MASARYK
UNIVERSITY

FACULTY OF INFORMATICS

Difficulty Classification of Moonboard Bouldering Problems

Bachelor's Thesis

EDUARD MINKS

Brno, Fall 2022

MASARYK UNIVERSITY

FACULTY OF INFORMATICS

Difficulty Classification of Moonboard Bouldering Problems

Bachelor's Thesis

EDUARD MINKS

Advisor: RNDr. Petr Eliáš, Ph.D.

Department of Machine Learning and Data Processing

Brno, Fall 2022


FACULTAS ARTIS INFORMATICÆ
UNIVERSITAS MASARYKIANA

Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Eduard Minks

Advisor: RNDr. Petr Eliáš, Ph.D.

Acknowledgements

I thank my advisor RNDr. Petr Eliáš, Ph.D. for his neverending support.

Abstract

This work aims to apply existing approaches of deep learning to the task of the difficulty grade assignment to the bouldering problems set on a standardized MoonBoard wall. Four distinct modalities are used to classify the grade of bouldering problems: i) videos of climber climbing the problem, ii) skeleton sequences extracted from such videos, iii) detailed human-readable topographical images with highlighted holds (topos), and iv) compressed matrix-like representations of topos. Various deep neural network architectures are used to process these modalities. X3D deep convolutional network is used for videos, recurrent bi-directional LSTM network for skeleton sequences, and ResNet for topo-image classification. The contributions of this work are insights into whether a subjective task of difficulty assignment has any potential to be solved by a machine and a comparison of what modalities are more suitable for this task. Last but not least, as an effort to evaluate the grade assignment problem, five datasets have been manually collected and made publicly available to the academic community. The work delivers satisfying results in terms of accuracy (86.18%) using the topo-images and skeleton sequences (72.22%). However, the results of the skeleton approach were not stable during the evaluation, which is underpinned by the difficulty of the task that is often non-obvious even for humans and also by the lack of training data. Nevertheless, the skeletons generally performed better than random guessing, which was not the case with the video-based approach, which turned out to be the worst performing.

Keywords

MoonBoard, bouldering, bouldering grading system, deep learning, human pose estimation, image classification, labelled dataset, skeleton-based action recognition, video classification

Contents

List of Tables

List of Figures

Introduction

Bouldering is a rock climbing discipline that is gaining large popularity, being for the first time one of the disciplines featured in the Olympic games in Tokyo 2020. In the discipline, the climbers climb on short climbing routes called boulders or problems. There is no need for protection in bouldering except for the mats placed under the climbers. The bouldering started outdoors on the rocks, but after some time, specialized indoor gyms started to appear as early as the 1970s.

The boulders vary in difficulty. The difficulty is expressed by the climbing grade. The grade is a subjective measure with hard-to-define rules based on the experience of those who assign it. The grading is very complex because the climber who assigns the boulder a grade must consider many factors, such as the friction of the rock, the overhang, the shape of the holds, and their position.

Outdoors, the grade is assigned by the climber who first ascents the boulder, and indoors, it is assigned by the route setter who sets it from the artificial holds. Also, the grade may change based on the opinions of other climbers who have tried the boulder.

The existence of grades and their importance in route setting creates an opportunity for an automated grading tool to help assess the grade by an independent entity. Despite the complexity of the problem, which relies on human experience and is heavily subjective, similar tasks have been addressed with machine learning, such as speech recognition or sentiment classification.

However, machine learning algorithms need training data to learn how to solve the problem; unfortunately, there is a lack of publicly available datasets in the area of climbing grade classification. Nevertheless, there is a potential source of raw data – MoonBoard.

The MoonBoard is a standardized bouldering wall where the holds are placed on an even grid, each boulder being a combination of the standardized holds. The boulders are created by its community and shared via a mobile app. Moreover, the MoonBoard encourages its users to film and share their ascents. Due to its popularity among

climbers, the MoonBoard is an attractive option as the data source[1] for both video content and topo images.

Therefore the secondary contribution of this work is collecting, cleaning, and preprocessing data of various modalities, including video, movement, and image data. The main contribution is applying existing, well-established deep-learning models (ResNet, 3DX, and LSTM) and comparing their accuracy on the bouldering grade classification task performed on the collected datasets.

1. Still, even with the existence of MoonBoard (and other standardized walls), no publicly available datasets exist.

1 Related Work

1.1 Computer Vision

Computer vision (CV) is an interdisciplinary field of artificial intelligence with many objectives that can be summarized as an endeavour to make computers learn and recognize highly descriptive patterns from multimedia data. The field is far from being just theoretical; the applications of the field are vast (1). An example could be a self-driving car technology that uses CV to parse the sensors' visual input to recognize road signs and potential collisions with other vehicles.

The prevalent technique in the field is deep learning which started to dominate in 2012 when ALEXNET (2) outperformed all the other methods in ImageNet challenge (3).

The most popular neural network architectures are variants of convolutional neural networks (4). However, in a recent development, the transformers (5) emerged from the field of natural language processing, and in the CV, there is given a lot of attention to them.

The most relevant subfields of computer vision to this thesis are image classification, video classification, and human pose estimation. Image classification is arguably the most known subfield of CV, whose aim is to assign a label to an image input.

The video classification is a similar task to the image classification with the difference that the video data points have, in addition to the height and width axis, the temporal axis. The popularity of the video classification is smaller than that of the image counterpart. One likely reason is that video classification is more complicated in every aspect; the datasets are harder to label, the models are more complex, and the model training takes considerably more resources. Moreover, video classification requires input videos to be cut into samples for a classifier to classify them, making it an impractical choice for most real-life scenarios, such as classifying data from security cameras.

Human pose estimation (HPE) is a popular subfield of CV with many applications. The objective of HPE is to map human bodies in visual input to a simplified skeleton model of the human pose. The human body models vary across applications and research. Skeleton-

based human motion processing is, for example, used in sports analysis (6), game industry (7), and robotics (8).

1.2 Skeleton-Based Action Recognition

Skeleton-based action recognition is a sub-field of action recognition. Action recognition is generally about assigning a class to data which captures an agent doing a specific action. For example, assign class "rope jumping" to a video of an athlete doing this particular task. Another example could be to assign the label "crouching" to the data recorded by the specialized HW like Kinect, which captures a human crouching. The skeleton-based action recognition classifies the data represented as the skeletons. For the video-based action recognition, the skeletons of the agents are extracted using the HPE techniques and then classified. There are several ways to classify the skeletons data. Below, the most relevant approaches are discussed.

1.2.1 Time Series

For this thesis, the most related approach is to see the skeletons as a multivariate time series classification problem. Time series classification is the problem of classifying the data sequence indexed by time. The only difference in multivariate time series classification is that there are more variables than one. The typical approach to solving the multivariate time series classification is Dynamic Time Warping(9). However, there are many others; their comparison was studied in (10).

1.2.2 C3D

As the authors of (11) demonstrated, it is possible to encode the skeleton as an image and use the consequent images as an input to the 3D convolutional neural network (C3D), the CNN, which has instead of 3 dimensions (width, height, colour) 4 dimensions (width, height, colour, time).

1.2.3 GCN

Graph convolutional neural network is a well-known graph neural network architecture that takes the idea of convolutions from the standard CNN architecture. In general, graph neural networks are good when applied to the data represented as a graph, thus making them a logical choice for classifying the skeletons. A big stream of works in the literature utilizes the concept of GCNs, usually proposing their own variants of the architecture. An example of such modern architecture is ST-GCN(12).

2 Difficulty Grading and MoonBoard

2.1 Difficulty Grading

The grade is a discrete value describing the boulder difficulty. There is no rigorous way boulders are graded; rather, the boulders are graded subjectively. Even for experienced route setters and climbers, grading is challenging because the grade tries to represent all the factors of a boulder in a single number. Such factors are; overhang, size of holds, the position of holds, friction of the rock, etc.

There are two major grading systems for bouldering(13): V-scale and Fontainebleau grading system. Both scales grade harder problems with increasing numbers. Even though they have a few differences, they are translatable and, in most cases, interchangeable. The Figure 2.1 depicts the translation between the two systems. This work uses the Fontainebleau grading system because it is the native grading system of the MoonBoard.

Fontainebleau grading system originated in the 1960s in the area with the same name. It is an open-ended system, which means the upper limit can change with the new climbs. The Histogram 2.2 shows the frequency of concrete grades on outside boulders in Europe in 2017.

2.2 MoonBoard

MoonBoard is a standardized bouldering wall 2.5 meters wide and 4 meters high. The holds on the MoonBoard are placed in an 11x18 square grid. Under the main wall is a kickboard, which is a part used exclusively for the feet placement. The boulders are created by its community and shared via mobile application. The illustrations of the MoonBoard and the app are depicted in the Figure 2.3.

Moreover, there are three parameters that affect the MoonBoard's overhang, the used hold set, and the rules of feet placement.

