features are routed to the corresponding subclass head using the ground-truth superclass.
* **Unlabelled samples** and **UDA**: even though subclass labels are missing, superclass labels remain available. To exploit this data, we use Unsupervised Data Augmentation (UDA), which is based on the principle that a model’s predictions should remain consistent under strong input perturbations.
  + For each unlabelled image, two predictions are computed:
    - One for the original image
    - One for the augmented image (RandAugment)

UDA enforces consistency between these two predictions and is useful in this problem given that sports classification tasks with subclasses like ours have a high level of similarity between the data of different subclasses.

To generate strong perturbations without the need of manual tuning, we used RandAugment. This encourages the model to learn augmentation-invariant representations and to treat both images the same way. RandAugment was chosen over AutoAugment as it avoids costly search and labelled validation data, while providing most of AugoAugment’s benefits with far less complexity.

In these unlabelled samples, we did not compute and include the superclass loss that we apply in the labelled samples because our hypothesis was that not including this part we forced the model to encode the inputs and the augmentations by only considering the KL divergence loss in this second part. With this technique we aimed to enhance the consistency regularisation term by not weakening its effect in the back propagation step by including the superclass loss in the calculation of the gradients.

## Penalization term

The total loss per batch becomes .

Where is the penalization term that increases the loss when the superclass prediction is incorrect, in order to reduce routing errors to incorrect subclass heads and is the penalisation weight.

# Experimental results

## Baseline

To evaluate the effectiveness of our proposed model, we first trained a baseline model. This model was trained from scratch (random initialization), without utilizing any pre-trained weights, warmup periods, or our proposed penalization loss. This baseline serves as a lower bound, allowing us to quantify the specific performance gains achieved by our proposed training strategies and architectural modifications.

This model achieved a superclass accuracy of 68% and a subclass accuracy of 25% using 50 epochs.

## Training from scratch + warmup and penalization

When trained from scratch, the model required approximately 50 epochs to converge to a superclass training accuracy of about 90% and a subclass accuracy of around 85%.

The warmup phase played a critical role when training from scratch. During warmup, the model’s superclass accuracy increased from near-random performance to approximately 40%, allowing the backbone and projection layers to acquire meaningful contextual representations. As a result, when semi-supervised training began, the model already possessed an understanding of the superclass structure.

Even after convergence, the performance was:

* Superclass accuracy ranged between 70% and 73% depending on the penalization\_weight value
* Subclass accuracy ranged between 44% and 45% depending on the penalization\_weight value

The penalization value was tested with the following values: [0.0, 0.5, 0.8, 1.0]

Training time was significant, making experimentation slow and computationally expensive.

## Pretrained Backbone

To address this, we initialized the backbone with ResNet-18 weights. This resulted in a significant improvement in both convergence speed and final performance:

* Only ~15 epochs total (including warmup)
* Near perfect training accuracy
* Validation accuracy between 92–94% for superclasses and 83–84% for subclasses
  + All tested with the penalization\_weight values as well

## Effect of unlabelled percentage

We analysed the impact of the proportion of unlabelled data used during training. Experiments were conducted using 20%, 50%, and 70% unlabelled samples, while keeping the total dataset size fixed and all other hyperparameters unchanged.

Across all configurations, the observed performance differences were minor. As illustrated in Figures 8 and 9, both validation superclass accuracy and subclass accuracy remained within a narrow range with small fluctuations.

These results suggest that, within the explored range, changing the amount of unlabelled data does not substantially improve performance. The model is already able to extract most of the useful supervisory signal from the available labelled samples, and UDA primarily acts as a regulariser. Once pretrained representations are used, the system appears to be relatively insensitive to the ratio.

## Effect of penalization

We evaluated the impact of the penalisation weight for both training settings: from scratch and with pretrained weights.

For the pretrained backbone, variations in the penalisation weight produced only marginal changes:

* Superclass accuracy ranged from 92% to 94%
* Subclass accuracy ranged from 83% to 84%

For the model trained from scratch, performance remained consistently low regardless of penalisation strength:

* Superclass accuracy ranged from 70% to 73%
* Subclass accuracy ranged from 44% to 45%

These results suggest that, while penalisation helps stabilise hierarchical training, its effect is secondary compared to the impact of pretrained feature representations. Without strong initial features, penalisation alone is insufficient to recover fine-grained performance.

## Effect of warmup when using pretrained weights

In addition to changing the penalization weight, we also tested the impact of the warmup phase when using pretrained backbone weights. We trained the model with and without warmup epochs with both penalization and no penalization.

