ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

IGP
Institute of Geodesy and Photogrammetry

Agnieszka Rożniak

# Drone Images and Deep Learning for River Monitoring in Switzerland

**Semester Project**

Institute of Geodesy and Photogrammetry
Swiss Federal Institute of Technology (ETH) Zurich

**Supervision**

Prof. Konrad Schindler
Dr. Jan Dirk Wegner
Nico Lang

in collaboration with: Hunziker, Zarn& Partner

May 2019

# Abstract

Due to human activity, mainly hydro power plants, the balance of the sediment in Swiss rivers is diminished. This disruption causes negative effects on bed stability and aquatic habitats. A key indicator for the sediment dynamics of a river system is a grain size distribution. Therefore, it is an important aspect to be monitored, so that the balance within river dynamics can be retrieved. However, current methods evaluating the grain size distribution are cumbersome and not efficient. This work introduces a new approach to estimate the grain size distribution over large spatial areas from high-resolution drone images. The method omits the explicit detection of individual grains and instead obtains the grain size distribution directly from the image texture, using supervised deep learning with regression. Within the scope of this project, a novel CNN architecture based on the state of the art ResNet-50 model was created.
The training data was collected and labeled by experts from the hydrological engineering company Hunziker, Zarn& Partner. The tiles in the final data set depict gravel bars along 15 river basins in the northeastern part of Switzerland.
The outcome of the proposed approach is promising and was evaluated with various quantitative and qualitative techniques. The suggested solution has a potential to replace the current methods of estimating grain size distribution and consequently to simplify and improve the process of the sediment monitoring.

**Keywords:** grain size distribution, deep learning, CNN, remote sensing, drone images.

# Acknowledgment

I would like to take this opportunity to express my appreciation to everyone who supported me during this project. I would like to thank especially:

- **Prof. Dr. Konrad Schindler** and **Dr. Jan Dirk Wegner** for making this thesis possible.
- **Roni Hunziker** and all his employees for the collaboration opportunity and providing the previously prepared data set.
- **Andrea Irniger** for her expertise in the field of river monitoring and her support in solving numerous issues regarding the data.
- **Nico Lang** for continuous support and guidance during the duration of this project. Our rewarding discussions and his valuable inputs contributed significantly to the final outcome of this work.
- **Andyn Omanovic** for proofreading.

# Contents

# Chapter 1

# Introduction

Without water, no life could exist. Many essential as well as excessive activities of people would not be possible without the use of healthy fresh watersheds. Considering only various human ways of fresh water exploitation, multiple areas in which the water is marginal can be listed. Just a few examples include domestic and industrial usage, irrigation, agriculture, or energy generation. Water systems are also a natural habitat for vast numbers of plants and animals, as well as a vital part of the entire Earth ecosystem. Hence, ensuring the high quality of the water and keeping an ecological balance are crucial tasks for the society.

There exist multiple ways in which watersheds can be monitored, and one of them is to monitor grain size distribution of the gravel bars along rivers. Grain size distribution is a key indicator for the sediment dynamics of a river system. Currently, due to the human activity the balance of the sediment is disrupted in almost one-third of all watercourses in Switzerland. This imbalance brings negative consequences and can affect bed stability and aquatic habitats, as well as lead to floods. In a response to this problem, the Swiss Federal Office for the Environmentt operates a nationwide monitoring network for the sediment transport in Swiss watercourses [1]. The overall goal is to retrieve the balance within the river dynamics, as a prerequisite for successful water systems revitalization projects.

However, development of a completely acceptable method for assessing the grain size distribution in gravel-bed rivers has been made difficult by the multiscale heterogeneity of river-bed sediment [2]. The majority of current techniques has been manual measurements. Those methods are not only cumbersome and time consuming, but also lack in generalization abilities and pose risk to human safety. Therefore, new methods are required by fluvial scientists. Since remote sensing approaches have revolutionized the fluvial data acquisition over the last two decades [3], those technologies could deliver a satisfying alternative to the classical measurement techniques. Multiple approaches were already tried, such as aerial photosieving [4, 5] or a Basegrain MATLAB tool [6]. Both present certain limitations. Photosieving requires high-resolution images and additional field calibration, and is also limited by pixel size. While working with the Basegrain software, the user needs to perform numerous additional manual adjustments, the software is highly unstable, and fails in the case of overlapping grains or shadows.

The purpose of this work is to overcome limitations of existing methods, by developing an approach which will make it possible to estimate the grain size distribution over large spatial regions from high-resolution drone images. By that, sediment monitoring can be greatly simplified and improved. This novel methodology is based on the recent advances of convolutional neural networks. It bypasses the explicit detection of individual grains and instead estimates the grain size distribution directly from the image texture, using supervised deep learning with regression.

# Chapter 2

# Related Work

In the recent years, the growing popularity of the remote sensing approaches can be seen [3]. Hence, various approaches using digital aerial photography were developed for the determination of the grain size distribution, to overcome the limitations of existing manual techniques. Aerial photo-sieving employs high-resolution imagery and image texture analysis to estimate the characteristic parameters of the grain size distribution over large areas. [4, 5] It focuses on recognizing texture patterns related to grain size, using tools such as geostatistical analysis and linear regression. However successful, this method is limited by light conditions, sediment color, as well as pixel size and the need for a field calibration. Another approach is presented in [2] and uses aerial imagery and a Structure-from-Motion photogrammetric technique to extract 3D point clouds, from which the grain characteristics can be derived. Its limitation is again represented by the required field calibration.

Other approaches focus on the estimation of the grain size distribution with image processing technique. [7] developed four image-segmentation procedures, which were successfully tested on measurements of fluvially deposited gravel in order to segment individual grains. Based on this method, the MATLAB-based algorithm, so called Basegrain [6], was created. It estimates the grain size distribution by analyzing the digital top-view photographs, using an optimized object detection approach. Each detected grain is replaced with an ellipse, which size is then measured. This information can be further transformed to a grain size distribution, using methodology described in [8]. However, this approach still requires manual adjustments and is unstable, since it fails in case of overlapping grains or presence of shadows.

While many methods were developed in order to estimate the grain size distribution, they are either limited by the existing need for a field calibration, or by limitations related to the explicit object segmentation.

However, while deriving a grain size distribution, the main interest is in finding the distribution of the object sizes in the image. Information about single objects is not required. [9] introduces a convolutional neural network architecture which predicts their size distribution without explicit instance segmentation and can be mapped to the problem of estimating the grain size distribution. The solution presented in [9] learns histograms of object sizes directly from the images. Employed data sets contain objects belonging to one category, such as soldier fly larvae or cancer cells. Obtained distributions are useful in biological research or for medical use, respectively.

# Chapter 3

# Theoretical Foundations

This chapter presents theoretical foundations, which are necessary to understand the problem, as well as its solution proposed within the scope of this project. The basic concepts and reasons behind river monitoring are summarized in Section 3.1. Current development of remote sensing is briefly explained in Section 3.2, with special emphasis given to Unmanned Aerial Vehicles (UAV). Subsequently, the basic theory associated with the Convolutional Neural Networks is presented in Section 3.3. It is assumed that the reader is familiar with the basics in the field of image processing.

## 3.1 River monitoring

Grain size distribution of a riverbed sediment in gravel-bed rivers have been in the area of interest for fluvial scientists and engineers since a long time. It is a key indicator for the investigation of the dynamics in river systems. One of the fundamental components of the properly functioning water system is its sediment, which is transported together with the water flow. It can be carried in two different forms, as a bed load or as suspended soils, and is naturally balanced in ecologically untouched rivers and streams. Grain size distribution provides fundamental insights on the interactions between water flows, sediment transport, and the morphodynamic evolution of rivers.

Understanding the changes in river dynamics helps us to investigate the response of rivers subject to both environmental forces and human activities. Currently, the sediment balance has been disrupted in almost one-third of watercourses in Switzerland, mostly due to river control structures and hydro-power plants. The disrupted sediment dynamic can have negative effects on the bed stability, as reduced load transport leads to unnatural erosion. There is also an increased risk of floods, since the water level may change depending on the speed of the river, affected by the roughness of the watercourse substrate. Finally, the imbalance affects the aquatic habitats of water organisms and plants. An example can be a disturbance of suitable spawning grounds for fish. Reactivation of the original dynamics is a prerequisite for successful revitalization of the water system. [1]

Therefore, a development of a satisfactory method for the measurement of the grain size distribution and surface roughness in gravel-bed rivers is required. The difficulties arise from the substantial heterogeneity of the sediment, which necessitates sampling over multiple patches of similar texture and grain size for an accurate representation of the reach-scale variability. Currently, one of the conventional and well-established measurement procedures has been a manual pebble count on exposed gravel bars. The method is called line sampling, as grain sizes are evaluated along a line. A measuring tape is placed in a few arbitrarily chosen locations on the surface of the river bank and the sizes of 100 to 150 stones along the tape are assessed. The measured size is the so-called b-axis, which is the middle dimension of a stone, nor the shortest neither the longest extent. The survey is performed with a simple ruler and each stone is categorized based on its b-axis. The

procedure results in the counted number of grains per defined grain classes, what can serve as a basic information for the derivation of the grain size distribution. Further parameters characterizing the grains in the measured area, such as fraction-weighted mean diameter (dm) can be derived from obtained distribution. [10, 8]

The pebble count method, although widely popularized among fluvial scientists and also relatively reliable, has several limitations. The measurement is highly time consuming and tiresome, and also heavily dependent on the observer and hence subjective. Furthermore, the method is limited in its ability to provide high resolution information, as unique measurements in specified locations are not sufficient for the accurate representation of the spatial variability of the river sediment. The technique is also destructive, what reduces the potential for repetitive measurements and affects the habitat. Finally, the exposed gravel bars often arise as unreachable river islands or patches with limited access due to e.g. steep terrain in mountain torrents.

Consequently, fluvial scientists and engineers require a faster and more reproducible technique for grain size measurements, which is capable of providing fast and accurate results at larger scales. Thus, these needs were addressed in recent years by numerous remote sensing approaches, such as photosieving or Basegrain, described in the previous chapters. These new technologies could deliver satisfactory alternatives to existing classical approaches.

## 3.2 Remote sensing

Rapid development of UAVs, commonly known as drones, as well as continuous advancement of on-board installed sensors and instruments, have tremendously increased the usage of this technology in many scientific areas. In 2007 the European Commission has defined a number of applications in which UAVs are or can potentially be employed in the future. The exemplary areas of interest include agriculture, forestry, surveillance, environmental monitoring, photogrammetry, archaeology and urban environments. As UAVs capture information at distance, without any direct contact with an object, measurements performed by aerial systems are considered as remote sensing techniques.

The huge success of this technology comes from its numerous advantages. Firstly, UAV measurements provide an outstanding opportunity to close the gap between traditional field observations and air- and space-borne remote sensing. Moreover, spatial, temporal and spectral requirements can be captured with relatively small cost. Finally, the high adaptability and flexibility of drone measurements enable rapid and repeated data retrieval adapted specifically to particular use cases. Due to the numerous advantages and growing popularity of UAVs, plenty of methods, procedures, and strategies were developed and disseminated. Scientists are also becoming more and more aware of UAVs and contributing new ideas, while companies are undertaking new challenges and developing personalized solutions. At the same time, the range of performance and applications is extended.

Depending on the application, UAVs may be equipped with various sensors. One of the popular choices is a digital camera combined traditionally with remote surveys through photogrammetry, which is a science of obtaining measurements from photographs. With proper instruments, UAVs are able to obtain photographs accurate enough to constitute a base for further processing and analysis. Before a flight, a host of parameters must be taken into consideration, such as sensors to be used, geographical extent of a study, height of the flight, ground sampling distance, meteorological forecast and local regulations concerned with drone flights. All of those parameters, as well as additional factors such as pilot expertise, affect the final characteristics of the acquired data and the successive data processing. The accurate absolute orientation of the images needs to be defined in a process referred to as a georeferencing or registration. Usually it is done either by a positioning sensor installed on the UAV or by establishing ground control points (GPCs), whose coordinates are fixed with a higher-order control method, such as the Global Navigation Satellite System. Georeferenced images can be a source of orthophotos, which can serve for further image analysis. [11]

## 3.3 Convolutional Neural Networks (CNNs)

This section provides a short overview of Convolutional Neural Networks, as well as introduces basic terms and concepts related to training the model. Firstly, general ideas behind traditional Neural Networks and Convolutional Neural Networks are illustrated in Section 3.3.1, and Section 3.3.2. The basic building blocks of a CNN are listed and characterized in Section 3.3.3. Subsequently, Section 3.3.4 describes the training, and is followed by an introduction to the model parameters (Section 3.3.5), which are necessary to build and train a CNN. Section 3.3.6 presents 4 different loss functions which can be used, when solving regression problems. Finally, a short description of the batch normalization is provided in Section 3.3.7. The content of this section is based on the lecture notes from Karpathy [12], if not listed differently.

### 3.3.1 General idea behind Neural Networks

Artificial Neural networks are machine learning algorithms inspired by the way information is processed by the biological neurons in the human brain. They are designed to learn and recognize various patterns in the input data and predict the outcome based on it. A basic computational unit of a neural network is a single neuron, called also a node or a unit. Neurons are arranged into layers. Furthermore, connections are created between nodes from adjacent layers, but nodes within single layer function independently and do not share connections. All the interconnected layers build the complete network. The simple exemplary architecture can be seen in Figure 3.1.

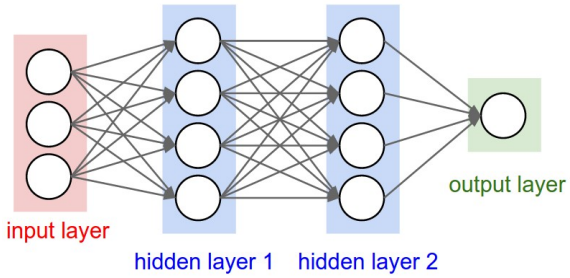


Figure 3.1: Simple example of the neural network architecture with the input, output and two hidden layers. Taken from: [12]

The first layer of the network is called an input layer and contains input neurons. The name of the last layer is output layer and respectively, it consist of output neurons. Finally, the middle layers are addressed as hidden layers, since the neurons are neither inputs nor outputs. The reasoning behind the layer-wise organization structure of the network is strictly mathematical, as the calculations and evaluations can be efficiently and simply represented using matrix operations.

The information within a neural network structure is passed forward through the layers. Each neuron receives some inputs, performs a defined mathematical operation and passes the resulting single outcome to the neurons in the following layer. Every input has parameters, which are its weight and bias and can be understood as the importance of the input for the following output. The final outcome may also depend on an activation function, which introduces non-linearity into the network. In this way the neurons are able to learn non-linear representations. Different activation functions will be further discussed in the following subsections. The whole network from the input to the output data is represented by a differentiable function.

### 3.3.2 From Neural to Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are Neural Networks, which assume explicitly that the inputs are images. This premise encodes certain property within an architecture, which is a parameter sharing. While regular neural nets are built only out of fully connected layers, in which all neurons in one layer are connected to all neurons in the following layer, this idea does not scale

well to images. With images giving a lot of input data, the number of parameters would blow up quickly. That is why CNNs use convolutions with filters, which share weights across different parts of the input data. As a result, deep networks with many parameters can be successfully trained.

The input is passed to the network as a volume, e.g. RGB image. While training, the CNN learns features from the input images. The recognition starts with simple elements, such as edges or corners. With the increasing depth of the network, the extracted information becomes more complex and more complicated structures are built. The final layer returns an output, e.g. an array of class scores in classification. For this process to succeed, the architecture is built out of different types of layers, described below.

### 3.3.3 CNN architecture

Each CNN architecture consists of basic building blocks: convolutional layers, pooling layers, and fully connected layers. Additional layers may apply depending on the architecture.

**Fully connected layer (FC)** is the standard layer used in every neural network. In FC layer neurons have full connections to all activations from the previous adjacent layer. The calculations are simple matrix multiplications between the input arrays and the weight vectors, followed by an addition of a bias and eventually interwoven with the activation function. In CNNs, the FC layer is typically used at the end of the network to connect all the neurons with the final predictions.

**Convolutional layer (CONV)** is what distinguish a CNN from a regular artificial neural network and it is a key element of the CNN concept. The primary function of the convolution is feature extraction. The image is convoluted with sets of learnable filters, which share parameters. The reasoning behind this comes down to the assumption that if one feature is useful at some spatial position, it can be also profitable at another location in the image. Therefore, filters are connected only to local regions in the input image, and not to all neurons, as in the case of the FC layer. The spatial extend (width and height) of the filter is called a receptive field and is related to the size of the feature to be recognized. It is usually defined by a small squared kernel with an odd size, such as 3x3. Even though the 2D size of the filter is small, it is connected to the entire depth on the input, e.g. all 3 channels of an RGB image. The local connectivity through the entire volume is presented in Figure 3.2.

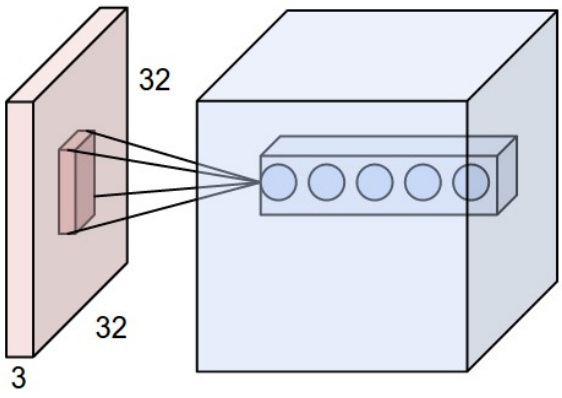


Figure 3.2: Each neuron in the convolutional layer is connected only to a local region in the input volume. Taken from: [12]

Each filter slides across the image and performs matrix operations. The calculations are the same as in the FC layer and comprise the dot product computed between the input and the weights, followed by addition of a bias and a possible non-linearity. Those operations results in the 2D activation map, which depicts the response to the filter in a specific location in the image. All the

maps are stacked together in the third dimension. Hence, the depth of the output volume depends on the number of filters.

**Pooling layer (POOL)** is used to reduce the height and width of the input image and as such do not contain learnable parameters. Pooling helps to reduce the number of parameters and computations in the network and hence increases the computational performance. This type of layer divides the image into sets of rectangles and outputs a single value for each sub-area. The output value may be defined with different strategies. Common choices are max pooling, which outputs the maximum value within the region and average pooling, which averages all the sub-pixels. The intuitive understanding of pooling is that exact location of the feature in the image is not as important as a rough location in relation to other features. Additionally, the decreased number of learnable parameters reduces the chances of overfitting. Pooling layers are usually located between the consecutive CONV layers.

**Activation layer** introduces non-linearity within the network. It is applied to the output of CONV and FC layers and it maps the resulting values into desired range. Two exemplary activation functions are described below.

Rectified Linear Unit Layer (ReLU) is a simple calculation that returns 0 if it receives a negative input, or the input itself, if its value is positive. ReLU is defined with the function,

$$f(x) = max(0, x). \tag{3.1}$$

Softmax function is also known as normalized exponential function. It outputs a probability distribution, which means that a vector of arbitrary real-valued scores get squashed in a vector of values between zero and one, which all sum to one. It is defined by the following equation:

$$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_k e^{x_k}}, \tag{3.2}$$

where $\vec{x}$ is an input vector of values.

### 3.3.4 Training

Training the network can be understood as finding such parameters of all layers, that the attained prediction is as close as possible to the given label. The error between the predicted and the true value is quantified by a defined loss function, which is a function of learnable parameters. The loss is minimized in the iterative training process and each iteration results in small updates to the model weights and biases. The values of updates are defined with an optimization algorithm such as gradient descent. In this method, the gradient is computed with respect to the parameters. It is a vector of partial derivatives, pointing to the direction of the steepest increase of the loss function. Since the loss should be minimized, each update is a small step in the opposite direction of the gradient. In this way the loss decreases gradually until it converges to some local minima. CNNs are constructed as a collections of layers, and parameters of each single layer need to be updated. To do so, the local gradient is back propagated through the network from the top to the bottom layer, such that all the weights and biases within all layers are adapted. Back propagation is a recursive application of a chain rule.

### 3.3.5 Model parameters

Model parameters define the architecture of the network, and the variables which determine how the network is trained (so called hyper parameters). This subsection introduces the following parameters: learning rate, mini-batch size, dropout rate and number of epochs. The concepts related

to those them are also shortly described. Although more parameters could be mentioned, there are not of a main interest within this project.

**Learning rate** defines the step size of the parameter update during training. The gradient calculation guarantees a decrease of the loss function if infinitely small steps are taken. However in practice, the infinitely small steps are approximated with the learning rate. It can be kept constant or adapted throughout a training process. The common practice is to decrease the learning gradually in order to approach an optimal minimum without overshooting.

**Mini-batch training** is a variation of a gradient descent algorithm and it is commonly used in the field of deep learning. In this implementation the exact derivative over the entire data set is not calculated. Instead, the gradient is evaluated on a subset of training samples, called a mini-batch. This method is effective, as all the training samples are correlated. Since the gradient is calculated only on a small subset of a data, the loss will fluctuate and be much more 'noisy'. However, it will decrease gradually over time. The size of the mini-batch is another hyperparameter within the network, and defines the number of images present in one mini-batch.

**Dropout** is a simple regularization technique introduced by [13], which prevents the neural network from overfitting. Model capacity is reduced by excluding certain units (i.e. neurons) from the training during a particular forward or backward pass. Hence, their parameters are not considered in the current calculation. The neurons to be excluded are chosen at random in every iteration, and the number of omissions depends on the defined dropout rate. More specifically, nodes are kept in the network with the given probability $p$. As a result, interdependent learning among the neurons is reduced.

**Number of epochs** is the number of times the whole training data set is shown to the network while training. It should be defined such that a sufficiently low value of the loss function can be obtained.

### 3.3.6 Loss functions

Loss function is a key element of a CNN, as it defines an objective against which the performance of the model is measured. During each iteration, the current loss is computed and estimates the current state of the model. For the model to learn the correct mapping from inputs to outputs, the chosen loss function needs to match the framing of that specific modeling problem. What is more, for the loss function to return proper outcome, the output layer must have an appropriate form. Four examples of loss functions are described below.

**L1 and L2 losses** are two standard loss functions, widely used in regression problems, where the target variables take a continuous value. L1, also commonly known as Mean Absolute Error (MAE), is the average of the absolute differences (errors) between target and estimated values. Its mathematical formula is given in Equation 3.3. L2, also referred to as a Mean Squared Error (MSE), is the average of squared differences (errors) between target and estimated values. L2 is calculated according to the formula presented in Equation 3.4.

$$L1 = \frac{1}{n}\sum_{i=0}^{n}\left|y_i - \hat{y_i}\right|, \tag{3.3}$$

$$L2 = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - \hat{y_i}\right)^2, \tag{3.4}$$

where $n$ is a number of samples, $y_i$ is a target variable of sample $i$ and $\hat{y_i}$ is a predicted target variable of sample $i$.

The difference between those two loss functions is their sensitivity to errors. Since in L2 the error is squared, the model is punished more for making a big mistake. In other words, L2 is less robust to gross errors than L1. When finding an outlier, L2 loss will be adjusted to minimize that single case at the cost of other common samples, which will reduce overall model performance. On the other hand, L2 provides a more stable solution.

**Kullback-Leibler divergence (KL divergence)** is a measure of how one probability distribution diverges from a second expected probability, and it was introduced by [14]. The KL divergence from $P$ to $Q$ is often denoted $D_{KL}(P \parallel Q)$. It is calculated as a difference between the entropy and cross entropy, what is equal to the formula,

$$D_{KL}(P \parallel Q) = \sum P(x) log\left(\frac{P(x)}{Q(x)}\right), \tag{3.5}$$

where $P(x)$ and $Q(x)$ are discrete probability distributions. KL divergence is not a distance measure, what means that the $D_{KL}(P \parallel Q)$ is not equal $D_{KL}(Q \parallel P)$. A divergence of 0 indicates that two distributions are identical.

**Intersection over Union (IoU)** is a measure of similarity between two finite sample sets, and is defined as size of an intersection divided by the size of the union of two sets. In the case of geometrical figures, the metric is constructed accordingly as the area of the intersection divided by the area of the union. IoU takes values between 0 and 1, as it is by definition a simple ratio. A value equal to 1 means that two sets are identical, while 0 indicates no common elements. The more similar two sets are, the higher the resulting value of IoU. Hence, while training a neural network, a $-IoU$ or $1 - IoU$ should be used as a loss function which can be minimized.

### 3.3.7 Batch normalization

Batch normalization deals with a problem referred to as internal covariate shift, which is defined as a change in the distribution of inputs to the current layer as a reaction to the change of parameters of the previous layer. It occurs during training and forces the current layer to repeatedly readjust to new distributions. The deeper the network, the more severe the problems, as small changes in the bottom hidden layers will be amplified during their propagation through the network, resulting in a considerable shift in the top hidden layers. Batch normalization was introduced in [15], as a solution to the internal covariate shift. The problem is addressed by applying the normalization on the values in hidden units per each mini-batch. The output of a previous activation layer is altered by subtracting the batch mean and dividing by the batch standard deviation. As a result, the unwanted shifts within the network are reduced. Respectively, the stability, speed, and performance of a network are increased. Additionally, higher learning rates may be used without vanishing or exploding gradients. What is more, batch normalization introduces a slight regularization effect, which helps to generalize the model.

# Chapter 4

# Methodology

This chapter presents the methodological approach used within this project. The proposed method, as defined in the description of the project, predicts the distribution of the grain sizes in the Swiss rivers without the detection of single objects. Instead, the histograms are concluded from the overall texture, captured in an image. This task is approached by training a Convolutional Neural Network, employing a set of drone images collected by the hydraulic engineering company Hunziker, Zarn & Partner. Specifically, as the tiles are labeled, the proposed machine learning solution is a supervised learning method. In this project two consecutive designs are presented. The first one is an exploratory testing of the feasibility of histogram predictions by adapting an algorithm HistoNet proposed by [9]. This solution employs a modified ResNet-50 network [16]. The second concept includes creating a novel architecture based on [9], adapted and tuned for the specific problem discussed within this project.

This section is organized as follows: 4.1 introduces preliminary tests performed on HistoNet. Subsequently, novel architecture developed in the course of this project is illustrated in Section 4.2. The network is trained in accordance with the settings and parameters described in Section 4.3. Finally, various evaluation methods used to assess the obtained results are given in Section 4.4, followed by a description of a preprocessing techniques employed on the input data (Section 4.5).

## 4.1 Feasibility testing with HistoNet

This approach aims to predict a global data statistics in the crowded scenes. The proposed solution performs two separate tasks. Firstly, the count of larvae instances in the image is derived, followed by the prediction of the object size distribution via drawing a histogram. This approach includes training a CNN architecture with two separate branches (Figure 4.1). The input layer and the two first hidden layers are shared between them. The architecture used for the histogram prediction is the adaptation of the ResNet-50 and is depicted in the upper branch. The final fully connected layer from the original ResNet architecture is replaced by two convolutional and two fully connected layers, interspersed with two dropouts. To train both branches of the network for the multi-task prediction, loss functions on both tasks are imposed. The loss function defined for the object count is a L1 loss. Prediction of the histogram uses two loss functions having equal impact, namely Kullback-Leibler divergence and weighted L1 loss, where weights are assigned to the center of respective bins. Kullback-Leibler divergence and L1 loss influence the histogram differently, such as both shape and scale of the predicted distribution are captured during training. As the main focus of this project is a histogram prediction, the implemented counting functionality does not play a leading role. Therefore, the training of this branch is omitted by setting the adequate L1 loss to 0.

As all the algorithms were made available directly by the author, this creates an opportunity to adapt and test the source code for testing on the river data set. The code is written in Python and utilizies the Theano library for efficient use of multi-dimensional arrays, which are defined as tensors. Lasagne is chosen as the machine learning library used on the top of Theano to build and train neural networks. The original input to the algorithm are larvae images with a size of 256x256 pixels. For proper utilization of the architecture, the tiles from the river data set are re-sampled from its original rectangular form to squares containing 256x256 pixels. Further on, the final solution presented within the paper predicts the 16-bin histogram, gradually increasing its resolution from 2, through 4 and 8 bins, using Deeply Supervised Nets [17]. For the exploratory prediction on the river data set, the intermediate solution from [9] is used, which omits deep supervision layers. Instead, it employs the adapted ResNet-50 architecture as presented in Figure 4.1, discarding the DSN blocks visible on the top of the network. At last, the size of the model output is adapted from the 16-bin histogram to the 22-bin histogram, representing the histogram of the grain size distribution.

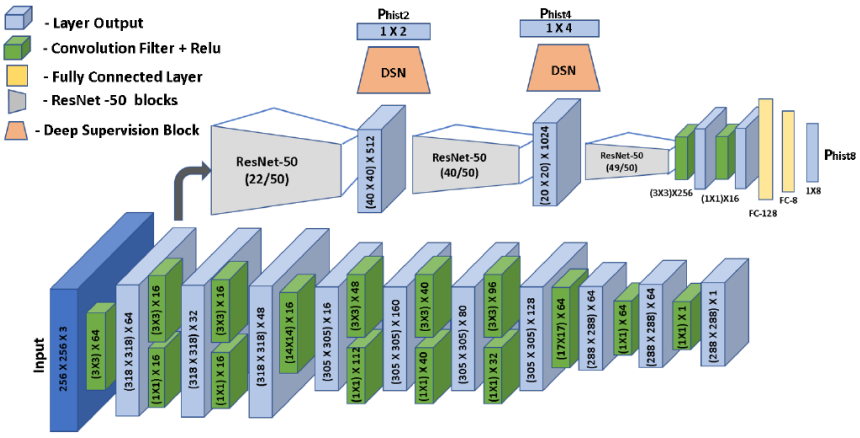


Figure 4.1: Adapted ResNet-50 architecture, used in the HistoNet project [9]

Considering loss functions, 3 different experiments are set up. As already mentioned, L1 loss responsible for the object count is set to 0 in each training session. The remaining loss functions are arranged as following: the first experiment employs the utilization of L1 loss and the second one uses KL divergence. The final trial use both L1 and KL loss functions with equal weights. The goal of those settings is to test the performance of the network depending on the chosen loss function. As expected, training with L1 loss predicts histograms in the correct scale (with gradually increasing values between 0 and 1), however their shape is approximating the overall mean histogram for the data set with minimal adaptations of the individual bins. On the other hand, KL loss produces higher variety of forms, yet the scale is far off the intended range. The results of the third experiment resemble the outcome of the first trial.

## 4.2 Network architecture

The architecture created for the derivation of a grain size distribution adopts the network proposed by [9], based on the ResNet-50 introduced by [16]. This solution is used, as the feasibility testing described in the previous section indicates the potential of that architecture to succeed in predicting histograms. Furthermore, adapting already existing architectures, which are proven to work for similar problems is a common approach while building a CNN model.

The general process behind the architecture design is based on the guidelines listed in [18]. What is more, certain theoretical assumptions, representing state of the art theory related to training CNNs [19, 20], influence the adaptation of the Resnet-50 architecture. Predicting a grain size distribution

Table 4.1: Final architecture. Layer types: CONV = convolution, MAX pool = max pooling, AVG pool = average pooling, FLAT = flatten layer, FC = fully-connected layer. The output of the FC layer corresponds to the number of bins in the target distribution.

| Name | Type | Filters | Kernel Size | Output Dim. |
|---|---|---|---|---|
| input_layer | Input | – | – | $(500, 200, 3)$ |
| conv_1 | CONV | 64 | $3x3$ | $(500, 200, 64)$ |
| max_pooling_1 | MAX pool | – | $3x3$ | $(250, 100, 64)$ |
| res2a_branch2a | CONV | 64 | $1x1$ | $(250, 100, 64)$ |
| res2a_branch2b | CONV | 64 | $3x3$ | $(250, 100, 64)$ |
| res2a_branch2c | CONV | 256 | $1x1$ | $(250, 100, 256)$ |
| avg_pooling | AVG pool | – | $5x5$ | $(50, 20, 256)$ |
| conv_2 | CONV | 256 | $3x3$ | $(50, 20, 256)$ |
| conv_3 | CONV | 16 | $3x3$ | $(50, 20, 16)$ |
| max_pooling_2 | MAX pool | – | $3x3$ | $(25, 10, 16)$ |
| flatten | FLAT | – | – | $(4000)$ |
| fc_histogram | FC | – | – | $(22)$ |

composes a theoretically easy machine learning problem, since not many features are required to get a correct output. Texture represents a main feature of interest and provides an information about the spatial arrangements of colors and intensities in an image. Using mainly this information can potentially lead to a successful prediction. Therefore, the complexity of the problem is relatively low and the depth of the basis ResNet-50 architecture can be greatly reduced. After series of trials and errors, only 1 convolutional block out of 15 modules within the original architecture is kept. As the adapted network is relatively shallow, the skip connection can be potentially removed, however the experiments have shown better results with skip connection preserved within the network.

Table 4.1 provides an overview of the final architecture. The bottom layers of a network corresponds to the ResNet-50. The first layer is a CONV layer with 64 filters, followed by batch normalization, ReLU activation and max pooling. The first convolutional block, as defined in ResNet-50 architecture, comes after. On this stage only 1 parameter, namely the kernel size in the first convolution, is changed. The original model uses a kernel of size 7x7, which is altered to 3x3. As the main feature of interest in the river data set is texture, there is no need to look at the area of 7x7 pixels, which would be more appropriate for detecting blobs. Each CONV layer within the block is followed by batch normalization and ReLU activation. The remaining convolutional and identity modules are skipped, and after the first block the average pooling is applied. On top of that, two CONV layers are added in order to increase the network capacity. The parameters of those CONV layers are set the same as in HistoNet. Next, the spatial extent of the output is further decreased with a max pooling and squished with the flatten layer. The flatten layer is followed by a dropout layer, designed to introduce regularization within the architecture. The final FC layer outputs a vector of 22 elements, which forms the target histogram. Softmax is used as an activation function, such that all outcome values sum up to 1. This way, the resulting histogram represents a proper probability distribution. Thanks to this procedure, the Kullback-Leibler divergence can be properly employed as a loss function. Furthermore, KL divergence is used as a primary loss function in the experiments, as it is proven to work with the given problem. Finally, an ablation study is designed to train the model with 3 additional loss functions: L1, L2 and IoU.

## 4.3 Training

The proposed architecture is trained using a gradient-based optimization method called Adam [21], which is designed specifically for training deep neural networks. Adam is an adaptive learning rate method, which means that individual learning rates are adapted for different parameters. Most of the hyper parameters specified for Adam are kept during training as proposed default values. The

base learning rate is set to $3e-4$, as proposed by [18]. The mini-batch size is set to 32, which is the maximum number still possible to be stored in the memory during training. The single input image has a size of 500x200x3, where numbers correspond to the height, width and number of channels respectively. The three bands of the input tile speaks for red, green and blue (RGB) colors. The output of the network, or so called label, is a vector of size 1x22, which corresponds to a 22-bin histogram, containing information about the grain size distribution in a single labeled tile.

The input data is divided in three sets: training, validation and test set. 96 images are assigned for both test and validation, and all the remaining images become a part of a training set. Before the division, data set is randomly shuffled, to assure that tiles from all rivers basins are spread across the batches. What is more, the input data is preprocessed as specified in Section 4.5. Additionally, data augmentation is introduced. This technique trains the model with additional synthetically modified data in order to increase the variety of training samples and improve the generalization capabilities of the network.

The model is trained with the initial random weights and biases given by the initializers provided in the employed deep learning library. Training process is monitored by observing the loss of the train and validation sets. The weights and biases from the epoch with the lowest validation loss are chosen as final model parameters. For the most of the experiments the loss functions in both training and validation sets tend to flatten out after 150 epochs, hence the optimal training duration of 200 epochs is chosen for the succeeding experiments. Additionally, regularization is imposed by introducing a dropout layer with a 40% dropout ratio to increase the generalization capabilities of the network.

The parameters listed in this and previous subsections are used in all the reported experiments, if not specified differently in the experiment description.

## 4.4 Evaluation strategy

The model is evaluated with diverse methods, comparisons and metrics. A vast range of evaluation methods is used due to the complex assessment of the regression model, which returns a multi-output label per each single input image.

In order to get a better feeling of how certain data characteristics can influence the outcome, the general statistics of the data set are prepared. This analysis comprises a base for a more complete interpretation of the outcome, since it facilitates deeper understanding of the general tendencies and characteristics within the data. Both qualitative and quantitative results may be enhanced due to this evaluation.

The first actual evaluation method is a n-fold cross-validation, described in Section 4.4.1. Subsequently, the model performance is compared to the one of a human labeling. More details are given in Section 4.4.2. This analysis is followed by testing the model's generalization capabilities, as described in Section 4.4.3. Finally, the error case analysis of individual, characteristic tiles is presented (Section 4.4.4).

### 4.4.1 N-fold cross validation

In order to test the overall performance of the model and the influence of random data splits into training and test sets, n-fold cross validation is carried out. To perform a 4-fold cross validation, the data is shuffled and randomly divided into 4 equal parts. Then the model is independently trained 4 times, each time holding out a different set for testing, and training on the remaining three parts merged together, as shown in Figure 4.2

The evaluation metrics are averaged on all folds and represents the overall performance. The respective variances express the impact of the data split. As folds are non-overlapping, each image occurs only once. Hence, cross-validation enables the assessment of the entire reference data set.

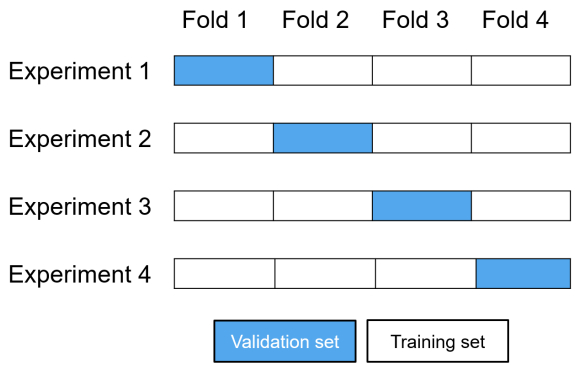


Figure 4.2: 4-fold cross validation

### 4.4.2 Comparison of the model to human performance

Uncertainty can be introduced partially from the labeling method itself. During this process, expert quasi randomly selects and measures the stones along the measurement line. Depending on the person performing this task, different stones would be chosen, what affects the final histogram. Hence, the human labeling is affected by a certain error. On that account the model performance is evaluated in comparison to the performance of a human.

### 4.4.3 Generalization capabilities

The actual application of the trained network depends on its generalization ability, which determines whether a model is effective or not. A network capable of generalizing performs well on unseen images. However, those photographs must have certain degree of similarity to the training set, such that the learned features can be used in order to make the correct prediction.

### 4.4.4 Error case analysis

This analysis includes the inspection of all the predictions in a search for erroneous cases. It enables in-depth comprehension of the reasons behind poor predictions. The repetitive cases of problematic tiles, as well as single dubious images can be pointed out and the adequate solution may be applied.

## 4.5 Input data preprocessing

Preprocessing refers to all transformations on the raw data before it is fed to the learning algorithm. In this project images are transformed with two subsequent procedures. The first one consists of geometric transformations in order to unify input data size and resolution. Moreover, invalid tiles are filtered out and removed from the data set. The second procedure normalizes the set and leads to the improvement of the model's training capabilities. Next, the labels are transformed from their original form of the cumulative histograms to the regular probability distributions. In this way, a single label provides a right supervision signal, which is not biased towards right side, as in the case of cumulative histogram.

### 4.5.1 Geometric transformations and filtering

Multiple tiles are obtained from one single drone data acquisition, covering the whole river basin area. Rectangular tiles of the size 1.25 x 0.5m are cropped out of the large orthophotomap. Each of them is saved in .tiff format and comprises the input format of the preprocessing flow. In the first step, the ground sampling distance of every tile is set to 2.5 mm. After this procedure, the image height varies between 500 and 502 pixels and the width between 200 and 202 pixels. This inaccuracy arise from the prior cropping from the orthophotomap and from differing pixels number

due to rounding effects. Hence, the tiles are cropped to a size of 200 x 500 pixels. Finally, each tile is flipped to be vertical. As the orthophotomap is not rectangular, the tiles obtained from the boundaries may contain areas of white pixels (hence including no information). Images, in which the fraction of white pixels exceed 20% are filtered out from the data set.

### 4.5.2 Data normalization

There exist standard preprocessing techniques for the CNN input data, where raw pixel values are altered. One of them is data normalization, which involves subtracting the mean and dividing by the standard deviation. The mean subtraction has the effect of centering the data around 0. The division by the standard deviation unifies the scale of all image features, such that the standard deviation of the data set equals 1. Any preprocessing statistics must only be computed on the training data and then applied to all sets, namely training, validation, and test sets. Hence, the mean and standard deviation of the preprocessed validation and test data might slightly differ from the target values of 0 and 1. Data normalization results in faster convergence while training the network.

# Chapter 5

# Experiments

This chapter presents the data set, as well as a set of experiments, which were performed upon it. The detailed description of the data can be found in Section 5.1. Subsequently, Section 5.2 introduces the statistics of the entire data set. Later, four performed experiments are described. In Section 5.3, the usage of 4 different loss functions is presented, followed by the results of cross validation in Section 5.4. The performance of the model in comparison with the performance of the human can be found in Section 5.5 and the analysis of the generalization capabilities is introduced in Section 5.6. Finally, the overall results are assessed visually, and some erroneous tiles are pointed out in Section 5.7.

## 5.1 Data set

The river data set is provided by the hydraulic engineering company Hunziker, Zarn & Partner and was collected at 15 various locations of 6 different rivers in northeast regions of Switzerland: Aare, Emme, Grosse Entle, Kleine Emme, Reuss and Rhone. The images were taken between April 2018 and March 2019, using an UAV. Each drone acquisition is completed with the following settings: the flight is performed approximately 10 m above the terrain and with the minimal possible speed. The camera is set vertical to the ground and the images are taken with 80% overlap. The images are georeferenced using ground control points, whose coordinates are measured with a GPS receiver. All photos from a single measurement campaign are patched together into an orthomosaic, visualizing the grain size distribution within a river basin. All information considering a single river basin is saved in one folder with an assigned name, as presented in Table 5.1

Before the actual labeling, further processing is required. Firstly, the orthomosaic is clipped from the left upper corner into the dimension of a multiple of 2.5 m in both width and height. Then, it is divided into squared tiles of size 2.5x2.5 m. Each square image can be further cropped and serve as a source of two final tiles of size 1.25x0.5 m, cut along the middle line either horizontally or vertically. The choice between the horizontal or vertical cut is defined based on the direction of the river flow. Extraction of tiles is presented graphically in Figure 5.1

The rectangle patches of size 1.25x0.5 m are saved into .geotiff format and selected tiles are manually labeled. The labeling process is digitized and performed with an assistance of a computer. It maps the pebble count method, described in the Section 3.1. Labeling is completed by drawing the rim of stones along the longer center line of the tile. The following requirements must be met: the rim should be drawn such that the entire stone is outlined. In the case of occlusions the drawer imagines the continuation of the grain's blueprint. Next, a rectangle is fit into each shape. Based on the b-axis of all the rectangles in the image, the grain size distribution is derived according to the procedure described in [10]. There are multiple methods for histogram derivation and its further empirical adaptations, however they will not be described within the scope of this report.

Table 5.1: Presentation of the collected data set

| River name | Folder name | Acquisition date | Number of labeled tiles |
|---|---|---|---|
| Aare | Aare | 20.04.2018 | 114 |
| Aare | Aare_1 | 25.10.2018 | 32 |
| Aare | Aare_Bern3 | 25.10.2018 | 60 |
| Aare | Aare_Bern4 | 25.10.2018 | 42 |
| Aare | Aare_MB1 | 13.02.2019 | 5 |
| Aare | Aare_NR1 | 13.02.2019 | 12 |
| Emme | Emme_Altisberg | 29.06.2018 | 116 |
| Emme | Emme_Biberist | 29.06.2018 | 190 |
| Emme | Emmebirre | 29.06.2018 | 78 |
| Grosse Entle | GrosseEntle1 | 11.07.2018 | 76 |
| Grosse Entle | GrosseEntle2 | 11.07.2018 | 90 |
| Kleine Emme | KLEmme_Hasle | 11.07.2018 | 212 |
| Reuss | Reuss-FIGO | 21.02.2019 | 30 |
| Rhone | Rhone_Brigerbad | 23.12.2018 | 106 |
| Rhone | Rhone_Praz | 04.03.2019 | 74 |

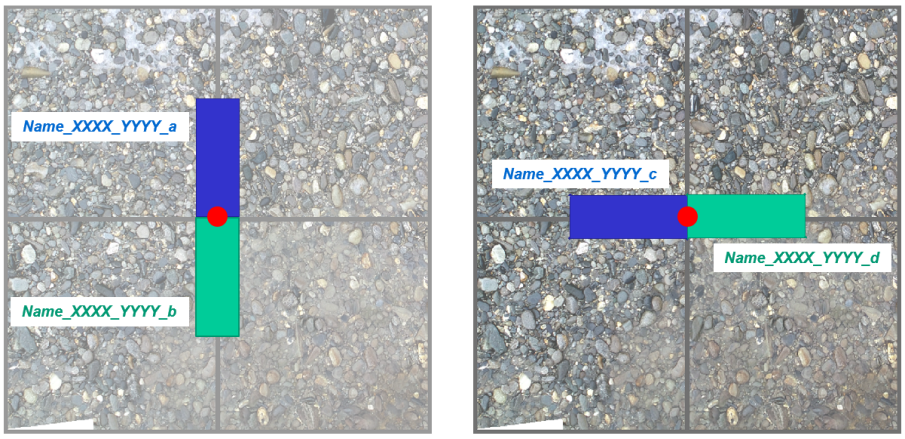


Figure 5.1: Final tiles derivation from the orthophotomap

The label for each tile is given in a form of 22-bin cumulative histogram. The values within bins are in range between 0 and 1, as a distribution is described. The sizes of the bins are not dispersed linearly. The higher limits of all the bins are as following (defined in meters): 0.01, 0.02, 0.03, 0.04, 0.06, 0.08, 0.10, 0.12, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.50, 0.60, 0.80, 1.00, 1.20, 1.50, 2.00. The final outcome of the data acquisition process are 1237 labeled tiles.

## 5.2 Overall data statistics

Firstly, mean histogram for each river basin is drawn. There are two major shapes which are distinguished from all the results. One, manifested by the majority of the rivers, is described as follows. The range of grain sizes covers diameters between 1 and 25 cm. A large section of the stones (above 30%) are small ones with a diameter within a range of 1-2 cm. The contribution of the bigger grains is smaller and decrease exponentially with increasing grain size until reaching a 25 cm borderline, as visualized in Figure 5.2. The following rivers depict this scheme: Aare, Emme, and Reuss and constitute 10 out of 15 locations in which the data was collected. Additionally, 1 out of 2 locations along Rhone (Rhone_Brigerbad) corresponds to the first scheme as well, resulting

in total 11 out of 15 river basins having similar characteristics. For this group, the average value of the mean diameter falls in the range of 2.87 cm to 5.23 cm, with the mean of 3.92 cm and a standard deviation of 1.04 cm. Hence, the mean diameter is rather small with narrow standard deviation, what is in accordance with the described shape of the histogram.

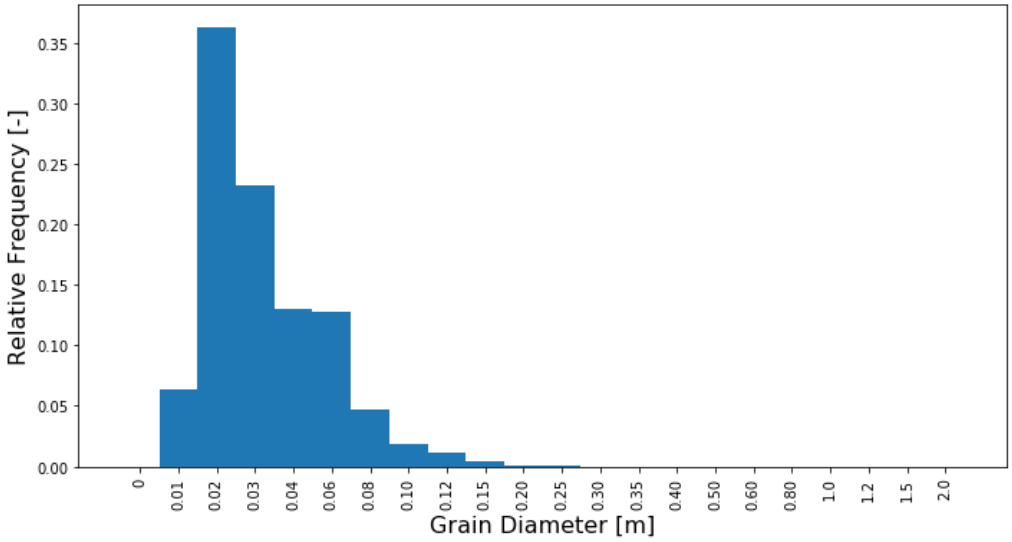


Figure 5.2: Exemplary histogram shape of the river belonging to the first type

Different characteristics are observed for the remaining rivers: Grosse Entle, Kleine Emme, as well as a single location at Rhone (Rhone_Praz), which together constitute 4 out of 15 river basins. For those rivers the stones are generally of larger sizes than in the previous scheme, what results in the histogram shifted towards the right side, as seen in Figure 5.3. Since bigger grains are also present, the range of values is between 1 and 60 cm. Also, stones of sizes up to 8 cm constitute significant parts in the distribution.

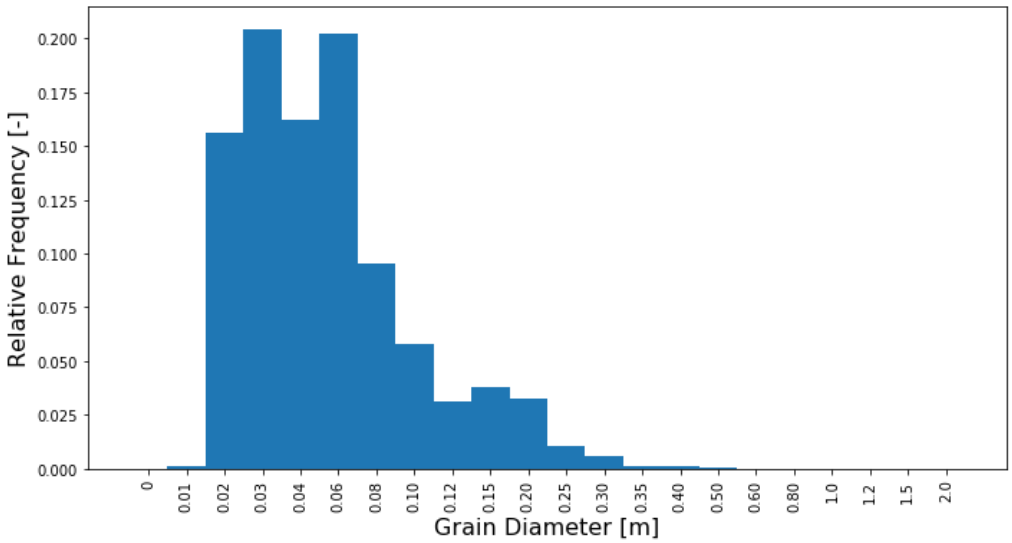


Figure 5.3: Exemplary histogram shape of the river belonging to the second type

For this group, the average value of the mean diameter falls into values between 8.79 cm and 13.36 cm, with the mean of 10.99 cm and a standard deviation of 3.39 cm. Thus, the average

mean diameter is not only almost 3 times larger than in the first scheme, but also shows higher variations.

Finally, the distribution of the mean diameter along all tiles is shown in Figure 5.4. For the majority of rivers, the mean diameter is below 6 cm, and the fraction decreases with the increasing value of dm. There are only single images of the mean diameter larger than 15 cm present in the data set. The red line shows the average mean diameter of the entire data set.

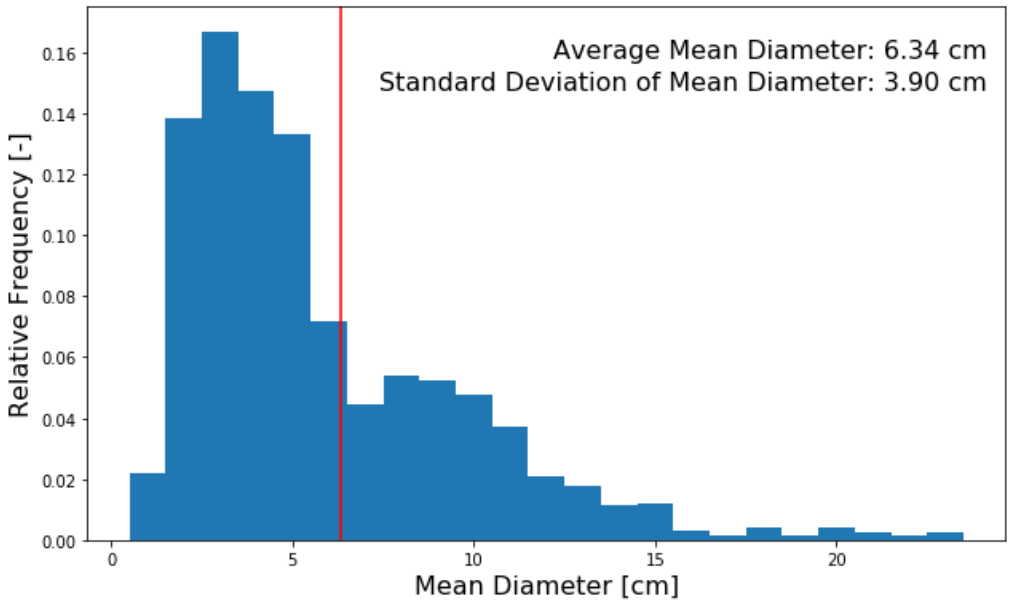


Figure 5.4: Overall dm distribution in all tiles

## 5.3 Ablation study: losses

Within this project 4 different loss functions are tested. The chosen losses are: Kullback-Leibler divergence, Intersection over Union, Mean Absolute Error and Mean Squared Error, and the results are presented in Table 5.2. All the hyper parameters of the model are set as described in Section 4.3. However, the dropout rate is changed from 0.4 to 0.2, since when training the model with the MSE loss, the larger dropout prevents the model from training. For proper comparison based only on the chosen loss function, the dropout ratio is set to 0.2 in all the tests. For additional comparison, the default training model with KL divergence loss and dropout of 40% is added in the last row of table.

Table 5.2: Results of the ablation study, featuring 4 different losses.

| Loss function | KLD | IoU | MAE | MSE | Average $\Delta$ dm [cm] | Dropout rate |
|---|---|---|---|---|---|---|
| KLD | 0.1461 | -0.6992 | 0.0167 | 0.0016 | 1.20 | 0.2 |
| IoU | 0.1512 | -0.7112 | 0.0159 | 0.0014 | 1.58 | 0.2 |
| MAE | 0.1513 | -0.7098 | 0.0160 | 0.0014 | 1.32 | 0.2 |
| MSE | 0.1590 | -0.6841 | 0.0176 | 0.0015 | 1.38 | 0.2 |
| KLD | 0.1308 | -0.7215 | 0.0153 | 0.0013 | 1.60 | 0.4 |

For an additional visual comparison one tile is arbitrarily selected from the test set. The grain size distribution of the chosen image is predicted 4 times, each time by the model trained with a

different loss function. The comparison is presented in the Figure 5.5. Resulting predictions are almost the same, regardless of the chosen loss function.

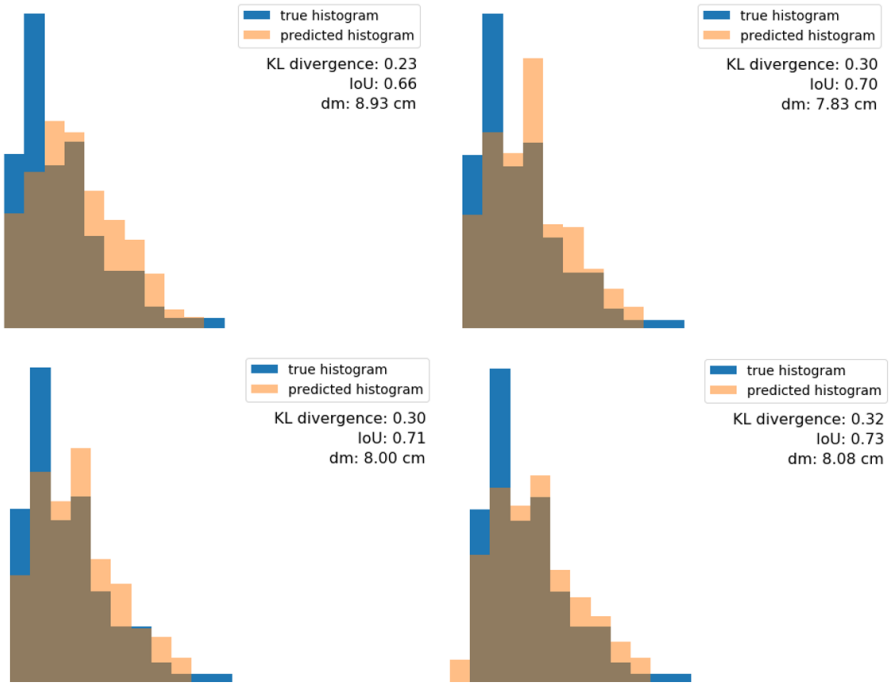


Figure 5.5: Visual comparison between predictions from models trained using different loss functions. Top-left: IoU, top-right: KL, bottom-left: MAE, bottom-right: MSE

## 5.4 4-fold cross validation

For the 4-fold cross validation, the model is trained 4 times, each time holding out different randomly selected part of the data. The hyperparameters are in accordance with the settings described in Section 4.3. Even though KL divergence is used a loss function, the IOU, MAE and MSE are computed as additional metrics. Additionally, the average of the absolute differences between predicted and true mean diameters is computed. Table 5.3 presents the outcome of the cross validation.

Table 5.3: Results of the 4-fold cross validation

| Fold | KLD | IoU | MAE | MSE | Average Δ dm [cm] | Std Δ dm [cm] |
|---|---|---|---|---|---|---|
| Fold 1 | 0.1269 | -0.7150 | 0.0155 | 0.0012 | 1.39 | 1.73 |
| Fold 2 | 0.1356 | -0.7068 | 0.0160 | 0.0014 | 1.54 | 2.10 |
| Fold 3 | 0.1468 | -0.6904 | 0.0171 | 0.0015 | 1.25 | 1.90 |
| Fold 4 | 0.1455 | -0.6949 | 0.0168 | 0.0014 | 1.41 | 2.25 |
| Average value | 0.1387 | -0.7018 | 0.0164 | 0.0014 | 1.40 | 1.99 |
| Standard deviation | 0.0093 | 0.0112 | 0.0007 | 0.0001 | 0.12 | 0.23 |

## 5.5 Comparison of the model to human performance

In this experiment, the uncertainty of the model is compared to the uncertainty introduced by human labeling. For this purpose, 5 employees of Hunziker, Zarn & Partner agreed to label 6 various tiles independently from each other. To obtain the prediction, selected 6 tiles are placed in

the test set and all the remaining images are shuffled and divided into training, test and validation sets in the usual manner. The results are put together and assessed quantitatively and qualitatively.

### 5.5.1 Quantitative assessment

The quantitative assessment is completed based on the comparison of the resulting mean diameter value.

Table 5.4: Mean diameter values for 6 different tiles, based on labels made by 5 people and the model prediction

| Tile name/Labeling source | P1 | P2 | P3 | P4 | P5 | Prediction | Mean value [cm] | Std. dev. [cm] |
|---|---|---|---|---|---|---|---|---|
| Aare_Bern3_0000_0001_b | 3.04 | 3.18 | 3.10 | 2.97 | 3.12 | 2.68 | 3.02 | 0.08 |
| GrosseEntle1_0005_0006_b | 6.80 | - | 6.25 | 6.22 | 7.28 | 7.75 | 6.64 | 0.51 |
| KLEmme_Hasle_0004_0016_b | 7.96 | 6.35 | 6.63 | 6.96 | 6.38 | 6.16 | 6.86 | 0.66 |
| Rhone_Brigerbad_0005_0002_c | 6.86 | 5.92 | 5.80 | 7.80 | 5.87 | 3.80 | 6.45 | 0.87 |
| Rhone_Brigerbad_0009_0009_c | 4.10 | 4.80 | 3.79 | 5.32 | 5.07 | 3.64 | 4.62 | 0.65 |
| Rhone_Brigerbad_0021_0013_d | 2.74 | 2.89 | 2.66 | 2.74 | 2.92 | 2.32 | 2.79 | 0.11 |

In Table 5.4, the values of mean diameter for all tiles and all experts are listed. The mean value calculated for the prediction is placed in the last column. All values are expressed in centimeters. The expected value of the mean diameter is computed as the average value from all expert measurements. The standard deviation for each tile is calculated and the tolerance is set as mean value $\pm 2$ standard deviations. Only 2 values in the prediction column fit within a given range.

Table 5.5: The average absolute error of the mean diameter values for 6 different tiles, based on labels made by 5 people and the model prediction

| Tile name/Labeling source | Person 1 | Person 2 | Person 3 | Person 4 | Person 5 | Prediction |
|---|---|---|---|---|---|---|
| Aare_Bern3_0000_0001_b | 0.02 | 0.16 | 0.08 | −0.04 | 0.11 | −0.34 |
| GrosseEntle1_0005_0006_b | 0.16 | - | −0.39 | −0.41 | 0.65 | 1.11 |
| KLEmme_Hasle_0004_0016_b | 1.10 | −0.51 | −0.23 | 0.11 | −0.48 | −0.70 |
| Rhone_Brigerbad_0005_0002_c | 0.41 | −0.53 | −0.65 | 1.35 | −0.58 | −2.65 |
| Rhone_Brigerbad_0009_0009_c | −0.51 | 0.19 | −0.82 | 0.70 | 0.45 | −0.98 |
| Rhone_Brigerbad_0021_0013_d | −0.05 | 0.10 | −0.13 | −0.05 | 0.13 | −0.47 |
| Average absolute error | 0.38 | 0.25 | 0.38 | 0.44 | 0.40 | 1.04 |

Further, the error for each measurement is calculated by subtracting the mean value. The results are presented in Table 5.5. After averaging all the errors, it is seen that the average absolute error for a human does not exceed 0.5 cm. On the other hand, the resulting average error for the model prediction equals approximately 1 cm. What is more, the majority of the prediction errors are negative.

### 5.5.2 Qualitative assessment

In this part the visual difference between grain selection is assessed. Figure 5.6 shows the digital labeling done by 2 experts on the same tile.

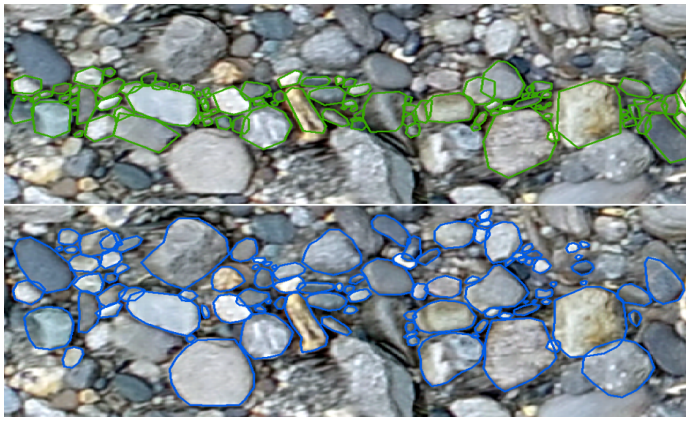

Figure 5.6: Comparison of digital labeling performed by 2 different experts

The person working on the upper tile tries to choose grains located as close as possible to the middle line. On the other hand, the stones labeled in the lower tile cover bigger area of the image.

Further visual evaluation encompasses comparison of the obtained distributions. Histograms based on the labeling of 5 experts are drawn next to the histogram predicted by the model. Visualizations for 2 tiles are shown below in the Figure 5.7 and Figure 5.8.

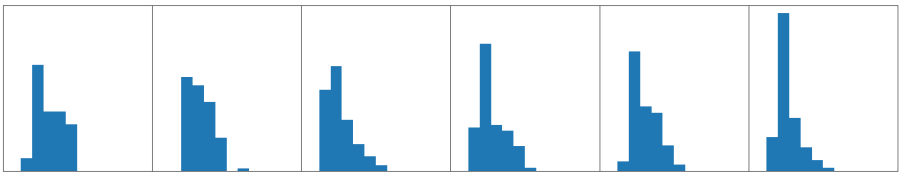

Figure 5.7: Comparison of histograms created for Aare_Bern3_0000_0001b tile by 5 experts and the model. The last histogram on the right is a model prediction.

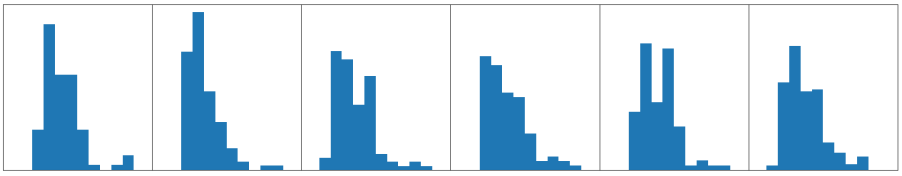

Figure 5.8: Comparison of histograms created for KLEmme_Hasle_0004_0016_b tile by 5 experts and the model. The last histogram on the right is a model prediction.

## 5.6 Generalization capabilities

To analyze the generalization capability of the network it is trained on 14 out of 15 of the river banks, and tested on the one remaining. This process is repeated 15 times to obtain the good summary over the entire data set. The results are firstly evaluated quantitatively, which is followed by the qualitative assessment of the potentially interesting images.

### 5.6.1 Quantitative assessment

Firstly, the KL loss for each river bank is plotted on a graph, as seen in Figure 5.9. The red line is drawn on the level of the mean loss resulting from the cross validation and represents the averaged

performance of the entire data set. The majority of loss values are located slightly above the red line. Exceptionally high values can be noticed for basins Aare_1, Aare_MB1 and GrosseEntle1.

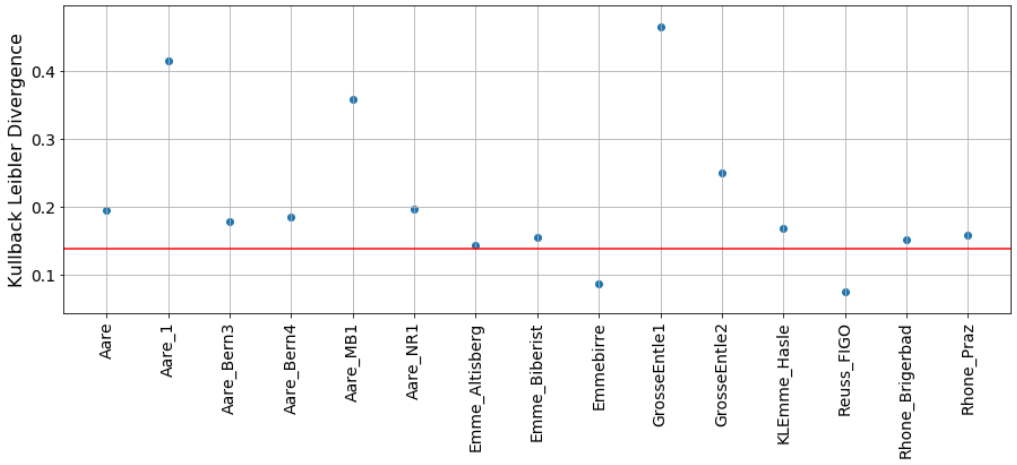


Figure 5.9: Kullback-Leibler divergence of individual rivers basins resulting from the generalization testing

Second qualitative evaluation involves the assessment of the resulting mean diameter. The results are shown in Figure 5.10. To that end, the average absolute difference of mean diameters per single water basin is computed. Resulting values are plotted on the graph together with their standard deviations. The majority of the results circles around the 1 cm difference, which is comparable to the value of 1.40 cm emerging from the cross validation. Relatively high values are attained for the following rivers: GrosseEntle1, GrosseEntle2, KLEmme_Hasle and Rhone_Praz.

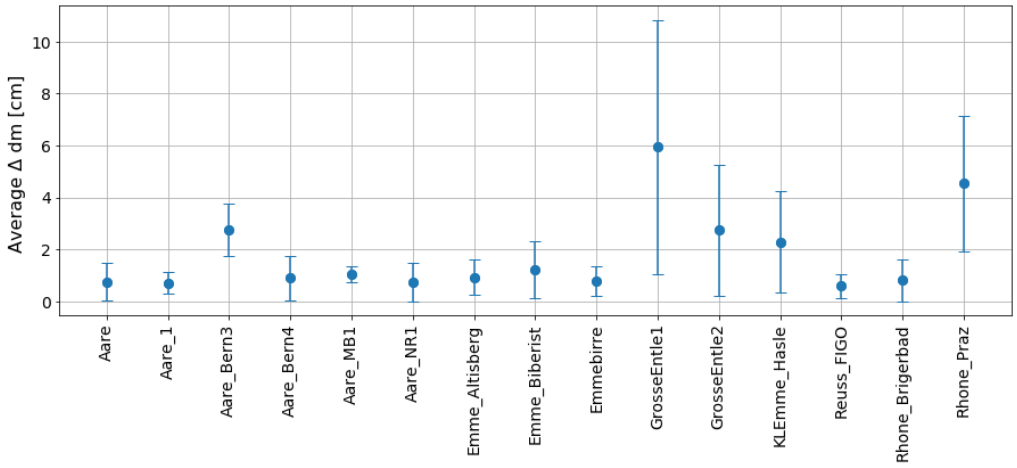


Figure 5.10: Average difference of the mean diameter of individual rivers basins resulting from the generalization testing

### 5.6.2 Qualitative assessment

In this subsection, individual river basins are described along with the chosen examples of tiles. The evaluation includes a simple visual examination of the results, compared to the ground truth. Each selected image has some distinctive characteristics, which results in unexpected prediction far

from the ground truth. Figure 5.11 shows a tile from the Aare_1 folder, for which a KL divergence is high.

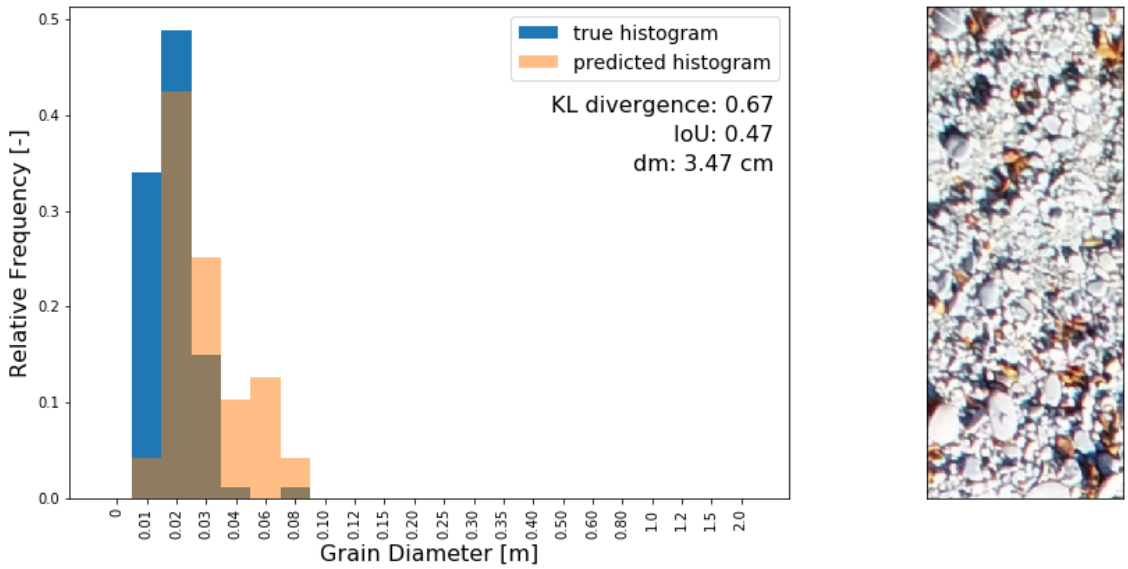


Figure 5.11: Exemplary prediction from the generalization testing: Aare 1. Tile: Aare_1_0001_0014_b

As can be noticed, heaps of orange leaves are present in the image. Although the overall distribution looks roughly adequate, the sizes between 3 and 8 cm are overestimated, and the smallest ones with diameter below 1 cm are underestimated.

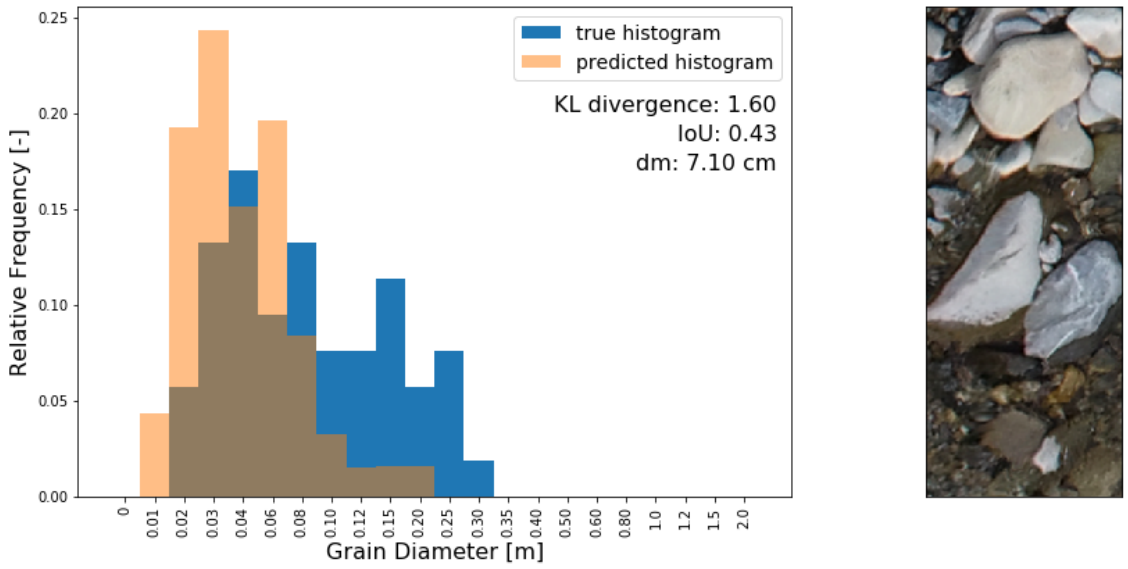


Figure 5.12: Exemplary prediction from the generalization testing: GrosseEntle1. Tile: GrosseEntle1_0002_0005_a

Figure 5.12 introduces next group of images, for which the outcome is questionable, notably tiles with grains partially hidden under water surface, as it is a case within Grosse_Entle_1 river basin.

For this folder, both KL divergence and average dm values from the quantitative analysis are worse than for other basins. As can be seen in the Figure 5.12, the prediction is rather misguided. For comparison, the resulting prediction for the same tile generated during the cross validation is presented in Figure 5.13.

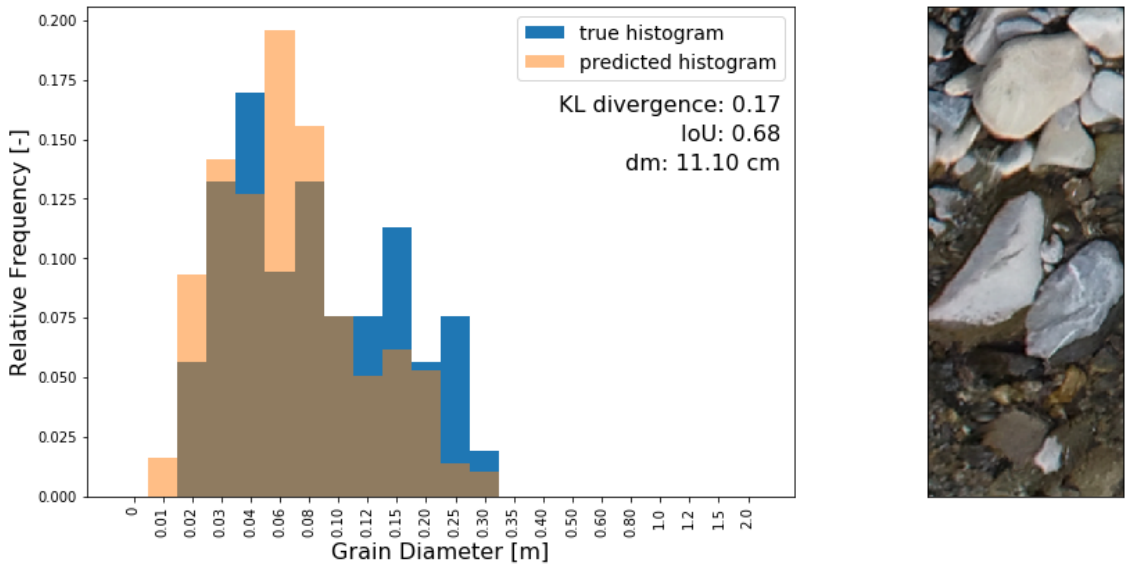


Figure 5.13: Exemplary prediction from the cross validation: GrosseEntle1. Tile: GrosseEntle1_0002_0005_a

## 5.7 Visual assessment and error case analysis

This section presents visual examples of the accurate, as well as the misguided predictions. Firstly, some tiles delivering the proper results are shown. The next part deals with the error case analysis. The tiles which result in misguided prediction repeatedly in different experiments are shown and described.

### 5.7.1 Successful predictions

The majority of tiles results in satisfying visual outcome. To give the reader a feeling about what type of outcome is qualitatively evaluated as good, some examples assessed as good predictions are available in the Appendix A. All the selected results come from the cross validation experiment, described in Section 5.4.

### 5.7.2 Error case analysis

In this part, the tiles resulting in misguided predictions are shown and described. Firstly, the categories of images resulting in misguided predictions are introduced. Within this context, category is understood as a group of tiles, which share similar characteristics and all fail to output a correct prediction. The representatives of 3 separate categories are shown in Figure 5.14.

The first tile on the left represents the case in which single stones with significantly large diameter are present. The middle image represents the group of tiles with the majority of the area covered by sand. The third image is an example of a tile containing a high variety of grain sizes with relatively large size, however with no dominant diameter.

Secondly, images shown in Figure 5.15 represent gross errors, found within the data. In this context, the gross errors are understood as images corrupted with artifacts, which preclude obtaining a

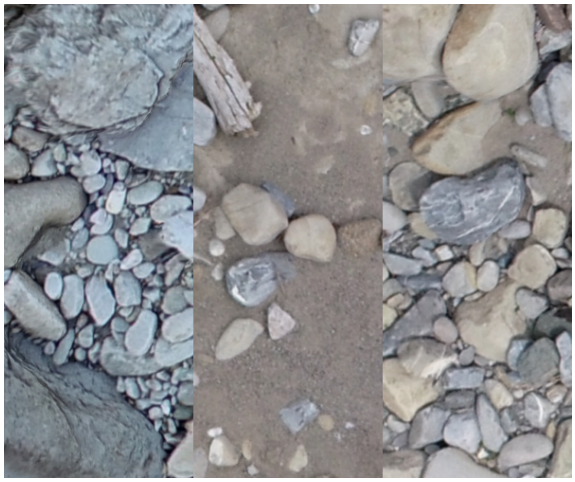

Figure 5.14: Representatives of the problematic image categories

correct prediction. Three types of disrupted images are found. The gross errors include tiles lacking proper illumination, as shown in the first image from the left. The grains are not visible and therefore cannot be recognized. The middle tile contains white pixels enclosing no information, which were cropped from the boundary of the orthophotomap. Finally, the third tile is disturbed by a blur. Thus, the real grain sizes are distorted and as a result the overall histogram does not depict the actual grain size distribution.

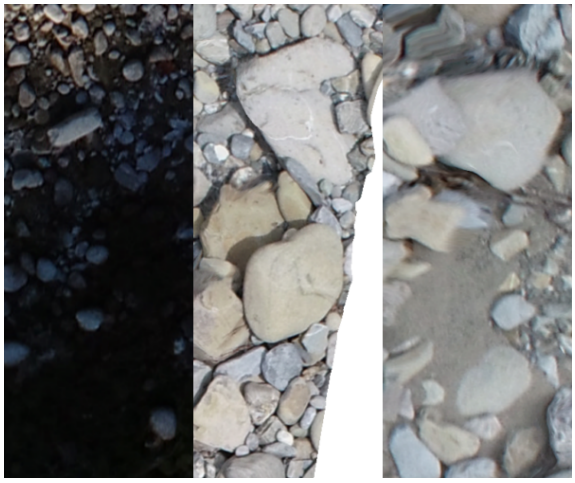

Figure 5.15: Representatives of the gross errors

# Chapter 6

# Discussion

In this project, ResNet-50 architecture is adapted to predict a grain size distribution from the images of the gravel bars along rivers. This model is chosen as a building basis, since the feasibility testing with HistoNet confirms that the network is able to learn given river data set samples, and there is a potential of adapting this solution wile predicting the grain size distribution.

The comparison between models trained with 4 different losses is given in Section 6.1. Multiple methods are employed to test the trained model quality and all the obtained results are discussed within the remaining sections.

## 6.1 Loss functions

Within this project, 4 different loss functions were tested: Kullback-Leibler divergence, Intersection over Union, Mean Absolute Error and Mean Squared Error. Surprisingly, neither quantitative nor qualitative comparison shows significant differences within the predictions. What is more, low resulting loss, as well as average difference of dm for each trained model indicates that all selected loss functions are capable of returning a correct histogram. This statement is confirmed by the visual comparison, as the distributions predicted for each loss function are almost identical. Hence, it is inferred that all chosen loss functions are a good choice for the problem being solved within this project.

## 6.2 Statistics

Based on the calculated overall statistics it is concluded that all the river basins are divided into two separate types. The categories are created based on the form of the histogram, as well as the value and standard deviation of the average mean diameter. In this report the categories are informally referred to as the first and second type.The first one, including 11 out of 15 river basins and around 64% ot the tiles, is characterized by smaller grains with less variety of size. It generally results in better predictions, as the distributions are more uniform and there is less variation in the dm values. The second type encompasses 4 out of 15 locations, what results in around 36% of the overall images. Those tiles contains stones of both small and large diameters, falling into bigger range of values. The high variety of size is evident and leads to larger average mean diameter with broad standard deviation. Within this category, the tiles differ much more from each other, even when descending from the same river bank. Respectively, the histograms and dm values present high variation, what results in labels more complex to predict and in general, less accurate obtained predictions. The reason behind this is the complexity of the grain size distribution, as well as possession of less training samples of this type (36% of tiles in comparison with the 64%

of the first type). The prediction for the second type of tiles theoretically should be improved by introducing more images of this type into the training data set.

## 6.3 Qualitative assessment

Based on the visual judgement, the vast majority is assessed as successful predictions, as the general histogram shape and its main characteristics are mapped correctly. What is more, this opinion is shared by the domain expert after visual inspection of the results.

On the other hand, general drawbacks of the model related to the data set might be acknowledged based on this type of analysis. =

Visual examination allows also to pinpoint the gross errors among the tiles. Three types of disrupted images are found, and in each case the artifacts are caused during the acquisition process. The gross errors include images with reduced illumination, blur or containing white pixels with no information within. All of the above may end up in unreliable predictions and should be permanently excluded from the data set. This way no false information, which can disturb either the training or the prediction is feed into the network. Improved performance for the regular samples is expected, if the fraction of the gross errors deleted from the data set is large.

Just by looking at the results, certain problematic characteristics of images are indicated. In general, presence of single or multiple stones with significantly large diameter within the tile or of large areas covered with sand, worsen the prediction. The reason behind this may be very few large grain sizes present in the data set. It might be solved by adding more training data including bigger stones. What is more, the majority of those type of images are present within the river basins with the second type of grain size distribution, what confirms the conclusions drawn from the statistical analysis of all tiles.

What is also noticed, are the tiles with stones partially hidden under the water surface. Firstly, correct labeling procedure for this type of images seems to be vague. When trying to map the exact pebble count method, the rim of the entire stone should be drawn, even in the situation when part of the stone is hidden under water. The reason behind this is as follows: if the measurement took place in the field, the grain would be simply taken out of the river and measured. However, digitization of the labeling process leads to the situation, in which the intersection of the grain with the water surface alters the image texture and introduces false information within a tile. Additionally, it is important to mention that the appearance of objects below water is distorted due to the rays being refracted on the water surface and hence the real sizes of objects do not correspond to the sizes visible on the image. For those 2 reasons, tiles including stones hidden under water should be avoided if possible.

## 6.4 Cross validation and generalization capabilities

Apart from simple visual analysis, various experiments with numeric results are performed. The first one is a cross validation. As is deduced from the results, the influence of the data split is noticeable, and may depend on the the distribution of the erroneous images within the train, validation and test data sets. However, neither relevant differences between 4 experiments nor any outstanding characteristics of a single experiment are present. Hence, the average values can be considered as representative for the entire reference data set.

Another experiment tests the generalization capability of the network. As expected, the majority of loss values were slightly higher than the average loss obtained in the cross validation. This behavior is perfectly explainable, since images from the new river basins have their own individual characteristics, but the overall resemblance of tiles is preserved. However, 3 river basins, namely Aare_1, Aare_MB1 and GrosseEntle1, result in unnaturally high losses.

Visual examination of Aare_1 river basin allows for a discovery of this folder's unique characteristic, namely heaps of orange leaves, which are noticed in the majority of images. As leaves are not to be found in any other river basin, the network does not have a chance to decipher this information during the training and faces problems to generalize properly. Here, the piles of leaves may be presumed to be bigger grains of undeniably discernible texture, and predicted as such in the distribution, what leads to larger error and also a higher value of KL divergence. However, the overall characteristics of this river basin are preserved and predicted quite well, as the resulting average difference of dm fall within a tolerable range. Yet, the full comprehension of the model prediction is not possible in the case of CNN and no image characteristics can be blamed for certain. However, elements which are not seen by the network during training have an excessive potential of resulting in worse predictions.

Next, the Grosse_Entle_1 folder is tested. It contains the majority of images including water within the entire data set. Therefore, issues related to the previously unseen features may occur, as in Aare_1. Indeed, as shown in Section 5.6.2, the prediction of this tile in the generalization experiment is rather misguided. To reassure that the problem lays in the model generalization capabilities, the result is compared with the prediction of the same tile obtained during the cross validation. In that case the network has a chance to see multiple images with grains located under water during the training process. The prediction is undoubtedly better, what confirms the previous conclusion.

For further quantitative assessment of generalization capabilities, the average absolute difference of mean diameters per single water basin is computed. Relatively high values are attained for the following rivers: GrosseEntle1, GrosseEntle2, KLEmme_Hasle and Rhone_Praz. To understand the possible reasoning behind the large dm difference, it is crucial to look at the overall data set statistics. All four river basins constitute the second type of grain size distribution. Those rivers are characterized by higher variety of grain sizes, as well as diversity of the resulting mean diameters. Hence the larger error of the model predictions is to be expected.

In conclusion, it is inferred that the model is capable of generalization and correct prediction in the case of newly introduced, previously unseen data set. However, special focus should be put on the unique tiles characteristics or features, which might affect the outcome negatively.

## 6.5 Comparison to human performance

Within that experiment the performance of the model is tested in comparison with the performance of the human. Although human labeling introduces extensive uncertainty, a human still performs better than the model. This cannot be concluded from the visual assessment, but is proven in the quantitative analysis. It is concluded accordingly, that the general shape of the predicted histogram is definitely close to expected, however there is not enough precision in the prediction. Hence, there is a place for further improvements of the model. Furthermore, the majority of the prediction errors are negative and hence the model may underestimate the contribution of the grains with bigger diameters.

Although this analysis provides certain feeling about the results, more samples should be tested for better statistical quality.

Even though the grain size distribution predicted by the model is less accurate than if labeled by an expert, the solution presented in this project has some features, which can overrun manual labeling process. Firstly, the results are scalable, such that grain size distribution in high resolution is easily available. Secondly, working time and manual labor are greatly limited. Ultimately, those advantages may turn out to be more fundamental than the certain accuracy of the prediction.

# Chapter 7

# Conclusion

This project tackles the problem of direct estimation of the grain size distribution form the drone images, using supervised deep learning with regression. The data comprises 1237 tiles, obtained from 15 individual river basins of Swiss rivers. A Convolutional Neural Network architecture is created by adapting the state-of-the-art ResNet-50 model. The input of the network is a tile od size 1.25x0.5 m, depicting the gravel bars along rivers, and the output is a 22-bin histogram, representing the grain size distribution within that tile. The outcome is evaluated qualitatively and quantitatively in several different experiments. For the qualitative evaluation, the values of the loss and mean diameter of grain sizes within one tile are used. The overall assessment of the results indicates that the project is recognized as a successful. Further more, the results from various experiments tend to be mutually implicated, what increases their reliability.

The performance of the network depends on the characteristics of the image. Although the majority of tiles gives reasonable predictions, the results obtained for the others are not so reliable. The challenging types of tiles are described within the scope of this report. That issue is potentially resolvable by adding more training data, representing the more demanding examples. Furthermore, some images are pointed out as gross errors due to the artifacts present within. Those should be removed from the data set to further improve both the training process and the resulting predictions.

The generalization capabilities of the model are also tested, by predicting the distribution of the individual river basins, previously unseen by the network. Primarily, the performance is slightly worse that the one of the model trained on the entire data set, which is randomly shuffled into training, validation and test set. This result is to be expected, as even though the river basins are similar, each of them is characterized by its own traits. In the presence of the elements unseen during the training, such as orange autumn leaves present only in the single river basin, the performance is further decreased. This behavior is rather easy to justify. Since the model has never seen those new features, it does not have an opportunity to learn how to interpret them.

Finally, it is crucial to mention that the performance of the model is still worse than of the expert, doing the manual the labeling. This leaves a room for further improvements of the proposed solution. Some ideas of how to advance the existing model are listed in the outlook. However, even though the model outcomes are less accurate, they scale much better than traditional methods for derivation of a grain size distribution. With the approach proposed in this project, multiple tiles from large spatial areas can be assessed within seconds, what is not achievable with a manual work.

To sum up, the proposed solution can provide novel capabilities, which have a great potential to replace the current methods of estimating the grain size distribution. Deriving histograms from the drone images would not only help to save time and money, while acquiring the information,

but also can provide results in scales and on the level of generalization, which are not achievable with the ongoing manual approach.

# Chapter 8

# Outlook

The results achieved within this project look very promising, however there are several ideas with which the outcome of this project can be improved.

Firstly, the architecture itself could be adapted to allow the model to generalize better. The visualization of the activation maps proves that some filters tends to learn the same features, which means that there are too many parameters within the model. Also, there is still a big discrepancy between the training and validation loss. This causes a risk of over fitting to the training examples. Further methods to prevent over fitting could be also tested, such as adding regularizes within the model.

Next, most of the hyper parameters are set to their default values, and their best configuration is not searched for. Different trials with grid or random search may reveal better combinations of parameters, leading to better results.

Although 4 different losses are tested, there exist a potential improvement in creating a custom loss as a combination of multiple losses, as it is done in a successful HistoNet project [9].

Finally, there is more data collected by the Hunziker, Zarn &Partner, which is not used and hence not described within the scope of this project. The examples include the digital terrain model or characteristic diameters, giving the grain diameter for predefined percentiles. This information has a potential to be used and hence improve the overall performance.

# Appendix A

# Successful predictions

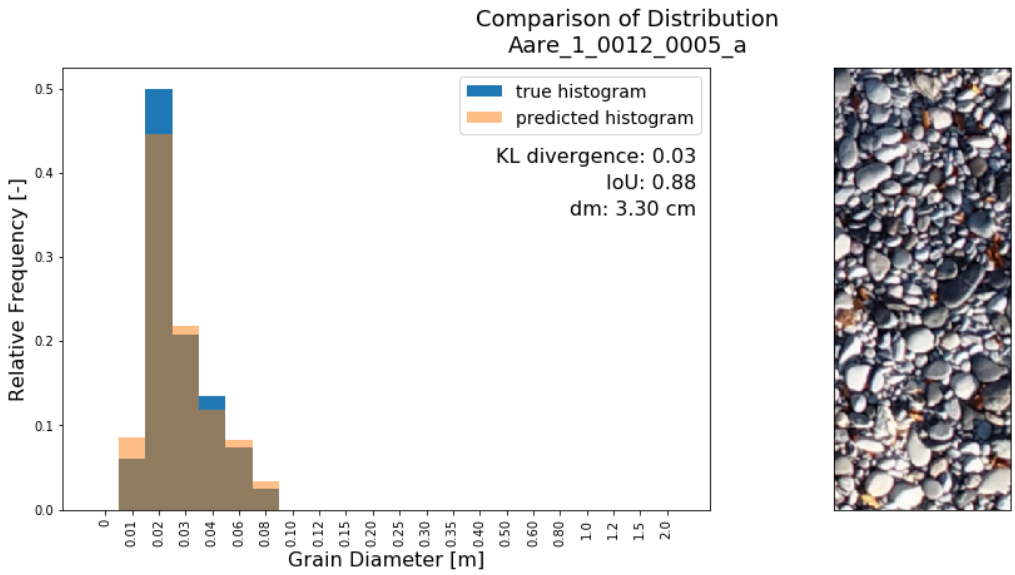


Figure A.1: Prediction close to perfect with resulting KL divergence: 0.03, IoU: 0.88

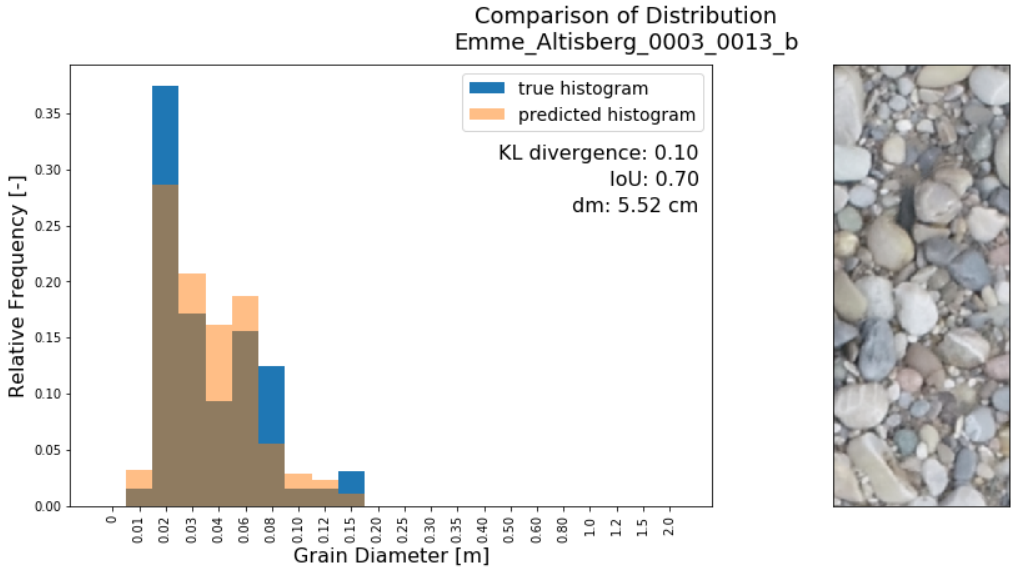


Figure A.2: Good prediction of more complex shape with resulting KL divergence: 0.10, IoU: 0.70

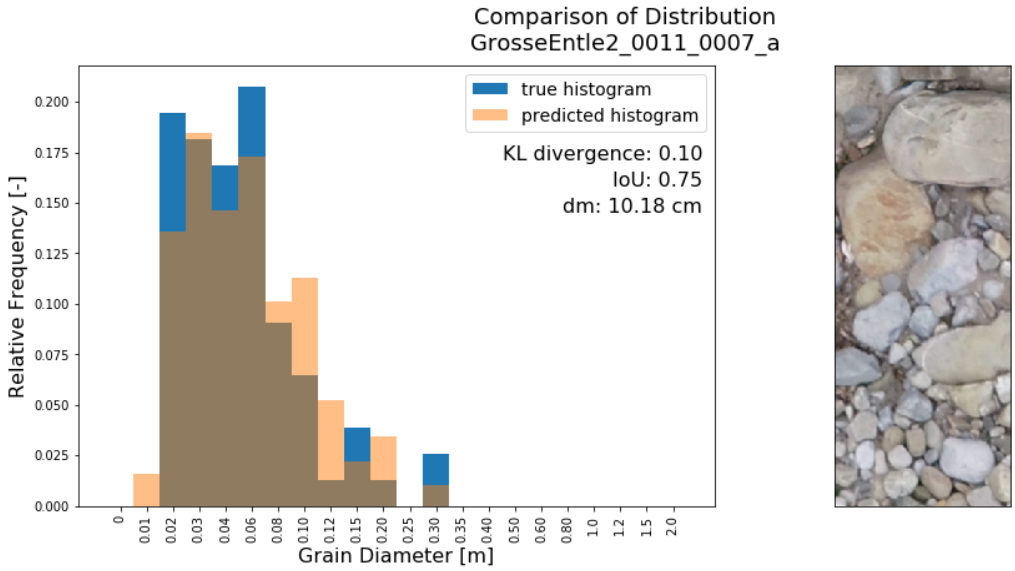


Figure A.3: Good prediction for a difficult tile with resulting KL divergence: 0.10, IoU: 0.75

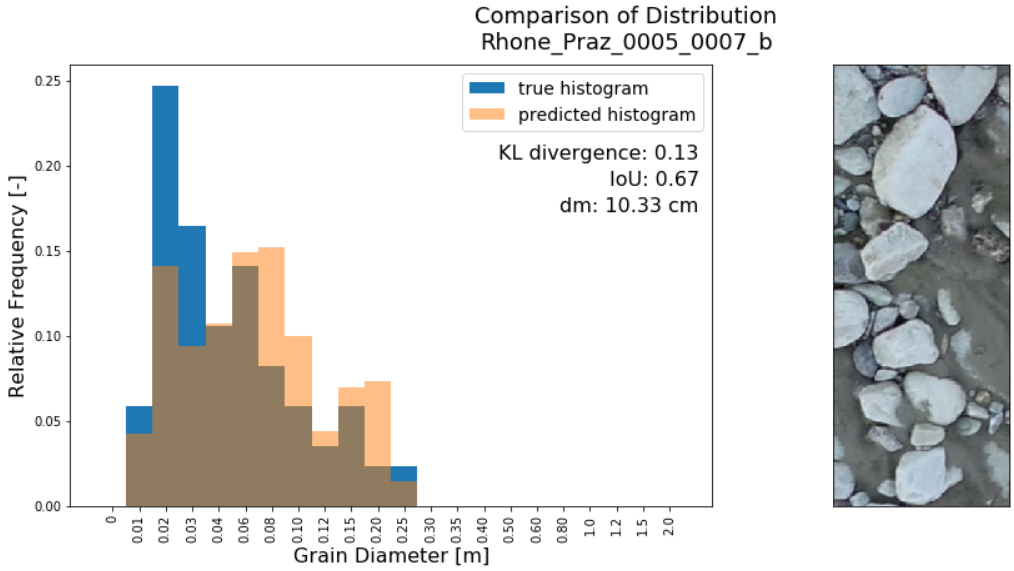


Figure A.4: Good prediction for another difficult tile with resulting KL divergence: 0.13, IoU: 0.67

# Bibliography


[1] BAFU. (2019) Bed load and suspended solids in watercourses. [Online]. Available: https://www.bafu.admin.ch/bafu/en/home/topics/water/info-specialists/state-of-waterbodies/state-of-watercourses/bed-load-and-suspended-solids-in-watercourses.html

[2] D. Vázquez-Tarrío, L. Borgniet, F. Liébault, and A. Recking, "Using uas optical imagery and sfm photogrammetry to characterize the surface grain size of gravel bars in a braided river (vénéon river, french alps)," *Geomorphology*, 2017.

[3] J. Brasington, D. Vericat, and I. Rychkov, "Modeling river bed morphology, roughness, and surface sedimentology using high resolution terrestrial laser scanning," *Water resources research*, vol. 48, no. W11519, 2012.

[4] H. Ibbeken and R. Schleyer, "Photo-sieving: a method for grain-size analysis of coarse-grained, unconsolidated bedding surfaces," *Earth Surface Processes and Landforms*, vol. 11, pp. 59–77, 1986.

[5] J. M. Verdú, R. J. Batalla, and J. A. Martínez-Casasnovas, "High-resolution grain-size characterisation of gravel bars using imagery analysis and geo-statistics," *Geomorphology*, vol. 72, pp. 73–93, 2005.

[6] M. Detert and V. Weitbrecht, "Automatic object detection to analyze the geometry of gravel grains - a free stand-alone tool," *River Flow*, 2012.

[7] D. J. Graham, I. Rice, and S. P. Reid, "Automated sizing of coarse-grained sediments: image-processing procedures," *Mathematical geology*, vol. 37, no. 1, pp. 1–28, 2005.

[8] R. Fehr, "Einfache bestimmung der korngrössenverteilung von geschiebematerial mit hilfe der linienzahlanalyse (simple detection of grain size distribution of sediment material using line-count analysis)," *Schweizer Ingenieur und Architekt*, vol. 105, no. 38, pp. 1104–1109, 1987.

[9] K. Sharma, "Histonet: Predicting size histograms of object instances," 2019, manuscript submitted for publication.

[10] G. R. Bezzola, *Flussbau*, 2011.

[11] G. Pajares, "Overview and current status of remote sensing applications base on unmanned aerial vehicles (uavs)," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–329, 2015.

[12] A. Karpathy. (2019) Cs231n convolutional neural networks for visual recognition. [Online]. Available: http://cs231n.github.io/

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[14] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 19–86, 1951.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML'15 Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, 2015, pp. 448–456.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[17] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-Supervised Nets," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, vol. 38, 2015, pp. 562–570.

[18] A. Karpathy. A recipe for training neural networks. [Online]. Available: http://karpathy.github.io/2019/04/25/recipe/

[19] Y. Bengio, *Practical Recommendations for Gradient-Based Training of Deep Architectures*. Springer Berlin Heidelberg, 2012, pp. 437–478.

[20] I. Zafar, G. Tzanidou, R. Burton, N. Patel, and L. Araujo, *Hands-On Convolutional Neural Networks with TensorFlow: Solve Computer Vision Problems with Modeling in TensorFlow and Python*. Packt Publishing, 2018.

[21] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *ICLR*, 2015.