1. **MoonBoard configuration** is an overhang of a MoonBoard. Even though the MoonBoard can be built with any overhang, the application supports just two modes: 25° and 40°. Based on the

V Scale	Font Scale
V0	4
V1	5
V2	5+
V3	6A/6A+
V4	6B/6B+
V5	6C/6C+
V6	7A
V7	7A+
V8	7B/7B+
V9	7B+/7C
V10	7C+
V11	8A
V12	8A+
V13	8B
V14	8B+
V15	8C
V16	8C+
V17	9A

THE WANDERING CLIMBER

Figure 2.1: The translation table between the Fontainebleau and V-scale grading systems, taken from (14).

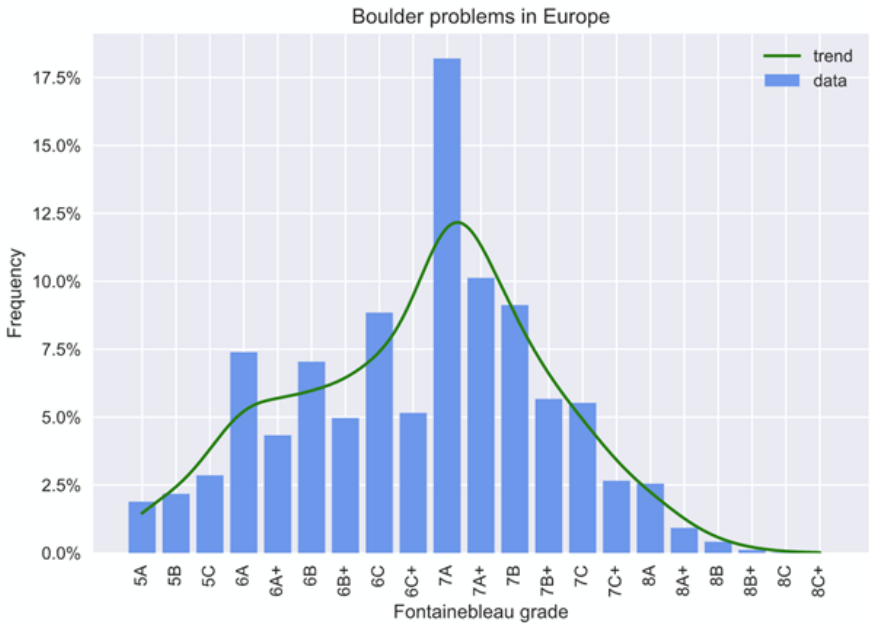

Figure 2.2: The bouldering grade distribution on the European rocks, taken from (15).

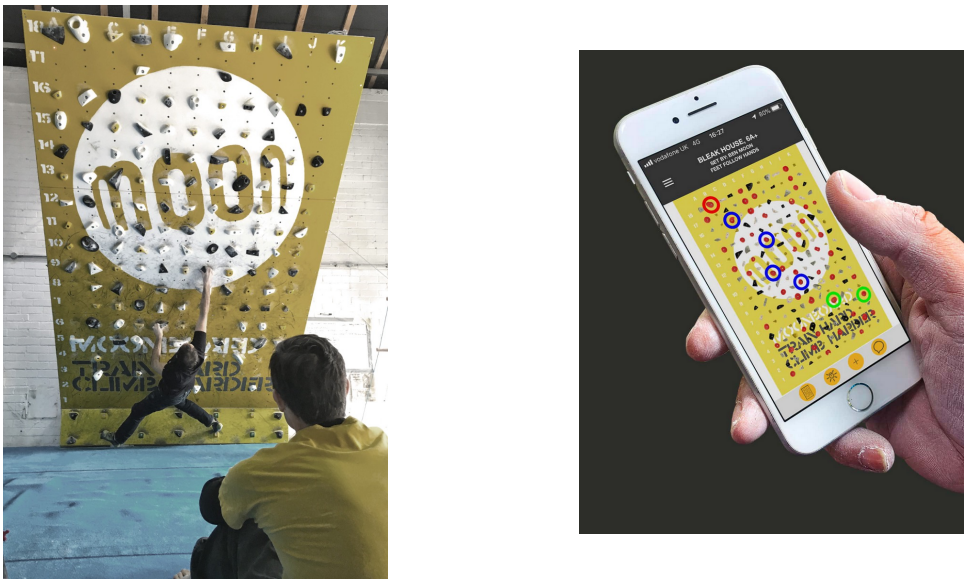

Figure 2.3: MoonBoard and MoonBoard app, taken from [16] [17].

number of boulders in a category, the second is far more popular within the community.

2. **MoonBoard hold setup** is the combination of standardised holds and their position on a grid. As of April 2022, there are three hold setups for MoonBoard. Figures 2.4 show the difference between all three distinct hold setups.

3. **Method** furthermore specifies foot placement rules. The most used variant is "Feet follow hands", denoting that climber can place their feet on the holds which she/he uses for hands and the kickboard. There are a few other methods. For example, the "Footless + kickboard" option forbids placing feet on any hold except the kickboard.

Each MoonBoard hold setup has its MoonBoard benchmarks. These boulders were graded carefully by a wider group of people. The benchmarks are used as a guideline for users to decide how to grade new boulders and thus prevent difficulty variation within the grade. Within this thesis, we use them as one of the sources of data.

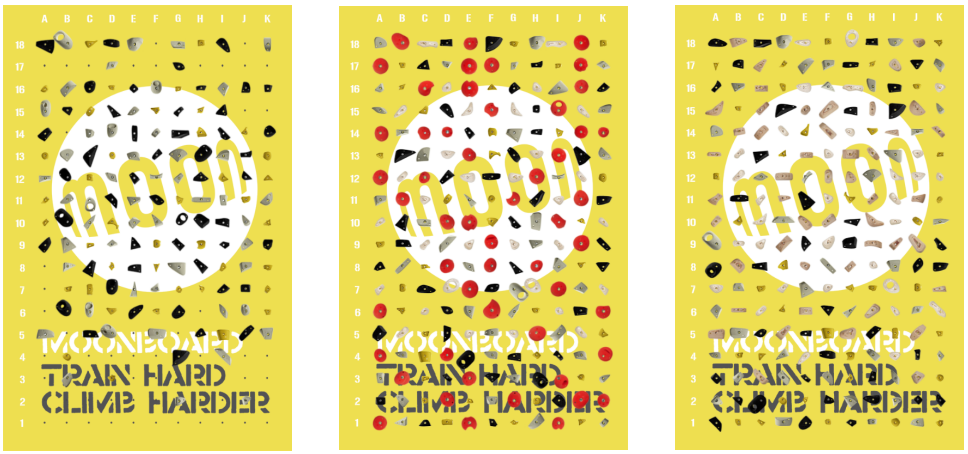

Figure 2.4: Comparison of different MoonBoard layouts, taken from [17].

3 Data Modalities

It is possible to approach the problem of classifying the difficulty of boulders from different data modalities, the relevant for this work are videos, skeleton sequences, topos, and compressed topos. Each modality has its advantages and can be used in different problem settings. The sections of this chapter describe the used modalities and discuss their applicability in various climbing grade classification settings.

3.1 Video

The video of the climber climbing a boulder encoded in .mp4 format with resolution 1080x1080[1] and 48 fps. The Figure 3.1 shows one frame from the dataset.

The videos have the advantage of being the most general modality discussed in this thesis. The videos preserve all the information captured on the camera, and they do not need any intricate preprocessing.

1. The resolution of the videos in the dataset, to train the model much smaller resolution is used.

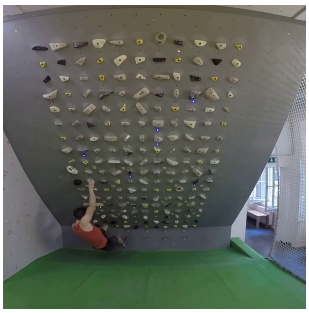

Figure 3.1: Sample frame from the video dataset.

However, these advantages come at a cost; the data has an enormous size compared to the other modalities. The model has to have more trainable parameters and be trained on a larger dataset; this takes more resources and time.

3.2 Skeletons

The skeletons modality is an ordered sequence of simplified poses with a fixed frame rate. Each pose of the human-body model used in this work is formally an undirected graph consisting of 16 nodes (joints) and 15 edges (bones). 2D XY-coordinates denote each pose within the captured scene space without the knowledge of depth information. Figure 3.2 depicts the example of a skeleton.

The advantage of the skeletons over the video is that the skeletons are very light-weight but still carry very particular information about the movement. In addition, the reduction in the data size results in less demanding computations. However, some useful information is lost, for example, the shapes and sizes of the climbing holds, the overhang of the wall, the position of fingers etc. Moreover, the extracting tool is needed to extract the skeletons.

3.3 Topo

Topo is an image representation of a MoonBoard boulder. Figure 3.3 shows an example.