The results showed no significant difference between the settings. All of them achieved comparable convergence speeds and nearly identical validation accuracies (with differences of about 0.001).

This indicates that with a strong backbone, the warmup phase becomes unnecessary as the model already knows semantically meaningful representations.

# Results analysis

The results obtained highlight that using pretrained weights significantly improves the training times and is a dominant factor influencing the convergence speed and final accuracies. Hierarchical decomposition simplifies the fine-grained classification task, however, subclass performance remains fundamentally bounded by superclass routing quality. Penalisation provides limited gains once the backbone representations are sufficiently strong.

# Limitations and future extensions of our work

Despite the encouraging results obtained, several limitations of our approach remain. First, the hierarchical architecture inherently depends on the correctness of the superclass prediction. Errors at this level propagate directly to the subclass prediction through incorrect routing, which fundamentally bounds fine-grained performance. While the penalisation term partially mitigates this effect, it cannot fully compensate for incorrect high-level decisions.

Second, although Unsupervised Data Augmentation enables the exploitation of unlabelled subclass data, its effectiveness appears limited when the backbone representations are weak. In the training-from-scratch scenario, neither UDA nor penalisation were sufficient to recover competitive subclass accuracy, highlighting the strong dependency of semi-supervised learning techniques on feature quality.

From a computational perspective, training the model from scratch is costly and slow, restricting extensive hyperparameter exploration and architectural experimentation.

Several future directions could address these limitations. A natural extension would be the integration of pseudo-labeling or self-training strategies for unlabelled subclass data, potentially combined with confidence-based filtering. Furthermore, replacing the backbone with larger or more expressive architectures, such as deeper convolutional networks or vision transformers, may significantly enhance representation quality.

# Conclusions

In this work, we studied the problem of sports image classification when fine-grained labels are only partially available. To address this challenge, we proposed a hierarchical multi-task learning approach that predicts both high-level superclasses and detailed subclasses, while also using semi-supervised learning to exploit unlabelled data.

Our results show that organizing the problem into a hierarchy helps the model use contextual information and reduces the complexity of fine-grained classification. Unsupervised Data Augmentation proved useful for regularizing the model when subclass labels were missing. However, the experiments clearly indicate that the most important factor for good performance is the quality of the feature representations. Models using a pretrained backbone converged much faster and achieved significantly higher accuracy than models trained from scratch.

We also observed that, when pretrained features are used, additional techniques such as warmup training and penalisation have only a minor impact on the final results. This suggests that strong visual representations already provide enough structure for stable hierarchical learning.

In conclusion, hierarchical semi-supervised learning is an effective approach for large-scale sports classification with incomplete annotations, but its success strongly depends on using pretrained models. These findings highlight the importance of representation learning for fine-grained visual classification tasks in real-world settings.

# References

[1] G. Duttakiit, *Sports Identification Dataset*. Kaggle, 2020.  
 Available at:<https://www.kaggle.com/datasets/gauravduttakiit/sports-identification>

[2] G. Piosenka, *Sports Classification Dataset*. Kaggle, 2021.  
 Available at:<https://www.kaggle.com/datasets/gpiosenka/sports-classification>

[3] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, “Unsupervised Data Augmentation for Consistency Training,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.  
 Available at: <https://arxiv.org/abs/1904.12848>

