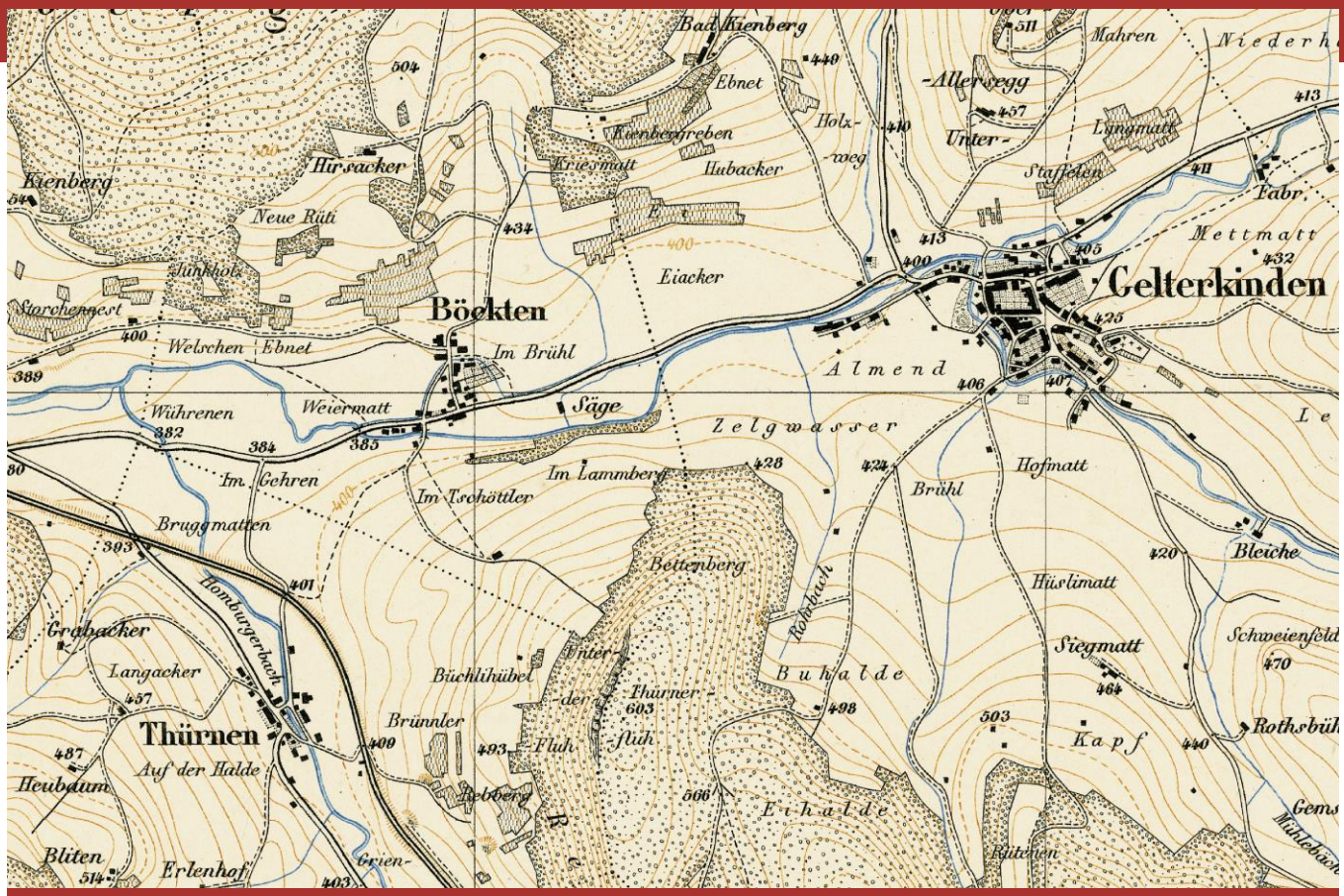
ETH zürich



Samplingstrategien zur Trainingsdatenerzeugung für tiefe Segmentierungsmodelle

Bachelor-Arbeit

Autor:
Joël Bender, Geomatik und Planung BSc
jbender@student.ethz.ch

Leiter:
Prof. Dr. Lorenz Hurni

Betreuer:
Dr. Magnus Heitzler

Frühlingssemester 2020
Abgabedatum: 29.05.2020

IKG
Institut für Kartografie
und Geoinformation

Vorwort

Da mich verschiedene Bachelorarbeiten ansprachen (zwei im Bereich Geoinformationssysteme und die vorliegende), erhielt ich bereits vor dem Beginn der Arbeit einen Einblick in die verschiedenen Aufgabenstellungen. Dabei stellte ich fest, dass das Verständnis von neuronalen Netzwerken und deren konkrete Anwendung in der Kartografie und Geoinformatik ein sehr spannendes Vertiefungsthema für eine Bachelorarbeit ist. Folglich entschied ich mich dafür, mich mit dem vorliegenden Thema zu befassen.
Für die Ermöglichung und das Anbieten dieser interessanten Bachelorarbeit, welche mir einen sehr guten Einblick in die Welt der neuronalen Netzwerke und deren Anwendung für Geoinformationszwecke gewährt hat, bedanke ich mich bei Prof. Dr. Lorenz Hurni.
Für die Erstellung der diversen Frameworks, die grosse Unterstützung im Rahmen des Arbeitsprozesses, die diversen Meetings mit hilfreichen Erklärungen und die rasche Beantwortung von Fragen im Laufe des gesamten Arbeitsprozesses bedanke ich mich bei meinem Betreuer Dr. Magnus Heitzler.
Für das Gegenlesen der Arbeit bedanke ich mich bei Caroline Bender und Christof Gemperle.

Zusammenfassung

Eine Möglichkeit zur Digitalisierung historischer Karten stellen neuronale Netzwerke dar, welche auf Basis von Trainingsdaten Parametermodelle erstellen, mit denen sich spezifische Featureklassen wie beispielsweise Seen extrahieren lassen. Um ein Training der gesamten Datenmenge zu vermeiden, kann auf Undersampling zurückgegriffen werden. Damit die Trainingsergebnisse erfolgreich sind, muss die Datenbasis, welche mittels Undersampling aus den Grundlagedaten extrahiert wird, eine möglichst hohe Güte aufweisen.
In der vorliegenden Arbeit werden verschiedene QGIS-Workflows für Undersampling erstellt. Auf Basis eines bestimmten historischen Datensatzes, der Siegfriedkarte, werden die Workflows zum Training des Fully Convolutional Neural Networks U-Net verwendet und nachfolgend in ihrer Güte miteinander verglichen und bewertet. Dabei liegt der Fokus der Arbeit sowohl auf der Erstellung der Workflows als auch auf deren Vergleich bezüglich Performanz und den daraus resultierenden Empfehlungen. Im Rahmen dieser Arbeit wird konkludiert, dass Workflows für ein optimales Undersampling die Grundlagedaten berücksichtigen müssen und dass aufgrund statistischer Effekte mehrere Instanzen der jeweiligen Workflows trainiert werden müssen. Der fehlerbasierte Workflow «EP2F_mix» erweist sich dabei als geeignetster Workflow für die vorliegende Extraktionsaufgabe.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Im Jahre 1868 wurde Hermann Siegfried durch Bundesgesetze mit der Erstellung eines «Topographischen Atlas der Schweiz 1:25 000/1:50 000» beauftragt, welcher von 1870 bis 1926 erschien. Dieser Atlas, gemeinhin auch unter dem Namen «Siegfriedkarte» bekannt, bildet eine wichtige Grundlage für die kartografischen Informationen aus dieser Zeit (swisstopo, 2020).
Weltweit wurden bereits in den vergangenen Jahren historische Karten wie die Siegfriedkarte in grossem Masse digitalisiert und liegen nun in Archiven vor. Um die entsprechenden Informationen in einem Geoinformationssystem (GIS) weiter zu verarbeiten, beispielsweise für eine Analyse der Veränderung der St. Petersinsel im Bielersee, müssen die jeweiligen topografischen Objekte klassifiziert beziehungsweise extrahiert werden. Dabei existieren für diese Digitalisierung verschiedene mögliche Vorgehensweisen, es können beispielsweise Farbsegmentation oder Formdeskriptoren verwendet werden (Chiang, Leyk, & Knoblock, 2014, S. 7). Im Rahmen dieser Arbeit wird der Ansatz mittels des Fully Convolutional Neural Networks (FCNN) U-Net gewählt, um aus den Siegfriedkarten die existierenden hydrologischen Featureklassen zu extrahieren. Dies umfasst konkret Bachläufe, Feuchtgebiete (zum Beispiel Moore), Flüsse und Seen. Die Vorhersagegenauigkeit eines FCNN hängt unter anderem von der Güte der Trainingsdaten ab, weshalb für deren Erstellung (Undersampling) verschiedene Samplingstrategien verglichen werden müssen. Dies ist insbesondere notwendig, da bei den vorliegenden Featureextraktionen ein Klassenungleichgewicht auftritt.

1.1 Ziel der Arbeit

Das Ziel der Arbeit ist es, verschiedene Workflows für die Trainingsdatenerzeugung zu generieren und diese mithilfe bestimmter Metriken und Methoden zu vergleichen. Durch diesen Vergleich soll auf verschiedene, bei der Workflowerstellung vorkommende Parameter Rückschluss gezogen werden, woraus sich entsprechende Empfehlungen für optimale Workflows ableiten lassen. Die Resultate dieser Arbeit sind:

- Verschiedene Workflows zur Generierung von Trainingsdaten
- Modellparameter für U-Net basierend auf verschiedenen Trainingsdatensätzen
- Vorhersagen pro Workflow und Kartenblatt
- Güteparameter für jeden Workflow
- Empfehlungsrahmen für die Workflowerstellung

2 Grundlagen

2.1 Grundlagedaten

Für die Bachelorarbeit liegen die verschiedenen Kartenblätter der Siegfriedkarte der Jahre 1879 und 1880 im Massstab 1:25 000 als Tagged Image File-Datei (TIF) vor. Für die Herstellung der gesamten Kartenblätter über ein ganzes Jahr liegen entsprechende virtuelle Rasterdateien im Format «vrt» vor, welche aus verschiedenen TIF-Dateien durch Georeferenzierung ein Mosaik definieren. Diese Daten weisen eine Rasterauflösung von 1.25 m / Pixel auf und liegen im RGB-Farbformat mit den Farbkanälen rot, grün und blau vor. Ausserdem existiert ein vierter Kanal (Alpha), welcher im Bereich der Kartenblätter den Wert 255 aufweist, während er überall sonst den Rasterwert 0 aufweist (Leerwert).
Für das Jahr 1880 liegen 17 über die Schweiz verteilte Kartenblätter vor (016, 029, 031, 056, 057, 058, 059, 070, 072, 073, 126, 133, 139, 141, 142, 233, 282). Für das Jahr 1879 liegen zehn Kartenblätter vor (008, 013, 015, 065, 067, 114, 116, 128, 134, 136). Alle vorliegenden Kartenblätter weisen eine Ausdehnung von 7000×4800 Pixel auf, was zu einer Breite von 8.75 km und einer Höhe von 6 km führt.
Zudem liegt für das Jahr 1880 ein (manuell) vektorisierter Groundtruthdatensatz vor, welcher die gesamte Ausdehnung der drei zu extrahierenden Featureklassen Stream (Bachläufe), Wetland (Feuchtgebiete) und Riverlake (Flüsse und Seen) als Shapefile wiedergibt. Hierbei sind alle Daten vom Typ «Polygon». Ausserdem ist auch für das Jahr 1879 pro vorkommendes Kartenblatt ein Shapefile mit jeder der drei Featureklassen vorhanden, welche als Groundtruthdatensatz für die Überprüfung der Vorhersagen des FCNN U-Net benutzt werden. Hierbei liegt die Featureklasse Stream als Liniendaten vor, während Wetland und Riverlake als Polygondaten gegeben sind.
Vorgegeben ist im Rahmen dieser Arbeit, dass die Daten des Jahrs 1880 als Trainingsdaten fungieren, während die Daten des Jahres 1879 zur Vorhersage und nachfolgenden Evaluation benutzt werden.
Alle Grundlagedaten unterstützen im Rahmen der Verarbeitung im Open Source-GIS QGIS das Koordinatenreferenzsystem LV03 bzw. EPSG 21781 (epsg.io, 2019). Dabei beschreibt LV03 die Landestriangulation, welche bis 2016 die offizielle Grundlage für Vermessung der Schweiz war (swisstopo, kein Datum).

2.1.1 Featureklassen

Die vorliegenden Featureklassen sind Stream, Wetland und Riverlake. Flüsse und Seen werden dabei aufgrund ihrer ähnlichen Symbolisierung bereits vereinigt als Featureklasse Riverlake zur Verfügung gestellt.
Die Klasse Stream wird in den vorliegenden Kartenblättern jeweils durch eine blaue Linie symbolisiert (vgl. Abbildung 1). Die Symbolisierung der Klasse Wetland erfolgt durch parallele blaue Striche verschiedener Länge, die in unterschiedlichen Abständen gruppiert sind. Die Klasse Riverlake wird entweder durch parallele Linien, welche dem Randverlauf des entsprechenden Objekts (Fluss oder See) folgen, oder aber durch blaue Punkte symbolisiert.

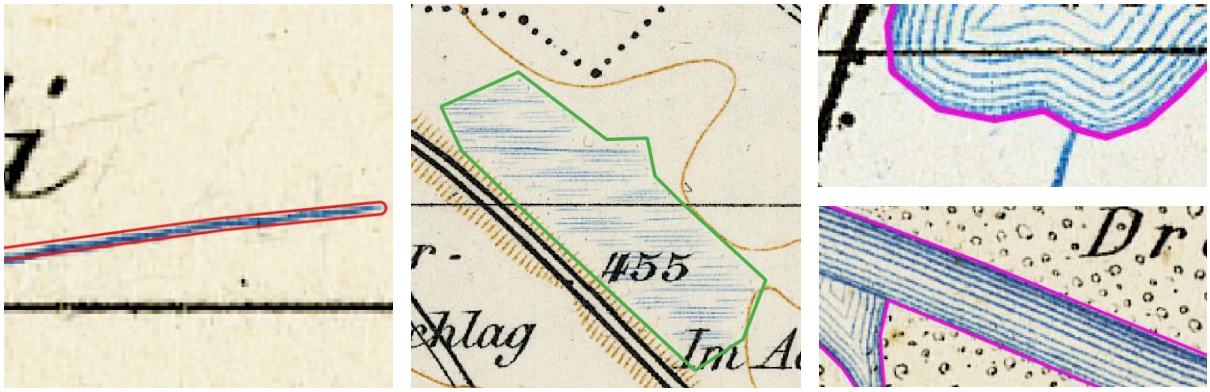


Abbildung 1: Featureklassen Stream, Wetland und Riverlake (v. l. n. r.)

Dabei gilt es zu beachten, dass diese Featureklassen im Rahmen dieser Arbeit als invariant angenommen werden, die Einteilung dieser Klassen wird nicht verändert. Flüsse und Bäche werden folglich in verschiedene Kriterien eingeteilt, wobei die Unterscheidung auf der Anzahl der Striche beruht. Ein Wasserlauf bestehend aus mehreren parallelen Strichen wird als River klassifiziert, ein Wasserlauf mit einem Strich als Bach.
Aufgrund eines geometrischen Fehlers im vorliegenden Datensatz «Riverlake» muss dieser jeweils vor Verwendung entweder gepuffert werden (mit Distanz 0 oder einem positiven Distanzwert), oder aber mithilfe der Funktion «Geometrien reparieren» repariert werden.

2.2 Theoretische Grundlagen

Für die Extraktion der Features aus historischen Karten existieren im Rahmen derer Digitalisierung verschiedene Möglichkeiten. Einerseits kann mit Methoden wie beispielweise Template Matching als Korrelationsverfahren (Stengele, 1995, S. 67) oder Kantenextraktion (Stengele, 1995, S. 93) eine Rasterkarte bearbeitet und allenfalls auf Features überprüft werden. Diese Bildverarbeitungsmethoden können auch für eine Extraktion mit neuronalen Netzwerken die Grundlage bilden. Nicht alle Extraktionen sind aber mit den beschriebenen Verfahren mit einem eher niedrigen Level an künstlicher Intelligenz (Stengele, 1995, S. 67) machbar. Für beispielsweise Flüsse oder Seen mit sich verändernden Formen und Ausdehnungen ist Template Matching nicht optimal geeignet, da «im Prinzip nur Muster derselben Grösse und einheitlicher Ausrichtung gesucht werden» können (Stengele, 1995, S. 67).[1]
Für eine umfassendere Featureextraktion ist es aus diesem Grund notwendig, auf eine Methode zurückzugreifen, welche «lernen» kann, wie die zu extrahierenden Featureklassen aussehen. Deshalb wird im Rahmen dieser Arbeit ein künstliches neuronales Netzwerk verwendet.

2.2.1 Machine Learning

Machine Learning bezeichnet laut Samuel (1959) das «Studienfeld, das Computern die Möglichkeit zu lernen gibt, ohne explizit [dafür] programmiert zu sein» (Kurzhals, 2020, S. 19). Angewendet auf die konkrete Problemstellung bedeutet dies, dass mit einer entsprechenden Machine Learning-Vorgehensweise Bachläufe, Feuchtgebiete und Flüsse und Seen erkennt werden können, ohne dass der Detektionsalgorithmus explizit darauf abgestimmt wurde, genau diese topografischen Objekte aus der Karte zu extrahieren und ausschliesslich dafür verwendbar ist.

[1] Rotationen und Skalierungen sind nach heutigem Stand der Technik auch möglich. (Chiang, Leyk, & Knoblock, 2014, S. 13)

2.2.2 Supervised und Unsupervised Learning

Im Rahmen der Mustererkennung wird oft zwischen Supervised Learning und Unsupervised Learning unterschieden. Dabei bezeichnet das Supervised Learning den Fall, dass für die Trainingsdaten bereits die korrekt erkannten Muster für ein Datenset mit Trainingsdaten vorliegen und das neuronale Netzwerk davon lernen kann (Michie, Spiegelhalter, & Taylor, 1994, S. 85).
Diese liegen im Rahmen dieser Arbeit vor, womit es sich um Supervised Learning handelt (Kurzhals, 2020, S. 22).
Unsupervised Learning bezieht sich auf den Fall, dass keine klassifizierten Groundtruthdaten vorliegen. Somit müssen Strukturen in den Daten beispielsweise durch Wahrscheinlichkeitsdichtefunktionen oder Clustering erkannt werden (Michie, Spiegelhalter, & Taylor, 1994, S. 85). Dazu existieren verschiedene Algorithmen wie beispielsweise K-Means oder DBSCAN. (Kurzhals, 2020, S. 39)

2.2.3 Künstliche neuronale Netzwerke

Eine Möglichkeit, wie ein Computer im Rahmen von Machine Learning «lernen» kann, eine bestimmte Aufgabe zu erledigen, stellen die künstlichen neuronalen Netzwerke dar. Deren Architektur ist von den neuronalen Netzen im menschlichen Gehirn inspiriert, welche beschreiben, wie das Gehirn Informationen prozessiert. Künstliche neuronale Netzwerke bestehen aus einer grossen Anzahl von Knoten, welche (normalerweise in Ebenen angeordnet) miteinander verbunden sind und durch welche die Daten im Laufe des Trainings transformiert werden. Jeder Knoten weist pro eingehende Verbindung eine spezifische Gewichtung auf, welche am Anfang des Trainingsprozesses zufällig initialisiert wird. Die so prozessierten Daten werden dann während des Trainings jeweils mit der Gewichtung des Knotens multipliziert und abhängig von einer bestimmten Aktivierungsfunktion weitergegeben. Im Rahmen von Supervised Learning liegt für die Daten ein bereits klassifiziertes Trainingsbeispiel vor, wobei im Verlauf des Trainings die Gewichte so angepasst werden, dass Inputdaten mit der gleichen Klassifizierung auch den gleichen Output ergeben. Für die Optimierung der Gewichte können gradientenbasierte Ansätze verwendet werden, welche auf dem mathematischen Prinzip des Gradienten einer skalaren Loss-Funktion basieren (Goodfellow, Bengio, & Courville, 2016, S. 80). Damit diese Loss-Funktion optimiert werden kann, muss deren Gradient mithilfe eines Backpropagation-Algorithmus auf die jeweiligen Parameter zurückgeführt werden. Dabei beschreibt Backpropagation nur den Algorithmus der Informationsübertragung rückwärts entlang des neuronalen Netzwerks. (Goodfellow, Bengio, & Courville, 2016, S. 200)

2.2.4 Fully Convolutional Neural Networks

Convolutional Neural Networks (CNN) sind neuronale Netzwerke, welche an Stelle von Matrixmultiplikation Faltungsoperationen verwenden (Goodfellow, Bengio, & Courville, 2016, S. 326). Falls sie ausschliesslich aus Faltungsebenen bestehen, so nennt man sie Fully Convolutional Neural Networks (FCNN).
Diese sind im Rahmen der Bilderkennung von Bedeutung. Sie sind konkret darauf ausgelegt, mit Bildern der Dimension $h \times w \times d$ zu arbeiten, also mit einem Bild der Ausdehnung $h \times w$ und der Tiefe bzw. Kanalanzahl d (Long, Shelhamer, & Darrell, 2015, S. 2). Dabei werden diese Bilder durch aufeinanderfolgende Faltungsoperationen und Pooling in ihrer Ausdehnung verkleinert und zu Feature Maps umgebildet. Für die Faltungsoperationen treten dabei verschiedene Parameter auf (beispielsweise Kernels), welche im Rahmen des Trainings verändert werden.