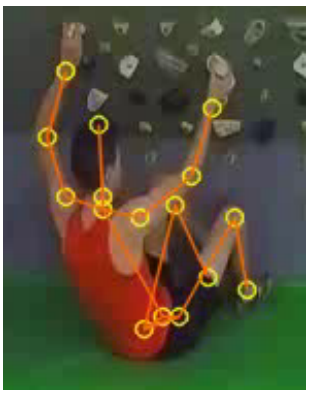

Figure 3.2: Visualised skeleton.

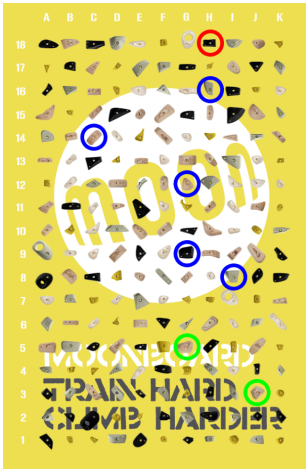

Figure 3.3: Sample from the topo dataset.

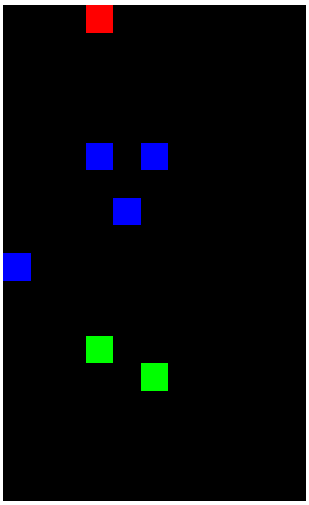

Figure 3.4: Sample from the compressed topo dataset.

All the encircled holds are the holds the climber can use for both hands and feet in his climb. The goal is to get from the starting holds (green), using intermediate holds (blue) to the top hold (red).

The topo is a simple yet highly specialized modality compared to the already discussed. The modality assumes a fixed layout which is not the case with the videos and skeletons. This premise limits the transferability of the model to other climbing grade classification problem settings.

3.4 Compressed Topo

A compressed topo is a small image with proportions of 11x18 pixels. It is a small representation of a MoonBoard boulder. Every pixel represents one hold on the MoonBoard. The colour of pixels has the same semantics as in the classical topos. Figure 3.4 shows an example.

The compressed topo modality is even more complex and specialized than topos. Compared to the topos, compressed topos assume not only a fixed layout but non-changing holds that are placed on a square grid. This limits the transferability even more.

4 Overview of Selected Deep-Learning Approaches

This chapter first discusses the theoretical deep-learning models used in the experiments (ResNet, LSTM, 3DX), then the tool used in the process of the extraction of skeletons from the videos (HRNet) and lastly, briefly summarizes the method of finetuning pretrained models.

We chose the selected models because they are well-established in the deep-learning community and are often used as benchmarks.

4.1 ResNet

Residual neural network (ResNet) (18) is a type of CNN. The ResNet solved the vanishing gradient problem in deep convolutional neural networks by the introduction of skip connections. The skip connection in the feed-forward neural net is when there is a connection which skips a few layers. The vanishing gradient problem is when the gradient used to update the model's weights shrinks too much. Hence, the update makes no difference. This problem occurs when the backpropagation algorithm has too many steps, i.e. when the neural network is too deep or with the recurrent neural network. Solving the vanishing gradient in CNNs made it possible to train a network as deep as 152 layers; to add perspective, the famous AlexNet had only 8 layers.

The ResNet won the ImageNet 2015 challenge (3) with the top-5 error of 3.57% (18), overperforming humans with the estimated top-5 error of 5.1% (19).

4.2 LSTM

The Long short-term memory (LSTM) (20) is a recurrent neural network architecture (RNN). The recurrent architectures differ from feed-forward architectures by the recurrent connections. The feed-forward neural network takes one data point (e.g. an image) as input. On the contrary, RNNs take the whole sequence (e.g. a video) and go over the entire sequence at one point (e.g. a frame) per pass. During the computation, the pass of a single point creates a hidden state that memorizes the temporal context and is added to the input of the next

pass. The LSTMs are effective with the sequence data, for example, time series forecasting, seq2seq translation, and speech recognition.

4.3 X3D

The 3XD (21) is a family of convolutional neural networks specialized in the video domain that expands the classical 2D convolutional neural network architecture, which specializes in the image domain. Unlike the classical approach, i.e. directly extending the image CNN to spacetime, so it takes the whole video at once as an input, the 3XD takes a stepwise approach. The 3XD extends the classical architecture gradually along the temporal duration, frame rate, spatial resolution, width, bottleneck width, and depth axis to achieve the best accuracy-to-complexity ratio.

4.4 HRNet

Hight-Resolution net (HRNet) (22) is CNN which can be used for multiple tasks such as pose estimation, image classification, semantic segmentation, and object detection.

The HRNet trained on the MPII dataset (23) is used for the skeleton extraction since HRNet proved effective in the (24) (25), (6), the works which all dealt with the speed climbing data, the data that are more or less similar to the data in this work.

The preliminary results of the videos-to-skeletons conversion were not satisfactory. As the author of (26) pointed out, the problem is that the major human pose estimation datasets do not include climbing poses. In the speed climbing applications, as was (24), (25), and (6), the problem was less severe because, in the speed climbing, the climber is climbing on a wall with little overhang and has visible limbs most of the time; thus human pose estimation algorithm has better information and performs better. The situation is different in this work because the input videos are recorded at an angle, and the whole body is not always visible.

However, the same problem is expected to occur with the alternative pose estimation models because of the same underlying issue; there is no climbing data in the pose estimation datasets.

4.5 Finetuning

Finetuning is a technique of training a neural network which was already pretrained on a different, usually larger dataset. The neural network procedurally transforms input; firstly, it selects the low-level features, for example, in the CV, the edges, arcs etc., then combines these low-level features to find more complex features, for example, the nose of the dog. The theory is that the low-level feature filters are transferable through machine learning tasks. Hence, if the already trained low-level features are used in the new training task, the training is simpler because most of the network is already trained and thus needs less training data. When finetuning the model, the last layers (or their weights) of the trained model are discarded and replaced with fresh to-be-trained layers. The finetuning is especially good in scenarios with insufficient training data samples for regular learning. Within this work, this method is used to train all the selected computer-vision models.

5 Datasets

This chapter is dedicated to the datasets which were created for this work.

5.1 Videos

The video dataset serves two purposes: an input for the video classification experiment and input to the HRNet to create a skeleton dataset. The source for the videos is a YouTube channel [1], which provides the videos in better quality than other alternative sources. The videos are cut into 224 individual video samples. Each sample is a video of a climber climbing one boulder from start to top. The samples are divided into 10 classes 6A+, 6B, 6B+, 6C, 6C+, 7A, 7A+, 7B, 7B+, 7C. Lastly, the videos are cropped to keep the whole climbing wall while omitting unnecessary pixels.

5.2 Skeletons

The skeleton sequences were extracted from the individual video samples using HRNet. Due to the complexity of the pipeline (object detection→tracking→pose estimation), the extracted raw data contained errors caused by the individual steps or their combination.

The first problem was that for some videos, there were more detected people. The corrupted files containing the skeleton sequences fell into three categories.

1. Files in which climber detection was split into more skeleton sequences. This problem occurred because the algorithm somewhere missed many frames in a row, and the subsequent detection was far enough to be classified as a different person. The solution was to merge all the sequences into one.

2. Files generated from the videos with more people in. The solution was to find which sequence corresponds to a climber and delete the rest detections.

1. https://www.youtube.com/user/freshydumbeldore1/videos

3. A compound of the previous two. There were eight samples in this category. The two of them were fixed because it was possible to identify which sequence corresponds to whom. The rest (6 samples) was deleted.

The second problem was that the skeleton detection was missing for some of the frames within the sequence. If the problem was not compensated, the model used for classification would see the climber jump from one position to a faraway position in just one frame. The inputting method was to fill the empty frames with zero coordinates. The idea is to denote to the model that there are no data and leave the solution for the algorithm.

5.3 Topos

The previous datasets have the same underlying information; both are created from one set of boulders: the subset of benchmarks from the 2019 MoonBoard hold setup with 40° overhang and the method "Feet follow hands". This dataset contains the same boulders as the previous two to keep this consistency. The topos were downloaded from the official MoonBoard website (17).

5.4 Compressed Topos 1

The compressed topos were generated from the full-size topos. The process to achieve it was as follows. (1) Calculate the pixel position of the holds in a full-size topo image. (2) For each hold, decide if some colour encircles it. If yes, then colour the corresponding pixel in the compressed topo image with that colour.

5.5 Compressed Topos 2

The sharing policy of the MoonBoard company changed recently; most boulders are accessible only through the official mobile application. On GitHub, there is a project (27) which uses not-to-date scrap of the database from before the policy has changed. The database is used with the consent of the author of the project.

