Determine wind direction and force by applying machine learning algorithms on windsock images of the webcam at the paragliding take off "Möntschelen"

Conceptual Design Report

31 October 2022

Abstract

Wind direction and force are two critical measures for paragliding. Before take-off the pilots must check five points, the fourth point is to check the wind conditions. At the official take-off and landing places there is often a windsock that helps the pilots estimate the wind direction and force just by looking at the windsock.

In this project we want to determine if a machine learning model can be used to determine these measures by analyzing the windsock in the webcam images of the paragliding take off "Möntschelen". Because we already know what the model should predict we use a supervised learning approach. First, the webcam images are downloaded, and then get labelled by a person using our own developed Python script. In a second step we define the relevant pixels and color channels that carry the most information for the considered task to represent the independent features used in the model. We then transform the data so that it can be processed with the scikit-learn models. Finally, we test different models starting with linear regression to find out which model is suitable for this task.

This conceptual design report (CDR) describes the used methods, data flows and models in this project. Furthermore, we give an outlook on the analysis that could be done if the model reaches the minimum targeted precision of 80%.

Table of Contents

1 Project Objectives

As a paraglider pilot it's often hard to find enough time to paraglide besides working, family, and all the other responsibilities. Unlike other recreational activities paragliding is very dependent on the weather conditions. One popular paragliding take-off site around Bern is in Blumenstein on the alp Möntschelen. There is no weather station at this site, but a webcam taking a picture of the take-off area, including the windsock, every five minutes. Especially for pilots who want to fly very safely, it is often difficult to tell if it is safe to fly on a given day. Usually, one must walk up for about 1.5 hours to the take-off spot. Therefore, it is not nice to hike up only to find out that there is too much wind to take off.

In this project, we want to analyze the image data provided from the webcam to obtain statistical information about the wind conditions at the take-off site. Besides information about clouds and precipitation, the most critical weather factor for paragliding is wind force and direction. One widely used measuring tool is a windsock. It hast the advantage that the pilots can check the wind conditions during flight preparations and immediately before the take off just by looking at the windsock, without the need for digital instruments. It is quite easy for humas to interpret the wind direction and force just by looking at the windsock. The first goal of this project is to find out if a machine learning algorithm can be trained to do the same thing. If successful, the algorithm will be used to classify the wind force and direction over an extended period of time (at least three month) to find typical patterns for wind conditions at this location.

2 Methods

The used infrastructure are Google Collab and local Python installations (Visual Studio Code, Anaconda Distribution for Python) on private clients. Depending on the required interval of data downloads, switching to a server-based infrastructure in the future would further reduce manual tasks.

The following modules used in the present project are:

1. Requests[1] – Scraping the images of the webcam.
2. OS[2] – Labelling the webcam images and removing faulty images.
3. Skimage[3] – Loading image to a Numpy ndarray.
4. Matplotlib.pyplot[4] – Plotting images and figures.
5. Pandas[5] – Importing and exporting csv files.
6. Time[6] – Showing the image for a specific time during labelling.
7. IPython.display[7] – Clearing the figure during labelling.
8. Google.colab – Mounting Google drive.

9. OpenCV[8] – Import images as arrays and detecting objects in the images.
10. Scikit-learn[9] – Applying machine learning methods
11. TensorFlow[10] – Further machine learning functionalities
12. Seaborn[11] – Data plotting
13. Plotly[12] – Data plotting

First, the webcam images are uploaded to a local drive or Google Drive using web scraping methods. Next, the labels are generated using our own developed labeling-script. This labeling-script demands user interaction (wind direction and wind force labels are determined by a person looking at each image). The labels are then stored in a dictionary (key=image-name, value=List of labels indicating the wind direction and force) and written to a CSV-file on Google Drive.

In the next step, the images are cropped to the required area, if necessary, the resolution is further reduced (depending on if the complexity of the model is too high), and the image data is put into a NumPy-array format so it can be processed by the functions of the machine learning modules. In this format, each pixel-value of an image is a column (for single color images or gray scale images there is one value per pixel, which corresponds to one column per pixel), and each image is row. The label-values (wind direction and force) are converted to numerical representations and are stored in a separate NumPy-array. Machine learning methods demand a training-set to estimate the unknown coefficients in the model, and a validation-set to evaluate these estimates with new data. To randomly split the data into a training-set and a validation-set, the *train_test_split* function from the Scikit-learn module was applied. This function lets the user decide, to what proportion of the data each set is made of. We followed the convention of allocating 80% to the training-set and 20% to the validation-set.