2.2.5 U-Net

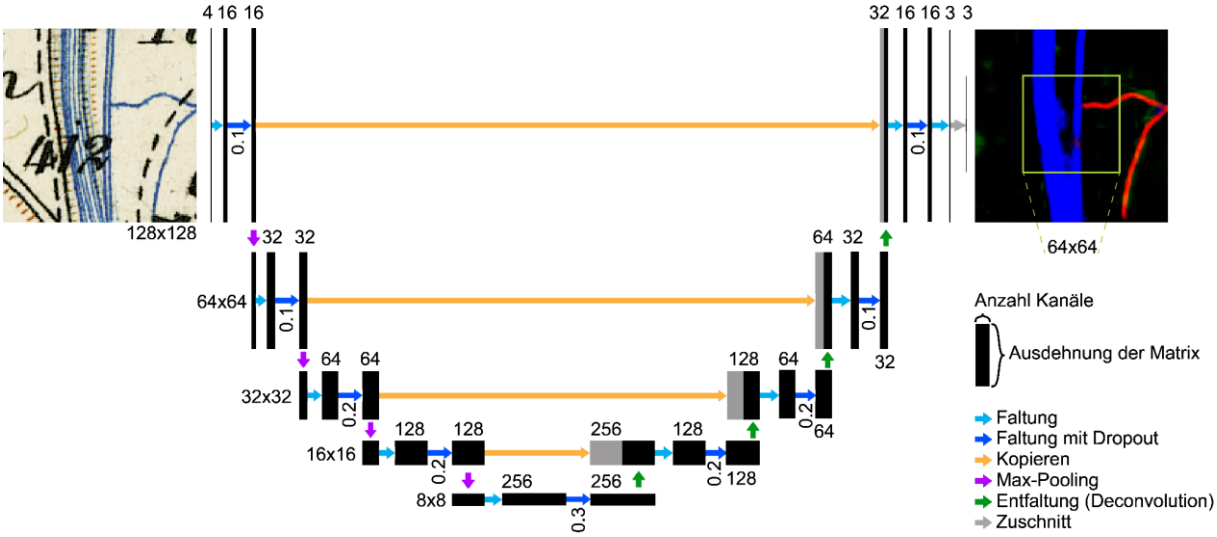


Abbildung 2: U-Net-Framework zur Extraktion von Features aus historischen Karten

Im Rahmen dieser Arbeit wird für das Training mit U-Net eine spezifische U-förmige Architektur eines FCNN gewählt. Diese FCNN-Architektur wurde an der Universität Freiburg zum Zwecke der Segmentation von biomedizinischen Bilddaten entwickelt (Ronneberger, Fischer, & Brox, 2015, S. 1). Dabei werden die Inputdaten durch Faltungsoperationen und Pooling zu einer Segmentation Map ummodelliert. Die verwendete Implementation von U-Net wird dabei als Modell in Abbildung 2 dargestellt. Diese Implementation weist insgesamt 1 941 283 zu trainierende Parameter auf, welche alle im Laufe der Faltungen bzw. Entfaltungen vorkommen.[2] Es wird mit dem Modell «U-Net-Batch-Multi» gearbeitet. Dabei werden symmetrisch um die Trainingspunkte quadratische Kacheln der Grösse 128×128 Pixel erzeugt und als Inputdaten für U-Net verwendet.[3] Daraufhin werden viermal in Folge zwei Faltungsoperationen und eine 2×2 Max-Pooling-Operation, bei welcher die Ausdehnung des Bildes halbiert wird, ausgeführt. Schliesslich erfolgt bei der kleinsten Ausdehnung nochmals eine zweimalige Faltung. Anschliessend an diese erfolgen viermal jeweils eine Entfaltung (Deconvolution) mit Überspringen eines Pixels (2×2 Stride), eine Zusammenfügung mit dem Zwischenergebnis derselben Stufe (grauer Pfeil) und zwei Faltungen. Bei all diesen Faltungen wird als Aktivierungsfunktion «eLu» verwendet, eine Mischung aus einer exponentiellen und einer linearen Funktion (vgl. Abbildung 3), die sich wie folgt definiert (Layer activation functions, kein Datum):

$$eLu(x) = \begin{cases} e^x - 1, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Diese wird jeweils auf das Ergebnis der Faltung angewendet. Als letzter Schritt wird ausserdem eine weitere Faltung mit der Sigmoidfunktion

$$sig(x) = \frac{1}{1 + e^{-x}}$$

als Aktivierungsfunktion angewendet (Layer activation functions, kein Datum) und das entstehende Bild wird auf die passende Grösse zugeschnitten.[4] Dabei ist zu beachten, dass das Ausgabebild eine Grösse von 64×64 aufweist. Dies ist im Rahmen der Segmentierung von Bedeutung, da bei Faltungsoperationen am Rand jeweils Information extrapoliert wird, was das Ergebnis verschlechtern könnte. Bei allen Faltungsoperationen in Folge wird

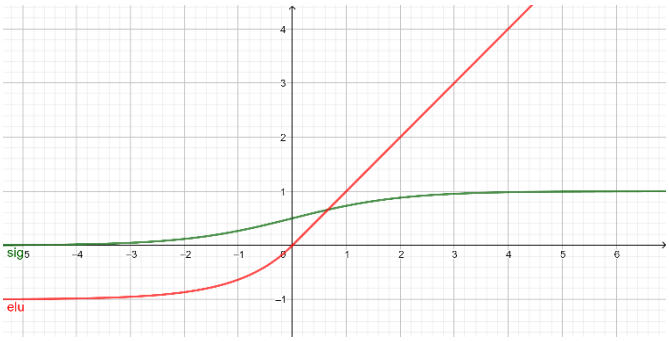


Abbildung 3: Aktivierungsfunktionen sig(x) in Grün und eLu(x) in Rot (Plot erstellt mit Geogebra)

[2] vgl. in der Datei model_summary.txt Zeile 94
[3] vgl. im Code von model.py Zeile 45
[4] vgl. im Code von model.py Zeilen 39-108

dazwischen noch ein Dropout durchgeführt. Dropout bedeutet, dass der in Abbildung 2 angegebene Anteil an Parametern nicht durch das Training an diesem Punkt verändert wird. Dies schützt vor Overfitting auf die Trainingsdaten (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014, S. 1930).
Für die Loss-Funktion wird die binäre Kreuzentropie verwendet. Diese definiert sich nach Ronneberger, Fischer, & Brox, 2015, S. 4 als

$$\sum_{x\epsilon\Omega} \omega(x) * \log\left(p_{l(x)}(x)\right),$$

wobei l(x) das wahre Label für das jeweilige Pixel ist, ω(x) die spezifische Gewichtung jedes Pixels im Rahmen des Trainings und Ω die Menge aller Pixel (Ronneberger, Fischer, & Brox, 2015, S. 5). Als Optimierungsfunktion wird Adam verwendet, ein Algorithmus, welcher auf dem lokalen Gradienten und dem Moment erster und zweiter Ordnung basiert (Kingma & Ba, 2015, S. 2).[5]

2.2.6 Gründe für Sampling

Die vorliegenden Daten, auf denen das neuronale Netzwerk trainiert werden soll, sind historische Karten der Schweiz. Das Training, bei welchem das Netzwerk «lernen» muss, die vier Klassen «Stream», «Wetland», «Riverlake» und «nichts davon» zu unterscheiden, wird mithilfe der Kartenblätter aus dem Jahre 1880 durchgeführt. Dabei sind diese Featureklassen nicht gleichmässig verteilt. Es liegen viel mehr Pixel mit «nichts davon» als beispielsweise mit «Wetland» vor, da es sich um Originaldaten handelt (vgl. Tabelle 1).

Klasse	Stream	Wetland	Riverlake	Nichts davon	Gesamt
Fläche [km^2]	4.607	6.108	9.607	872.272	892.5

Tabelle 1: Fläche der verschiedenen Klassen

Insofern tritt bei neuronalen Netzwerken das Problem des Klassenungleichgewichts auf, welches dazu führen kann, dass die überwiegende Klasse beim «Lernen» zu stark berücksichtigt wird, während kleine Klassen vernachlässigt werden (Johnson & Khoshgoftaar, 2019, S. 2).[6]
Gelöst werden kann dieses Problem durch verschiedene Ansätze, beispielsweise durch Oversampling oder auch durch Undersampling. Beim Oversampling werden Daten der kleineren Klassen künstlich erzeugt, was bei Datenknappheit sinnvoll sein kann. Da im Rahmen dieser Arbeit aber genügend Daten vorliegen, wird hier der Ansatz des Undersamplings verfolgt, bei welchem nur ein Teil der Daten als Trainingsdaten verwendet wird. Auch aufgrund der Trainingszeit empfiehlt sich bei grossen Datenmengen Undersampling (Johnson & Khoshgoftaar, 2019, S. 22)

[5] vgl. im Code von model.py Zeilen 117-118

[6] Dieser Sachverhält wird in der Quelle für binäre Klassifikation beschrieben. Die sich daraus ergebenden Grundlagen lassen sich aber auch für Klassifikation mit mehr als zwei Featureklassen verwenden (Johnson & Khoshgoftaar, 2019, S. 3).

3 Methode und Vorgehen

3.1 Framework

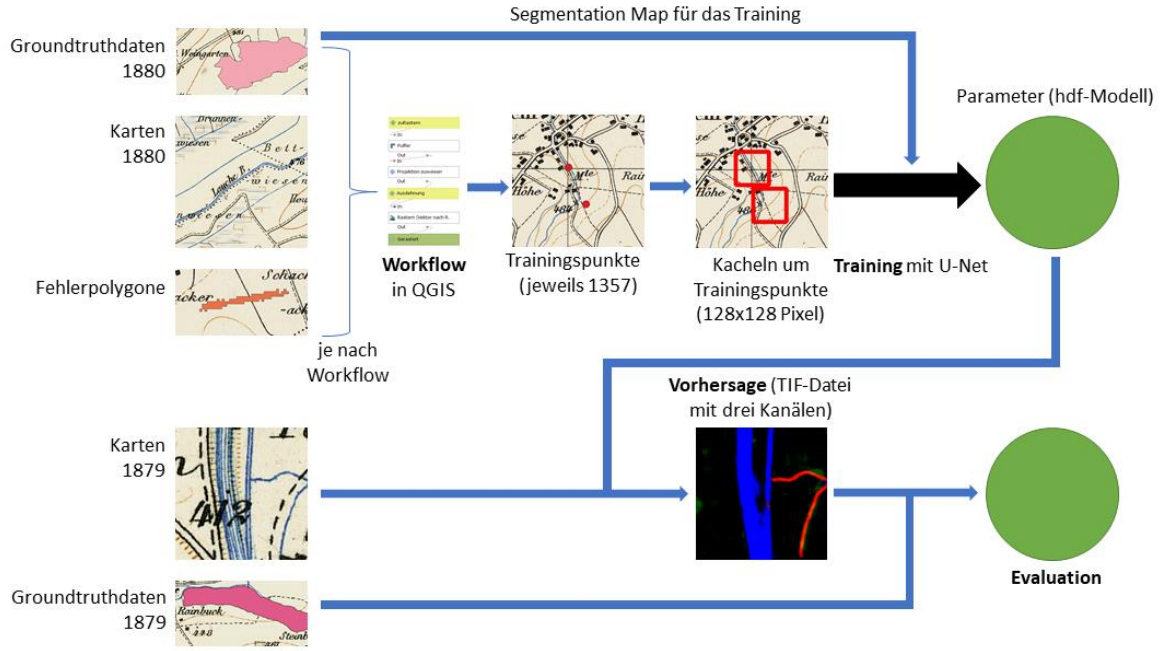


Abbildung 4: Übersicht über den Ablauf der verschiedenen Schritte

Das Framework, welches im Rahmen dieser Arbeit verwendet wird, besteht aus mehreren Schritten (vgl. Abbildung 4). Die Grundlagedaten liegen bereits vor und werden in Kapitel 2.1 beschrieben. Mit den Grundlagedaten von 1880 und allenfalls generierten Fehlerpolygonen (je nach Workflow, vgl. Kapitel 4.1.4) als Input, werden durch verschiedene, selbst konzipierte Workflows Trainingspunkte im Shapefile-Format (.shp) erzeugt. Um diese Punkte werden nachfolgend durch ein vorliegendes Framework Quadrate mit einer Seitenlänge von 128 Pixeln generiert, welche U-Net als Grundlage für das Training dienen. Im Laufe des Trainings mit den Groundtruthdaten aus dem Jahre 1880 als Segmentation Map werden die Parameterwerte in U-Net angepasst (vgl. Kapitel 2.2.5) und als Output wird ein Parametermodell generiert. Dieses dient dem bereits existierenden Vorhersageframework als Input, sodass aus den Testdaten (Kartenblätter des Jahres 1879) eine Vorhersagedatei entsteht. Die Güte dieser Vorhersage wird anschliessend mit verschiedenen Metriken evaluiert, welche implizit auch die Güte der Workflows beschreiben.

3.2 Generierung der Workflows

Um das Undersampling durchführen zu können, müssen die entsprechenden Punktdaten aus den Kartendaten erzeugt werden. Dafür wird im Rahmen dieser Arbeit die grafische Modellierung des Open Source-Programmes QGIS verwendet. Dabei werden je nach Workflowgruppe verschiedene Inputdaten verwendet. Als Outputdaten entstehen jeweils Punktlayer mit 1357 Punkten, welche von der Punktanzahl der Gitterpunkte herrühren (vgl. Kapitel 4.1.1) und den Attributen «type», «year», «scale» und «origin». Da die Workflows sowohl ein Ergebnis darstellen als auch die Grundlage für die weiteren Ergebnisse bilden, werden sie in Kapitel 4.1 ausführlich erklärt.

3.3 Trainieren

Das Training mit den durch die verschiedenen Workflows generierten Punktlayern erfolgt auf Basis des Frameworks mit U-Net, das im Rahmen der Arbeit zur Verfügung gestellt wird. Dabei existieren weitere Parameter, welche im Rahmen dieser Arbeit als invariant angenommen werden. Dazu zählt die Epochenanzahl, welche auf 500 gesetzt wird. Eine Epoche beschreibt dabei generell «einen Durchgang durch die gesamten Daten» (Keras FAQ, kein Datum). Der Wert von 500 wird aber bei keinem vorkommenden Workflow ausgereizt. Vielmehr kommt die Early Stopping Patience zum Tragen. Diese beschreibt, nach wie vielen Epochen ohne Verbesserung der Monitorfuktion (in vorliegenden Fall Binary Accuracy[7]) das Training nicht mehr weitergeführt werden soll (Early Stopping, kein Datum).
Als Ergebnis des Trainings wird eine Hierarchical Data Format-Datei (hdf) erzeugt, welche die Parameter der Trainings abbildet und vom Vorhersage-Framework aufgerufen wird.
Das Training dauert jeweils etwas mehr als 2 Minuten pro Epoche, ein Training mit 500 Epochen würde also mehr als 16 Stunden dauern. Summiert über alle 21 Trainingsinstanzen echter Workflows (vgl. Kapitel 4.1.6) werden 4777 Epochen ausgeführt, was zu einer durchschnittlichen Epochenanzahl von 227.48 und einer durchschnittlichen Dauer von mehr als 7.5 h führt[8]. Aus diesem Grund können auch nicht bei allen vorliegenden Workflows jeweils drei Instanzen trainiert werden, sodass sich das Training von drei Instanzen auf die Workflows ZP, ZPmin227, BPmin227, 2F und EP2F beschränkt.

3.4 Vorhersagen

Mit der generierten hdf-Datei des Trainings einer bestimmten Instanz werden darauffolgend für alle vorliegenden Kartenblätter des Jahres 1879 Vorhersagen mit einem entsprechenden Python-Framework getroffen. Diese werden vom Netzwerk prozessiert und haben ein TIF-File mit der gleichen Ausdehnung und Rasterweite wie die Inputkarte als Resultat, welches aus drei Kanälen besteht: Im RGB-Modell steht rot für Stream, grün für Wetland und blau passenderweise für Riverlake. Dabei wird pro Pixel und Kanal jeweils eine Wahrscheinlichkeit, entsprechend den Axiomen von Kolmogorow also ein Wert zwischen 0 und 1, gespeichert.

3.5 Evaluieren

Die durch das Vorhersagen generierten Rasterdaten müssen mit entsprechenden Groundtruthdaten verglichen werden, um die Güte der Rasterdaten bewerten zu können.

3.5.1 Generieren der Groundtruthdaten

Die Groundtruthdaten liegen bereits als Vektordatei vor. Im Rahmen der Evaluation müssen sie aber im Rasterformat vorliegen und werden dementsprechend je nach Featureformat in das jeweilige Rasterformat gerastert. Ein entsprechender QGIS-Workflow zur Darstellung der Methodik wird in Abbildung 5 dargestellt. Dabei wird die Klasse Stream um 2 m gepuffert, um Konsistenz mit den Inputdaten zu gewährleisten.
Beim Workflow, welcher Flüsse und Seen vereinigt (vgl. Abbildung 5), werden die Geometrien jeweils aufgelöst. Dadurch wird sichergestellt, dass alle Polygone vollständig gerastert werden.

[7] vgl. im Code von segmentation_manager.py Zeile 98 und die Beschreibung von Accuracy im Kapitel 3.5.2
[8] vgl. im Code von Datenvergleich_Plots.m Zeile 15

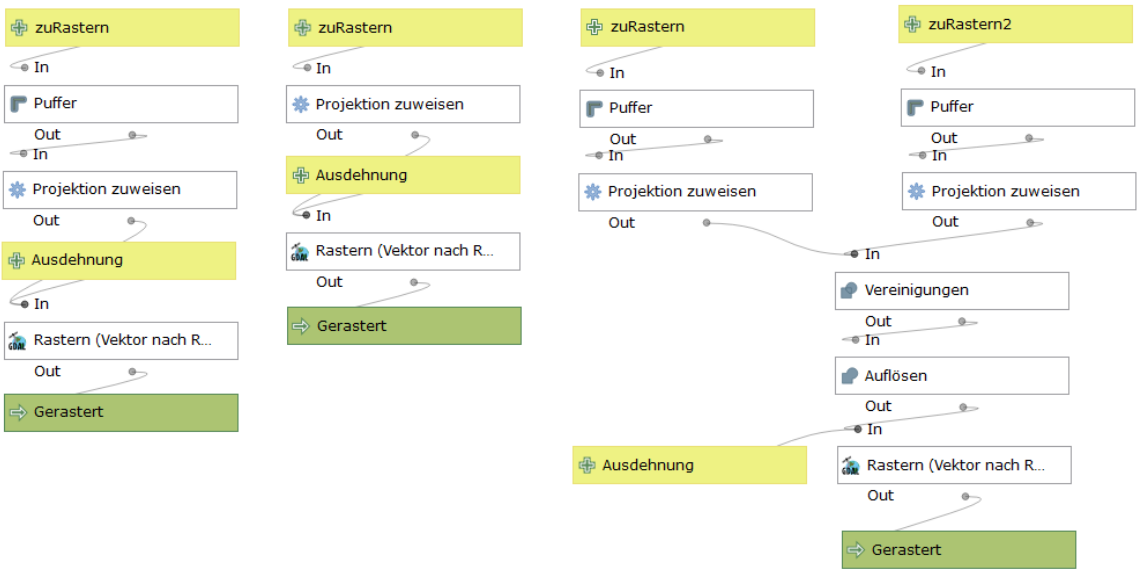


Abbildung 5: Workflows zur Generierung der Groundtruthdaten für Stream, Wetland und Riverlake (v. l. n. r.)

3.5.2 Verwendete Gütemasse

Für die Beurteilung der Güte der verschiedenen Vorhersagen und implizit auch der verschiedenen Workflows werden sechs Metriken verwendet, die alle einen unterschiedlichen Fokus aufweisen, wodurch je nach Anwendungsfall verschiedene Metriken betrachtet werden können. Für den Vergleich werden dabei die Operationen der Python Bibliothek Scikit-Learn verwendet (scikit-learn 0.23.0 documentation, 2019). Dabei gelten die Daten dann als «true», wenn die jeweilige Wahrscheinlichkeit einen Schwellenwert von 0.5 übersteigt.
Um diese Metriken und deren Bedeutung zu zeigen, wird nachfolgend jeweils ein Beispiel im Rahmen verschiedener Mengen gezeigt.
Sei D nachfolgend die Menge der detektierten Pixel des durch das FCNN trainierten Datensatzes und R die Menge der «richtigen» Pixel entsprechend des Groundtruthdatensatzes, welcher durch das Institut für Kartografie und Geoinformation (IKG) digitalisiert und überprüft wurde.

Precision

Die «Precision», auf Deutsch «Präzision», beschreibt den Anteil der richtigen Pixel in der Menge der detektierten Pixel. Die Präzision bestimmt sich also durch folgende Formel (scikit-learn 0.23.0 documentation, 2019):

$$Pre = \frac{R \cap D}{D}.$$

Konkret angewendet bedeutet eine hohe Präzision folglich, dass ein detektierter Pixel stimmt. Wenn aber beispielsweise in den Groundtruthdaten 50 korrekte Pixel und in den trainierten Daten 2 Pixel auftreten, welche beide in der Menge dieser 50 Pixel liegen, so liegt eine Präzision von 100 % vor, obwohl das Training und somit die trainierten Daten bei dieser Datenlage ohne Zweifel verbesserungswürdig sind. Die Präzision als alleinige Metrik für die Daten kommt deshalb nicht in Frage.

Recall

Der «Recall», auf Deutsch am treffendsten «Trefferquote» genannt, beschreibt den Anteil der detektierten Pixel in der Menge der richtigen Pixel. Dieses Mass definiert sich deshalb durch folgende Formel (scikit-learn 0.23.0 documentation, 2019):

$$Rec = \frac{R \cap D}{R}$$

Konkret bedeutet ein hoher Recall, dass ein hoher Anteil an den richtigen Pixeln detektiert wurde. Wenn aber beispielsweise alle existierenden Elemente detektiert werden, dann liegt der Recall bei 100 %, obwohl eine solche Detektion für die vorliegende Aufgabe nicht zielführend ist, sodass auch dieser als alleiniges Mass nicht ausreicht.

F1

Aus diesem Grund bietet es sich an, Präzision und Recall zu kombinieren, sodass sich sowohl zu viele Fehldetektionen als auch zu viele nicht detektierte Richtige negativ auf den Wert des entstehenden Gütemasses auswirken. Insofern ergibt sich die F1-Metrik, welche als das harmonische Mittel von Präzision und Recall definiert wird (vgl. Abbildung 6). Das harmonische Mittel hat gegenüber dem arithmetischen Mittel insbesondere den Vorteil, dass sobald einer der beiden Werte 0 ist, es auch den Wert 0 annimmt (scikit-learn 0.23.0 documentation, 2019).
Konkret berechnet sich die F1-Metrik durch die Formel

$$F1 = \frac{2}{Rec^{-1} + Pre^{-1}} = \frac{2 * Rec * Pre}{Rec + Pre}.$$

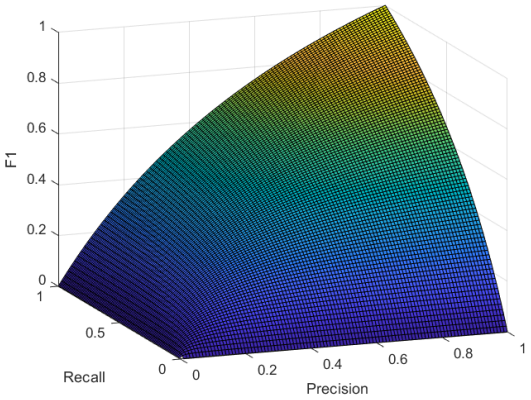


Abbildung 6: Plot von F1 gegenüber Precision und Recall

Accuracy

Die «Accuracy», auf Deutsch «Genauigkeit», bestimmt sich durch den Anteil an übereinstimmenden Pixeln, also den Anteil an korrekt detektierten Pixeln vereinigt mit den korrekt nicht detektierten Pixeln an der Gesamtmenge aller Pixel n. Als Formel ausgedrückt ergibt sich folgendes (scikit-learn 0.23.0 documentation, 2019):

$$Acc = \frac{1}{n}\sum_{i=1}^{n}(v_{i,groundtruth} = v_{i,predicted})$$

wobei letztere Gleichung 1 ist, wenn die Pixel übereinstimmen und 0, wenn die Pixel nicht übereinstimmen.
Im Unterschied zu den Metriken Precision und Recall werden hier also explizit auch die korrekt nicht detektierten Pixel einbezogen. Dies führt dazu, dass bei Rasterdaten von grossem Ausmass, bei welchen detektierte Features (z.B. Riverlake, Wetland oder Stream) eher selten vorkommen (hohes Klassenungleichgewicht), der Wert dieser Metrik automatisch sehr hoch wird.

Jaccard

Das «Jaccard-Mass» oder auch «Intersection over Union», bezeichnet eine weitere Metrik, um die Trainingsdaten mit den Groundtruthdaten zu vergleichen. Mathematisch ist sie wie folgt definiert (scikit-learn 0.23.0 documentation, 2019):

$$Jac = \frac{R \cap D}{R \cup D}$$

Sie betrachtet also die richtig detektierten Pixel und setzt sie in Relation zu den entweder richtigen oder detektierten Pixeln. Von der Wirkung her ist Jaccard vergleichbar mit der F1-Metrik, da der Wert sowohl bei einer zu tiefen Precision als auch bei einem zu tiefen Recall sinkt.

Average Precision

Die «Average Precision», übersetzt die «durchschnittliche Präzision», ist ein Mass für den Verlauf der Präzision gegenüber verschiedenen Recall-Werten. Stellt man sich die Präzision als eine Funktion vor, welche vom Recall abhängt, so ergibt sich dabei der typische Verlauf der Precision-Recall-Kurve (vgl.

Abbildung 7). Von einem Recall von 0 (noch keine detektierten Pixel bzw. Treffer) bis zu einem Recall von 1 (alle Elemente wurden detektiert) wird das folgende Integral gebildet:

$$AvP_{theor} = \int Pre(Rec)dRec$$

Da dieses Vorgehen aber eher rechenintensiv ist und zu optimistisch sein kann, wird deshalb auf die diskretisierte Formel

$$AvP_{prax} = \sum_{n} Pre(n) * [Rec(n) - Rec(n-1)]$$

ausgewichen, wobei n alle Orte bezeichnet, an denen Pre(Rec) eine Schwelle bildet (scikit-learn 0.23.0 documentation, 2019). Die geometrische oder mengentheoretische Vorstellung dieser Gleichung ist schwierig, sie bildet aber ähnlich der F1-Metrik den Trade-Off zwischen der Precision und dem Recall ab und wird oft als Mass dafür verwendet, um die Güte eines Algorithmus zu messen (Davis & Goadrich, 2006, S. 6).