Table 5.1: Distribution analysis

Attribute	Value	Count
Hold setup	MoonBoard 2016	27340
Hold setup	MoonBoard Masters 2017	30093
Hold setup	MoonBoard Masters 2019	1132
Method	Feet follow hands	55495
Method	Feet follow hands + screw ons	2428
Method	Footless + kickboard	376
Method	Screw ons only	266
MoonBoard configuration	40	27460
MoonBoard configuration	25	3756
MoonBoard configuration	NIL	27340

The data are encoded in a .json file, including 58565 boulders. These are further divided by the **hold setup**, **configuration**, and **method**. The table 5.1 shows a count of boulders with a given value of an attribute.

Two identical compressed topo images can correspond to the two different boulders, e.g. by having different overhangs. To overcome this issue, only boulders from the set of MoonBoard Masters 2017 with the method "Feet follow hands" and with a 40° overhang are used. This combination of parameters is used for two reasons (a) it gives the most boulders (b) the data are similar to the existing compressed topo dataset. Both have the same overhang and method. After the filtration, there were 23787 boulders which were used to generate the compressed topo images.

5.6 Summary

The first four datasets: the MoonBoard benchmarks video dataset, the MoonBoard benchmarks skeleton dataset, the MoonBoard benchmarks topo dataset, and the MoonBoard benchmarks compressed topo dataset, are all datasets of the same data but a different modality. The four datasets each contains 218-224 samples labelled with 10 labels, namely 6A+, 6B, 6B+, 6C, 6C+, 7A, 7A+, 7B, 7B+, and 7C. The histogram 5.1 shows the distribution of the classes.

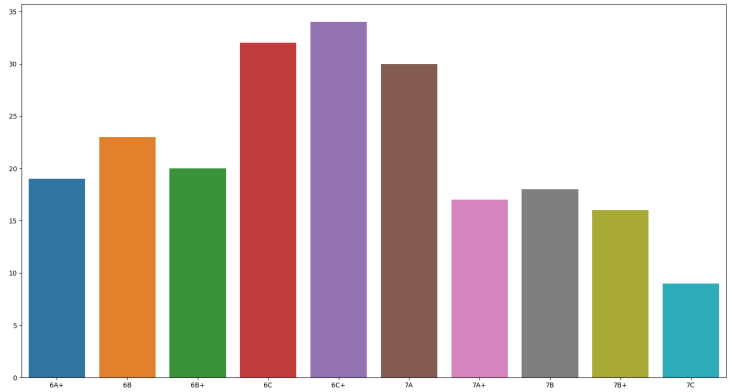

Figure 5.1: Distribution of classes in MoonBoard benchmarks datasets.

The fifth dataset: the MoonBoard database compressed topo dataset, is a dataset generated from the not-to-date MoonBoard database. It contains 23787 samples with the classes ranging in [6A+; 8B+] The histogram 5.2 shows the distribution of the classes.

All the datasets are available to the public on `www.kaggle.com/datasets/eddous/moonboard` under the public domain licence (CC0).

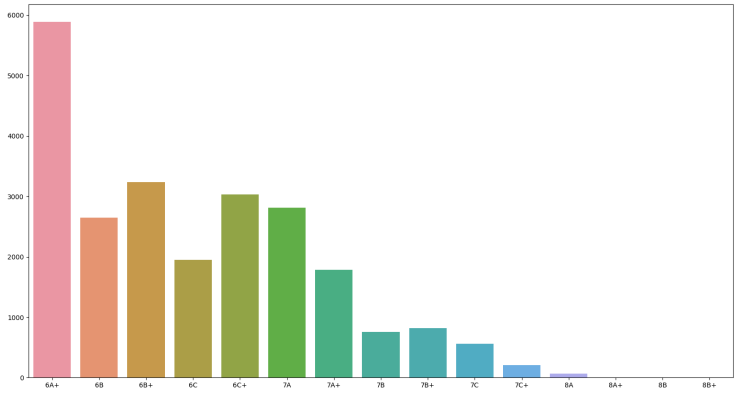

Figure 5.2: Distribution of classes in MoonBoard database dataset.

6 Evaluation Protocol

This chapter discusses why and how the experiments are designed. It also describes the parameters of the algorithms used for evaluation.

6.1 Splits

The experiments use 5-fold cross-validation splits for the four MoonBoard benchmarks datasets and an 80/20 split for the MoonBoard database dataset. For the MoonBoard benchmarks datasets, the splits are identical, i.e. if the boulder is in the train set of the skeleton dataset, it is in the train set of every other MoonBoard benchmarks dataset and visa versa.

6.2 Dealing with Class Imbalance

As Figure 5.1 shows the classes in the MoonBoard benchmarks datasets are imbalanced. Furthermore, the number of samples in each category is small. To compensate for this issue, the grading system is simplified to only two classes: "easy" and "hard". The problem does not

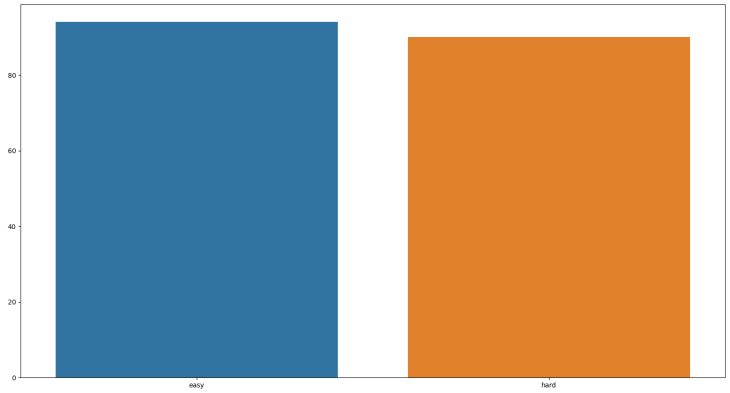

Figure 6.1: Distribution of classes in MoonBoard benchmarks datasets after the merge of classes.

change with the simplified grade because the grading is an artificial system where each grade is an interval on the underlying continuous difficulty and the simplified grade is only a different view of the underlying value. The original grades [6A+; 6C] are merged into the "easy" grade, and classes [7A;7C] into the "hard" grade, so the 6C+ is left completely.[1]

The Figure 6.1 shows the distribution of classes in benchmarks datasets after the class merge.

The same class transformations were applied to the MoonBoard database dataset to have the same starting conditions for the experiments. The histogram 6.2 shows the distribution of classes in the MoonBoard database dataset after the merge of classes. It also reveals

1. The idea of leaving 6C+ out is a reduction of the noise. This noise arises from the fact that the grading is subjective - people often do not agree if a particular boulder is graded correctly. The most usual grading error is by one grade, e.g. 7B is graded 7A+ or 7B+. Hence, in addition to the simplified grade, the gap removes the majority of misclassified-by-one boulders. The only possible misclassified boulders in the new settings are true 6C+, classified as 6C (easy) or 7A (hard). Another advantage is the simplification of the problem; the model has to learn a less subtle border between classes. The disadvantage of this approach is that some data suitable for training are deleted.

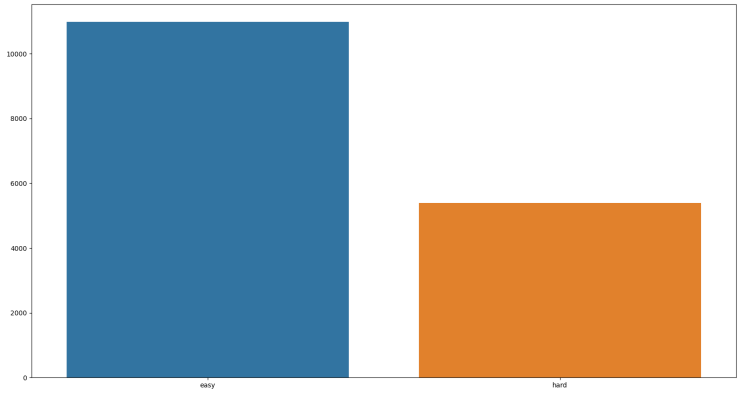

Figure 6.2: Distribution of classes in MoonBoard database dataset after the merge of classes.

that the transformation kept the problem of imbalanced classes. To compensate for this issue, the "hard" class in a train set is oversampled to be even with the "easy" class because, as it was shown in (28), it is an effective technique to solve the problem of unbalanced classes.

6.3 Videos

The videos experiment uses the 3XD-xs implementation from the Lighting Flash framework. This implemented model is pretrained on Kinetics 400 dataset (29). In the experiment, finetuning is used as the technique of training.

Since the framework provides out-of-the-box models, the experiment uses all the default hyperparameters, which are not limited by the technical capabilities of a machine which was used for training. The limiting factor for some is the size of the GPU memory. The video resolution and batch size were reduced to compensate for the limited GPU memory.

The Table 6.1 shows the hyperparameters used in the video experiment.

Table 6.1: The hyperparameters of the video experiment.

video size	126x126
batch size	1
clip sampler	uniform
clip duration	2
backbone	x3d_xs
pretrained	true
loss function	cross entropy
optimizer	Adam
learning rate	1e-3
epochs	100

6.4 Images

The experiments (the MoonBoard benchmarks topo dataset, the MoonBoard benchmarks compressed topo dataset, and the MoonBoard database compressed topo dataset) use ResNet18 implementation from Lightning Flash framework, pretrained on ImageNet-1K dataset (3), the dataset with 1000 classes and 1,281,167 training images.

Similarly to the videos experiment, the default hyperparameters are used. Unlike it, the images experiments are more lighweight to computing resources, so there was no need for tweaking the hyperparameters.

Moreover, the experiments run with two different fine-tuning strategies: freeze and no-freeze. Freeze is a finetuning strategy that freezes the not discarded layers, and the update of weights is allowed just in the replaced last layers. No-freeze, on the other hand, allows the weights to update even in the not discarded layers of the neural network. The no-freeze strategy needs more data because the model is larger; thus, it needs more data to not be overfitted. The only dataset with such quantity is a MoonBoard database compressed topo dataset. However, the strategy is evaluated against the others to help conclude the experiments as a whole.

The Table 6.2 shows the hyperparameters used in the images experiments. The set notation next to the "strategy" indicates that both were tried.

Table 6.2: The hyperparameters of the images experiments.

image size	196x196
batch size	4
backbone	resnet18
pretrained	true
loss function	cross entropy
optimizer	Adam
learning rate	1e-3
epochs	100
strategy	{freeze, no-freeze}

6.5 Skeletons

In contrast to computer vision, skeleton-based action recognition is not so well studied; thus, there are less well-known procedures in data preprocessing, training, and the choice of hyperparameters. Another reason is that the LSTM is more sensitive to nuances in hyperparameters than CNNs. Hence the skeletons experiments are more thorough. Within the experiments are used variants of the hyperparameters and different data normalization methods.

Unlike images and videos experiments with skeletons, the experiments are run on the two splits because the preliminary results indicated that a lot of data would be needed to draw conclusions. Also, skeletons experiments use slightly different metrics. Top accuracy is the best validation accuracy throughout all epochs. Top10 accuracy is the average of the 10 best validation accuracies throughout the training. Average accuracy is the average of all validation accuracies.

The Table 6.3 shows the hyperparameters used in the skeletons experiments. The set notation indicates that the hyperparameter did change within the evaluation.

6.5.1 Skeleton Normalization Methods

This section discusses the aforementioned skeleton normalization methods.

Table 6.3: The hyperparameters of the skeletons experiments.

learning rate	{5e-7, 5e-6, ..., 5e-3}
embedding	{0, 32, 64}
normalization	{none, mid top, left top, first root, all root}
epochs	100
hidden state size	1024

Left Top Corner

This normalization method shift coordinates so the hold in the left top corner of a MoonBoard has the image coordinates (0,0). With this normalization, all coordinates in the data have non-negative values.

Mid Top

Mid top normalization has a similar idea as the left top corner normalization. The method shifts the skeleton coordinates, so the hold on the MoonBoard grid (6,17) has the image coordinates (0,0). This normalization makes the x-coordinate evenly distributed around zero.

First Root

The first root normalization removes the information about the absolute starting position. Every skeleton sequence after the normalization starts with its root node (pelvis) at the coordinates (0,0), and other nodes are shifted to preserve the same relative distance. The rest of the skeletons in a given sequence are processed to have the same relative distance from the first position as in the original data.

The following pseudocode is a complete description of the normalization.

```
for skeleton_sequence in data {
   first_pose := skeleton_sequence[0]
   root_coords := first_pose[root_index]
   for pose in skeleton_sequence {
      subtract_vector_from_all_nodes(pose, root_coords)
   }
}
```

All Root

All root normalization preserves only information about the relative position of the nodes within the single skeleton. Every skeleton in the sequence has a root node set to (0,0) and other nodes accordingly.

The following pseudocode is a complete description of the normalization.

```
for skeleton_sequence in data {
   for pose in skeleton_sequence {
       root_coords := pose[root_index]
       subtract_vector_from_all_nodes(pose, root_coords)
   }
}
```

7 Evaluation

7.1 Skeletons

The boxplots 7.1 show the distribution of a top accuracy in both splits. As the first boxplot shows, the median of top accuracy is 60%, and in some experiments, the metric exceeds 70%. However, the top accuracy on the validation set can be a cherry-picked value, and it is not safe to draw a conclusion from it alone.

Figure 7.2 shows the boxplots of the average accuracy in the experiments. The boxplot shows that the observed average accuracies are close to 50%, which is an expected accuracy of a random classifier. The bootstrap(30) is used as the statistical method for deciding whether the observed data could be generated by chance. The null hypothesis is that the observed data are generated by random guessing. For both splits, the bootstrap is used to generate a confidence interval with a significance level of 95% for the mean of the average accuracy metric. For the first split, the bootstrapped confidence interval is $(49.05, 50.98)$, and the observed mean value of average accuracy is 51.847. For the second split, the confidence interval is $(48.96, 50.99)$, and the observed mean value is 51.1. Both tests rejected the null hypothesis. Therefore, the LSTM (even on average) is better than random guessing.

Figure 7.3 shows the relation on top10 accuracy between experiments on a first and a second split. The correlation between the splits is 0.71, indicating that the training is replicable to some extent.

The boxplots 7.4, 7.5, 7.6 shows top10 accuracy in a relation to the optimized hyperparaters. They reveal that LSTM on average [1] is performing better when the learning rate is set to 5e-04. The performance is the best without the embedding. The normalization method does not seem to have a significant effect.

Figure 7.7 shows the best performing trained models based on the top10 metric. The red is trained on the first split, and the blue is trained on the second split. The x-axis is the number of epochs, and the y-axis

1. Boxplots captures only the relation between accuracy and given parameter. It is incorrect to deduce the best combination of hyperparameters from the boxplots alone because the accuracy can depend on the combination.

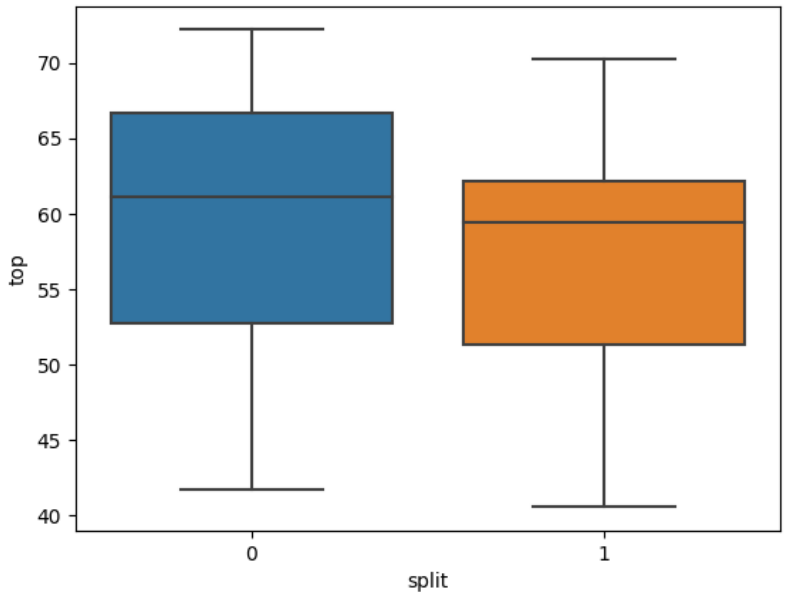

Figure 7.1: The top accuracy across the splits.

is the validation accuracy in a given epoch. The Figure indicates that the network does not converge.

The confusion matrices 7.8, 7.9, 7.10, 7.11, 7.12, 7.13 shows how the best performing models based on top accuracy classified validation set during their best epoch. The first three matrices are from the first split, and the rest are from the second. The x-axis represents the original boulder grade, and the y-axis represents the guessed class, where 0 is easy and 1 is hard. The confusion matrices show that the model does not perceive the climbing grade similarly to humans. The expected confusion matrix for such a model is a matrix where most of the errors are in class 6C+ mapped to 3 and 7A mapped to 5 because 6C+ is the hardest of the easy class, and 7A is the easiest of the hard class.

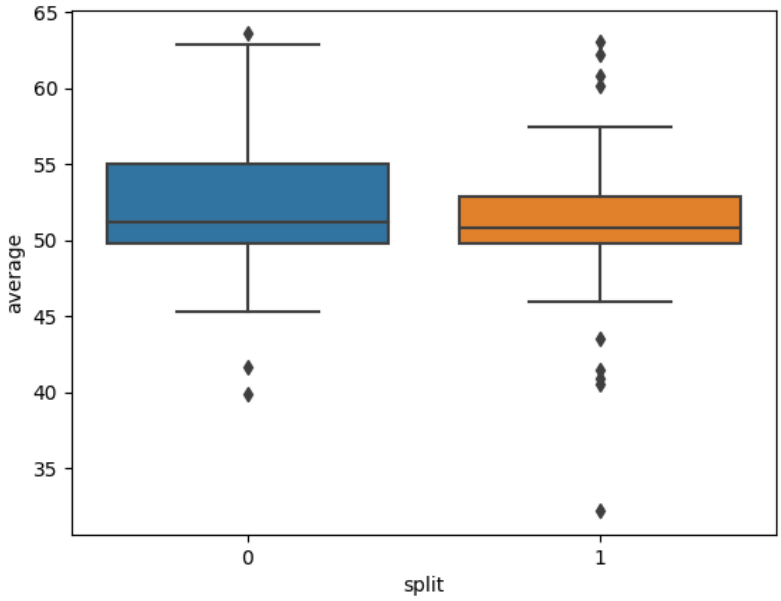

Figure 7.2: The boxplot of average accuracy across the splits.