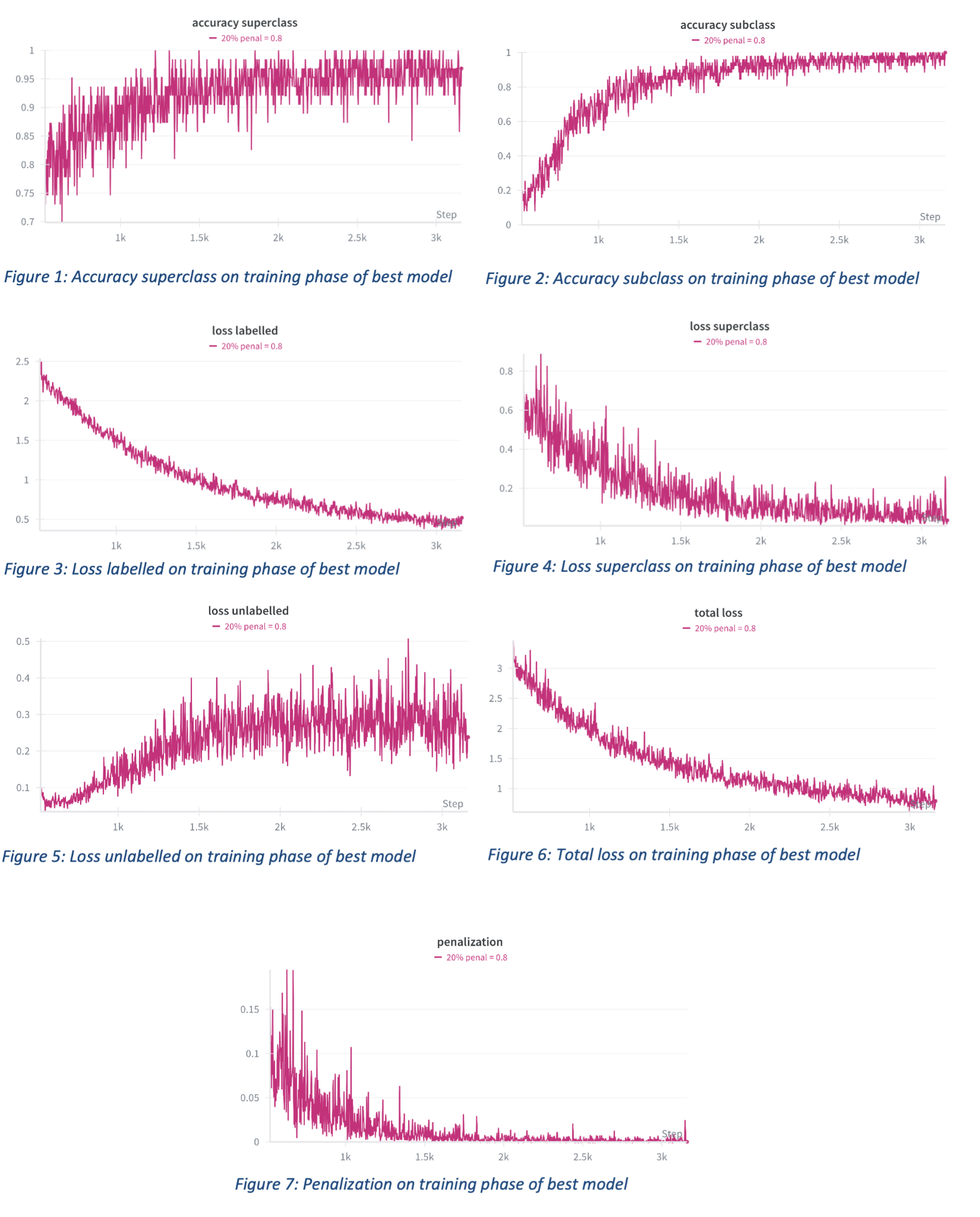
[4] Google Research, *Unsupervised Data Augmentation (UDA) – Official Implementation*, 2019.  
 Available at:<https://github.com/google-research/uda>

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Available at: <https://arxiv.org/abs/1512.03385>

[6] S. Ruder, “An Overview of Multi-Task Learning in Deep Neural Networks,” *arXiv preprint arXiv:1706.05098*, 2017. Available at: <https://arxiv.org/abs/1706.05098>