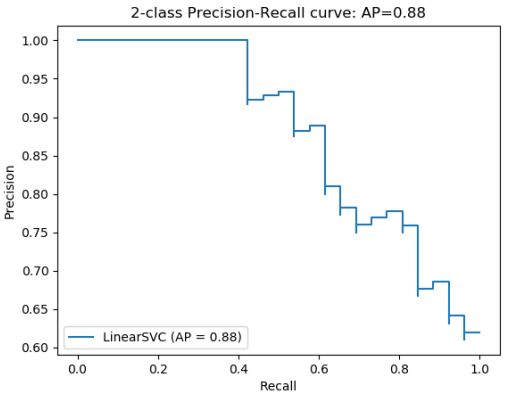


Abbildung 7: Typischer Verlauf einer Precision-Recall-Kurve (Plot Precision Recall, kein Datum)

Vergleich

Grundsätzlich beschreiben alle der zuvor aufgeführten Metriken einen Anteil und liegen deshalb zwischen 0 und 1. Rein mathematisch existiert eine Vergleichbarkeit aller Metriken. Ein Vergleich ergibt aber je nach Metrik nur begrenzt Sinn, da die vorliegenden Daten ein hohes Klassenungleichgewicht aufweisen (vgl. Kapitel 2.2.6). Die Accuracy beispielsweise wird immer viel höher sein, da sie auch nicht detektierte, falsche Pixel miteinbezieht. Precision und Recall sind aber sehr gut vergleichbar und werden gesamthaft durch F1 abgebildet. Eine Vergleichbarkeit ergibt sich auch zwischen Precision/Recall und Jaccard, da alle drei Metriken Mengen beziehungsweise Anteile abbilden.

3.5.3 Metriken für Kartenblätter

Um sicherzustellen, dass Ausreisser gut erkannt werden, sollen die Metriken für die Kartenblätter einzeln generiert werden. Dabei können insbesondere auch Rückschlüsse darüber gezogen werden, ob gewisse Workflows über alle Kartenblätter stabil funktionieren oder ob starke Schwankungen auftreten.

3.5.4 Gesamtwert über alle Kartenblätter des Jahres 1879

Für Vergleiche zwischen den Daten soll ausserdem ein Gesamtwert über alle Kartenblätter des Jahres 1879 gebildet werden. Dabei soll jedes Kartenblatt gleich gewichtet werden. Somit eignet sich das arithmetische Mittel.

4 Ergebnisse

4.1 Workflows

Das erste Ergebnis stellen die verschiedenen Workflows dar, welche mithilfe der grafischen Modellierung von QGIS erzeugt werden.
Die Algorithmen zur Generierung von Punkten in bestimmten Grenzen setzen das Vektorformat voraus. Da die Kartenblätter des Jahres 1880 im Rasterformat vorliegen, müssen die Kartenblätterränder als erstes vektorisiert werden. Zu diesem Zweck dient bei den vorliegenden aus vier Rasterkanälen bestehenden Kartenblätter der vierte Kanal (Alpha). Dieser besitzt entweder denn Wert «0» oder «255», je nachdem ob die Karte am entsprechenden Rasterpunkt existiert oder nicht. Insofern wird auf Basis dieses Rasterkanals der Algorithmus «Vektorisieren» angewendet, um die Kartenblattgrenzen im Vektorformat zu erhalten.
Allgemein gilt für alle Workflows, dass sobald die Punkte vorliegen, diese noch mit den entsprechenden Attributen versehen werden müssen. Notwendigerweise ist dies das Attribut «type», da jeder Punkt entweder «training» oder «validation» als Typ-Attribut aufweisen muss.[9] Ebenfalls notwendig für die weitere Prozessierung (Speicherung) sind die Attribute «year», womit das Jahr des zugrundeliegenden Kartenblattes in den Punktdaten gespeichert wird und «scale», womit der entsprechende Massstab als Ganzzahl ohne Tausender (1:25 000 als 25) gespeichert wird, beide im String-Format.[10]
Hinzugefügt wird ausserdem das Attribut «origin», sodass in den Punktdaten sichtbar ist, wie diese generiert wurden. Dieses wird im Python-Code nicht verwendet.
Es werden 80 % der Trainingspunkte jeweils mit «training» klassifiziert, diese werden für das Training verwendet. Die restlichen 20 % der Daten werden für die Validierung verwendet. In den Workflows wird dies beim Feld Modulo durch die Default-Expression «id%5=0» implementiert. Dies setzt voraus, dass ein «id»-Feld existiert.

4.1.1 Kartenbasierte Workflows

Als erstes sollen Workflows ohne Einbezug der digitalisierten Daten erstellt werden. Dies soll einerseits im Rahmen eines regelmässigen Gitternetzes und andererseits im Rahmen von Zufallspunkten in den bestehenden Kartenblattgrenzen umgesetzt werden. Damit ein Vergleich möglich ist, soll die Anzahl an Punkten übereinstimmen.
Allgemein gilt, dass für die entstehenden Workflows in QGIS das entsprechende Koordinatenreferenzsystem LV03 bzw. EPSG 21781 (epsg.io, 2019) verwendet werden sollte, um Probleme bei der Generierung von Punkten zu vermeiden. Dies wird als Grundannahme für die entsprechenden Workflows vorausgesetzt.

[9] Vgl. im Code von segmentation_manager.py Zeile 78
[10] Vgl. im Code von segmentation_manager.py Zeilen 47-48

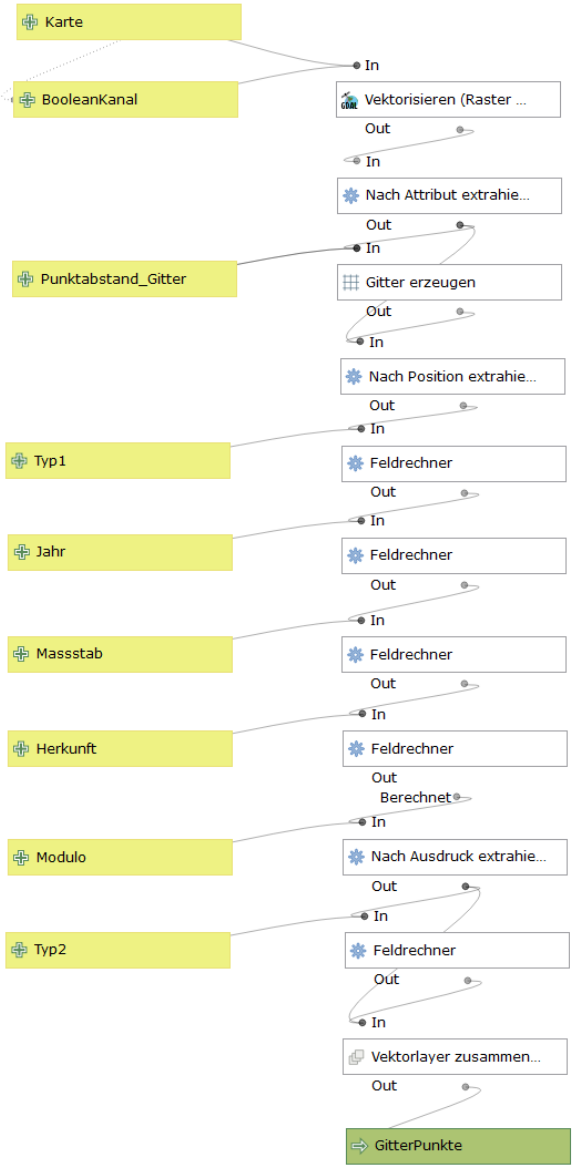


Abbildung 8: Gitterpunkte-Workflow (GP)

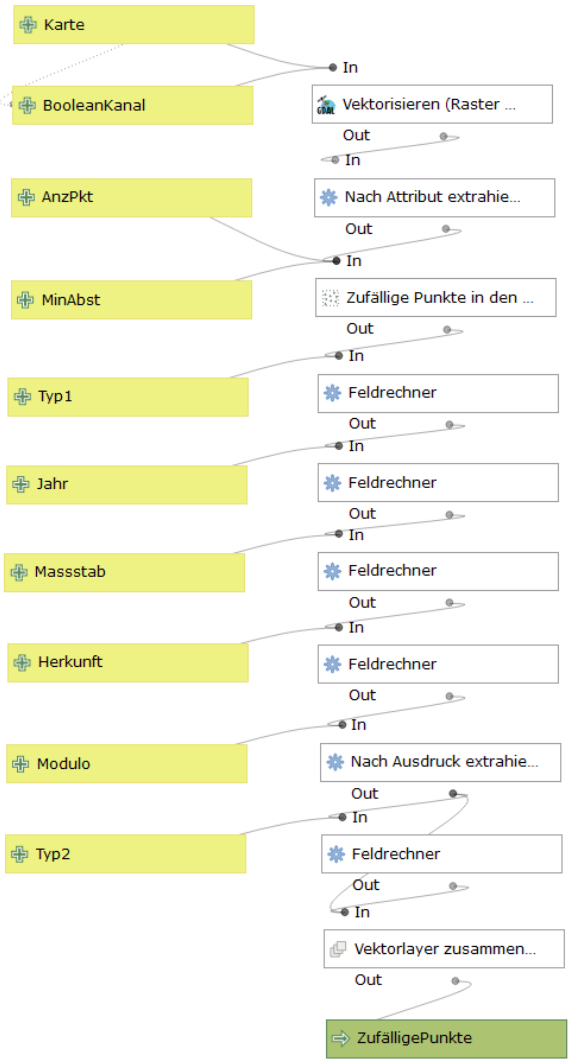


Abbildung 9: Workflow für zufällige Punkte (ZP und ZPmin227)

Punkte in einem regelmässigen Gitter (GP)

Als erster kartenbasierter Workflow dienen Punkte, welche in einem regelmässigen Gitter angeordnet sind. Dabei wird mit der Funktion «Gitter erzeugen» ein Punktegitter in der Ausdehnung des Layers generiert und nachfolgend werden die Punkte, welche innerhalb der Kartenblattgrenzen liegen, ausgeschnitten. Anschliessend werden den Punkten die notwendigen Attribute («scale», «type», «origin» und «year») hinzugefügt.

Parameter bei dieser Erstellung sind in Abbildung 8 zu sehen, die Anzahl an generierten Punkten wird dabei über den Gitterabstand definiert. Bei einem Abstand von 800 m und den Kartenausdehnungen von jeweils 6000 m auf 8750 m zwischen den Punkten ergeben sich so in den Kartenblattgrenzen 1357 Punkte. Dabei schwankt die Anzahl an Punktzeilen in Nord-Süd-Richtung zwischen 7 und 8. In West-Ost-Richtung bewegt sich die Anzahl an Punktspalten zwischen 10 und 11. Dies liegt daran, dass das Punktegitter jeweils nicht relativ zum Kartenblatt, sondern absolut ausgerichtet ist. Diese 1357 Punkte

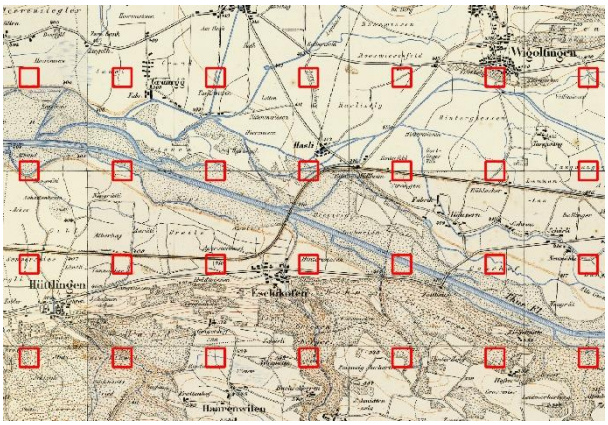


Abbildung 10: Kacheln um die mit dem Workflow GP erzeugten Trainingspunkte

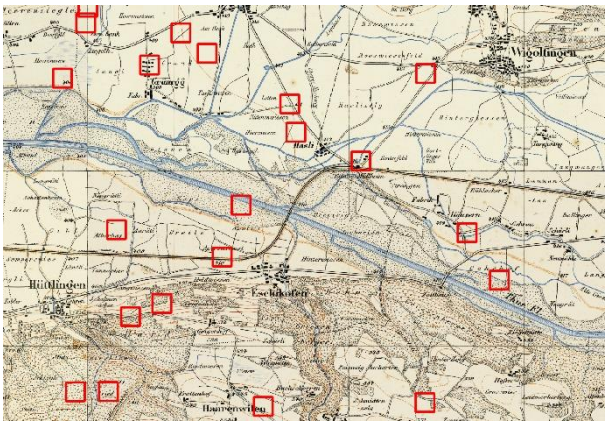


Abbildung 11: Kacheln um die mit dem Workflow ZP erzeugten Trainingspunkte

können für alle nachfolgenden Workflows als Referenzwert verwendet werden, sodass eine Vergleichbarkeit der Workflows gewährleistet ist.

Zufällige Punkte (ZP)

Der zweite kartenbasierte Workflow sind zufällige Punkte in der Ausdehnung der Kartenblattgrenzen. Für diese Aufgabe wird die Funktion «Zufällige Punkte in den Layergrenzen» verwendet. Dabei werden 1357 Punkte als Punktanzahl verwendet. Zusätzlich müssen hier wiederum entsprechende Parameter der Attributtabelle hinzugefügt werden (vgl. Abbildung 9).
Der Unterschied zwischen ZP und GP wird in Abbildung 10 und Abbildung 11 deutlich.

Zufällige Punkte mit Mindestabstand (ZPmin227)

Um einen Workflow von möglichst hoher Güte zu erhalten, muss die Vorgehensweise des Trainings von U-Net betrachtet werden. Dabei erstellt U-Net (wie beschrieben in Kapitel 2.2.5) 128 Pixel grosse Quadrate rund um die generierten Trainingspunkte. Bei der vorliegenden Rasterauflösung von 1.25 m / Pixel entspricht dies genau einer Länge von $h = 160\,m$. Wird davon ausgegangen, dass sich die Quadrate an keinem Punkt überlappen sollen (vgl. Abbildung 12), muss für die entsprechenden Punkte die Diagonaldistanz berechnet werden:

$$D = h * \sqrt{2} = 226.27\,m$$

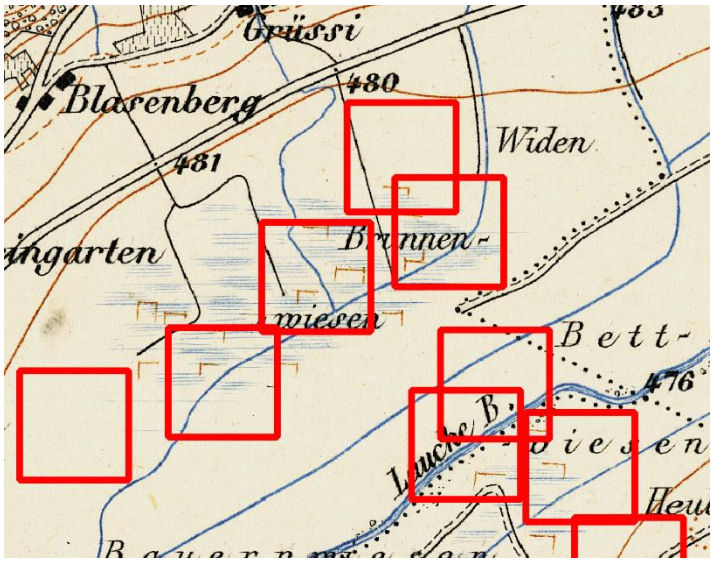


Abbildung 12: Sich teilweise überschneidende Quadrate der Grösse 160 x 160 m

Aufgerundet ergibt dies eine Mindestdistanz von 227 m zwischen verschiedenen Punkten, die durch die entsprechende Wahl der Parameter von «Zufällige Punkte in den Layergrenzen» erreicht werden kann. Ansonsten entspricht der Workflow ZPmin227 dem Workflow ZP.

4.1.2 Workflows auf Basis der digitalisierten Kartendaten

Im Unterschied zu den kartenbasierten Workflows bilden die Workflows auf Basis der digitalisierten Kartendaten nicht nur die Ausdehnung der Kartenblätter ab, denn die Datenpunkte werden gezielt an die entsprechenden Featureklassen angegliedert. Konkret bedeutet dies, dass im Rahmen der vorliegenden Daten die Datensätze «Stream», «Wetland» und «Riverlake» aus dem Jahre 1880 als Shapefile für die Erzeugung der Trainingspunkte verwendet werden.

Pufferpunkte (BP)

Als grundlegendes Beispiel für einen Workflow auf Basis der digitalisierten Kartendaten werden die drei vorliegenden Geometrien mit einer spezifischen Pufferdistanz gepuffert. Darauffolgend werden in der Ausdehnung der Vereinigung dieser drei so entstandenen Layer wiederum 1357 zufällige Punkte in den Layergrenzen hinzugefügt. Diese werden wiederum mit den entsprechenden Attributen versehen (vgl. Abbildung 15).
Dabei ist es wichtig, nach Anwendung der Funktion «Zufällige Punkte in den Layergrenzen» die Anzahl der generierten Punkte nochmals manuell zu prüfen, da nicht in jedem Fall die richtige Punktanzahl resultiert (vgl. Kapitel 5.2).
Die Pufferdistanz wird auf 80 m gesetzt, da dieser Wert genau der Hälfte einer Quadratseite entspricht. Somit kann geometrisch sichergestellt werden, dass ein Punkt im Extremfall genau an der Grenze seines Trainingsquadrates liegt und die umgebende Kachel noch entsprechende Featuredaten umfasst.

Pufferpunkte mit Mindestabstand (BPmin227)

Wie bei den zufälligen Punkten soll für die Pufferpunkte überprüft werden, ob ein Mindestabstand das Ergebnis der Vorhersage mit U-Net positiv oder allenfalls negativ beeinflusst. Insofern wird dem Workflow wiederum ein Parameter hinzugefügt, der den Mindestabstand beschreibt. Dabei wird auf den gleichen Mindestabstand zurückgegriffen wie bei den zufälligen Punkten (227 m). Auch die Pufferdistanz bleibt mit 80 m unverändert.
Ein Vergleich zwischen BP und BPmin227 ist in Abbildung 13 und Abbildung 14 ersichtlich.

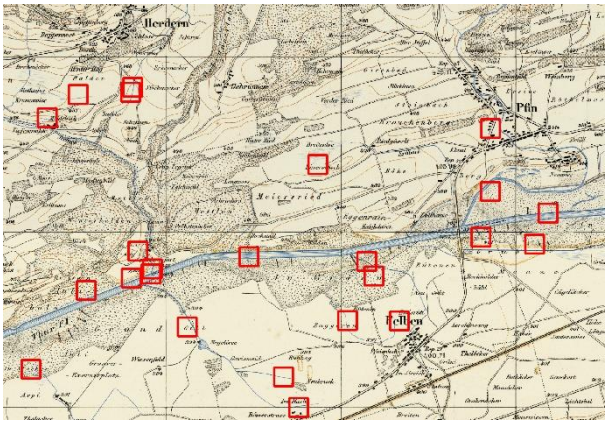


Abbildung 13: Kacheln um die mit dem Workflow BP erzeugten Trainingspunkte

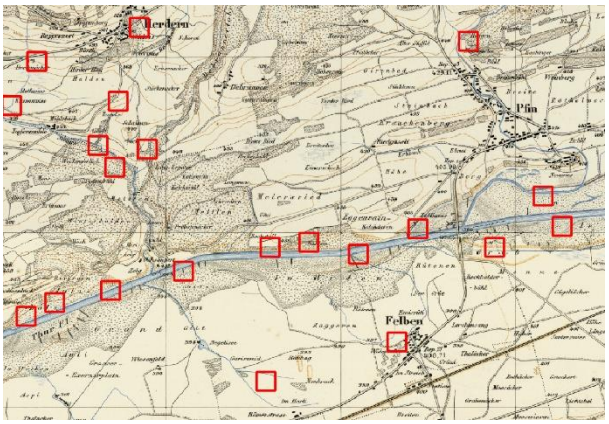


Abbildung 14: Kacheln um die mit dem Workflow BPmin227 erzeugten Trainingspunkte

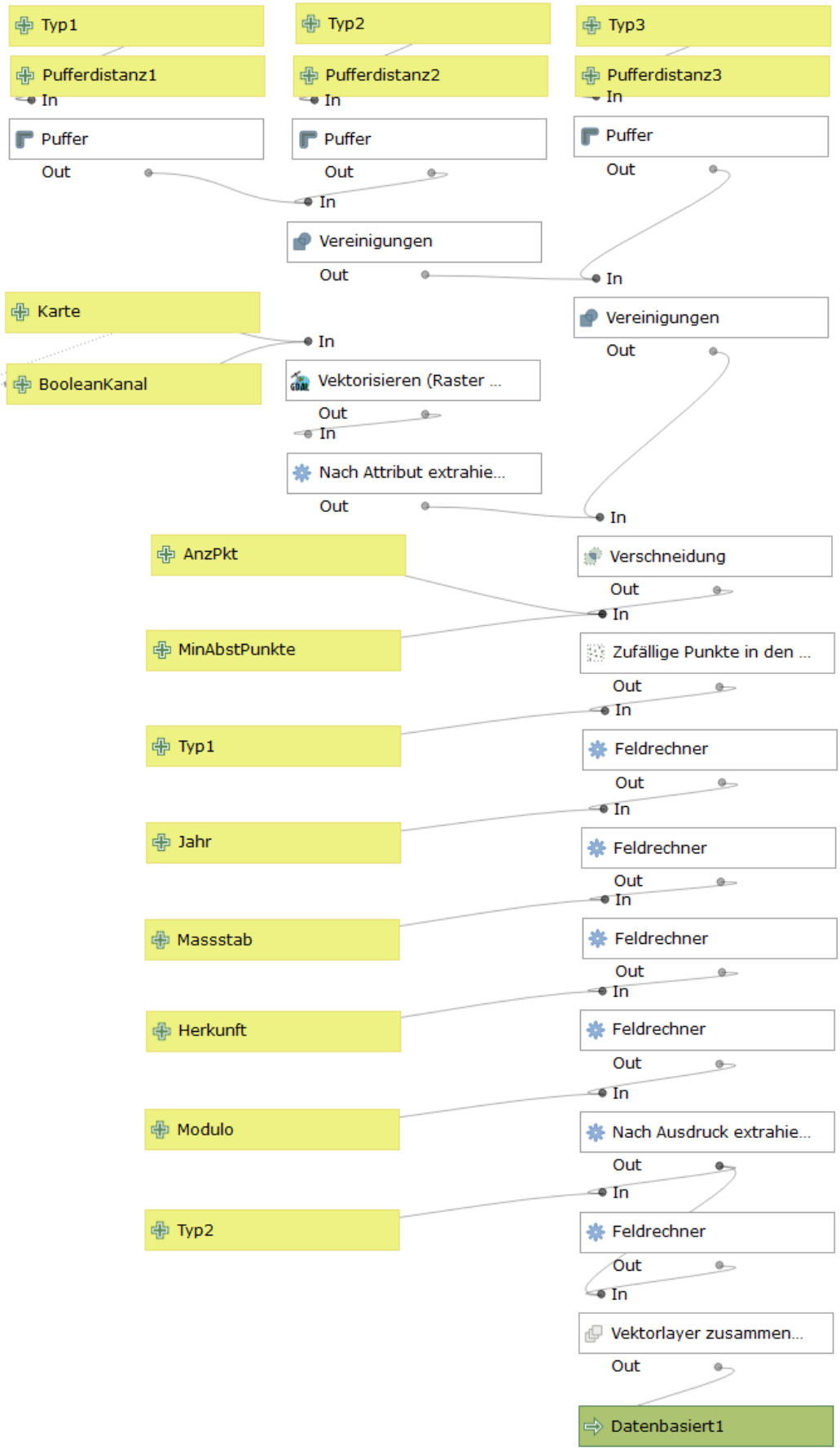


Abbildung 15: Pufferpunkte-Workflows (BP und BPmin227)

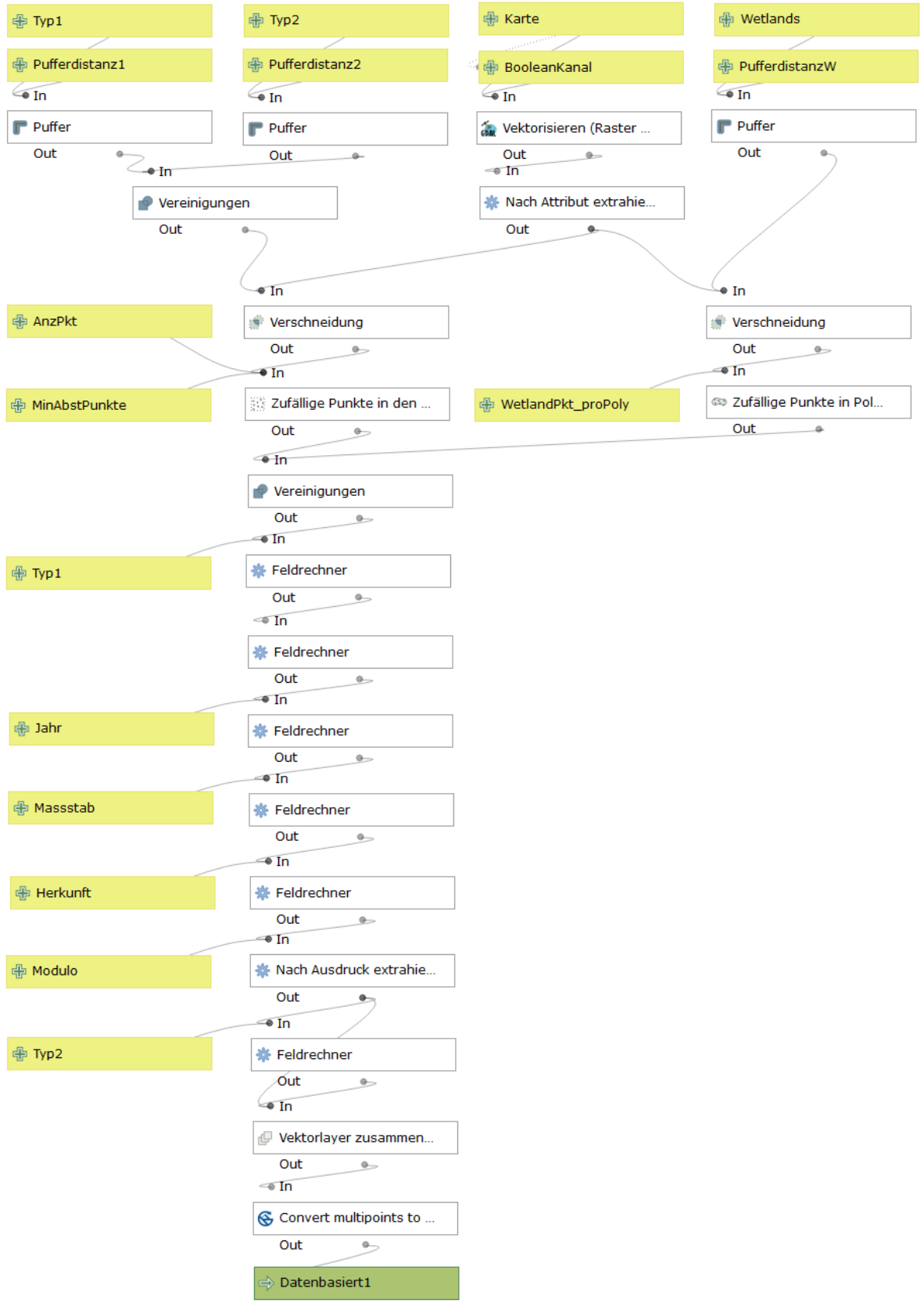


Abbildung 16: Wetlandpunkte-Workflows (WP3 und WP6)

4.1.3 Workflows auf Basis der digitalisierten Kartendaten mit Ergebniseinbezug

Im Rahmen eines iterativen Prozesses ist bereits bei ersten Evaluationsergebnissen erkennbar, bei welchen Featureklassen die Extraktion besser funktioniert. Dies widerspiegelt sich in höheren Werten bei den verschiedenen Metriken.

Wetland-basierte Punkte mit Faktor 3 (WP3)

Konkret werden die Ergebnisse bei der Featureklasse Wetland einbezogen (vgl. Kapitel 4.4). Aus diesem Grund werden beim vorliegenden Workflow die Punkte im Umkreis von Wetland explizit erzeugt, sodass für die Bereiche mit Wetland sichergestellt ist, dass ausreichend Trainingsdaten zur Verfügung stehen. Insbesondere sollen dabei sowohl grosse als auch kleine Wetland-Gebiete adäquat einbezogen werden, damit die Umsetzung nicht über die Funktion «Zufällige Punkte in den Layergrenzen», sondern über die Funktion «Zufällige Punkte in Polygonen» erfolgt (vgl. Abbildung 16). Dies hat den Vorteil, dass die Anzahl der Punkte pro Polygon deterministisch ist. Da die Anzahl an Polygonen der Grundlagedaten (134) konstant ist, verändert sich auch die Anzahl der Punkte für alle Wetland-Polygone nicht. Der im Titel beschriebene Faktor beschreibt die Anzahl an Wetland-Punkten pro Polygon und wird für die Berechnung der Anzahl der restlichen, zu generierenden zufälligen Punkte im Perimeter der weiteren Featureklassen (Stream, Riverlake) benötigt. Im konkreten Fall berechnet sich die Anzahl als

$$n_{rest} = 1357 - Faktor * 134 = 955.$$

Bei diesem Workflow werden die Punkte mittels zweier getrennter QGIS-Funktionen erstellt. Da bei der Zuweisung des Attributes «type» aber ein gemeinsames Feld «id» benötigt wird, ist noch ein zusätzlicher Schritt zur Zuweisung eines solchen Feldes notwendig.
Aufgrund der Vereinigung zweier verschieden generierter Punktlayer müssen die Punkte vom Typ «multipoint» zum Typ «point» konvertiert werden.
Im Rahmen dieses Workflows wird auf einen Mindestabstand zwischen den Punkten verzichtet.

Wetland-basierte Punkte mit Faktor 6 (WP6)

Damit für den WP3-Workflow eine Vergleichsbasis erzeugt werden kann, wird der gleiche Workflow nochmals verwendet und es wird ein Punktlayer mit 6 Punkten pro Wetland-Polygon erzeugt (vgl. Abbildung 19). Mathematisch bedeutet dies:

$$n_{rest} = 1357 - Faktor * 134 = 553$$

Punkte basierend auf zwei Featureklassen (2F)

Bei genauer Betrachtung der Daten ist ersichtlich, dass sich die Klasse Stream eher gleichmässig über die gesamten Kartenblätter verteilt. Aus diesem Grund werden beim vorliegenden Workflow nur zwei Featureklassen berücksichtigt, einerseits Wetland und andererseits Riverlake (vgl. Abbildung 17). Insbesondere in der Nähe von Wetland befinden sich oft Bäche. Bäche wiederum münden in Flüsse. Deshalb soll beim vorliegenden Workflow durch das Weglassen von verteilt vorhandenen Featureklassen wie Stream bei den restlichen Featureklassen eine höhere Genauigkeit erreicht werden.
Dies bedeutet, dass der BP-Workflow so modifiziert wird, dass Stream nicht mehr einbezogen wird. Ausserdem muss aufgrund der kleineren so entstehenden Polygone die Pufferdistanz auf 100 m erhöht werden, da sonst im Regelfall nicht genügend zufällige Punkte innerhalb des Ergebnislayers erzeugt werden können (vgl. Kapitel 5.2).

Mischpunkte (MP)

Im Rahmen einer vorgängigen Analyse der Workflows auf Basis der digitalisierten Kartendaten ist feststellbar, dass bei der Featureklasse «Wetland» oft zu viele Pixel detektiert werden. Dies äussert sich in einem im Vergleich zur Precision stark erhöhten Recall. Konkret bedeutet dies, dass ein grosser Teil der Feuchtgebiete (Wetland) erkannt wird, aber ein Grossteil der detektierten Feuchtgebiete in der Realität gar keine Feuchtgebiete sind (für entsprechende Zahlenwerte vgl. Kapitel 4.4).

Aus diesem Grund wird beim vorliegenden Workflow (Mischpunkte) darauf geachtet, dass für die Trainingsdaten genügend negative Beispiele beziehungsweise verteilte Punkte vorliegen, bei welchen keine Features auftreten, womit das neuronale Netzwerk «lernen» kann, nicht überall Wetland zu detektieren. Dies wird wiederum wie beim WP-Workflow durch eine Mischung von zwei generierten Punktesets erzeugt, einerseits ZP und andererseits BP (vgl. Abbildung 18). Konkret bestehen die 1357 MP-Trainingspunkte aus 1200 BP-Punkten und 157 ZP-Punkten. Das Verhältnis wird einseitig gewählt, da die zufälligen Punkte ansonsten die Ergebnisse der anderen Featureklassen zu stark beeinflussen könnten.

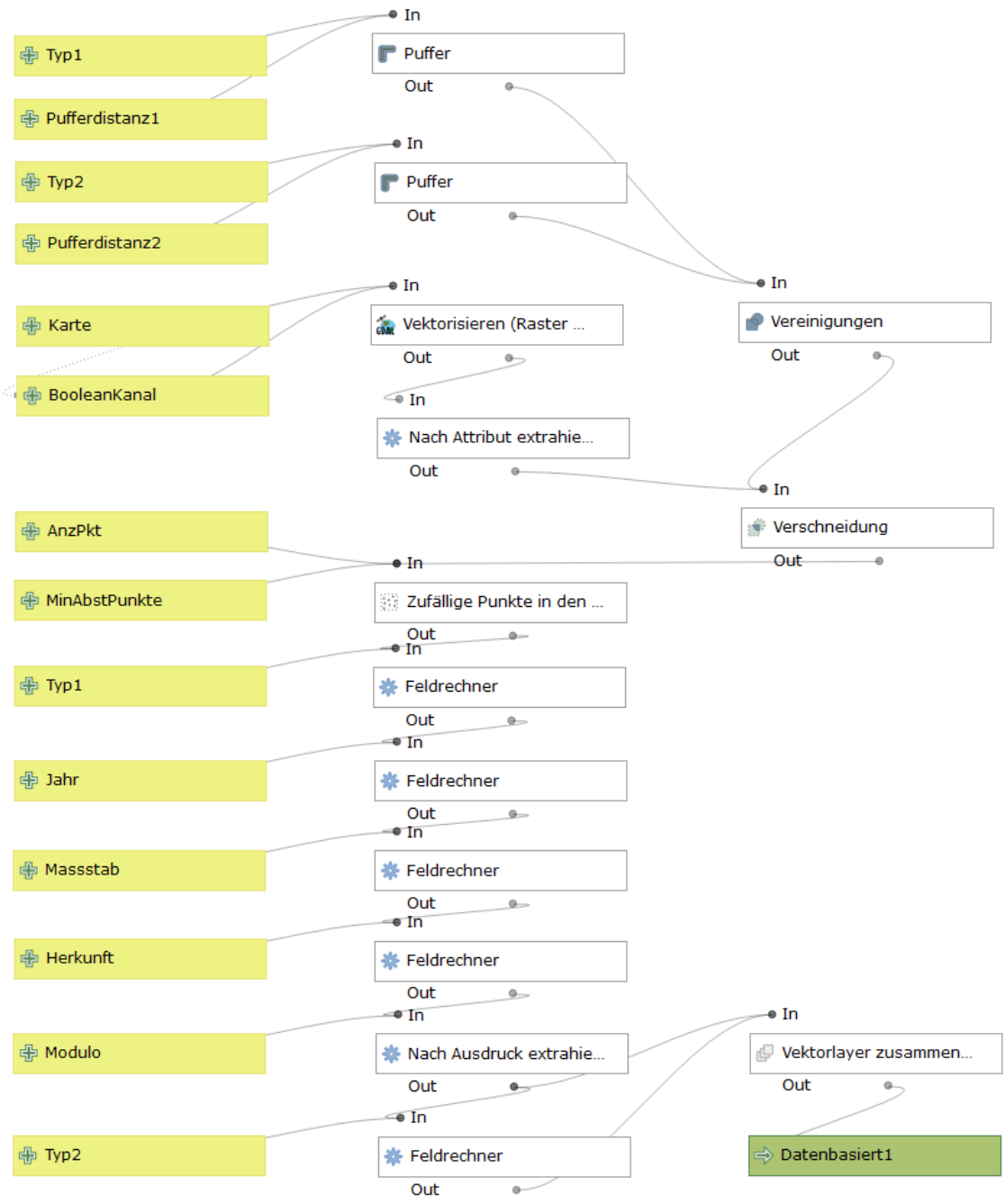


Abbildung 17: Workflow für die Zwei-Feature-Punkte (2F)

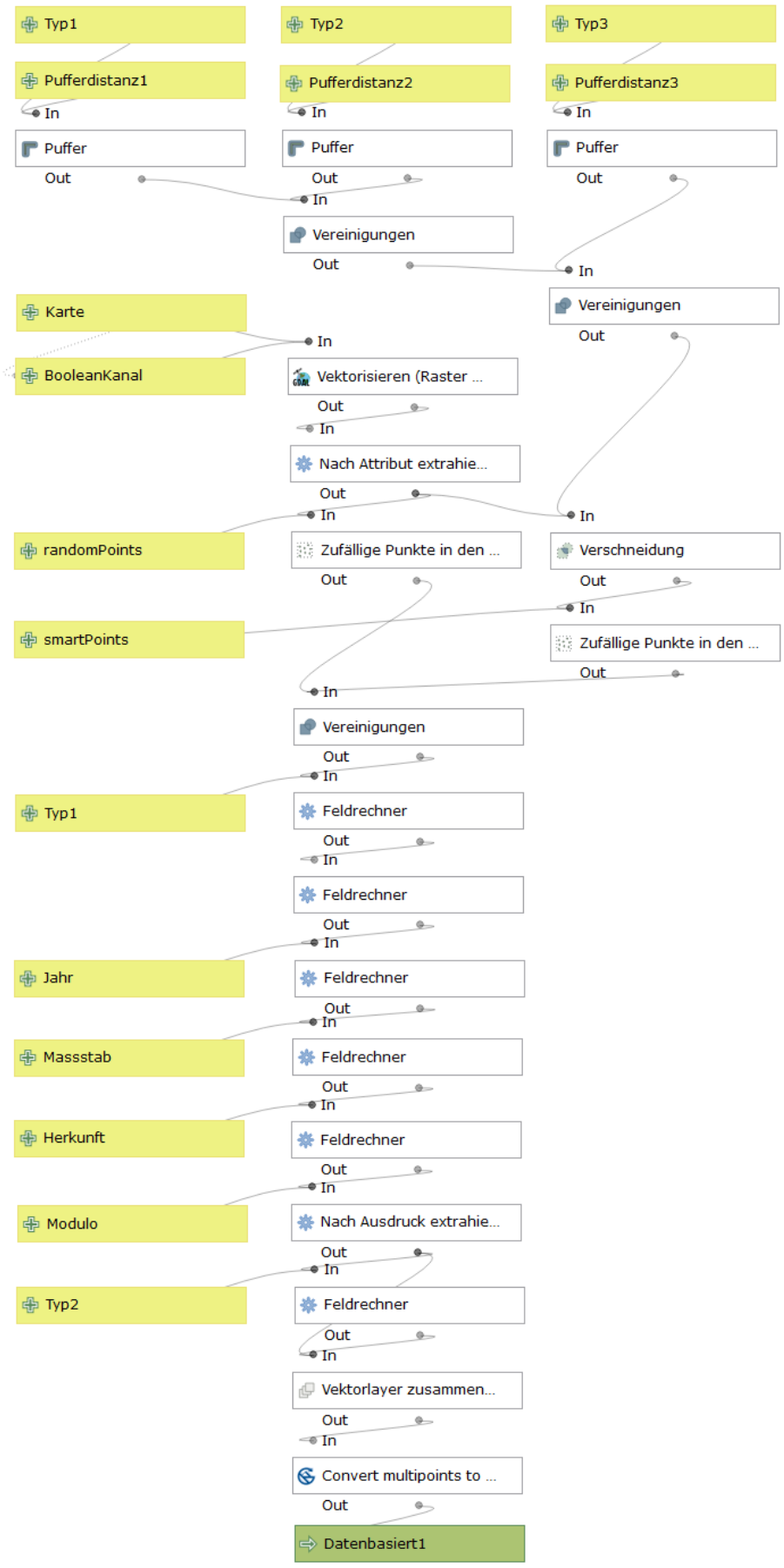


Abbildung 18: Workflow für die Mischpunkte (MP)

4.1.4 Workflows auf Basis von Fehlern

Bereits bei den Workflows auf Basis der digitalisierten Kartendaten werden bestehende Ergebnisse in die Workflowfindung miteinbezogen, dies aber immer nur auf Workflow-Ebene und nicht im Rahmen der verwendeten Daten. Die Workflows werden an den Ergebnissen orientiert, die Grundlagedaten bleiben aber die gleichen. Anders ist dies bei den nachfolgend beschriebenen Ansätzen, welche als fehlerbasiert bezeichnet werden können:
Hierbei wird aus den Fehlern bezüglich des Layers Wetland in QGIS ein neuer Polygondatensatz «Wetlandfehler» erzeugt, der die Bereiche beinhaltet, welche beim Workflow 2F angewendet auf die Trainingsdaten fälschlicherweise detektiert werden (detektiert, aber in der Realität nicht vorhanden). Dafür werden für die Kartenblätter des Jahres 1880 durch das erstellte Modell des Workflows 2F Vorhersagen bezüglich aller Featureklassen erzeugt. Diese werden über ein vrt-File (analog zur Speicherung der Grundlagekarten) miteinander verknüpft und durch die bestehenden Einstellungen automatisch mit einem Schwellenwert versehen, sodass Werte über 0.5 als 1 und Werte unter 0.5 als 0 angezeigt werden. Von diesem so entstandenen binären Rasterbild pro Kanal wird der «Wetland-Kanal» extrahiert. Aufgrund des Rechenaufwandes wird eine tiefere Rasterauflösung verwendet (10 m statt 1.25 m, da Berechnungen sonst mehrere Stunden dauern). Für das Ergebnis sind nur Regionen mit vielen Fehlern wichtig, deshalb hat diese tiefere Auflösung keine Nachteile zur Folge. Um ausserdem die Polygonisierung effizienter zu gestalten, wird der entstandene Datensatz mit der Funktion «Sieben» bereinigt, sodass nur Regionen von mindestens 20 Rasterpixeln in einer 8-Nachbarschaft polygonisiert werden. Nach diesem Bereinigungsschritt erfolgt der eigentliche Polygonisierungsschritt. Die so entstandenen Polygone werden anschliessend repariert und es wird eine Überlappungsanalyse bezüglich des Wetland-Layers (Groundtruthdaten) durchgeführt. Alle Polygone, die sich in der Realität mit einem Wetland-Polygon überschneiden, werden durch eine Extraktion eliminiert, sodass nur fälschlicherweise detektierte Regionen im Polygondatensatz «Wetlandfehler» verbleiben.

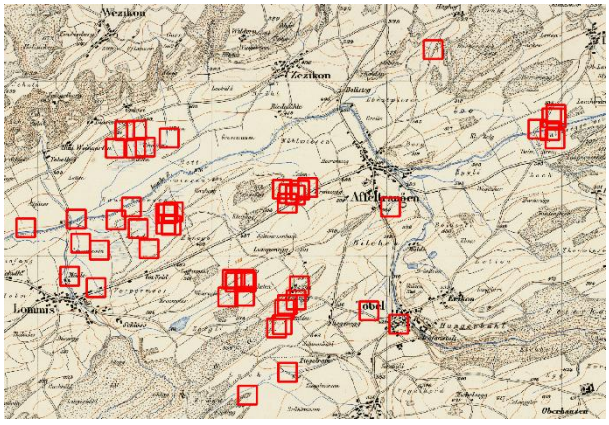


Abbildung 19: Kacheln um die mit dem Workflow WP6 erzeugten Trainingspunkte

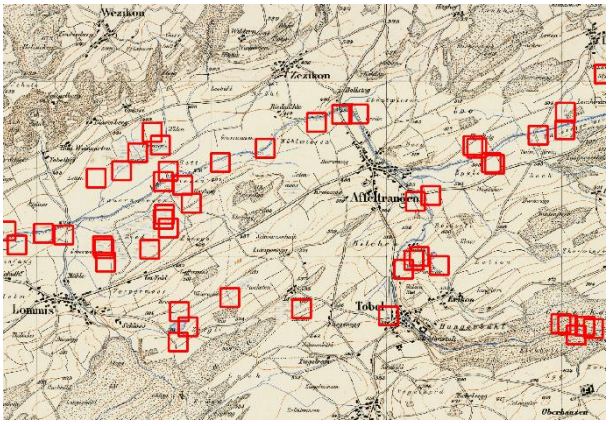


Abbildung 20: Kacheln um die mit dem Workflow EP2F erzeugten Trainingspunkte

Fehlerbasierte Pufferpunkte (EP)

Basierend auf den bisher verwendeten Datensätzen und dem hinzugekommenen Datensatz «Wetlandfehler» werden darauffolgend Pufferpunkte mit zusätzlichen Punkten im Bereich der «Wetlandfehler» erstellt. So soll das Netzwerk «lernen», dass bei diesen Punkten kein Wetland vorliegt, obwohl diese Bereiche mit der Instanz und dem Modell vom Workflow 2F als Wetland detektiert wurden. Der Layer Wetlandfehler umfasst 80 Polygone. Es soll pro Polygon ein Punkt generiert werden, was bedeutet, dass sich die fehlerbasierten Punkte (error-based points, EP) aus 80 Wetlandfehlerpunkten und 1277 Pufferpunkten zusammensetzen (vgl. Abbildung 21). Alle vorkommenden Layer werden ausserdem mit 80 m gepuffert (wie bei BP).

Fehlerbasierte Punkte basierend auf zwei Featureklassen (EP2F)

Bezüglich des Workflows, der auf zwei Featureklassen basiert, soll ebenfalls ein fehlerbasierter Workflow erstellt werden. Dabei wird dieser analog zum EP-Workflow erstellt, im Unterschied zu EP aber auf Basis des 2F-Workflows. Im Rahmen der Arbeit wird eine Instanz mit einer Pufferdistanz von 80 m generiert, während die Funktion «Zufällige Punkte in den Layergrenzen» bei den weiteren Instanzen nicht zuverlässig terminiert, weshalb dort eine Pufferdistanz von 100 m gewählt wird. Ein Beispiel für EP2F befindet sich in Abbildung 20, der entsprechende Workflow ist in Abbildung 22 dargestellt.

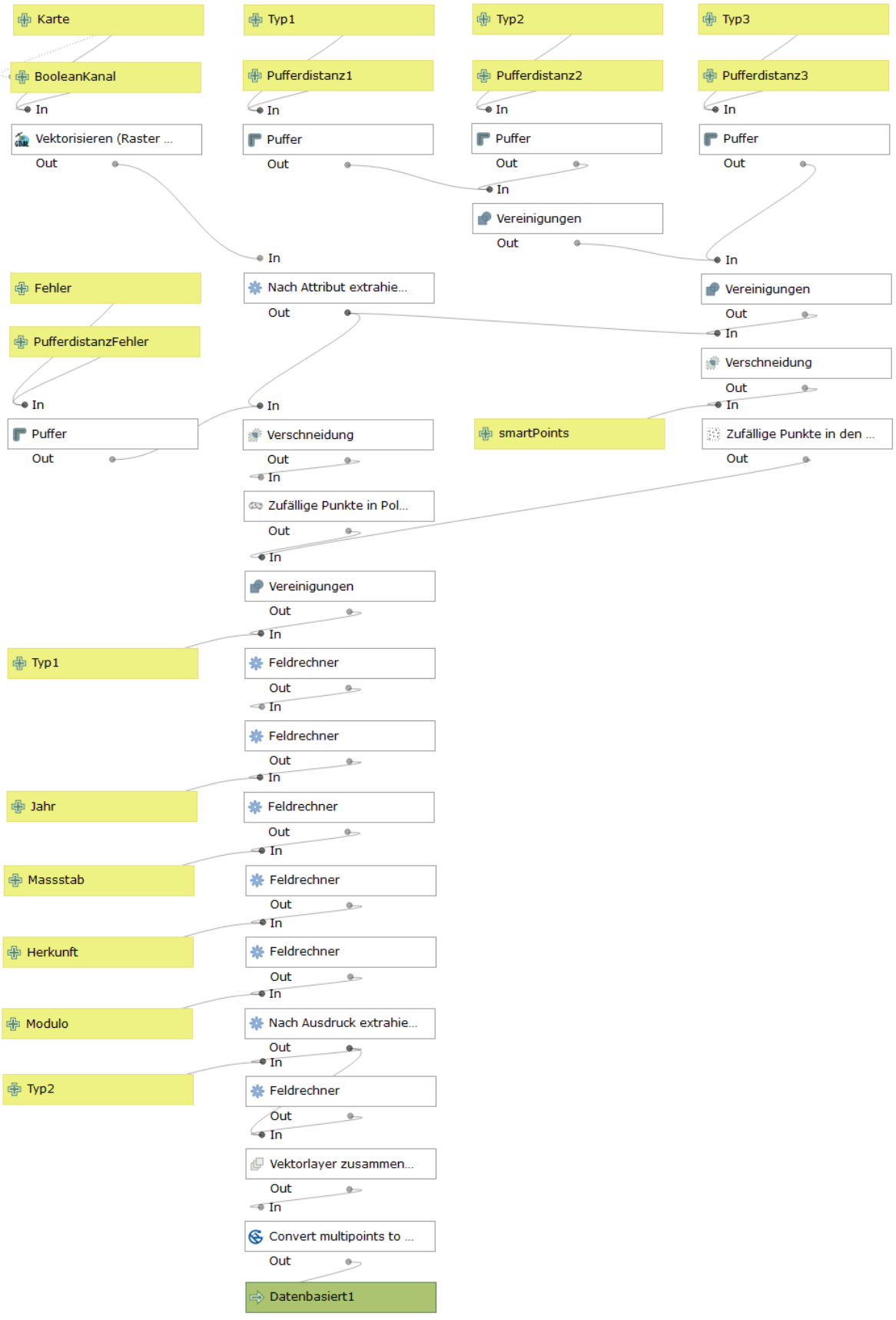


Abbildung 21: Workflows für die fehlerbasierten Pufferpunkte EP

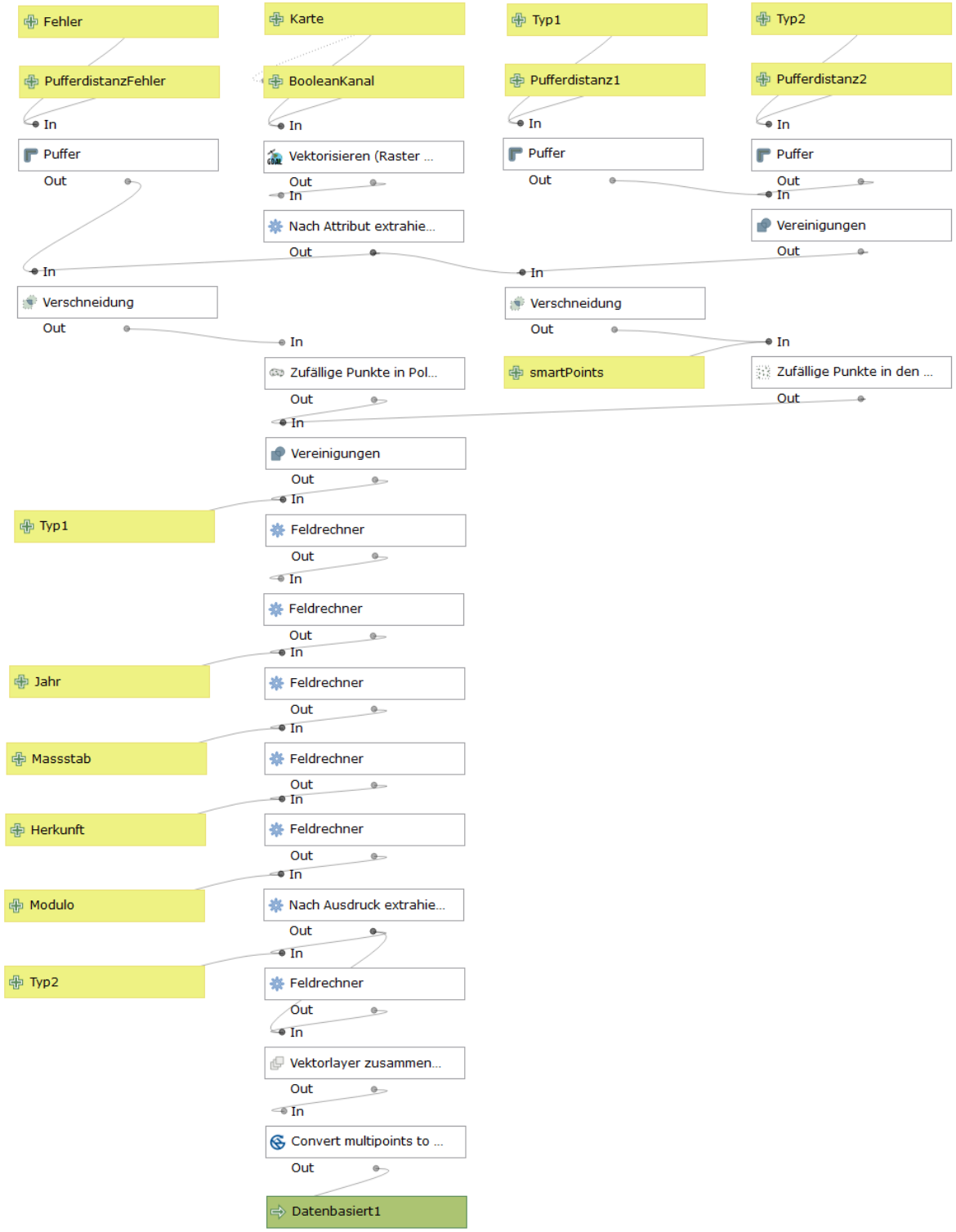


Abbildung 22: Workflows für die fehlerbasierten Zwei-Feature-Punkte (EP2F)

4.1.5 Unechte Workflows

Bei den Workflows 2F und EP2F werden nach dem Training dreier Instanzen die entstehenden Vorhersagen als TIF-Files mithilfe des QGIS-Workflows in Abbildung 23 gemischt, sodass eine neue Vorhersage entsteht. Dabei wird dieser für alle Kartenblätter als Batch-Prozess ausgeführt. Hierbei enzstehen die unechten Workflows 2F_mix und EP2F_mix.

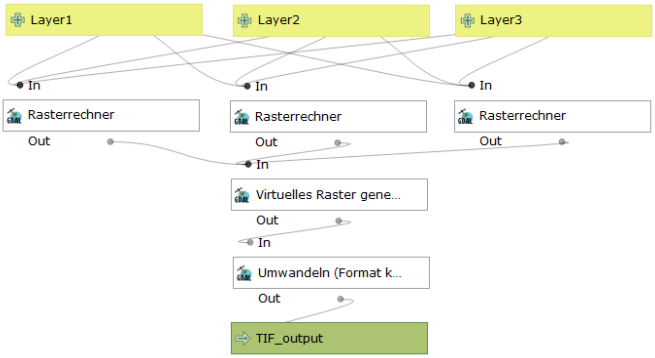


Abbildung 23: Workflow zur Generierung unechter Workflows

4.1.6 Übersicht über die Workflowgruppen

Tabelle 2 bietet eine Übersicht über die bestehenden Workflowgruppen. Dabei werden die unechten Workflows jeweils den zugrundeliegenden anderen Workflowgruppen zugeordnet. Verschiedene Instanzen desselben Workflows werden hierbei mit einem Unterstrich und dem entsprechenden Index benannt (bspw. ZP_1, ZP_2 und ZP_3).
Der Einbezug der Ergebnisse besteht bei der dritten Gruppe darin, dass bereits Trainingsergebnisse vorliegen beziehungsweise die Daten bezüglich der Verteilung ihrer Featureklassen betrachtet werden, während bei der zweiten Gruppe alle drei Featureklassen gleich behandelt werden.

Kartenbasiert	Datenbasiert ohne Einbezug von Ergebnissen	Datenbasiert mit Einbezug von Ergebnissen	Fehlerbasiert
ZP_1	BP	WP3	EP
ZP_2	BPmin_1	WP6	EP2F_1
ZP_3	BPmin_2	2F_1	EP2F_2
ZPmin227_1	BPmin_3	2F_2	EP2F_3
ZPmin227_2		2F_3	EP2F_mix
ZPmin227_3		2F_mix	
GP		MP	

Tabelle 2: Einteilung in Workflowgruppen

4.2 Workflowgüte

Im Rahmen dieser Arbeit werden verschiedene Dimensionen sowie Parameter erfasst und verändert. Als Ergebnis der Evaluation mit den verschiedenen Metriken stehen Werte als Funktion von mindestens vier Dimensionen:

- Workflow
- Datentyp (Stream, Wetland oder Riverlake)
- Kartenblatt (10 Kartenblätter) oder Mittelwert (harmonisch und arithmetisch)
- Verwendete Metrik

Diese Ergebnisse werden auf Basis der Kommandozeilenausgabe im Rahmen des bestehenden Frameworks in ein xls-File kopiert, sodass dieses mithilfe von Matlab ausgelesen, in eine 4D-Matrix gespeichert und nachfolgend ausgewertet werden kann. Insbesondere werden darauffolgend zur Sicherstellung und Überprüfung noch einige Testcases durchgeführt, sodass Fehler bei der Verarbeitung ausgeschlossen werden können.

Insgesamt liegen 23 verschiedene Evaluationen vor, von welchen zwei Mischungen aus vorhergesagten TIF-Files darstellen. Diese werden aus Gründen der Konsistenz jeweils auch «Workflow» genannt. Die übrigen 21 Vorhersagen basieren auf acht verschiedenen Workflows mit entweder unterschiedlichen Parametern oder aber verschiedenen Instanzen mit gleichen Parametern.
Nachfolgend sollen diese einerseits gesamthaft und andererseits pro Kartenblatt verglichen werden. Für die Erstellung des Empfehlungsrahmens sollen aber auch spezifische Workflows beziehungsweise Workflowgruppen gegeneinander abgewogen werden.
Beim Gesamtvergleich werden aus der vierdimensionalen Matrix alle Kartenblätter gleichgewichtet gemittelt[11], sodass die nachfolgend in den Tabellen (Tabelle 3, Tabelle 4 und Tabelle 5) vorkommenden Werte jeweils den Mittelwert einer spezifischen Metrik über alle Kartenblätter beschreiben. Dabei werden bei der Featureklasse Wetland die Kartenblätter «TA_114_1879» und «TA_116_1879» übersprungen, da diese in den Groundtruthdaten keine Features aufweisen. Bei der Berechnung der Metriken Precision, Recall, F1 und Jaccard ergibt dies automatisch einen Wert von 0, ungeachtet der Güte der Detektion, weshalb dieses Überspringen notwendig wird.
In den nachfolgenden Tabellen werden jeweils die Workflows mit dem höchsten Wert pro Spalte grün und diejenigen mit dem zweit- bzw. dritthöchsten Wert blau markiert. Die tabellierten Werte sind gerundet, der Vergleich erfolgt aber ungerundet. In den Ausschnitten aus den Trainingsdaten, die folgend zur Illustration verwendet werden, beschreiben rote Gebiete immer Stream, grün steht für Wetland und blau für die Kategorie Riverlake.

4.3 Gesamtvergleich Stream

Workflow	Precision	Recall	F1	Accuracy	Jaccard	Av. Precision
ZP_1	0.507	0.141	0.199	0.996	0.115	0.430
ZP_2	0.806	0.758	0.780	0.998	0.643	0.849
ZP_3	0.627	0.781	0.687	0.998	0.534	0.744
ZPmin227_1	0.821	0.006	0.011	0.996	0.006	0.476
ZPmin227_2	0.697	0.646	0.667	0.998	0.509	0.710
ZPmin227_3	0.836	0.757	0.792	0.998	0.659	0.854
GP	0.595	0.672	0.610	0.998	0.457	0.671
BP	0.815	0.824	0.818	0.999	0.695	0.892
BPmin227_1	0.768	0.827	0.792	0.999	0.660	0.867
BPmin227_2	0.827	0.820	0.823	0.999	0.700	0.890
BPmin227_3	0.807	0.763	0.783	0.999	0.646	0.861
WP3	0.787	0.828	0.806	0.999	0.678	0.875
WP6	0.723	0.858	0.781	0.998	0.646	0.845
2F_1	0.821	0.736	0.775	0.998	0.634	0.844
2F_2	0.904	0.438	0.576	0.998	0.414	0.807
2F_3	0.832	0.792	0.811	0.999	0.684	0.882
2F_mix	0.897	0.704	0.788	0.999	0.652	0.902

[11] Dabei könnte auch argumentiert werden, dass eine Flächengewichtung bezüglich Features notwendig ist. Dies würde aber dazu führen, dass Seen und grosse Wetlandflächen die einzigen «wichtigen» Featureklassen für die Metrikwerte wären, weshalb darauf verzichtet wird.

MP	0.791	0.812	0.800	0.999	0.668	0.874
EP	0.811	0.708	0.754	0.998	0.609	0.831
EP2F_1	0.844	0.789	0.815	0.999	0.689	0.892
EP2F_2	0.814	0.812	0.812	0.999	0.685	0.886
EP2F_3	0.801	0.809	0.804	0.999	0.673	0.875
EP2F_mix	0.863	0.818	0.839	0.999	0.724	0.919

Tabelle 3: Gesamtvergleich Stream

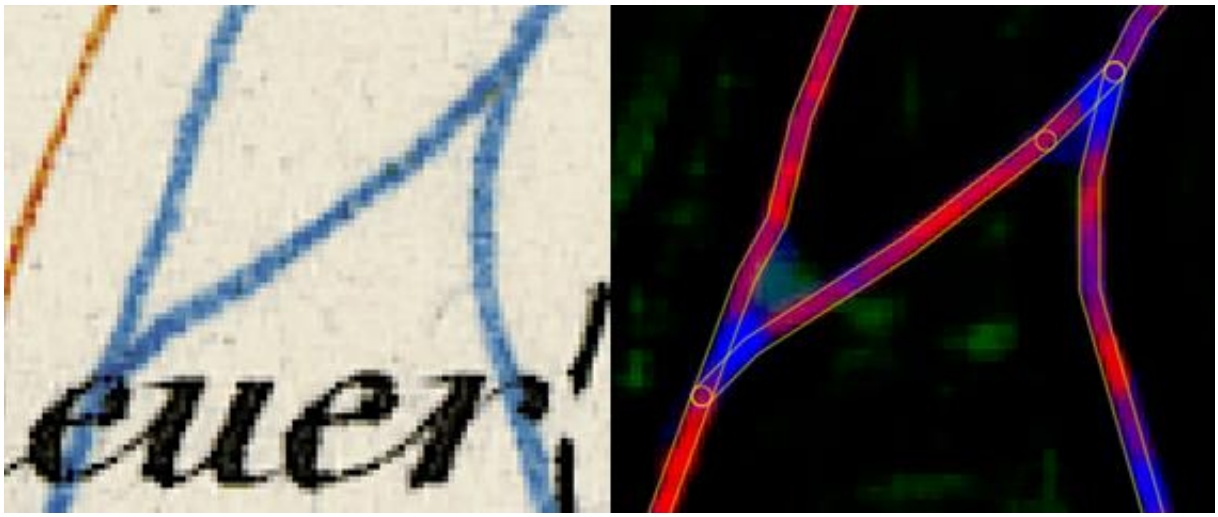


Abbildung 24: Detail der Extraktion von Stream. Gut zu erkennen ist die "Verwechslung" mit Riverlake bei Abzweigungen und breiten Stellen. Die Groundtruthdaten werden durch den schwachen gelben Rand im rechten Bildteil sichtbar.

Bei gesamthaftem Vergleich aller Workflows und Metriken bezüglich der Featureklasse Stream fällt auf, dass der Workflow EP2F_mix bezüglich vier Metriken den höchsten Wert aufweist. Nur bezüglich der Metriken Precision und Recall wird dieser Workflow durch die Workflows 2F_2 und WP6 übertroffen. Bezüglich der Kombination dieser zwei Metriken, dem F1-Score, ist aber der Workflow EP2F_mix am besten. Insofern kann daraus geschlossen werden, dass dieser Workflow sich am besten für die Detektion von Stream eignet. Bei EP2F_mix sind 86,3 % der detektierten Pixel richtig und 81,8 % der richtigen Pixel werden detektiert. Auch BP-Workflows sind zur Detektion von Stream gut geeignet, da sich davon zwei bezüglich der Metriken F1 und Jaccard unter den ersten drei befinden. Insofern überraschend ist das Ergebnis bezüglich der Güte von EP2F_mix, da Stream bei der Generierung der entsprechenden Trainingspunkte gar nicht einbezogen wurde. EP2F_mix setzt sich aus Wetland, Riverlake und den Wetlandfehlern zusammen. Dies kann dahingehend interpretiert werden, als dass der Einbezug von Stream für die Generierung der Trainingsdaten nicht notwendig ist, da in einem quadratischen Ausschnitt mit Riverlake und Wetland dementsprechend ausreichend Stream-Beispiele auftreten.
Beim Vergleich der Groundtruthdaten mit den detektierten Daten in einem zufällig ausgewählten Ausschnitt aus EP2F_mix fällt auf, dass die räumliche Ausdehnung der detektierten Daten sehr gut übereinstimmt, bei Verzweigungen von Bächen aber oft fälschlicherweise Riverlake detektiert wird (vgl. Abbildung 24). Dies erklärt zudem die hohen Precision-Werte, da die Detektion von Stream in bis zu 90.4 % der Fälle (Workflow 2F_2) korrekt ist. Die Recall-Werte sind aufgrund dieser Falschdetektion etwas tiefer und liegen bei maximal 85.8 % beim Workflow WP6.

4.4 Gesamtvergleich Wetland

Workflow	Precision	Recall	F1	Accuracy	Jaccard	Av. Precision
ZP_1	0.000	0.000	0.000	0.987	0.000	0.034
ZP_2	0.000	0.000	0.000	0.987	0.000	0.109
ZP_3	0.125	0.000	0.000	0.987	0.000	0.066
ZPmin227_1	0.000	0.000	0.000	0.987	0.000	0.112
ZPmin227_2	0.000	0.000	0.000	0.987	0.000	0.079
ZPmin227_3	0.361	0.757	0.409	0.974	0.302	0.461
GP	0.653	0.085	0.129	0.988	0.073	0.257
BP	0.392	0.759	0.428	0.977	0.323	0.525
BPmin227_1	0.400	0.696	0.423	0.980	0.319	0.433
BPmin227_2	0.378	0.716	0.406	0.978	0.310	0.439
BPmin227_3	0.617	0.020	0.035	0.988	0.018	0.209
WP3	0.372	0.888	0.441	0.971	0.348	0.562
WP6	0.283	0.871	0.344	0.971	0.267	0.522
2F_1	0.371	0.755	0.422	0.990	0.324	0.518
2F_2	0.273	0.532	0.290	0.972	0.196	0.327
2F_3	0.384	0.754	0.421	0.975	0.319	0.546
2F_mix	0.469	0.728	0.471	0.980	0.367	0.571
MP	0.391	0.679	0.404	0.975	0.296	0.436
EP	0.473	0.192	0.208	0.978	0.126	0.284
EP2F_1	0.429	0.783	0.466	0.975	0.359	0.524
EP2F_2	0.425	0.822	0.479	0.975	0.372	0.594
EP2F_3	0.257	0.805	0.322	0.971	0.236	0.470
EP2F_mix	0.500	0.818	0.530	0.975	0.426	0.621

Tabelle 4: Gesamtvergleich Wetland

Bezüglich der Feuchtgebiete (Wetland) ist auffallend, dass die Ergebniswerte der verschiedenen Metriken im Vergleich zu den Extraktionen bei Stream und Riverlake tiefer sind. Daraus kann geschlossen werden, dass es für das neuronale Netzwerk «schwieriger» ist, Wetland erfolgreich zu detektieren. Während der beste F1-Score bei Riverlake und Stream im Bereich zwischen 80 % und 90 % ist, liegt das Maximum für Wetland bei 53 %. Insbesondere ist ersichtlich, dass die Werte Recall, Jaccard und deshalb auch F1 für fünf der sechs zufälligen Instanzen (ZP_1, ZP_2, ZP_3, ZPmin227_1 und ZPmin227_2) 0 sind. Deshalb ist ZP für die Detektion beziehungsweise Extraktion von Wetland nicht ausreichend lernfähig und deshalb nicht geeignet.

Abbildung 25: Featureklassen Stream, Wetland und Riverlake (v. l. n. r.)

Für die Erklärung der schlechteren Metrikwerte und der Schwierigkeit, Wetland zu detektieren kann hierbei Abbildung 25 dienen. Während bei Stream und Riverlake die Grenzen im Rahmen der manuellen Digitalisierung (Erstellung der Groundtruthdaten) klar sichtbar sind und höchstens um einige Pixel abweichen, ist die Klasse Wetland nur durch sehr schwache Striche auf der Karte ausgeprägt. Insofern sind die Grenzen, wo Wetland beginnt und aufhört, nicht trivial zu bestimmen. Ausserdem handelt es sich bei Wetland um einen zusammengesetzten Signaturtyp, beispielsweise im Unterschied zur Featureklasse Stream. Zusätzlich stellen die unregelmässigen Abstände zwischen den verschiedenen Strich-Bereichen bei der Featureklasse Wetland einen weiteren Grund für die Schwierigkeit der Detektion im Rahmen von U-Net dar.

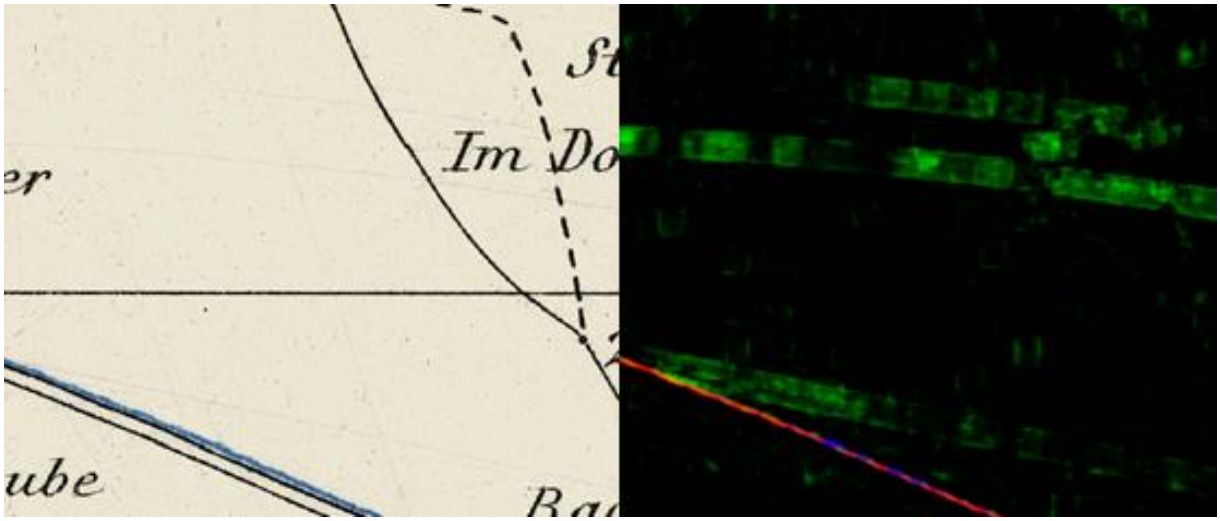


Abbildung 26: Extraktion von Wetland im Bereich schwarzer "Scanstriche"

Diese unregelmässigen Abstände führen zu einer weiteren möglichen Quelle von Problemen. Entweder im Rahmen des Kartendrucks oder beim Scanning sind auf der Karte vereinzelte Striche entstanden. Diese werden durch U-Net in einigen Fällen fälschlicherweise als Wetland detektiert (vgl. Abbildung 26).

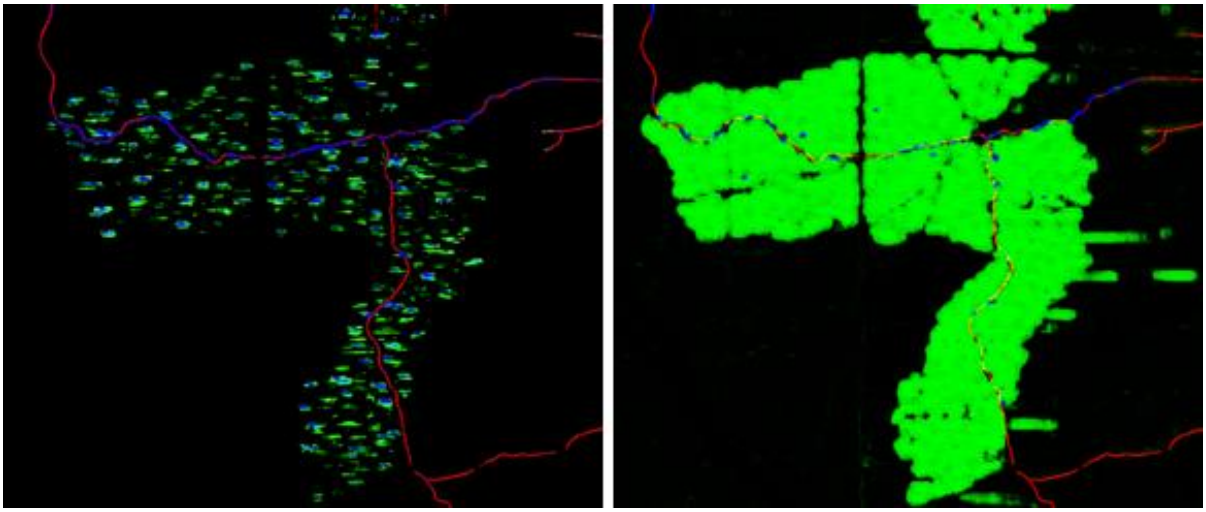


Abbildung 27: Vergleich der Wetlanddetektion von GP (links) und WP3 (rechts)

Weiter ist für eine hohe Precision bei Wetland der GP-Workflow am besten geeignet, da 65.3 % seiner detektierten Pixel richtig sind. Dieser basiert aber nicht auf den vorliegenden bereits digitalisierten Daten für das Jahr 1880, sondern wird durch eine regelmässige Anordnung in einem Gitter hergestellt.
Überaschenderweise sorgt dieses Gitter für eine regelmässige Verteilung und für ausreichend «Negativbeispiele», sodass die entsprechende Instanz lernt, Wetland eher defensiv zu detektieren. Bei diesem Workflow liegt aber der Recall bei nur 8.5 %. Somit werden gesamthaft nur 8.5 % der Wetlandpixel detektiert. Dies allein ist ein sehr schlechter Wert, bei Betrachtung des Bildes ist aber zu vermuten, dass eine aus den detektierten Pixeln lokal erzeugte Hülle um alle Wetlandpixel im Vergleich sehr gute Resultate liefern würde, dies kann aber im Rahmen dieser Arbeit nicht weiter untersucht werden. Insofern kann aber diskutiert werden, ob es für die entsprechende Nachbearbeitung nach der Anwendung des Machine-Learning-Algorithmus besser ist, einzelne Bereiche zu verbinden oder aber zu viel verbundene Bereiche zu trennen.
Der höchste Recall wird mit dem WP3-Workflow erreicht, welcher 88.8 % aller existierenden Pixel detektiert. Der Unterschied zwischen GP und WP3 wird in Abbildung 27 deutlich. Am rechten Bildrand des Bildes von WP3 ist ausserdem die falsche Detektion der Klasse Wetland ersichtlich.
Soll aus Precision und Recall der beste Kompromiss gefunden werden, was im Rahmen der F1-Metrik bewertet wird, empfiehlt sich für die entsprechende Extraktion der Ansatz mit dem fehlerbasierten Workflow EP2F_mix.

4.5 Gesamtvergleich Riverlake

Workflow	Precision	Recall	F1	Accuracy	Jaccard	Av. Precision
ZP_1	0.553	0.606	0.518	0.971	0.387	0.743
ZP_2	0.866	0.762	0.768	0.983	0.665	0.843
ZP_3	0.921	0.560	0.618	0.975	0.521	0.857
ZPmin227_1	0.677	0.461	0.475	0.973	0.362	0.730
ZPmin227_2	0.892	0.547	0.600	0.975	0.503	0.851
ZPmin227_3	0.851	0.668	0.693	0.976	0.608	0.743

GP	0.896	0.487	0.571	0.974	0.459	0.799
BP	0.930	0.699	0.735	0.978	0.653	0.783
BPmin227_1	0.900	0.783	0.806	0.983	0.708	0.866
BPmin227_2	0.906	0.699	0.728	0.977	0.633	0.928
BPmin227_3	0.883	0.735	0.769	0.985	0.659	0.898
WP3	0.949	0.642	0.703	0.977	0.615	0.843
WP6	0.937	0.756	0.813	0.986	0.711	0.877
2F_1	0.885	0.911	0.895	0.996	0.814	0.921
2F_2	0.756	0.800	0.736	0.982	0.601	0.912
2F_3	0.893	0.754	0.767	0.979	0.677	0.925
2F_mix	0.904	0.823	0.840	0.985	0.744	0.964
MP	0.895	0.677	0.705	0.977	0.616	0.766
EP	0.827	0.693	0.698	0.977	0.584	0.876
EP2F_1	0.906	0.756	0.771	0.978	0.689	0.840
EP2F_2	0.898	0.714	0.729	0.977	0.656	0.833
EP2F_3	0.842	0.753	0.724	0.977	0.646	0.779
EP2F_mix	0.912	0.742	0.749	0.977	0.686	0.859

Tabelle 5: Gesamtvergleich Riverlake

Im Rahmen der Riverlake-Detektion sticht bezüglich der Metriken ein Workflow spezifisch heraus. Anders als bei der Detektion von Stream und Wetland ist dies aber nicht EP2F_mix, sondern die erste Instanz des Workflows, der auf der Generierung von Trainingsdaten mithilfe der zwei Featureklassen Wetland und Stream basiert (2F). Gründe dafür finden sich vor allem bei der Betrachtung der verschiedenen Workflows aufgeschlüsselt nach Kartenblättern (vgl. Kapitel 4.6.3).
Vergleicht man die Metriken von WP6, 2F_1, 2F_mix und EP2F_mix ohne den Einbezug der Kartenblätter 134 und 136 bei der Mittelwertbildung, so erhält man folgendes Ergebnis (vgl. Tabelle 6).

Workflow	Precision	Recall	F1	Accuracy	Jaccard	Av. Precision
WP6	0.923	0.808	0.844	0.998	0.752	0.885
2F_1	0.860	0.919	0.886	0.999	0.800	0.913
2F_mix	0.886	0.916	0.898	0.999	0.819	0.961
EP2F_mix	0.917	0.915	0.913	0.999	0.846	0.964

Tabelle 6: Gesamtvergleich Riverlake ausgewählter Workflows ohne Einbezug der Kartenblätter 134 und 136

Ohne den Einbezug der Kartenblätter mit dem See schneidet folglich der Workflow EP2F_mix am besten ab (vgl. Abbildung 28).

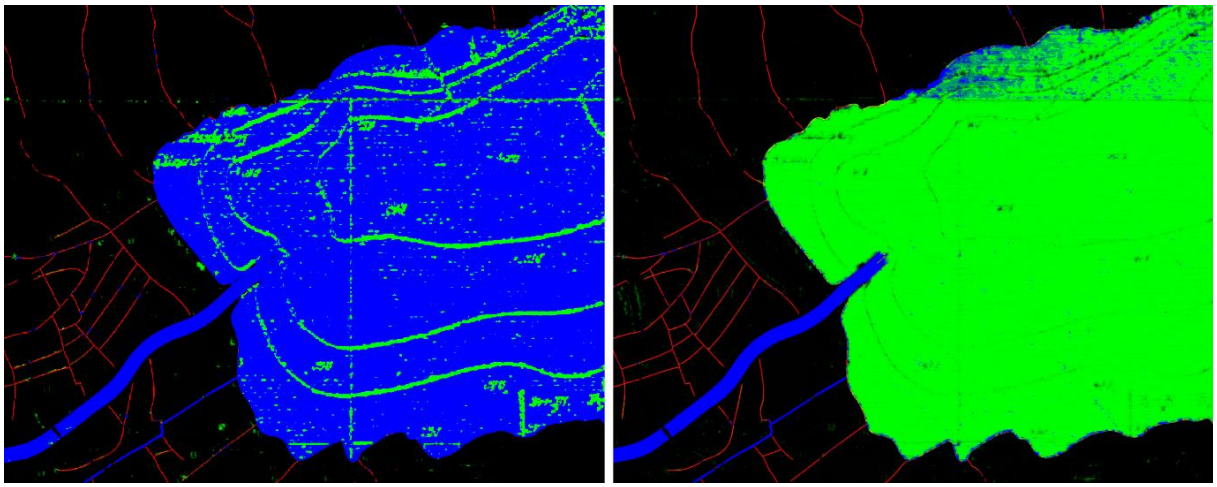


Abbildung 28: Optischer Vergleich der Seedetektion der Workflows 2F_1 (links) und EP2F_mix (rechts)

Allgemein ist auch bei der Featureklasse Riverlake zu vermuten, dass die «Verwechslung» mit Stream bei den Metrikwerten einen starken Einfluss hat. Um dies quantitativ zu ermitteln, wird ein Test durchgeführt mit der gemittelten Stream und Riverlake-Vorhersage des Workflows EP2F_mix im Vergleich zu den entsprechenden Testdaten («Vereinigung» aus den Groundtruthdaten Riverlake und Stream). Dies wird für das Kartenblatt 065 bei Winterthur durchgeführt, aufgrund der Mittelung der beiden vorhergesagten Layer mit einem Schwellenwert von 0.25 statt 0.5 bei der Evaluation (vgl. Kapitel 3.5.2). Die Ergebnisse dieses Tests sind in Tabelle 7 dargestellt.

Metrik	Precision	Recall	F1	Accuracy	Jaccard	Av. Precision
Stream	0.836	0.823	0.830	0.998	0.709	0.906
Riverlake	0.895	0.870	0.882	0.999	0.790	0.947
Riverlake ∪ Stream	0.900	0.906	0.903	0.998	0.823	0.962

Tabelle 7: Vergleich für einen Wegfall der Unterscheidung zwischen Stream und Riverlake für das Kartenblatt 65

Bei der Analyse wird ersichtlich, dass ohne die Unterscheidung von Riverlake und Stream die Genauigkeit der Featureextraktion verbessert werden könnte. Dies ist aber nur als hypothetische Untersuchung zu betrachten, da die Featureunterteilung im Rahmen dieser Arbeit als invariant betrachtet wird.

4.6 Vergleich nach Kartenblättern

Um Ausreisser oder Fehler, die nur in einzelnen Kartenblättern existieren gut erkennen zu können, werden auch alle Kartenblätter einzeln ausgewertet (vgl. Tabelle 8, Tabelle 9 und Tabelle 10). Dazu wird die F1-Metrik verwendet, welche Precision und Recall in Kombination abdeckt und sich in vielen Fällen ähnlich verhält wie Jaccard und Average Precision.

4.6.1 Stream

Workflow	8	13	15	65	67	114	116	128	134	136
ZP_1	0.404	0.187	0.012	0.138	0.159	0.142	0.095	0.270	0.331	0.250
ZP_2	0.795	0.783	0.648	0.763	0.830	0.826	0.785	0.836	0.728	0.809
ZP_3	0.681	0.740	0.546	0.705	0.784	0.605	0.450	0.835	0.764	0.762

ZPmin227_1	0.035	0.002	0.000	0.005	0.004	0.039	0.017	0.001	0.004	0.005
ZPmin227_2	0.663	0.713	0.439	0.675	0.743	0.631	0.519	0.794	0.742	0.747
ZPmin227_3	0.694	0.794	0.729	0.790	0.828	0.841	0.824	0.834	0.754	0.838
GP	0.540	0.701	0.546	0.698	0.727	0.416	0.247	0.790	0.733	0.701
BP	0.853	0.838	0.733	0.795	0.864	0.834	0.814	0.872	0.745	0.836
BPmin227_1	0.789	0.812	0.753	0.805	0.855	0.740	0.671	0.862	0.803	0.834
BPmin227_2	0.762	0.829	0.756	0.813	0.867	0.849	0.828	0.865	0.810	0.846
BPmin227_3	0.776	0.802	0.681	0.779	0.847	0.770	0.747	0.835	0.778	0.818
WP3	0.791	0.815	0.702	0.787	0.852	0.828	0.746	0.868	0.828	0.843
WP6	0.736	0.812	0.694	0.784	0.852	0.751	0.652	0.862	0.828	0.840
2F_1	0.718	0.769	0.770	0.800	0.823	0.738	0.731	0.816	0.774	0.814
2F_2	0.339	0.684	0.678	0.658	0.710	0.408	0.479	0.523	0.638	0.643
2F_3	0.725	0.797	0.772	0.815	0.844	0.839	0.824	0.854	0.810	0.834
2F_mix	0.695	0.775	0.769	0.808	0.829	0.783	0.796	0.818	0.781	0.827
MP	0.774	0.823	0.743	0.795	0.857	0.770	0.739	0.861	0.791	0.842
EP	0.683	0.787	0.730	0.778	0.822	0.712	0.648	0.808	0.799	0.777
EP2F_1	0.783	0.808	0.772	0.821	0.858	0.830	0.779	0.852	0.802	0.846
EP2F_2	0.807	0.802	0.744	0.801	0.843	0.843	0.819	0.848	0.795	0.821
EP2F_3	0.754	0.792	0.793	0.817	0.842	0.794	0.778	0.839	0.823	0.807
EP2F_mix	0.821	0.815	0.789	0.830	0.867	0.866	0.857	0.864	0.835	0.847

Tabelle 8: Vergleich der F1-Metrik über alle Kartenblätter bezüglich Stream

Bei der Betrachtung der Detektion von Stream aufgeschlüsselt nach Kartenblättern bestätigen sich die Resultate aus dem Gesamtvergleich. Sowohl EP2F_mix als auch die BP-Workflows BP und BP_min227_2 generieren die besten Ergebnisse. Die BP-Workflows weisen bei den Kartenblättern 8, 13, 67 und 128 bessere Ergebnisse auf, während der Workflow EP2F bei den restlichen sechs Kartenblättern Stream besser detektiert.

4.6.2 Wetland

Workflow	8	13	15	65	67	128	134	136
ZP_1	0	0	0	0	0	0	0	0
ZP_2	0	0	0	0	0	0	0	0
ZP_3	0	0	0	0	0	0	0	0
ZPmin227_1	0	0	0	0	0	0	0	0
ZPmin227_2	0	0	0	0	0	0	0	0
ZPmin227_3	0.057	0.108	0.382	0.865	0.583	0.553	0.686	0.036
GP	0.022	0.237	0.074	0.139	0.356	0.140	0.031	0.029
BP	0.083	0.059	0.521	0.879	0.523	0.559	0.762	0.040

BPmin227_1	0.044	0.080	0.556	0.840	0.515	0.499	0.811	0.040
BPmin227_2	0.029	0.052	0.672	0.881	0.461	0.363	0.754	0.041
BPmin227_3	0.017	0.021	0.085	0.038	0.013	0.030	0.060	0.018
WP3	0.021	0.078	0.721	0.903	0.742	0.311	0.713	0.040
WP6	0.010	0.035	0.248	0.868	0.522	0.180	0.840	0.052
2F_1	0.028	0.037	0.444	0.872	0.530	0.400	0.884	0.180
2F_2	0.019	0.025	0.285	0.709	0.398	0.279	0.576	0.029
2F_3	0.035	0.069	0.582	0.892	0.499	0.517	0.735	0.041
2F_mix	0.051	0.103	0.702	0.892	0.611	0.585	0.778	0.045
MP	0.093	0.075	0.446	0.858	0.504	0.546	0.678	0.036
EP	0.161	0.020	0.377	0.464	0.156	0.064	0.400	0.018
EP2F_1	0.061	0.115	0.593	0.891	0.626	0.698	0.705	0.038
EP2F_2	0.067	0.137	0.633	0.909	0.680	0.681	0.687	0.039
EP2F_3	0.024	0.022	0.147	0.850	0.404	0.410	0.685	0.038
EP2F_mix	0.101	0.212	0.746	0.924	0.716	0.805	0.696	0.039

Tabelle 9: Vergleich der F1-Metrik über alle Kartenblätter (ausser 114 und 116) bezüglich Wetland

Bei der Analyse der Klasse Wetland pro Kartenblatt ist auffallend, dass erneut der EP2F_mix-Workflow für alle Kartenblätter ausser 134 und 136 sehr gute Ergebnisse erzielt. Bei der geografischen Analyse fällt auf, dass genau diese beiden Kartenblätter Teile eines grossen Sees beinhalten. Für die Kartenblätter 134 und 136 stellt 2F_1 den besten Workflow dar (vgl. Kapitel 4.6.3). Allgemein liefern auch die WP-Workflows gute Ergebnisse.

4.6.3 Riverlake

Workflow	8	13	15	65	67	114	116	128	134	136
ZP_1	0.483	0.538	0.678	0.687	0.623	0.784	0.801	0.517	0.010	0.057
ZP_2	0.893	0.844	0.485	0.649	0.797	0.982	0.984	0.854	0.881	0.305
ZP_3	0.654	0.744	0.306	0.759	0.861	0.924	0.930	0.887	0.035	0.078
ZPmin227_1	0.528	0.413	0.106	0.676	0.694	0.851	0.849	0.560	0.016	0.062
ZPmin227_2	0.603	0.712	0.264	0.765	0.868	0.922	0.919	0.863	0.016	0.066
ZPmin227_3	0.764	0.855	0.668	0.813	0.855	0.963	0.969	0.920	0.024	0.101
GP	0.497	0.700	0.372	0.748	0.823	0.825	0.833	0.856	0.008	0.054
BP	0.946	0.884	0.589	0.804	0.852	0.978	0.982	0.928	0.257	0.132
BPmin227_1	0.918	0.864	0.672	0.777	0.850	0.981	0.982	0.920	0.762	0.330
BPmin227_2	0.822	0.864	0.658	0.822	0.847	0.958	0.973	0.916	0.273	0.143
BPmin227_3	0.847	0.767	0.407	0.663	0.832	0.981	0.980	0.867	0.876	0.467
WP3	0.848	0.809	0.596	0.812	0.899	0.957	0.945	0.937	0.085	0.140
WP6	0.865	0.864	0.493	0.816	0.884	0.971	0.967	0.892	0.846	0.529

2F_1	0.918	0.816	0.828	0.855	0.850	0.975	0.977	0.868	0.933	0.926
2F_2	0.814	0.693	0.744	0.722	0.734	0.943	0.951	0.630	0.651	0.478
2F_3	0.900	0.838	0.787	0.860	0.832	0.982	0.982	0.917	0.363	0.212
2F_mix	0.903	0.840	0.835	0.882	0.861	0.982	0.980	0.899	0.716	0.499
MP	0.875	0.850	0.543	0.784	0.837	0.981	0.979	0.909	0.148	0.141
EP	0.786	0.801	0.653	0.665	0.785	0.966	0.970	0.834	0.373	0.149
EP2F_1	0.942	0.839	0.784	0.862	0.868	0.984	0.983	0.925	0.386	0.136
EP2F_2	0.952	0.852	0.740	0.846	0.854	0.978	0.978	0.908	0.061	0.121
EP2F_3	0.943	0.793	0.815	0.861	0.833	0.969	0.971	0.872	0.059	0.125
EP2F_mix	0.961	0.863	0.815	0.882	0.887	0.984	0.983	0.931	0.062	0.123

Tabelle 10: Vergleich der F1-Metrik über alle Kartenblätter bezüglich Riverlake

Bei der Analyse der Riverlake-Workflows nach Kartenblättern ist ersichtlich, dass der Workflow EP2F_mix für fast alle Kartenblätter am besten oder gut geeignet ist, während er für die Kartenblätter 134 und 136 überhaupt nicht geeignet erscheint. Insbesondere fällt auf, dass sich die schlechten Werte so stark auswirken, dass im Gesamtvergleich von Riverlake der Durchschnitt von EP2F_mix deutlich schlechter ist als derjenige von 2F_1.

Abbildung 29: Kartenblatt (KB) 133 (links) und 136 (rechts) als Kartengrundlage für Abbildung 30

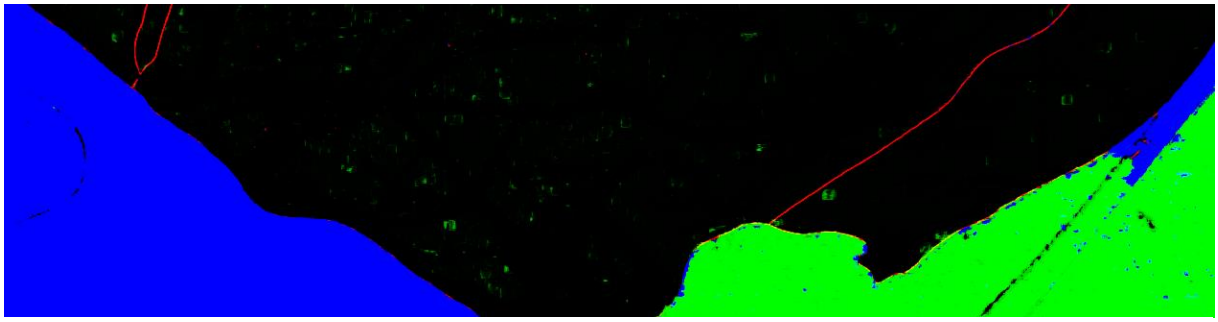


Abbildung 30: Vergleich von Seedetektion mit EP2F_mix für die Trainingsdaten (KB133, 1880, links) und die Testdaten (KB136, 1879, rechts)

Beim optischen Vergleich der detektierten Seeflächen von 2F_1 und EP2F_mix kann auch auf die niedrigen Metrikwerte des Workflows EP2F_mix für die Kartenblätter 134 und 136 geschlossen werden. Dabei werden bei diesen zwei Kartenblättern die Seen fast vollständig als Wetland klassifiziert, während dies beim Workflow 2F_1 nicht der Fall ist, welcher die grössten Teile des Sees richtigerweise als Riverlake klassifiziert (vgl. Abbildung 28). Augenscheinlich erreicht beim Kartenblatt 136 nur der Workflow 2F_1 einen F1-Wert von 92.6 %, während einige Workflows F1-Werte zwischen 40 % und 50 % aufweisen. Die meisten F1-Werte liegen zwischen 5 % und 15 %. Auch die Instanzen 2F_2 und 2F_3 erreichen hierbei nur F1-Werte von 47.8 % bzw. 21.2 %. Bei diesem sehr guten Wert für 2F_1 muss von einem Ausreisser im positiven Sinne ausgegangen werden. 2F_1 hat bei einem manuellen Vergleich der Samplingdaten nicht mehr Punkte im See als beispielsweise EP2F_2.

Dass die Performanz der meisten Workflows bei diesem Kartenblatt sehr schlecht ist, kann auf verschiedene Gründe zurückzuführen sein. Einerseits wird bei der Betrachtung der beiden Datengrundlagen sichtbar, dass sich die Blautöne zwischen dem Trainingsdaten-Kartenblatt 133 und dem Testdaten-Kartenblatt 136 unterscheiden, wahrscheinlich aufgrund des Drucks, der Lagerung oder des Scans (vgl. Abbildung 29 und Abbildung 30). Auch zwischen den beiden Testdatenkartenblättern 134 und 136 gibt es feine Unterschiede, welche den Unterschied bei der Detektion verursachen. Wenn für das Kartenblatt 133 der Trainingsdaten die Vorhersage angewendet wird, wird der See problemlos erkannt. Insofern kann dabei von Überspezifizierung auf die Trainingsdaten gesprochen werden.
Ausserdem existieren abgesehen von diesem Ausschnitt des Neuenburgersees in den Trainingsdaten keine weiteren grossen Seen. Dies ist insofern problematisch, als dass kleine Seen auf der Siegfriedkarte mit Abstandslinien symbolisiert werden (vgl. Abbildung 31), während grosse Seen eine uneinheitliche Symbolisierung aufweisen.

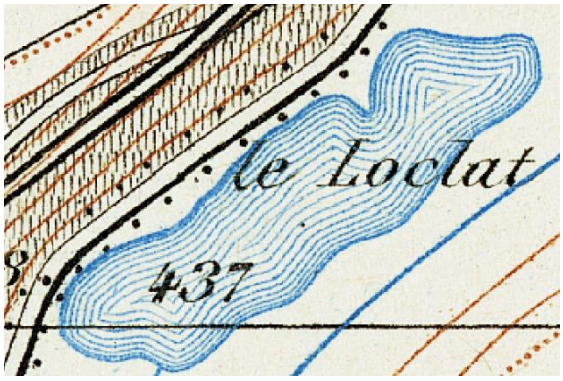


Abbildung 31: Beispiel eines kleinen Sees

4.7 Vergleich von Workflowgruppen

Um nicht nur Aussagen über den «besten» Workflow machen zu können, sondern auch fundierte Aussagen über günstige Einstellungen verschiedener Parameter bei der Trainingsdatengenerierung, sollen nachfolgend verschiedene Workflowgruppen miteinander verglichen werden.

4.7.1 Vergleich bezüglich Datengrundlage

Die existierenden 23 Workflows basieren zum Teil auf ähnlichen Vorgehensweisen. Eine eindeutige Unterscheidung kann anhand der Datengrundlage, welche für die Workflowgenerierung verwendet wird, getroffen werden (vgl. Kapitel 4.1.6).
Für die nachfolgende Analyse werden Box-Plots benutzt, wobei die Länge der Whisker auf das Anderthalbfache des Interquartilsabstands begrenzt ist. Alle Werte, die darüber liegen, werden als Outlier (rote Kreuze) charakterisiert. Hierbei werden alle Workflows aufgeschlüsselt nach den drei Datentypen betrachtet. Dazu werden das arithmetische Mittel über alle Kartenblätter und der F1-Score verwendet.

Stream

Für die Featureklasse Stream zeigt sich bei den kartenbasierten Workflows eine grosse Streuung (vgl. Abbildung 32). Die besten Werte befinden sich im Bereich der Werte der daten- und fehlerbasierten Workflows, der Median liegt aber deutlich unter jenen. Daraus kann gefolgert werden, dass kartenbasierte Workflows nicht wesentlich schlechter sein müssen, aber deren Ergebnis meist von der zufälligen Initialisierung bzw. dem Training abhängt.
Die datenbasierten Workflows mit Ergebniseinbezug schneiden etwas schlechter ab als diejenigen ohne Ergebniseinbezug. Die Differenz der beiden Mediane ist aber mit 0.017 klein und kann deshalb nicht ohne weiteres als statistisch

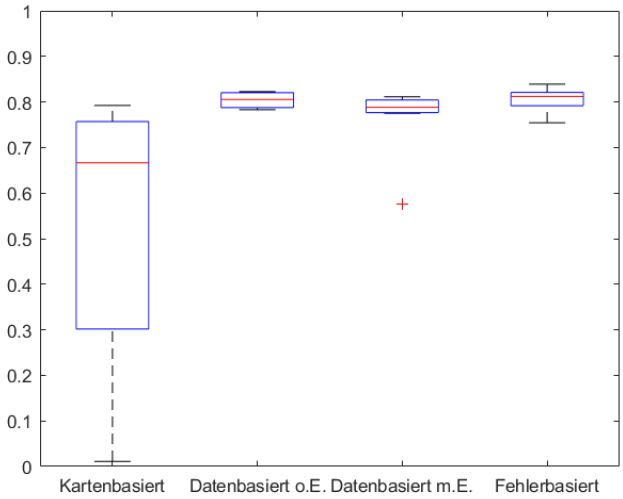


Abbildung 32: Box-Plot-Diagramm für die Workflowgruppen bezüglich Stream

signifikant betrachtet werden. Da die Gruppe der datenbasierten Workflows mit Ergebniseinbezug jeweils den Fokus auf Wetland oder Riverlake gelegt hat, ist dieses schlechtere Abschneiden nicht weiter überraschend. Der Nicht-Einbezug von Stream bei vier der sieben datenbasierten Workflows mit Ergebniseinbezug sorgt folglich für keine grosse Verschlechterung des Ergebnisses der Featureklasse Stream.
Auch auffallend ist, dass die Whisker der fehlerbasierten Workflows länger sind. Deren Werte sind folglich weiter gestreut.
Insgesamt kann konkludiert werden, dass alle Workflowgruppen ausser den kartenbasierten Workflows bezüglich Stream geeignet sind.

Wetland

Bezüglich der Klasse Wetland zeigt sich bei den kartenbasierten Workflows ein Medianwert von 0 (vgl. Abbildung 33). Dadurch sind diese zur Extraktion von Wetland nicht oder kaum geeignet. Den höchsten Medianwert und den höchsten der Mittelwerte weisen die fehlerbasierten Workflows auf. Die kleinste Streuung tritt bei den datenbasierten Workflows mit Ergebniseinbezug auf. Im Unterschied zu Stream tritt bei Wetland bei den datenbasierten Workflows ohne Ergebniseinbezug eine höhere Streuung auf als bei denjenigen mit dem Ergebniseinbezug. Mutmasslich liegt dies am expliziteren Einbezug von Wetland in die Workflowkonzeption. Während bei den Pufferpunkten (datenbasiert o. E.) die Klasse Stream jeweils viele Trainingspunkte erhält, wird durch das Erzwingen von Wetlandpunkten (WP) oder das Weglassen von Stream als Trainingsfeatureklasse (2F) erreicht, dass die Klasse Wetland besser trainiert werden kann.
Insgesamt eignen sich deshalb für die Wetlandextraktion vor allem fehlerbasierte und datenbasierte Ansätze mit Ergebniseinbezug. Auch die datenbasierten Workflows ohne Ergebniseinbezug sind in einem begrenzten Masse geeignet.

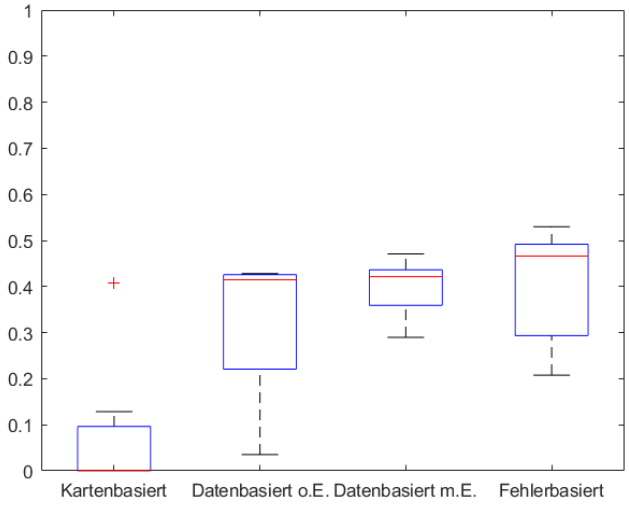


Abbildung 33: Box-Plot-Diagramm für die Workflowgruppen bezüglich Wetland

Riverlake

Bezüglich der Detektion von Riverlake zeigt sich ein ähnliches Bild (Abbildung 34). Die kartenbasierten Workflows erreichen zwar gute Ergebnisse (im Vergleich zu deren Ergebnisse bei Stream und Wetland), doch diese sind schlechter als diejenigen der daten- und fehlerbasierten Workflows. Der beste Absolutwert und der beste Mittelwert wird durch die datenbasierten Workflows mit Ergebniseinbezug erreicht. Die ist auf die gute Seedetektion von 2F_1 zurückzuführen (positiver Ausreisser).

Gesamthaft betrachtet sind die Analyseergebnisse für alle drei Datentypen Stream, Wetland und Riverlake ähnlich. Der deutlichste Unterschied existiert jeweils zwischen den kartenbasierten und den restlichen Workflows, da sich

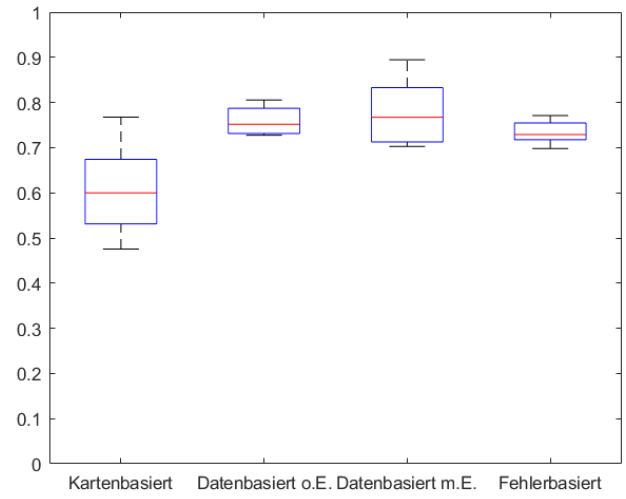


Abbildung 34: Box-Plot-Diagramm für die Workflowgruppen bezüglich Riverlake

die kartenbasierten Workflowstrategien als nicht zuverlässig für die Trainingsdatengenerierung erweisen. Die restlichen Workflowgruppen weisen in sich hohe Streuungen auf, sodass keine eindeutige Aussage über die Güte der jeweiligen Gruppen im gegenseitigen Vergleich getroffen werden kann.

4.7.2 Vergleich bezüglich Mindestabstand

Bei den Workflows ZP und BP werden jeweils zusätzlich Instanzen mit einem Mindestabstand zwischen den Trainingspunkten von 227 m generiert. Da bei ZP jeweils drei Instanzen ohne und drei Instanzen mit Mindestabstand vorliegen, werden diese gemittelt und anhand des F1-Scores und des arithmetischen Mittels aller Kartenblätter verglichen. Zudem wird der Mittelwert von BPmin mit der BP-Instanz verglichen. In Abbildung 35 wird der ZP-Vergleich dargestellt. Dabei fällt auf, dass die Mittelwerte jeweils im ähnlichen Bereich liegen. Werden die Werte von F1 der sechs Instanzen der zufälligen Punkte tabellarisch betrachtet (vgl. Kapitel 4.3, 4.4 und 4.5), ist ersichtlich, dass die Einzelwerte jeweils sehr stark gestreut sind. Insbesondere sind weder ZP noch ZPmin bezüglich aller Featureklassen besser als der jeweils andere Workflow. In Abbildung 36 wird der BP-Vergleich dargestellt. Dabei sind die Unterschiede zwischen den Metrikwerten wiederum klein. Insbesondere fällt auf, dass bei Riverlake ZP besser ist, beim BP-Vergleich hingegen BPmin. Genau entgegengesetzt verhält sich dies bei Wetland. Nur bei Stream ist die Variante ohne Mindestabstand bei beiden Workflowklassen leicht besser.

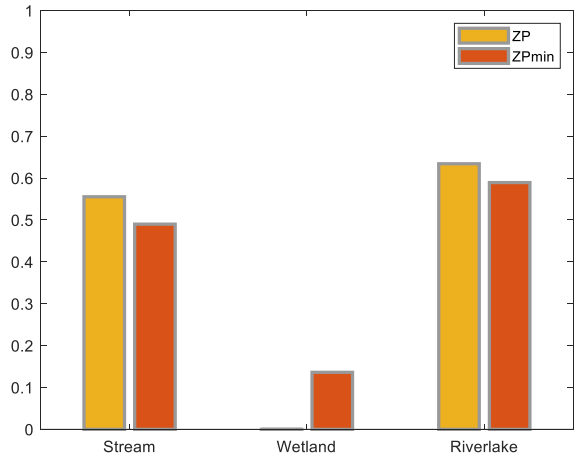


Abbildung 35: Vergleich des Mindestabstands bezüglich ZP

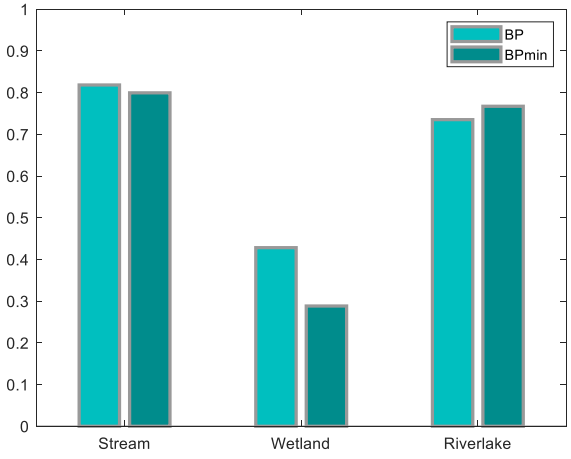


Abbildung 36: Vergleich des Mindestabstands bezüglich BP

Daraus kann gefolgert werden, der Mindestabstand für das Training keine Rolle spielt, unter anderem da die Bereiche bei den meisten Workflows bereits ausreichend gross sind. Nur einzelne Punkte liegen näher als 227 m beieinander.

4.7.3 Vergleich der Wetlandpunkte

Für das Training der Wetlandpunkte werden zwei verschiedene Parametereinstellungen gewählt. Beim Workflow WP3 werden drei Punkte pro Wetlandpolygon erzeugt, während beim Workflow WP6 jeweils sechs erzeugt werden. Beim Vergleich der Ergebnisse in Abbildung 37 (F1, arithmetisches Mittel über alle Kartenblätter) fällt auf, dass der Workflow mit mehr Wetlandpunkten bei der Detektion von Wetland schlechter ausfällt als derjenige mit weniger Wetlandpunkten. Da von beiden nur eine Instanz gemessen wird, kann dies von Messungenauigkeiten herrühren. Es ist aber auch möglich, dass das Modell dann beispielsweise auf die im Training verwendeten Wetlandgebiete

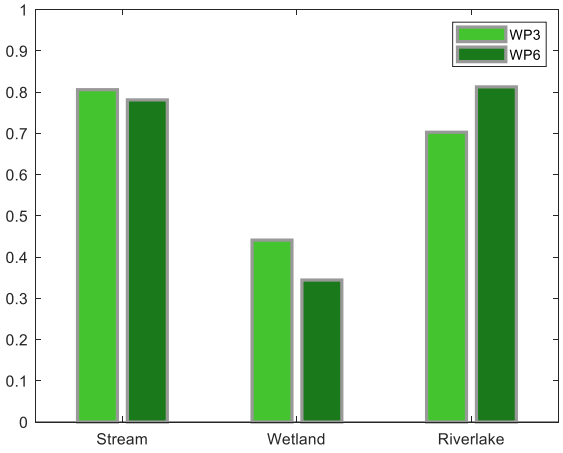


Abbildung 37: Vergleich der Wetlandpunkte-Instanzen

überspezifiziert und zu wenig generell trainiert wird. Insofern kann konkludiert werden, dass das sehr starke Vorkommen einer Featureklasse im Workflow wie bei WP6 mit 804 von 1357 Punkten in gepufferten Wetlandpolygonen nicht zwingend dazu führt, dass die entsprechende Featureklasse besser trainiert wird.

4.7.4 Vergleich der Mischungen nach der Vorhersage

Die «unechten» Workflows, also diejenigen, welche durch die Mischung der verschiedenen Vorhersagen erzeugt wurden, sollen mit den nicht gemischten drei Instanzen der jeweiligen Workflows verglichen werden. Dabei sollen die drei F1-Scores vom Mittelwert über alle Kartenblätter nochmals gemittelt werden, sodass ein Index entsteht. Dieser Index (Mittelwert) wird nachfolgend (in Abbildung 40 und Abbildung 39) mit dem jeweiligen F1-Wert vom unechten Workflow verglichen.
Bei diesem Vergleich fällt auf, dass der F1-Score der gemischten «unechten» Workflows in jedem Fall besser ist als der Mittelwert der drei Einzelinstanzen desselben Workflows. Der grösste Unterschied findet sich in beiden Fällen beim Datentyp Wetland. Dieser beträgt bei 2F 0.0932 und bei EP2F 0.1074. Somit ergibt sich durch Mittelung eine Verbesserung des F1-Scores um zirka 0.1. Daraus folgt, dass sich die Mischung durch Mittelung im Allgemeinen lohnt. Wie in Abbildung 38 erkennbar ist, unterscheiden sich die verschiedenen Workflows aber durchaus. Auf dem dargestellten Ausschnitt ist beispielsweise der Workflow EP2F_3 erfolgreich in der Vermeidung der falschen Wetland-Detektion. Für die korrekte Klassifikation von Stream und Riverlake bzw. deren Unterscheidung, könnte folglich ortsabhängig ein anderer Workflow verwendet werden. Dies könnte im Rahmen einer weiterführenden Untersuchung betrachtet werden.

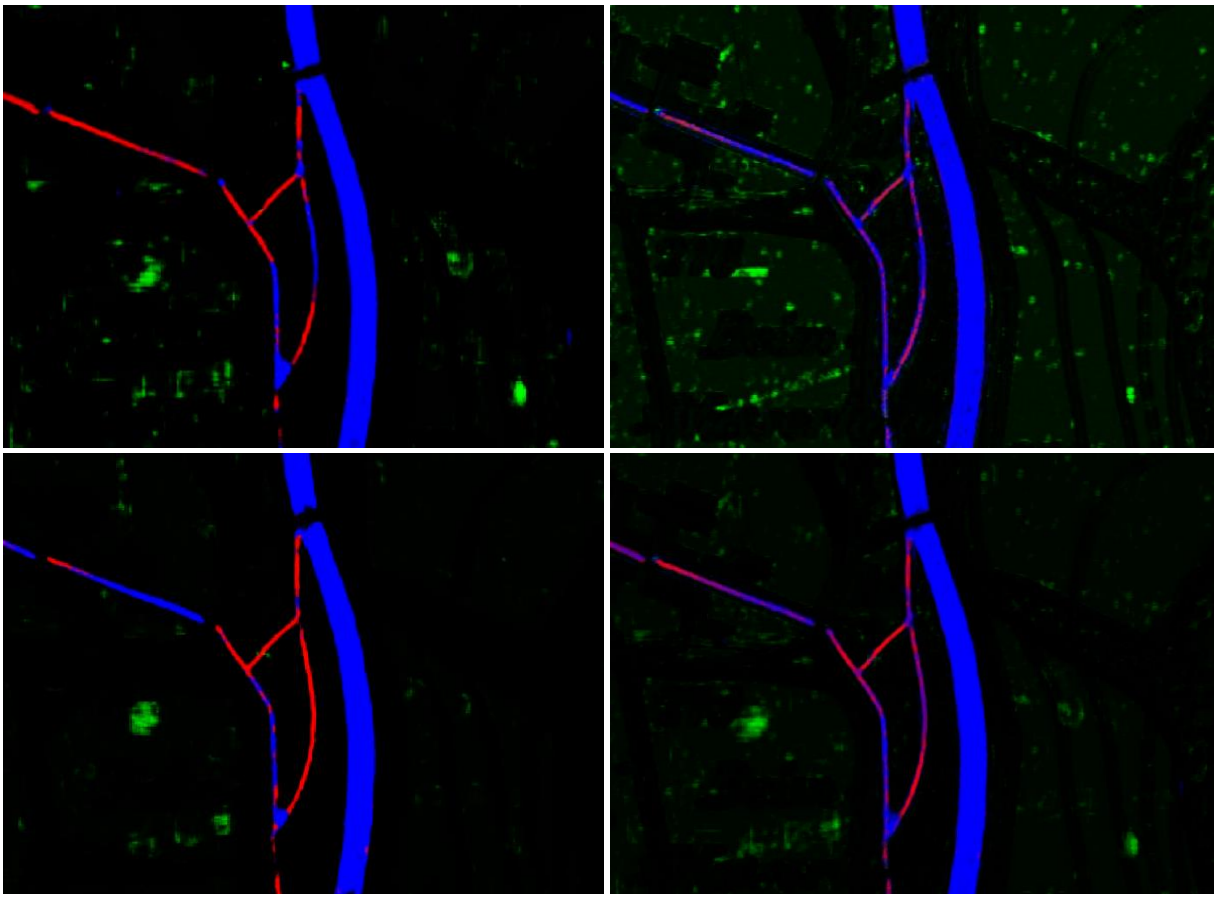


Abbildung 38: Vergleich von Mischungen nach der Vorhersage: EP2F_1, EP2F_2, EP2F_3 und EP2F_mix (von links oben nach rechts unten)

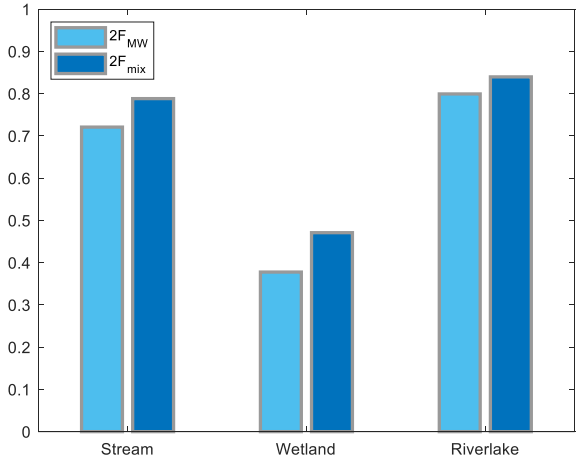


Abbildung 39: Vergleich der Mischungen der 2F-Vorhersagen mit dem entsprechenden Index (Mittelwert)

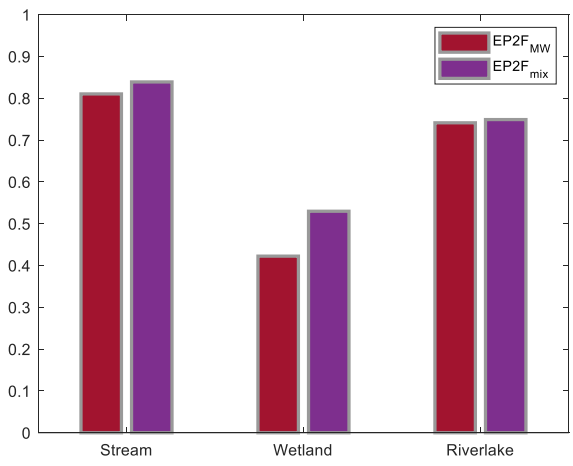


Abbildung 40: Vergleich der Mischungen der EP2F-Vorhersagen mit dem entsprechenden Index (Mittelwert)

4.7.5 Vergleich bezüglich fehlerbasierter Workflows

In Abbildung 41 und Abbildung 42 werden fehlerbasierte Workflows mit den entsprechenden nicht fehlerbasierten Workflows verglichen. Beim Vergleich zwischen 2F_mix und EP2F_mix kann festgestellt werden, dass EP2F_mix jeweils einen höheren F1-Score aufweist als 2F_mix. Dies ist bei Riverlake

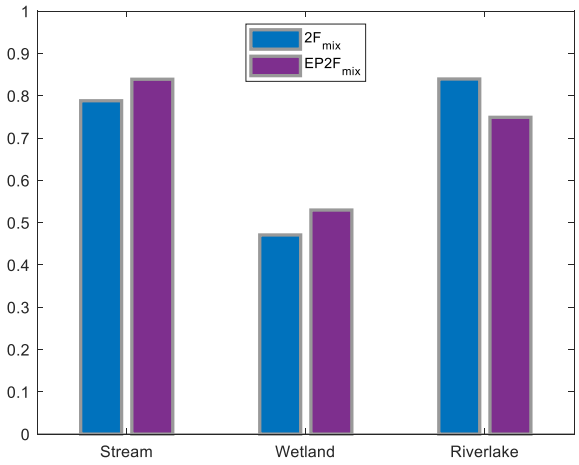


Abbildung 41: Vergleich zwischen 2F_mix und EP2F_mix

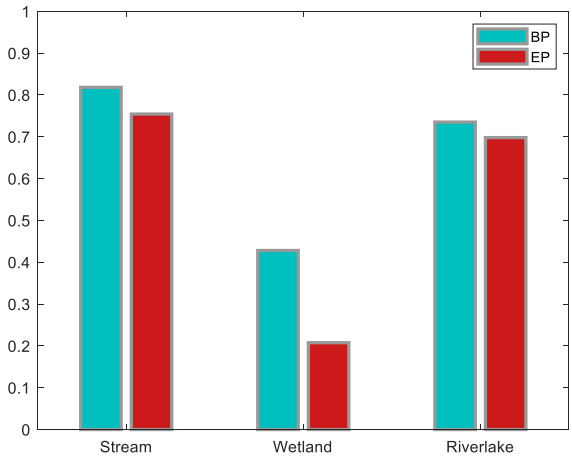


Abbildung 42: Vergleich zwischen BP und EP

aufgrund der Seedetektion anders. Unter der Bedingung, dass die Kartenblätter mit Seedetektion nicht berücksichtigt werden, schneidet aber auch hier EP2F_mix besser ab (vgl. Kapitel 4.6.3). Insofern lohnt es sich, im Rahmen der gemischten Layer 2F_mix und EP2F_mix auch fehlerbasierte Ansätze zu verwenden. Bei der Analyse des Vergleichs zwischen BP und EP wiederum relativiert sich dieses Ergebnis. Der Workflow EP weist für alle drei Featuretypen einen schlechteren F1-Wert auf als der entsprechende BP-Workflow. Eine eindeutige Aussage über die Wirksamkeit der Fehlerbasierung kann deshalb nicht getroffen werden. Je nach Instanz (zufällige Initialisierung) kann die Fehlerbasierung dazu führen, dass Features zu «defensiv» detektiert werden, wie dies bei EP bei Wetland der Fall ist (vgl. Kapitel 4.4). Zusätzlich liegt der Vergleich zwischen BP und EP für nur eine Instanz vor, während der Vergleich für 2F und EP2F für die Mischung aus drei Instanzen betrachtet wird. Insofern kann die Verbesserung durch die Fehlerbasierung als wahrscheinlich betrachtet werden.

4.8 Epochenanzahl

Das Training der jeweiligen Datenpunkte dauert für jede Instanz über eine unterschiedliche Epochenanzahl an. Die Epochen werden entweder durch eine manuelle Obergrenze oder durch die Early Stopping Patience definiert. Da das Training im konkreten Fall durch die Wirksamkeit der Early Stopping Patience die manuelle Obergrenze nie erreicht, kann untersucht werden, ob die Epochenanzahl ein Indiz für die Güte eines Workflows darstellt. Dazu wird zwischen der F1-Metrik und der Epochenanzahl pro Workflow jeweils der Korrelationskoeffizient gebildet, welcher einen linearen Zusammenhang untersucht. Dabei ergeben sich die Resultate in Tabelle 11 und Abbildung 43.

Stream	Wetland	Riverlake
0.440	0.730	0.656

Tabelle 11: Korrelationskoeffizienten zwischen F1-Score und Epochenanzahl über alle Workflows

Aus den Werten der Korrelationskoeffizienten kann gefolgert werden, dass für die vorliegenden Workflows bei Wetland und Riverlake ein schwacher linearer Zusammenhang zwischen der Güte der jeweiligen Workflows und der Epochenanzahl beim Trainieren existiert. Auch in Abbildung 43 ist dies erkennbar. Dies deckt sich mit der Intuition, dass bei gleichen Grundlagedaten das Ergebnis eines länger trainierten Workflows besser sein sollte als dasjenige eines nur kurz trainierten.

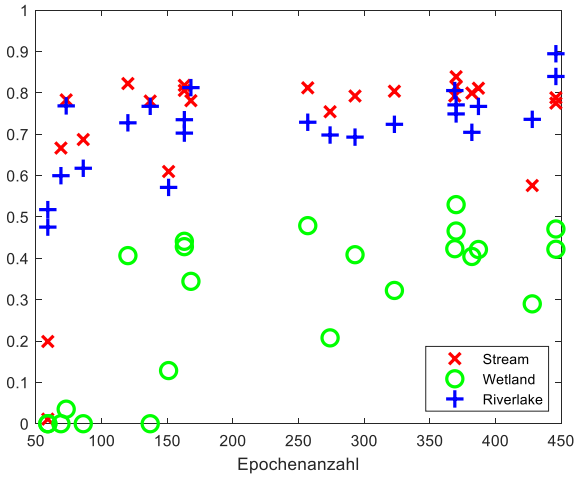


Abbildung 43: Punkte im Verlauf von F1-Score und Epochenanzahl für Stream (rot), Wetland (grün) und Riverlake (blau)

4.9 Empfehlungsrahmen

Auf Basis der Gesamtvergleiche und der Vergleiche nach Kartenblättern und Workflowgruppen kann ein Empfehlungsrahmen für das Sampling von Trainingsdaten zur Detektion von verschiedenen Featureklassen aus der Siegfriedkarte erstellt werden.
Im Allgemeinen lohnt es sich, daten- oder fehlerbasierte Workflows zu verwenden. Die ausschliesslich kartenbasierten Workflows beziehungsweise Trainingsdaten führen nur in einzelnen Fällen zu vergleichbaren Ergebnissen. Im Mittel sind sie schlechter als die daten- bzw. fehlerbasierten Ansätze. Bei der Konzeption der Workflows sollen je nach Komplexität der zu extrahierenden Daten unterschiedliche Zusammensetzungen gewählt werden. Featureklassen mit einer sehr klaren Struktur wie beispielsweise Stream können durch weniger Datenpunkte repräsentiert werden, während für die Detektion von komplexeren Featureklassen wie Wetland mehr Datenpunkte und auch fehlerbasierte Ansätze überprüft werden müssen. Dies kann aber dazu führen, dass die jeweiligen Punkte «zu defensiv» detektiert werden und die Metrikwerte bei einem allfälligen Test schlechter ausfallen. Dabei ist wichtig, das Ziel einer «guten» Detektion vorgängig zu definieren und zu beachten, dass bei gewissen Featureklassen noch Lücken geschlossen werden müssen. Dies kann aber je nach Problem einfacher sein, als Daten eliminieren zu müssen.
Bezüglich des Mindestabstands der Punkte kann konkludiert werden, dass dieser im Rahmen der Verwendung des QGIS-Algorithmus «Zufällige Punkte in den Layergrenzen» keine Rolle spielt, ein Mindestabstand ist folglich nicht notwendig.
Falls eine Featureklasse in den Trainingsdaten übermässig vertreten ist, kann dies dazu führen, dass «zu offensiv» detektiert wird. Dies ist anhand der Metriken im Rahmen eines erhöhten Recall und einer tiefen Precision sichtbar. Deshalb muss jeweils darauf geachtet werden, diese Übervertretung einer spezifischen Featureklasse beispielsweise durch fehlerbasierte Punkte mit ausreichend Negativbeispielen auszustatten.
Allgemein basieren die Generierung und das nachfolgende Training jeweils auf Zufallsvariablen. Um ein optimales Ergebnis zu erreichen, ist es deshalb unerlässlich, mehrere Instanzen zu trainieren, für diese jeweils Vorhersagen zu treffen und die entsprechenden Vorhersagen dann zu mitteln.
Für die Extraktion der Featureklassen Stream, Wetland und Riverlake kann konkludiert werden, dass die Klasse Stream als Datengrundlage nicht notwendig ist, um eine erfolgreiche Detektion zu gewährleisten. Die Featureklasse Riverlake wird am besten detektiert, wenn die Trainingsdaten ausreichend Beispiele zur Verfügung stellen. Insofern ist hier der Verzicht auf den Einbezug von Stream sinnvoll, da die restlichen Featureklassen in diesem Fall zahlreicher vertreten sind. Wichtig dabei ist es aber, möglichst von jedem vorkommenden Kartenblatt (auch der Testdaten) einen Ausschnitt ins Training miteinzubeziehen. Im vorliegenden Fall kann der See nicht gut detektiert werden, da für die Art des Sees wie sie im Kartenblatt von 1879 vorkommt (unterschiedlicher Blauton), keine Trainingsdaten vorliegen. Dies

bedeutet, dass bei den Trainingsdaten auf jeden Fall Überspezifizierung verhindert werden muss und möglichst aus verschiedenen Jahrgängen und verschiedenen Scans Trainingsdaten vorliegen sollen, um Featureklassen möglichst generell zu extrahieren.
Für die Extraktion von Wetland empfiehlt es sich, sowohl genügend Positiv- als auch Negativbeispiele zu generieren. Der Workflow EP2F_mix stellt die optimale Variante dar, da durch die zwei-Featureklassen-Strategie genügend Positivbeispiele vorkommen und durch die Fehlerbasierung ausreichend Negativbeispiele in den Daten vorhanden sind.

5 Diskussion und Ausblick

5.1 Grundlagedaten

Bei der Diskussion der Grundlagedaten ergeben sich verschiedene Aspekte, welche kurz diskutiert werden. Für die Ergebnisse dieser Arbeit können diese relevant sein. Dabei soll aber angemerkt werden, dass die Grundlagedaten als invariant angenommen werden.

Einerseits zeigt die Betrachtung der Kartenblätter in QGIS, dass an den Übergängen zwischen den jeweiligen Kartenblättern Unstetigkeiten auftreten (vgl. Abbildung 44). Für die Trainingsergebnisse ist dies kein stark beeinflussender Faktor, da das Training jeweils auf den Kartenblättern und den Featureklassen basiert. Nach erfolgter Extraktion kann dies aber für die Weiterverarbeitung relevant sein.

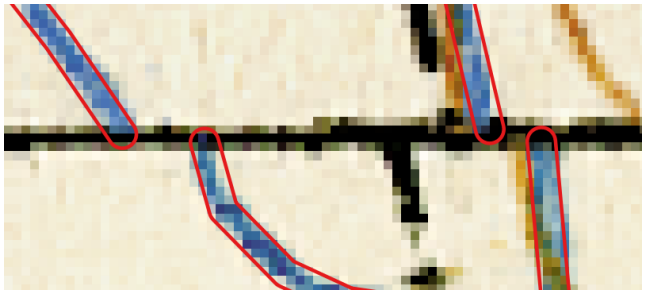


Abbildung 44: Versatz von Stream über Kartenblattgrenzen

Zudem fällt auf, dass die Unterscheidung zwischen Stream und Riverlake nicht in jedem Fall klar definiert ist. Insbesondere in Fällen wie in Abbildung 45 ist von Auge kein klarer Unterschied erkennbar. Semantisch betrachtet ist fraglich, inwiefern ein Fluss zum Bach werden kann und dann wiederum zum Fluss wird. Deshalb müsste geprüft werden, ob auch die Bäche gemeinsam mit den Flüssen und Seen eine Klasse bilden könnten. Die Trainingsergebnisse wären dabei nachweislich besser, da Verwechslungen zwischen Stream und Riverlake nicht mehr möglich wären (vgl. Tabelle 7). Da dies aber nicht nur im Rahmen dieser Arbeit von Relevanz ist, sondern generell für die Digitalisierung der Siegfriedkarte, kann diese Fragestellung im vorliegenden Rahmen nicht abschliessend beantwortet werden.

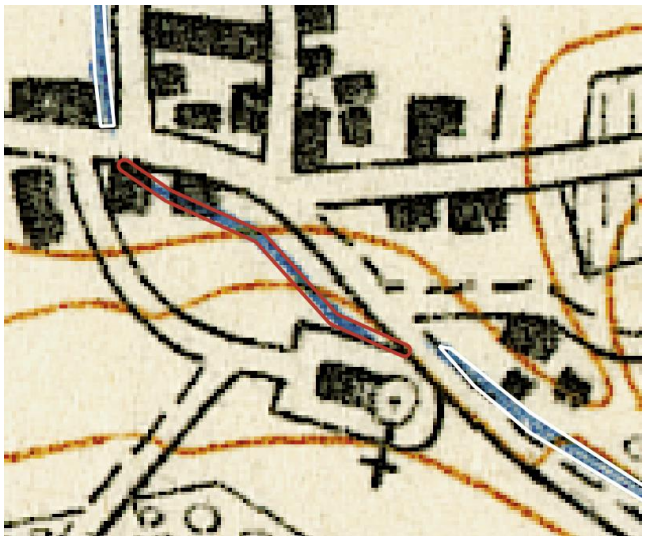


Abbildung 45: Klassifizierung von Stream und Riverlake in den Groundtruthdaten

Als weiterer Punkt muss auch die Überschneidung gewisser Featureklassen thematisiert werden. Wie in Abbildung 46 dargestellt, überschneiden sich die Featureklassen Wetland und Riverlake bzw. Wetland und Stream teilweise. Dies ergibt zwar semantisch Sinn, da ein Fluss durchaus quer durch ein Feuchtgebiet führen kann. Beim Training ergibt sich aber eine Doppeldeutigkeit, da Punkte existieren, welche sowohl Fluss als auch Wetland darstellen, obwohl sie eigentlich in der Flussmitte eines von Wetland umgebenen Flusses liegen. Inwiefern diese Doppeldeutigkeit das Training negativ (oder allenfalls auch positiv) beeinflusst, wäre ein möglicher Gegenstand weiterer Forschung.

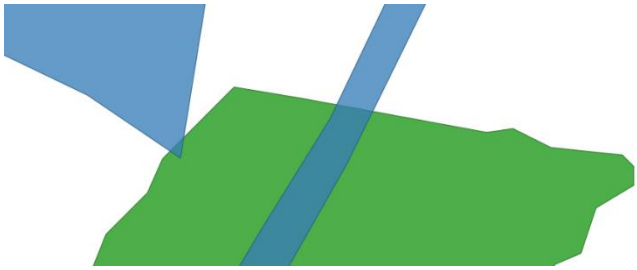


Abbildung 46: Überschneidung von Wetland und Riverlake bei den Trainingsdaten

5.2 Workflows, Training und Evaluation

Bezüglich der Workflows muss diskutiert werden, wie zuverlässig sich deren Ausführung gestaltet und inwiefern sie das Training beeinflussen.
Da die Workflows auf der grafischen Modellierung von QGIS basieren, sind die Ergebnisse von den jeweiligen QGIS-Funktionen abhängig. Semantisch als besonders nützlich erwiesen hat sich die Funktion «Zufällige Punkte in den Layergrenzen», da sie eine gewählte Anzahl zufälliger Punkte generiert, welche in den Polygonen eines gewählten Layers liegen.
Im Rahmen der Ausführung terminiert die Funktion aber in einigen Fällen nicht erfolgreich. Einerseits tritt in gewissen Fällen die Fehlermeldung auf, dass nicht genügend Punkte in den Layergrenzen erzeugt werden können. In diesen Fällen müssen normalerweise die Layergrenzen vergrössert oder die Punkte aus mehreren Ausführungen kombiniert werden.
Im Rahmen dieser Arbeit wird dabei der Ansatz gewählt, dass beispielsweise die Pufferdistanz rund um die entsprechenden Datengrundlagen erhöht wird, damit bei den folgenden Ausführungen der Funktion die Terminierung erfolgreich wird.
Auffallend ist, dass die Terminierung vor allem dann nicht erfolgreich verläuft, wenn die Geometrie aufgelöst und die Pufferdistanz klein ist. Dies lässt sich dadurch erklären, dass der Algorithmus mit der Bounding Box arbeitet und aus diesem Grund bei schmalen, verwinkelten und langen Geometrien oft nicht mit einer genügenden Anzahl an Punkten terminiert (Bruy, 2014). Insofern ist es wichtig, die Geometrie nicht aufzulösen und die Pufferdistanz vor allem bei Flüssen und Bächen nicht zu klein zu wählen. In vielen Fällen (ausser 2F) reicht die gewählte Distanz von 80 m aus, welche auf der Hälfte der Quadratseite einer Kachel des Trainings basiert.
In gewissen Fällen wird eine zu hohe Anzahl an Punkten generiert, obwohl derselbe Workflow zweimal hintereinander mit denselben Inputdaten ausgeführt wurde. Dabei generiert der Algorithmus bei einer gewählten Punktanzahl von 955 Punkten einmal einen Output von 1359 Punkten und einmal einen Output von 955 Punkten. Im Rahmen dieser Arbeit wird das nicht weiter untersucht. Wichtig ist für die Verwendung der vorliegenden Workflows aber, die Punktanzahl jeweils manuell zu überprüfen.
Zusätzlich zur Terminierung der Algorithmen, welche im Rahmen der Workflows verwendet werden, ist es zentral, die verschiedenen Dimensionen der vorliegenden Workflows zu diskutieren. Jeder Workflow weist bei seiner Ausführung eine bestimmte Anzahl von Parametern wie beispielsweise den Mindestabstand, die Anzahl Punkte pro Polygon oder aber Pufferdistanzen auf. Für eine abschliessende Untersuchung der einzelnen Workflows müssten alle diese Dimensionen berücksichtigt werden. Pro Parameter müssten mindestens drei oder vier Ausführungen mit verschiedenen Einstellungen durchgeführt werden. Zudem müssten für jede dieser Ausführungen mehrere Instanzen trainiert werden, um die Effekte des Zufalls möglichst gering zu halten. Bei beispielsweise zehn Parametern mit drei Ausführungen und drei Instanzen ergibt das bereits 90 Trainings. Da ein Training pro Epoche etwas länger als 2 Minuten dauert, ergibt sich bei Epochenanzahlen von 300 eine Trainingszeit von mehr als 10 Stunden. Dies entspräche einer reinen Trainingszeit von 37.5 Tagen (ohne Konzeption der Workflows, Vorhersage und Evaluation).
Aus diesem Grund wird im Rahmen dieser Arbeit ein Ansatz gewählt, bei welchem Workflows generiert und anschliessend ausgewertet werden, sodass iterativ vorgegangen werden kann. Somit können Ergebnisse bereits bei der Konzeption der folgenden Workflows einbezogen werden.
Des Weiteren ist die breite Anwendbarkeit der Workflows, wie sie durch Chiang et al. gefordert wird, zu diskutieren (Chiang, Leyk, & Knoblock, 2014, S. 4). Die generelle Anwendbarkeit ist bei den vorliegenden Workflows zwar gegeben, doch sie sind für den spezifischen Fall der Extraktion der hydrologischen Featureklassen Stream, Wetland und Riverlake konzipiert. Insofern ist es wichtig zu betonen, dass bei anderen Featureextraktionen die allgemeinen Prinzipien des Empfehlungsrahmens angewendet werden können, aber nicht alle Workflows in jedem Fall gut geeignet sind, um beispielsweise auf Gebäude oder Wald angewendet zu werden. Insbesondere das Weglassen von Stream oder der fehlerbasierte Ansatz bezüglich Wetland bewirken, dass der Workflow EP2F_mix spezifisch auf den Datensatz ausgelegt ist. Soll eine generelle Anwendung möglich sein, so empfehlen sich die datenbasierten Ansätze

ohne Einbezug der Ergebnisse (BP und BPmin), da diese alle Eingabelayer gleich behandeln und keine Datenanalyse in ihre Konzeption miteinbeziehen.
Bei der Gesamtbetrachtung kann festgestellt werden, dass die Extraktion von Featureklassen mit Hilfe von U-Net gut funktioniert und folglich weiterverwendbare Daten für die Digitalisierung der historischen Karten vorliegen. Nichtsdestotrotz muss angemerkt werden, dass die alleinige Extraktion durch U-Net mit den vorliegenden Metrikwerten ohne eine Form der Nachbearbeitung nicht ausreichend gut ist, um unkontrolliert in einem GIS als Daten für das Jahr 1879 verwendet werden zu können. Insbesondere die Extraktion von Wetland muss mit entsprechenden Nachbearbeitungsverfahren analysiert beziehungsweise verändert werden, damit ein ausreichend gutes Ergebnis erzielt werden kann.

5.3 Ausblick

Für eine weitere Untersuchung der vorliegenden Workflows müssten aus statistischen Gründen mehr Instanzen vorliegen. Insbesondere beim besten Workflow könnten allenfalls bei der Mischung aus nicht nur drei, sondern beispielsweise zehn Instanzen bessere Ergebnisse resultieren. Viel wichtiger gestaltet sich, die Trainingsdaten anzupassen, sodass sie für alle Kartenblätter auszugsweise vorliegen, für welche eine Extraktion durchgeführt werden sollte. Insbesondere bezüglich der Seen könnte damit die vorliegende Extraktion wesentlich verbessert werden.
Weitergehend kann auch untersucht werden, inwiefern die Daten weiterverarbeitet werden müssen, nachdem sie mit U-Net aus der historischen Karte extrahiert wurden. Dabei ist wichtig zu betrachten, welche Algorithmen jeweils für die endgültige Digitalisierung notwendig sind. Es kann einerseits sinnvoller sein, möglichst viele Daten zu generieren und so einen sehr hohen Recall zu erreichen oder eher andererseits möglichst defensiv zu detektieren und Lücken zu schliessen. Des Weiteren kann untersucht werden, welche zusätzlichen Algorithmen oder morphologischen Operationen auf die durch Machine Learning generierten Daten angewendet werden können. Zudem könnte auch untersucht werden, inwiefern eine Kombination des Machine Learning mit weiteren Digitalisierungsmethoden Sinn ergibt, sodass am Schluss ein optimales Digitalisierungsergebnis mit möglichst wenig menschlicher Detailarbeit notwendig ist.
In einem weiteren Schritt könnten der vorliegende Prozess und die sich ergebenden Workflows in ein Framework eingebettet werden, sodass verschiedene Workflows anwählbar sind, deren Punktanzahl automatisch kontrolliert wird. Dieses Framework könnte dann weitere Funktionalitäten aufweisen wie ein automatisches Training, die Generierung von Fehlerbereichen bei fehlerbasierten Ansätzen, eine automatische Vorhersage und die Mittelung verschiedener Vorhersagen. Anschliessend würden die so generierten Daten vektorisiert. Schliesslich läge beispielsweise ein QGIS-Plugin vor, welches aus einer Rasterkarte und einigen Samplingbereichen mit vorliegenden Groundtruthdaten verschiedene Featureklassen für die Weiterverwendung in einem GIS extrahieren kann (Heitzler & Hurni, 2020, S. 458).

6 Referenzverzeichnis

Bruy, A. (2014). *QGIS-Quellcode: RandomPointsLayer.py.* Abgerufen am 22. Mai 2020 von https://github.com/qgis/QGIS/blob/release-3_12/python/plugins/processing/algs/qgis/RandomPointsLayer.py

Chiang, Y.-Y., Leyk, S., & Knoblock, C. A. (2014). A Survey of Digital Map Processing Techniques. *ACM Computing Surveys, 47*(1), S. 1-44.

Davis, J., & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. In D. o. Informatics (Hrsg.), *International Conference on Machine Learning.* Pittsburgh, USA. Abgerufen am 17. Mai 2020 von https://www.biostat.wisc.edu/~page/rocpr.pdf

Early Stopping. (kein Datum). Abgerufen am 25. Mai 2020 von https://keras.io/api/callbacks/early_stopping/

epsg.io. (2019). Abgerufen am 25. Mai 2020 von http://epsg.io/21781

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press. Abgerufen am 17. Mai 2020 von https://www.deeplearningbook.org

Heitzler, M., & Hurni, L. (2020). Cartographic reconstruction of building footprints from historical maps: A study on the Swiss Siegfried map. *Transactions in GIS, 24*(2), S. 442-461.

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*(6).

Keras FAQ. (kein Datum). Abgerufen am 25. Mai 2020 von https://keras.io/getting_started/faq/#what-do-sample-batch-epoch-mean

Kingma, D. P., & Ba, J. L. (2015). Adam: A Method for Stochastic Optimization. *ICLR.*

Kurzhals, K. (2020). *GDA Vorlesung 3: Spatial Data Mining.* ETH Zürich, IKG.

Layer activation functions. (kein Datum). Abgerufen am 05. Mai 2020 von https://keras.io/activations/

Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully Convolutional Networks for Semantic Segmentation.* Berkeley.

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification.* Abgerufen am 23. Mai 2020 von https://www1.maths.leeds.ac.uk/~charles/statlog/whole.pdf

Plot Precision Recall. (kein Datum). Abgerufen am 22. Mai 2020 von https://scikit-learn.org/stable/_images/sphx_glr_plot_precision_recall_001.png

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation.* Freiburg.

scikit-learn 0.23.0 documentation. (2019). Abgerufen am 17. Mai 2020 von https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (Juni 2014). Dropout: A Simple Way to Prevent Neural Networks from. (Y. Bengio, Hrsg.) *Journal of Machine Learning Research*.

Stengele, R. E. (1995). *Kartographische Mustererkennung: Rasterorientierte Verfahren zur Erfassung von Geo-Informationen.* Dissertation, ETH Zürich, Zürich, Schweiz.

swisstopo. (2020). *Hintergrundinformation zur Siegfriedkarte.* Abgerufen am 28. April 2020 von https://www.swisstopo.admin.ch/de/wissen-fakten/geschichte-sammlungen/historische-kartenwerke/siegfriedkarte.html

swisstopo. (kein Datum). *LV03*. Abgerufen am 25. Mai 2020 von https://www.swisstopo.admin.ch/de/wissen-fakten/geodaesie-vermessung/bezugsrahmen/lokal/lv03.html

Die verwendeten Python-Frameworks und grundlegenden Geodaten wurden von Dr. Magnus Heitzler, IKG, ETH Zürich zur Verfügung gestellt.

Anhang

Eigenständigkeitserklärung

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Eigenständigkeitserklärung

Die unterzeichnete Eigenständigkeitserklärung ist Bestandteil jeder während des Studiums verfassten Semester-, Bachelor- und Master-Arbeit oder anderen Abschlussarbeit (auch der jeweils elektronischen Version).

Die Dozentinnen und Dozenten können auch für andere bei ihnen verfasste schriftliche Arbeiten eine Eigenständigkeitserklärung verlangen.

Ich bestätige, die vorliegende Arbeit selbständig und in eigenen Worten verfasst zu haben. Davon ausgenommen sind sprachliche und inhaltliche Korrekturvorschläge durch die Betreuer und Betreuerinnen der Arbeit.

Titel der Arbeit (in Druckschrift):

Samplingstrategien zur Trainingsdatenerzeugung für tiefe Segmentierungsmodelle

Verfasst von (in Druckschrift):
Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich.

Name(n):	**Vorname(n):**
Bender	Joël

Ich bestätige mit meiner Unterschrift:
- Ich habe keine im Merkblatt „Zitier-Knigge" beschriebene Form des Plagiats begangen.
- Ich habe alle Methoden, Daten und Arbeitsabläufe wahrheitsgetreu dokumentiert.
- Ich habe keine Daten manipuliert.
- Ich habe alle Personen erwähnt, welche die Arbeit wesentlich unterstützt haben.

Ich nehme zur Kenntnis, dass die Arbeit mit elektronischen Hilfsmitteln auf Plagiate überprüft werden kann.

Ort, Datum	**Unterschrift(en)**
Sissach, 23.05.2020	

Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich. Durch die Unterschriften bürgen sie gemeinsam für den gesamten Inhalt dieser schriftlichen Arbeit.