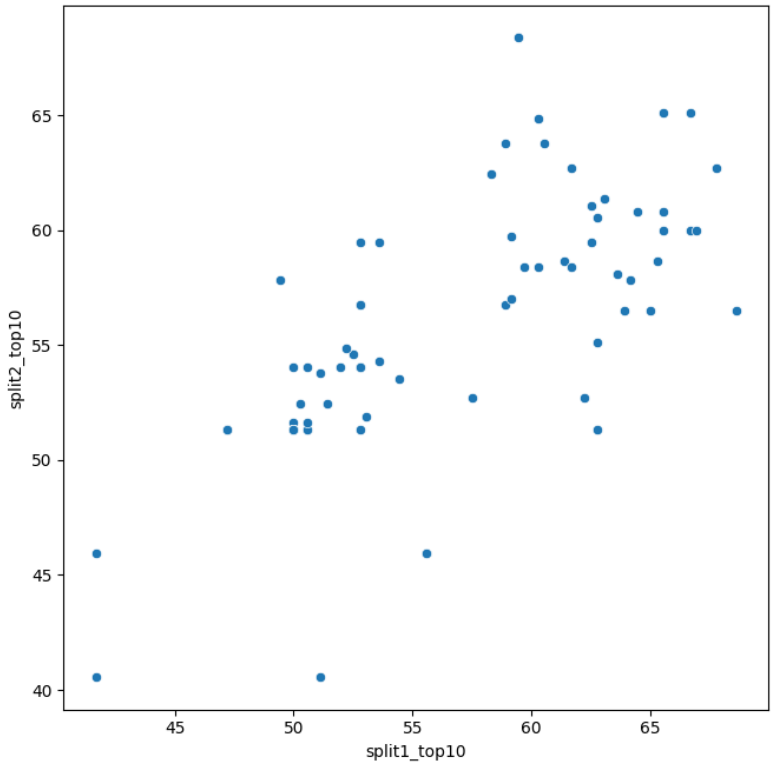

Figure 7.3: The scatterplot showing the relation between top10 accuracy of the first and the second split.

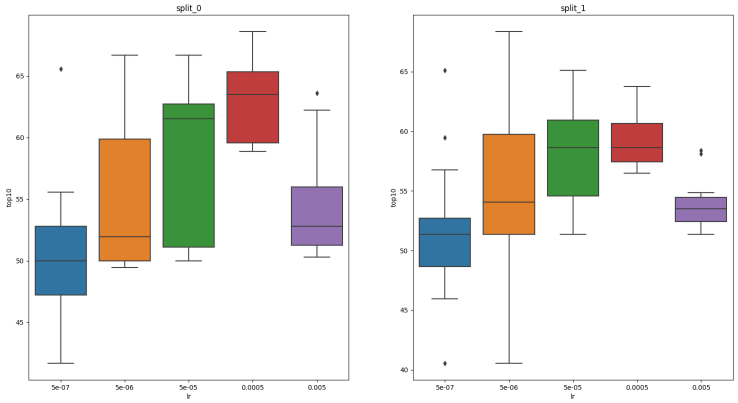

Figure 7.4: Effect of size of learning rate on top10 accuracy

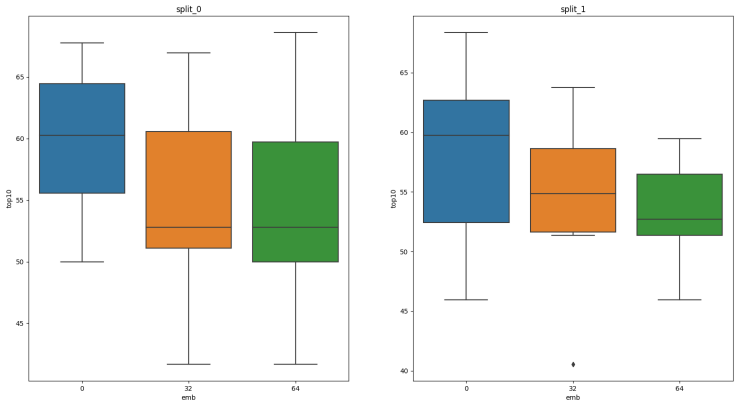

Figure 7.5: Effect of size of embedding on top10 accuracy

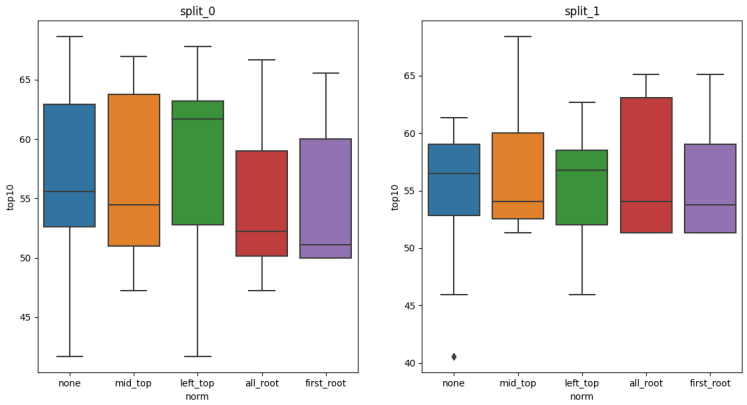

Figure 7.6: Effect of skeleton normalizations on top10 accuracy

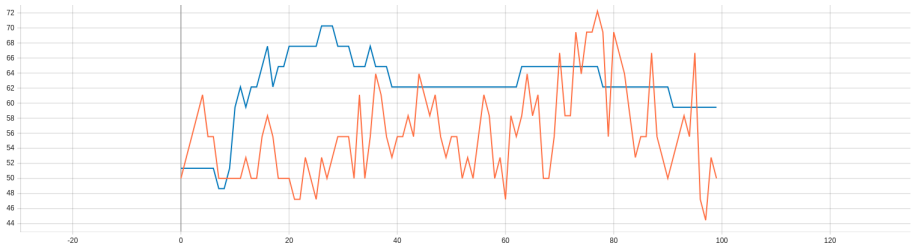

Figure 7.7: Best performing model on the first split and on the second split.

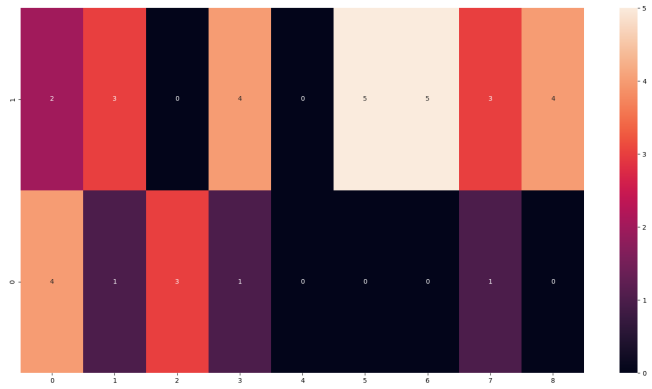

Figure 7.8: Confusion matrix of #1 model on 1 split.

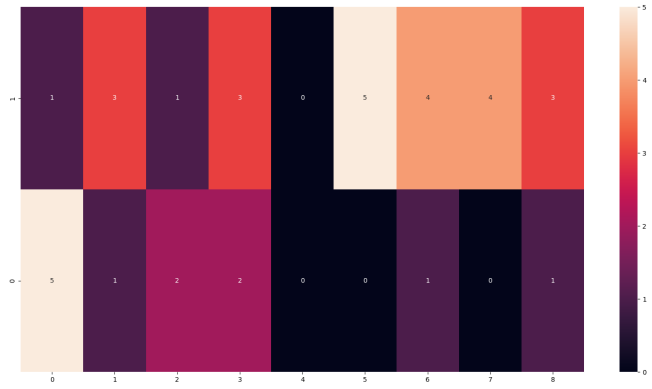

Figure 7.9: Confusion matrix of #2 model on 1 split.

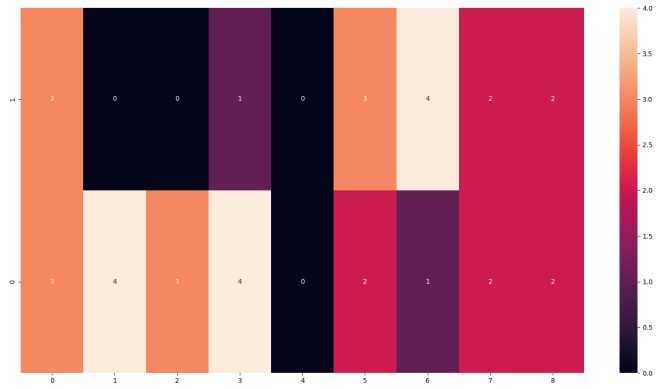

Figure 7.10: Confusion matrix of #3 model on 1 split.

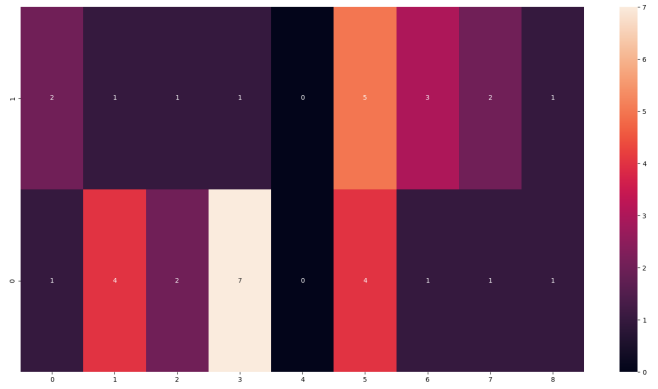

Figure 7.11: Confusion matrix of #1 model on 2 split.

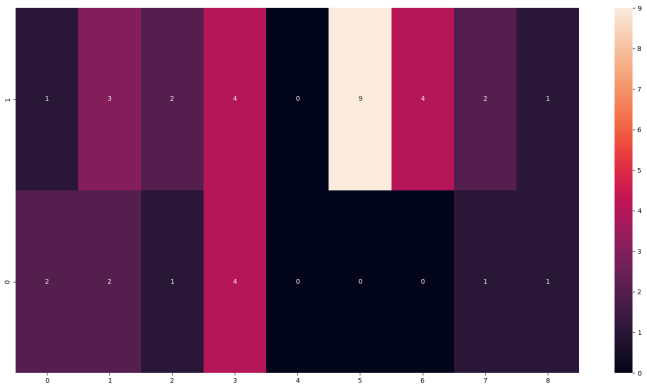

Figure 7.12: Confusion matrix of #2 model on 2 split.

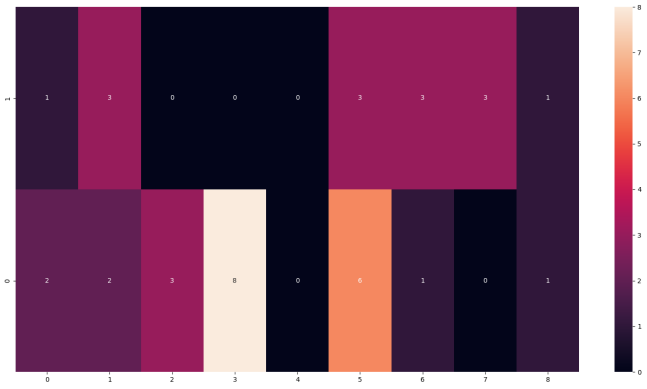

Figure 7.13: Confusion matrix of #3 model on 2 split.

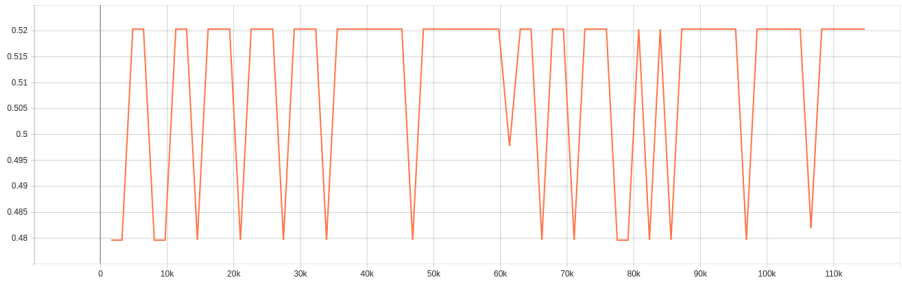

Figure 7.14: Accuracy of video classification through training steps.

7.2 Videos

Figure 7.14 shows the accuracy of the validation set throughout the training. The x-axis is the number of iterations, iteration being one update of network weights. The y-axis is the accuracy of the validation step.

The output is the same as random guessing. The inferior performance likely is due to the limited available resources, which resulted in rather limited hyperparameters (short clip sample size – 2s, small batch size – 1, and limited resolution – 126x126).

7.3 Images

The figures 7.15, 7.16, 7.17, 7.18 visualize metrics from the training process. The table 7.1 shows resulting accuracies.

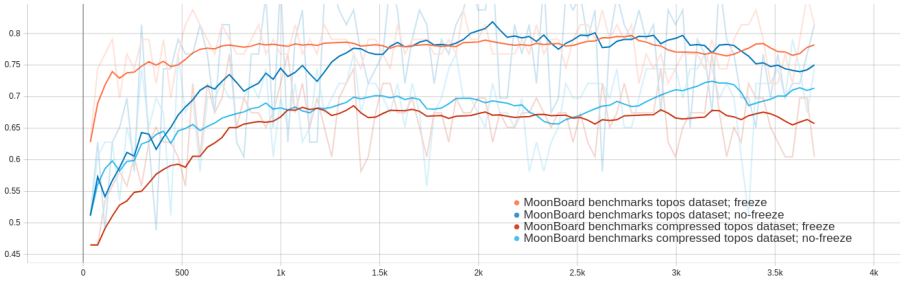

Figure 7.15: Validation accuracy through training steps on MoonBoard benchmarks topo dataset and MoonBoard benchmarks compressed topo datasets with two finetuning strategies: freeze and no-freeze.

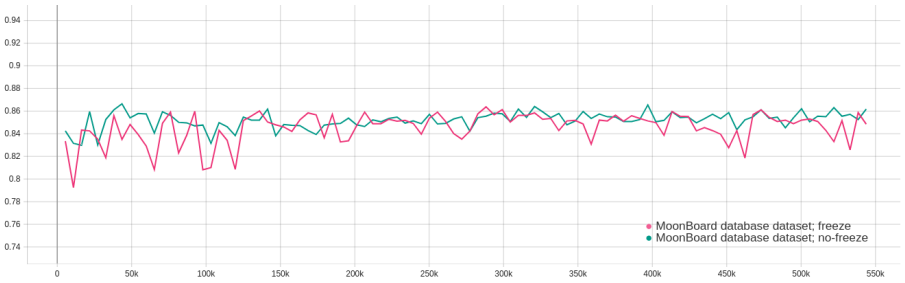

Figure 7.16: Validation accuracy through training steps on MoonBoard database dataset and with two finetuning strategies: freeze and no-freeze.

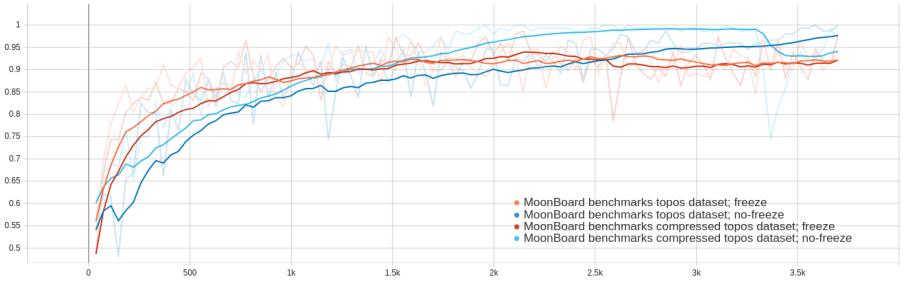

Figure 7.17: Train accuracy through training steps on MoonBoard benchmarks topo dataset and MoonBoard benchmarks compressed topo datasets with two finetuning strategies: freeze and no-freeze.

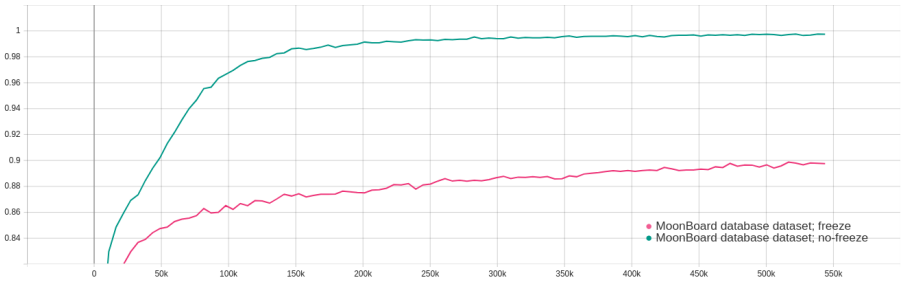

Figure 7.18: Train accuracy through training steps on MoonBoard database dataset and with two finetuning strategies: freeze and no-freeze.

Table 7.1: Evaluation of image classification experiments.

Dataset	Modality	Finetuning Strategy	Validation Accuracy	Train Accuracy
benchmarks	Topo	freeze	0.814	0.9392
benchmarks	Topo	no-freeze	0.814	1
benchmarks	Compressed	freeze	0.6047	0.9797
benchmarks	Compressed	no-freeze	0.7442	0.9662
database	Compressed	freeze	0.8484	0.8975
database	Compressed	no-freeze	0.8618	0.9975

All the Figures 7.15,7.16, 7.17, 7.18 show that the model did converge.

For topo modality, the finetuning strategy did not affect the validation accuracy. However, by validation accuracy metric, the no-freeze strategy performed better than freeze on the compressed topo modality. The no-freeze method likely performs better because when the model is finetuned, it uses already gained knowledge from previous training. In finetuning with the freeze method, this knowledge is more crucial than with no-freeze because, with the first, the network can not change the deeper layers and, by doing so, learn new patterns. Because the ResNet was pretrained on the ImageNet-1K dataset, which does not contain data similar to the compressed topo images, the network learning without frozen deeper layers had an advantage.

The Table indicates that the model finetuned with a no-freeze strategy is always overfitted. Overfitting suggests that the model was trained on an insufficient number of samples; this is surprising for MoonBoard database datasets because it consists of 20000+ samples. The freeze strategy also overfitted the MoonBoard benchmarks datasets. However, in the experiment on the MoonBoard database datasets, the training data were not overfitted. The reason why the freeze strategy did not overfit the train set is likely that the strategy has fewer parameters to optimize; thus, a smaller amount of data was required.

On the MoonBoard benchmarks datasets, the model performed better on topos than on compressed topos. The likely reason is similar to why the no-freeze strategy performed better than the freeze strategy on the compressed topo modality. The network was pretrained on unlike data to compressed topos. Thus it could not use previously gained knowledge.

The change between validation accuracy between MoonBoard benchmark compressed topo dataset and MoonBoard database dataset was 18% [2]. Again indicating that the MoonBoard benchmark datasets are too small.

2. Even though the boulders' source is different, comparing these results is possible because the only difference between the sets is the holds installed on the wall. Thus it is very similar problem.

8 Conclusion

This work focuses on the different deep learning approaches for classifying the MoonBoard bouldering problems.

The image classification (86.18%), the most straightforward and the least transferable to the general bouldering grade classification, was superior to the two more general approaches. Also, it was the only approach that produced satisfactory results regarding the evaluation metrics of the trained model.

The skeletons experiments (72.22%) were not unsuccessful as well though the evaluation metrics of the trained classifier were far from good. The experiments demonstrated that the LSTM network is capable of training on our small and arguably noisy dataset. The noise within the dataset, as discussed, was caused by the accumulated error of person detection, tracking, and pose estimation, where especially the available pre-trained models attributed issues with detecting people in complex climbing positions.

The video-based classification did not yield any significant results. This was likely due to the complexity of the video classification, followed by a lack of sufficient training dataset size and, lastly, by limited training parameters caused by available hardware. However, the video classification experiment showed that the human pose estimation approach is better.

To be able to compare the approaches, five novel datasets from three distinct raw sources were created for the purposes of this work; all are available at `www.kaggle.com/datasets/eddous/moonboard`.

Grading climbing problems by various classifiers turned out to be a challenging problem even in the simplified settings of the standardized climbing wall MoonBoard, where the approaches used did not need to consider many variables which would occur in the general settings, variables like different camera angles, wall profiles, climbing styles, the height of a climber, and many more.

Bibliography

1. SZELISKI, Richard. *Computer vision: algorithms and applications*. Springer Nature, 2022.
2. KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012, vol. 25.
3. RUSSAKOVSKY, Olga; DENG, Jia; SU, Hao; KRAUSE, Jonathan; SATHEESH, Sanjeev; MA, Sean; HUANG, Zhiheng; KARPATHY, Andrej; KHOSLA, Aditya; BERNSTEIN, Michael; BERG, Alexander C.; FEI-FEI, Li. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*. 2015, vol. 115, no. 3, pp. 211–252. Available from DOI: 10.1007/s11263-015-0816-y.
4. LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *nature*. 2015, vol. 521, no. 7553, pp. 436–444.
5. VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N; KAISER, Łukasz; POLOSUKHIN, Illia. Attention is all you need. *Advances in neural information processing systems*. 2017, vol. 30.
6. SALANIČOVÁ, Lucia. *Measuring and Visualizing Similarities and Discrepancies in Speed Climbing Performances* [*online*]. 2022. Available also from: https://is.muni.cz/th/yiruc/.
7. KOHLI, Pushmeet; SHOTTON, Jamie. Key developments in human pose estimation for kinect. In: *consumer depth cameras for computer vision*. Springer, 2013, pp. 63–70.
8. ZIMMERMANN, Christian; WELSCHEHOLD, Tim; DORNHEGE, Christian; BURGARD, Wolfram; BROX, Thomas. 3d human pose estimation in rgbd images for robotic task learning. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 1986–1992.
9. Dynamic Time Warping. In: *Information Retrieval for Music and Motion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 69–84. ISBN 978-3-540-74048-3. Available from DOI: 10.1007/978-3-540-74048-3_4.

10. ISMAIL FAWAZ, Hassan; FORESTIER, Germain; WEBER, Jonathan; IDOUMGHAR, Lhassane; MULLER, Pierre-Alain. Deep learning for time series classification: a review. *Data mining and knowledge discovery*. 2019, vol. 33, no. 4, pp. 917–963.

11. DUAN, Haodong; ZHAO, Yue; CHEN, Kai; LIN, Dahua; DAI, Bo. Revisiting skeleton-based action recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2969–2978.

12. YAN, Sijie; XIONG, Yuanjun; LIN, Dahua. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In: *AAAI*. 2018.

13. DRAPER, Nick. 14 Climbing grades. *The Science of Climbing and Mountaineering*. 2016, p. 227.

14. [https://www.thewanderingclimber.com/bouldering-vs-rock-climbing]. [N.d.]. Accessed:2022-11-04.

15. *Fontainebleau grades are unevenly hard* [https://etienne-pepin.medium.com/analysis-of-the-fontainebleau-grading-system-ce24fc27d5aa]. [N.d.]. Accessed:2022-9-05.

16. [https://eu.moonclimbing.com/News/post/important-information-2016-moonboard-set?___store=eu_uk&___from_store=default]. [N.d.]. Accessed: 2022-6-10.

17. [https://www.moonboard.com/]. [N.d.]. Accessed: 2022-6-10.

18. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

19. RUSSAKOVSKY, Olga; DENG, Jia; SU, Hao; KRAUSE, Jonathan; SATHEESH, Sanjeev; MA, Sean; HUANG, Zhiheng; KARPATHY, Andrej; KHOSLA, Aditya; BERNSTEIN, Michael, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*. 2015, vol. 115, no. 3, pp. 211–252.

20. HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-term Memory. *Neural computation*. 1997, vol. 9, pp. 1735–80. Available from DOI: 10.1162/neco.1997.9.8.1735.

21. FEICHTENHOFER, Christoph. *X3D: Expanding Architectures for Efficient Video Recognition*. arXiv, 2020. Available from DOI: 10.48550/ARXIV.2004.04730.

22. SUN, Ke; XIAO, Bin; LIU, Dong; WANG, Jingdong. Deep high-resolution representation learning for human pose estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5693–5703.

23. ANDRILUKA, Mykhaylo; PISHCHULIN, Leonid; GEHLER, Peter; SCHIELE, Bernt. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

24. POKORNÝ, Jan. *Mapping 2D Skeleton Sequences from Speed Climbing Videos onto a Virtual Reference Wall* [*online*]. 2021. Available also from: https://is.muni.cz/th/zp7vz/. Diplomová práce. Masarykova univerzita, Fakulta informatiky, Brno.

25. ŠKVARLOVÁ, Veronika. *Labeled Dataset of Speed Climbing Performances* [*online*]. 2021. Available also from: https://is.muni.cz/th/ll5rp/.

26. *Applying AI to climbing … Deep Learning meets the "odd human" dataset* [https://community.element14.com/members-area/personalblogs/b/blog/posts/applying-ai-to-climbing-deep-learning-meets-the-odd-human-dataset]. [N.d.]. Accessed:2022-6-15.

27. CAJANUS, Sakari. *moonboardsearch* [https://github.com/scajanus/moonboardsearch]. GitHub, [n.d.].

28. MASKO, David; HENSMAN, Paulina. *The impact of imbalanced training data for convolutional neural networks*. 2015.

29. KAY, Will; CARREIRA, Joao; SIMONYAN, Karen; ZHANG, Brian; HILLIER, Chloe; VIJAYANARASIMHAN, Sudheendra; VIOLA, Fabio; GREEN, Tim; BACK, Trevor; NATSEV, Paul, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*. 2017.

30. PHD, Josh Starmer. *Using Bootstrapping to Calculate p-values!!!* 2021. Available also from: https://www.youtube.com/watch?v=N4ZQQqyIf6k.