In a final step various machine learning methods are applied to all image data with the goal, to find a model that predicts the wind direction and the wind force in a completely new image correctly. To evaluate final performance of the models, new images will be labelled to create a test-set. Sufficient performance of the model at this stage is necessary, since the second goal of the present project is to analyze wind direction and wind force from images across an extended period of time, for which it would be impractical to label the images by hand.

3 Data

The main data source are images from the webcam on the alp Möntschelen owned by the Para-Deltaclub Stockhorn which are published on their website[1]. The website contains the 720 most

[1] https://www.pdcs.ch/fluggebiet/moentschelen/webcam/

recent images taken at a five-minute interval. On this website only images taken during daylight are published. So, the time span and number of images per day vary. To avoid the risk of missing images, the content of the website is written down every four days to our local disk. During the summer, this download rate must be reduced to three days due to the longer days. Since people may be recognizable on the images, no images will be published or uploaded to public platforms (e.g., GitHub) without anonymizing them first. At the time of writing this report, a total of 337 images were downloaded and labelled.

***Table 1**: Wind direction labels*

Label	Description
"l"	Flag showing to the left (wind coming from the right)
"r"	Flag showing to the right (wind coming from the left)
"u"	Flag showing up
"d"	Flag showing down
"n"	No wind
"0"	Not defined (due to weather conditions or unclear orientation of the windsock)

***Table 2**: Wind force labels*

Label	Description
"n"	No wind (windsock down)
"w"	Weak wind (windsock up to 45 degrees)
"m"	Medium wind (windsock from 45 degrees to horizontal)
"s"	Strong wind (windsock horizontal or almost horizontal)
"0"	Not defined (due to weather conditions or unclear orientation of the windsock)

Tables 1 and 2 display the labels for the wind direction and the wind force, respectively. Figures 1 and 2 illustrate two different wind conditions at the take-off spot. In Figure 1, the windsock indicates weak wind from the right side. For an average skilled pilot, these conditions would be unproblematic for a safe takeoff. The labels for this image are "l" for the wind direction (stands for "left", indication the direction of the windsock) and "w" for the wind force (stands for "weak").

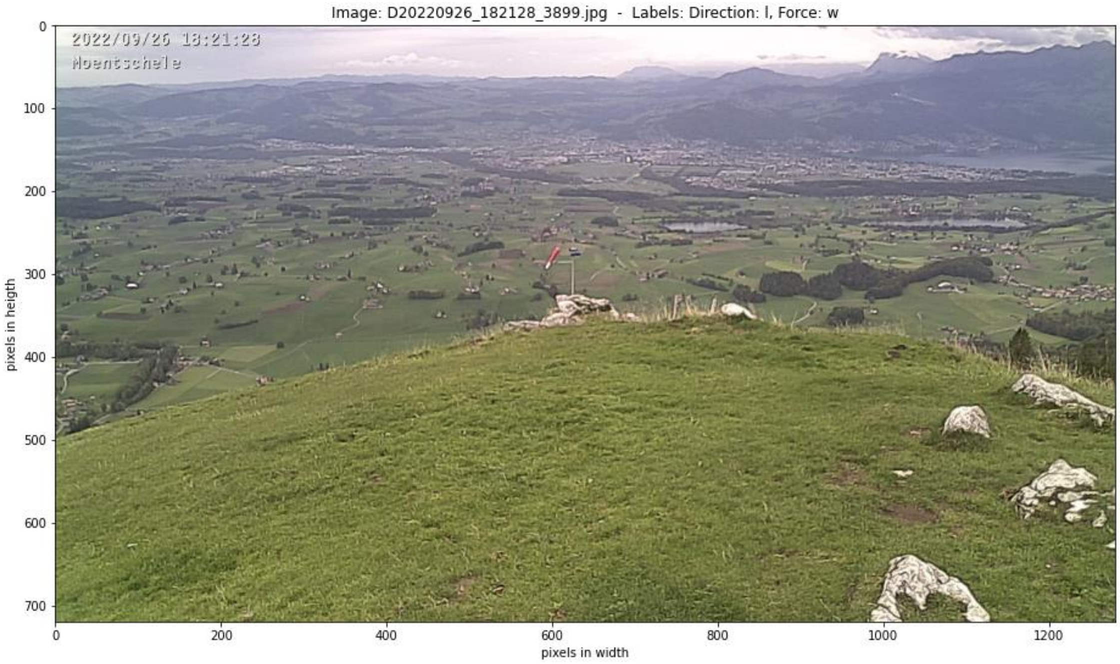

Figure 1*: Example image with weak wind from the right (windsock showing to the left).*

In Figure 2 the windsock points to the right and is almost horizontal indication strong wind coming from the left.

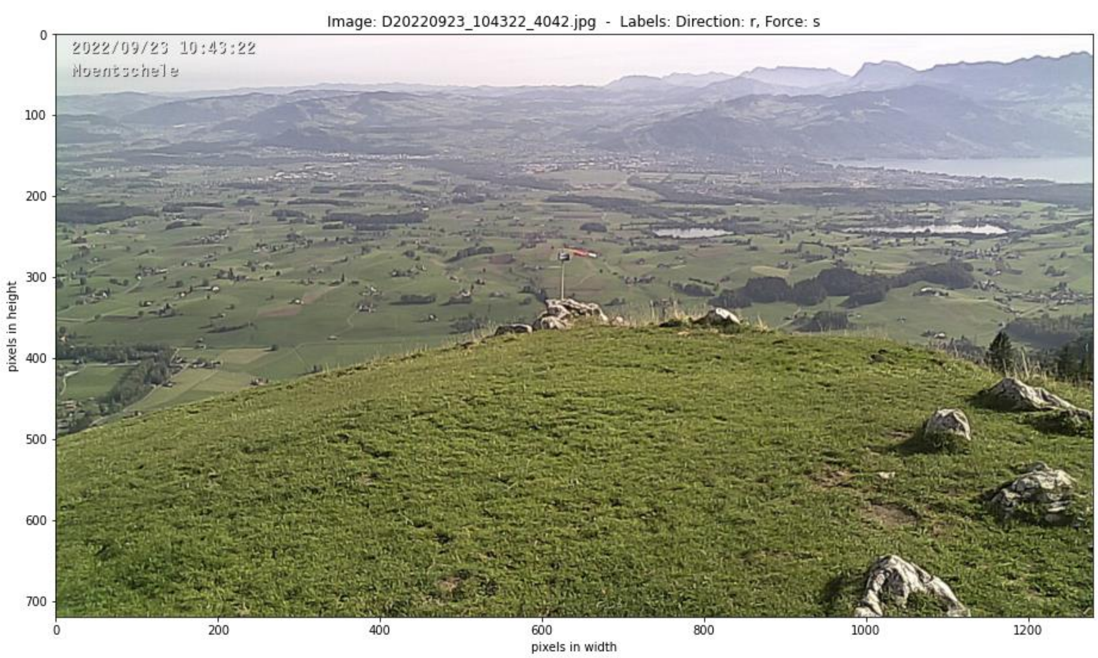

Figure 2*: Example image with strong wind from the left (windsock showing to the right).*

These conditions are more demanding for paragliding and for an averagely skilled pilot it would not be safe to take off. The labels for this image are "r" for the wind direction (stands for "right") and "s" for the wind force (stands for "strong").

However, since only a small part of the webcam image contains information about the wind direction and wind force, namely the area containing the windsock, only a small section of the image is considered for training the model. This reduces the complexity of the model and decreases the amount of irrelevant information.

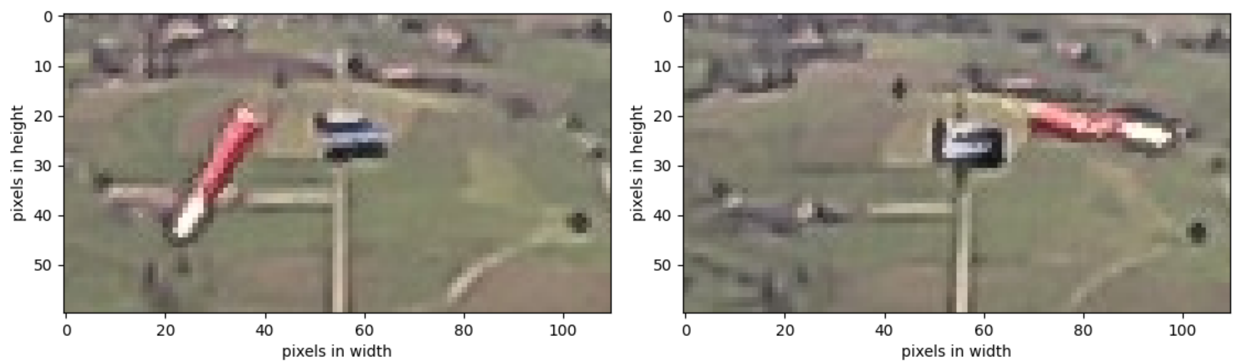

***Figure 3**: Example of the relevant area of the webcam image used in the learning of the model (left of Figure 1, right of Figure 2)*

Figure 3 displays the considered image part used in the machine learning model. Each image is 60 pixels in height and 110 pixels in width, resulting in a resolution of 6'600 pixels. However, since the image is colored, each pixel has three color values (one for red, green, and blue each), therefore the information of the cropped image is still high, with 19'800 information points. Since the windsock clearly stands out with its red color, and the remaining colors are negligible for determining the wind direction and wind force, only the red color channel is used for the machine learning models (6'600 information points per image). Figure 4 displays the values of the red color channel in the two example images (ranging from 0, meaning no red light, to 255, meaning maximum red light), mapped to a virids color palette. The white tip of the windsock stands out so clearly in the red color channel, since white is the presence of all three color channels.

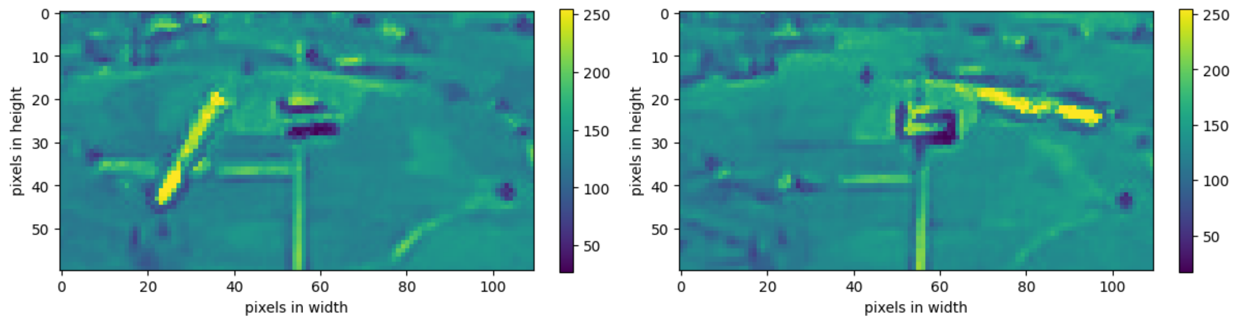

***Figure 4**: Example of the red color channel values used in the learning of the models (left of Figure 1, right of Figure 2)*

Figure 5 displays the distribution of wind direction across all labelled images available at the time of writing this report. In most of the labelled images the windsock is pointing to the right i.e., the wind is coming from the left. Figure 6 displays the distribution of the wind force across the labelled images. The vast majority of the images indicate weak wind.

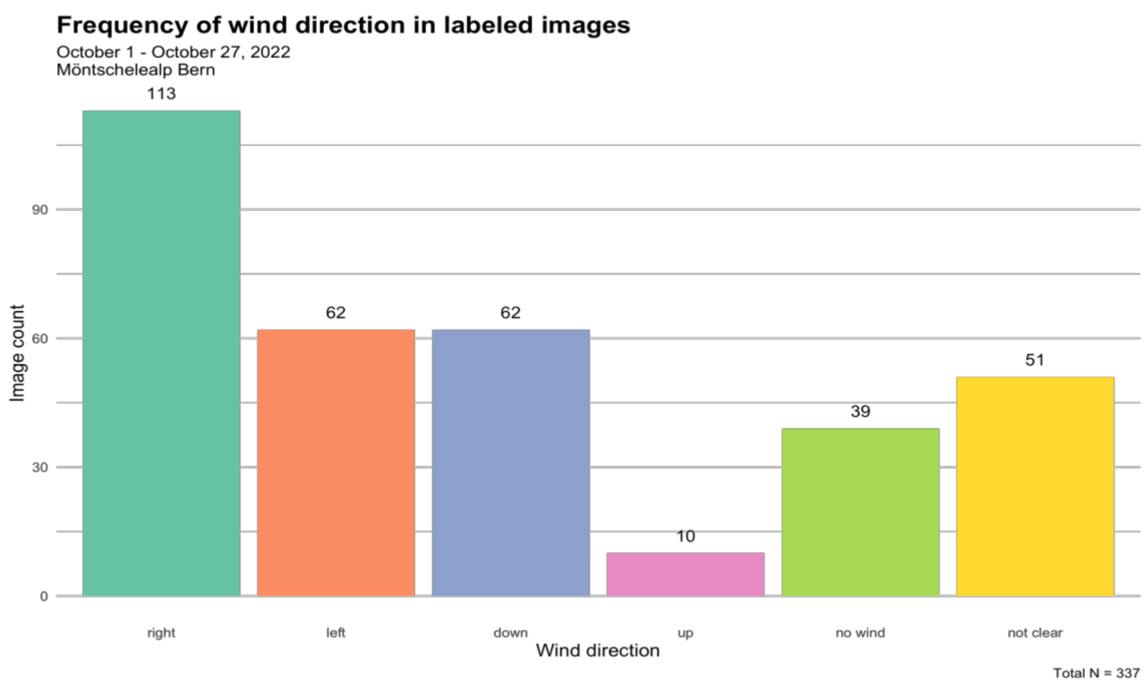

***Figure 5**: Distribution of the direction of the windsock of all available data at the time of writing this report*

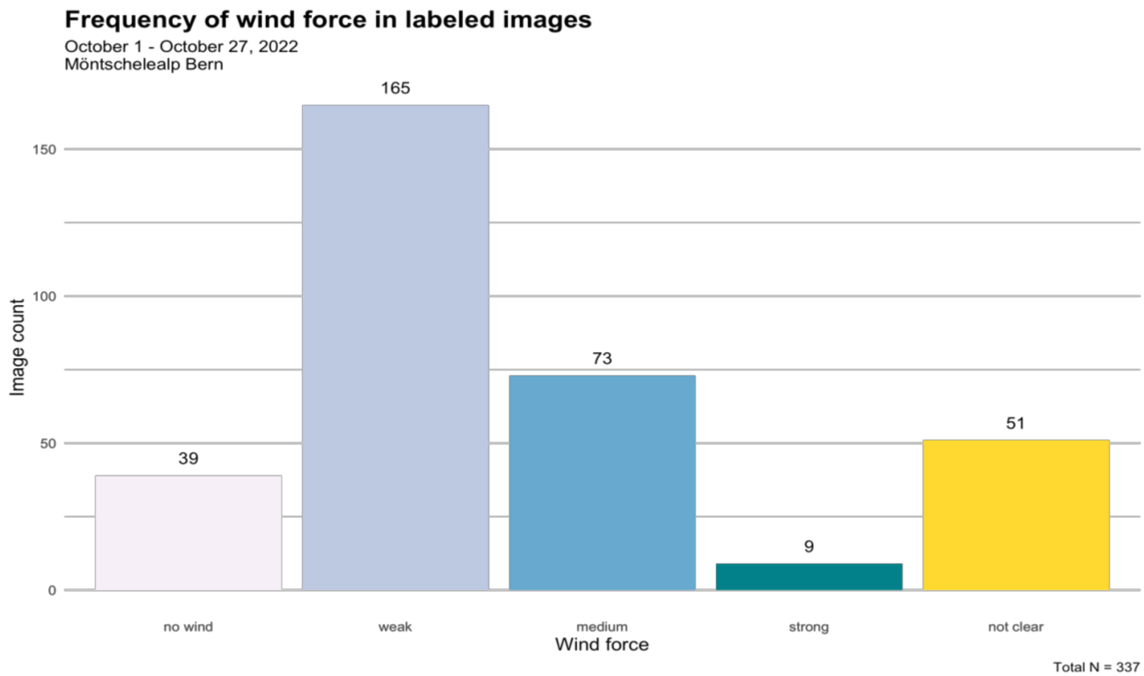

Figure 6*: Distribution of direction of windsock (windsock showing to the right – wind coming from left)*

4 Metadata

The main sources of metadata are the image names (e.g., "D20220923_104322_4042.jpg") as it contains the date and time when the image was taken, the image labels of wind direction and wind force, the coordination of the relevant image area that contains the windsock, and the scripts for scraping and labelling the data as well as the source code for the machine learning models and analyzes. Since all this data is not sensitive on its own, and no archive of the images is publicly available, all metadata is stored in our personal GitHub repository, where it is accessible by the public.

The images themselves will be stored on our personal clients, since these could contain sensitive information, e.g., people who are in the images could be recognizable and identified.

5 Data Quality

For the automatic detection of the wind force and wind direction, we require an accuracy of 80%. This means that the algorithm should determine the correct results for both wind direction and wind strength from 100 images for at least 80 images as they are described in Table 1 and 2.

The accuracy of 80% should be sufficient for further analyzing wind patterns and correlation of these patters with regional weather conditions. The model's predictions are not intended to influence in any way the paraglider pilots' decision whether to take off or descend. This decision remains entirely the responsibility of each pilot.

We expect that the data quality will be somewhat challenging, as the weather conditions and natural lighting of the images can be very heterogenous. There are images where the windsock is clearly visible, with a blue sunny sky in the background. But there are also images where the take-off spot is completely covered by clouds early in the morning in low light. If we have problems with these conditions, we could try to limit the analysis to images on sunny days, since these are the conditions that most pilots intend to fly. For this, we would need to develop an algorithm that can distinguish between sunny and cloudy days.

6 Data Flow

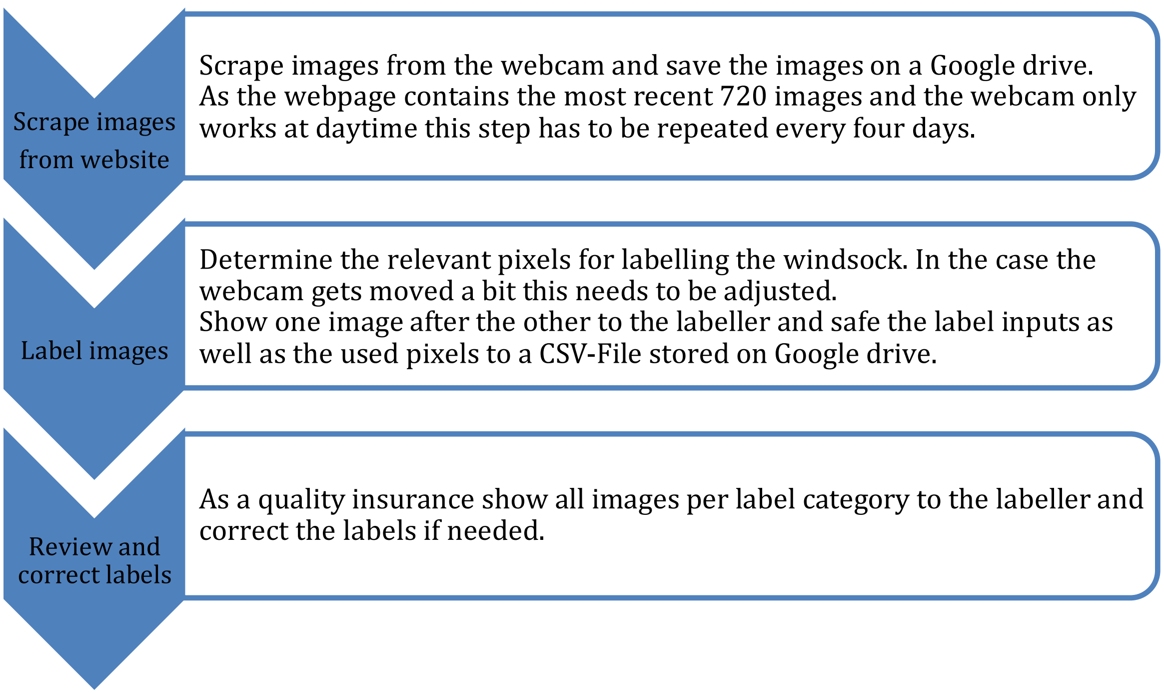

***Figure 7**: Labelling process.*

The first step is to label the images needed to train the machine learning model. The data flow above in Figure 7 illustrates the labelling process. In a second step we train our models with the manually labelled images and aim to label images over a longer time period with these models. The data flow below in Figure 8 illustrates this learning process.

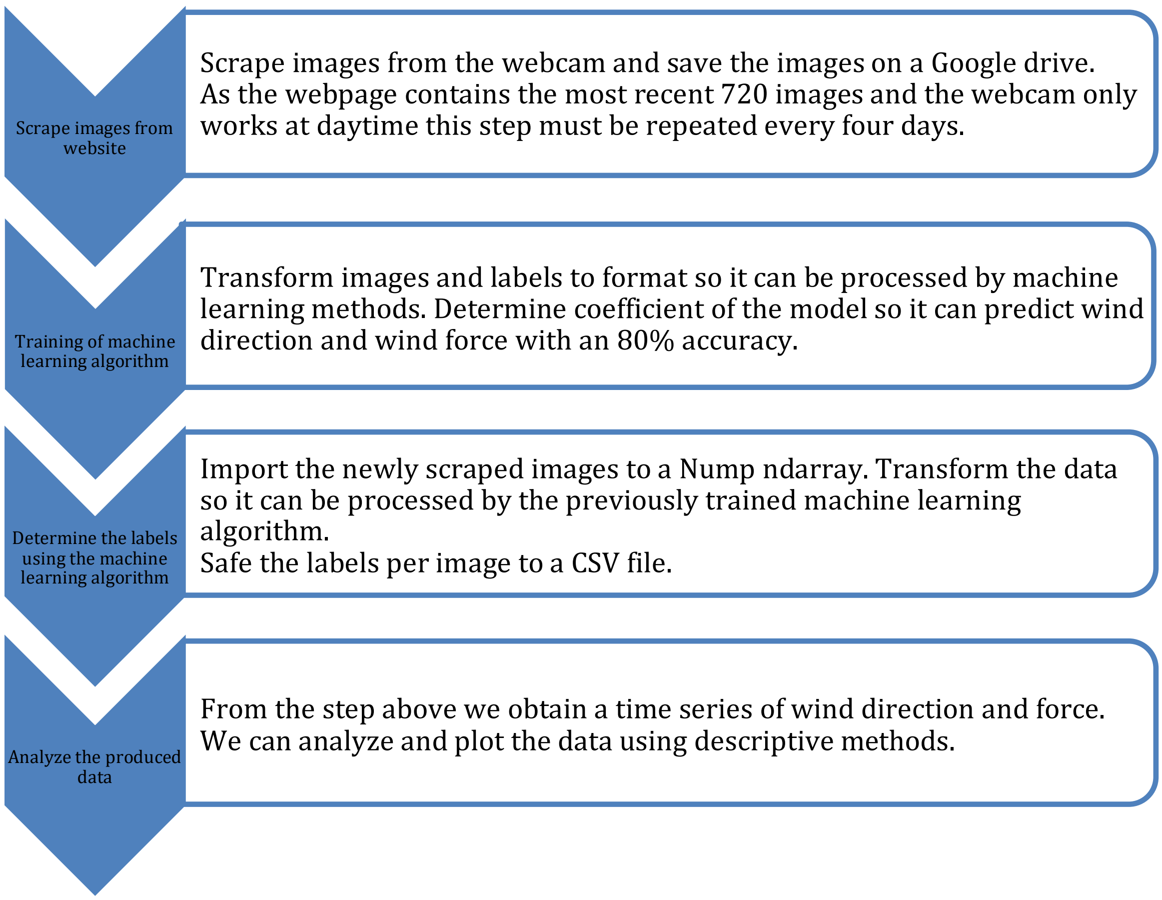

***Figure 8**: Machine learning process.*

7 Data Model

The data model at the conceptual level is to develop a tool to reliable determine the wind conditions at the take-off spot. At the logical level we intend to apply two models, a linear regression model and a neural network. The red color channel of the pixels in the defined area of the webcam images are the features used. Since the relevant area is 60 pixels high and 110 pixels wide, this results in 6'600 features. On the physical level, there are no special hardware requirements since the models are simple enough to be computed with conventional computers.

8 Risks

There is a risk that the webcam will stop working one day. There could be technical issues, or the host could decide to remove the camera from the take-off spot. In this case we would have to work with the already downloaded images.

On the other hand, it would not be possible to obtain the needed wind information over a longer time period if the webcam would stop working before we have images for at least three months. Another risk for this long-term analysis lies in the fact that we don't know if we will reach the needed precision with machine learning models. We assume that training neural networks would increase the precision, but they would also need more training data which would mean more manual labelling effort.

9 Conclusions

The development of a model that can forecast the wind conditions at the take-off spot is still at an early stage. However, the data quality of the already downloaded images, the project plan, and the preliminary results are promising. Webcam-based wind forecast could be an alternative at paragliding take-off spots, where a high-end weather station is not possible or economical feasible.

10 Outlook

By relating the common four wind categories (westerly winds, north wind, east wind, and south wind shown in Figure 9) in Switzerland, with local wind patterns at Möntschele, we could try to identify the weather conditions that are good for flying at this take-off spot.

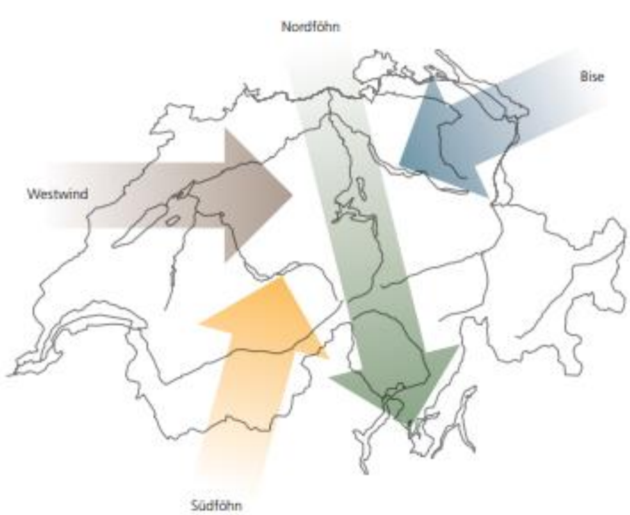

***Figure 9**: Typical wind conditions in the alps*[2]

[2] https://www.meteoswiss.admin.ch/content/dam/meteoswiss/de/service-und-publikationen/Publikationen/doc/Web_Wetterlagen_DE_low.pdf, *Page 13*

Furthermore, additional weather forecast data could be used to predict the next few days, for which flying conditions would be good.

The Möntschele take-off spot is mostly preferable when there is wind from northeasterly directions in the flatlands (Bise). A weather station that can be checked to estimate what the conditions are like is on the Bantiger. By looking at this weather forecast it is possible to get a sense about the wind force on the Möntschele at a given day. Thus, other interesting research questions could include: What is the correlation between the wind force on Bantiger and the wind patterns at Möntschele? Is there a maximum wind force at Bantiger that should not be exceeded to safely fly at Möntschele? Also, the interpretation area of the images could be extended. Not only the windsock could be analyzed, but the entire image. This would also allow further research questions such as: What is the minimum wind force and direction that leads to a significant higher number of pilots compared to the mean pilot amount on days clear of clouds. This would bring new challenges, such as recognizing and counting pilots without double counting them if they linger longer than 5 minutes or being able to distinguish them from hikers.

References and Bibliography

[1] Reitz, K. (2022). *Requests: HTTP for Humans* (2.28.1). https://requests.readthedocs.io/en/latest

[2] van Rossum, G. (2022). *os—Miscellaneous operating system interfaces*. https://docs.python.org/3/library/os.html

[3] van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). scikit-image: Image processing in Python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453

[4] Caswell, T. A., Lee, A., Droettboom, M., De Andrade, E. S., Hoffmann, T., Klymak, J., Hunter, J., Firing, E., Stansby, D., Varoquaux, N., Nielsen, J. H., Root, B., May, R., Elson, P., Seppänen, J. K., Dale, D., Jae-Joon Lee, McDougall, D., Gustafsson, O., … Kniazev, N. (2022). *matplotlib/matplotlib: REL: v3.6.1* (v3.6.1). Zenodo. https://doi.org/10.5281/ZENODO.592536

[5] The Pandas Development Team. (2022). *pandas-dev/pandas: Pandas* (v1.5.1). Zenodo. https://doi.org/10.5281/ZENODO.3509134

[6] Python Software Foundation. (2022). *time—Time access and conversions*. https://docs.python.org/3/library/time.html

[7] IPython Development Team. (2022). *IPython*. https://ipython.readthedocs.io/en/stable/index.html#

[8] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. https://pypi.org/project/opencv-python/

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012). *Scikit-learn: Machine Learning in Python*. https://doi.org/10.48550/ARXIV.1201.0490

[10] TensorFlow Developers. (2022). *TensorFlow* (v2.8.2). Zenodo. https://doi.org/10.5281/ZENODO.4724125

[11] Waskom, M. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

[12] Plotly Technologies Inc. (2015). *Collaborative data science*. Plotly Technologies Inc. https://plot.ly