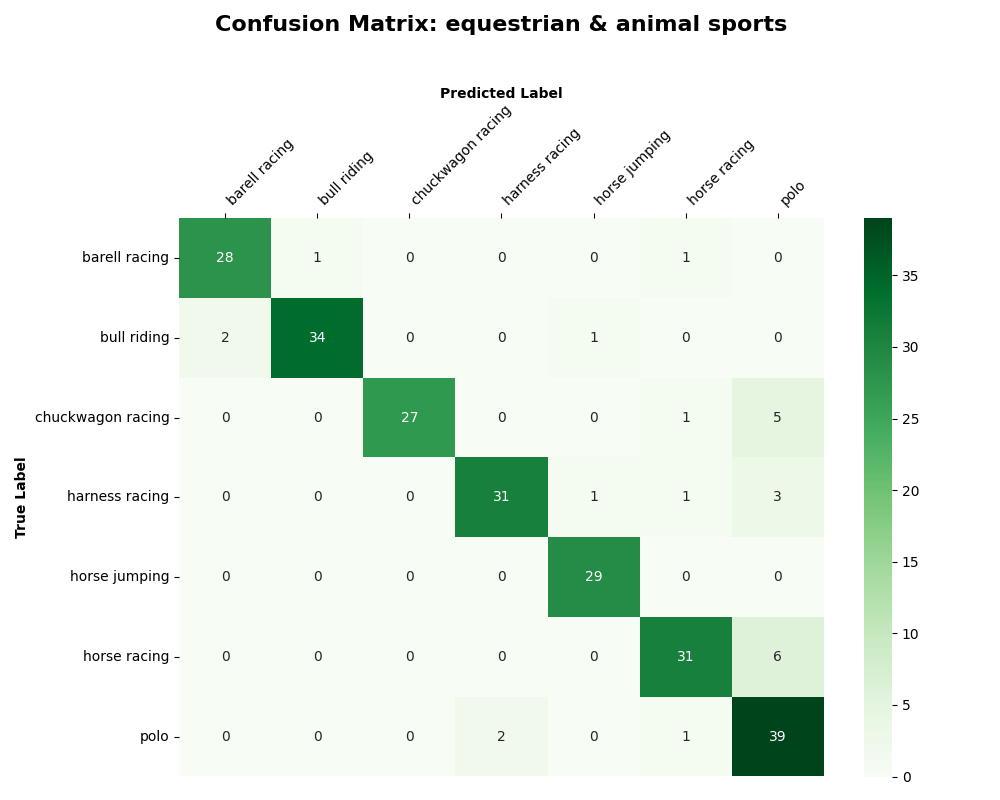
[7] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “RandAugment: Practical Automated Data Augmentation with a Reduced Search Space,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020. Available at: <https://arxiv.org/abs/1909.13719>  
  
[8] Our repository: <https://github.com/Martinacarretta/amlProject>

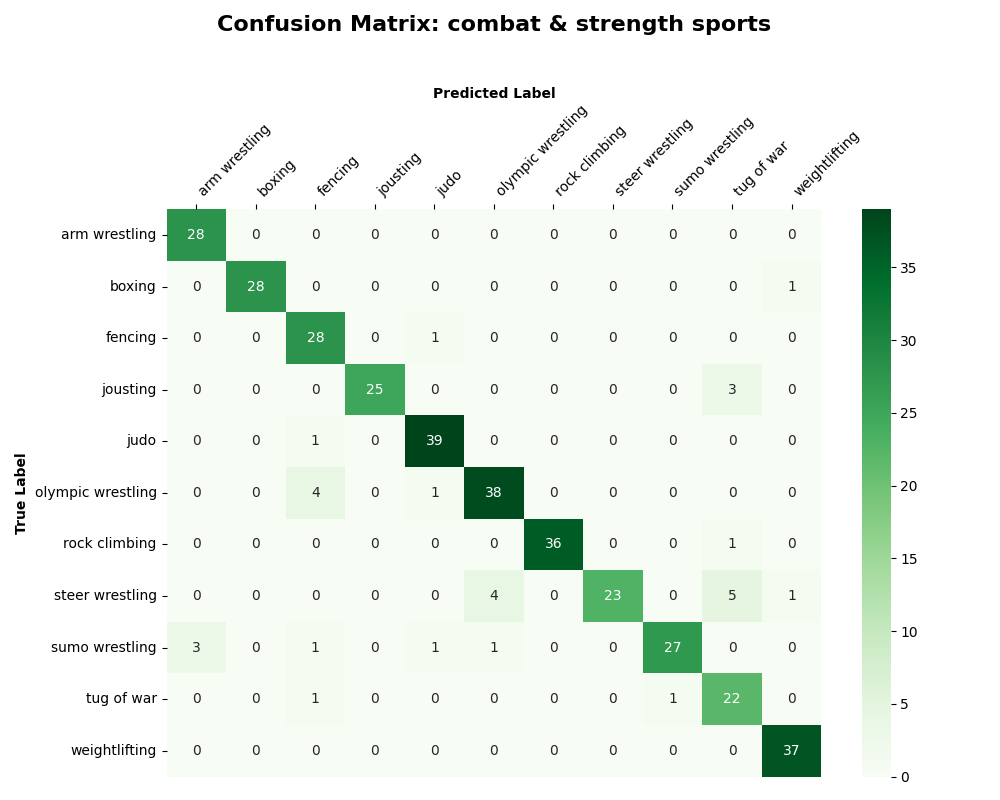
# Appendices







Examples of confusion matrix at subclass level:



Comparative with baseline, proposed solution and enhanced solution:

