

Kalliorakojen automaattinen mallintaminen ilmakuvista

Jonne Tamminen

Pro gradu -tutkielma  
Syyskuu 2020

MATEMATIIKAN JA TILASTOTIETEEN LAITOS  
TURUN YLIOPISTO

TAMMINEN, JONNE: Kalliorakotunnistuksen automatisointi  
Pro gradu -tutkielma, 38 s.  
Sovellettu matematiikka  
Syyskuu 2020

---

Tässä pro gradu -tutkielmassa kehitetään menetelmä kalliorakojen automaattiseen paikantamiseen kallioista otetuista kuvista. Kalliorakojen analysointi on oleellista, kun halutaan selvittää esimerkiksi kallioperän neste- ja hiilivetyvaroja ja saasteen siirtymistä. Analysoitavasta alueesta riippuen rakojen määrä vaihtelee sadoista aina muutamaan tuhanteen asti. Suurista määristä johtuen yhdenkin alueen rakojen mallintaminen käsin on tällä hetkellä erittäin työlästä. Tämän työn tavoitteena on luoda menetelmä, joka automatisoi rakojen mallinnusprosessia vähentäen runsaasti työtunteja ja antaa objektiivisen tulkinnan alueelle, minkä avulla rakojen topologinen jatkoanalyysi tarkentuu.

Työssä sovelletaan Python-kielellä toteutettuja signaalinkäsittelyyn ja syväoppimiseen pohjautuvia menetelmiä. Menetelmät yksinään eivät kuitenkaan anna kelpuuttavaa tulosta, sillä raot ilmaistaan murtoviivoina. Näiden murtoviivojen tulisi löytää oikea tasapaino sitä vastaavan raon mallin tarkkuuden ja käytettyjen datapisteitten välillä. Jos datapisteitten määrä kasvaa huomattavasti, rakojen topologinen jatkoanalyysi hidastuu liikaa, mitätöiden yhden menetelmän päämääristä.

Menetelmän tuottamat ratkaisut vaihtelevat tulkittavan alueen ja kuvien laadusta riippuen. Parhaimmillaan tulokset nopeuttavat rakoviivojen luontiprosessia hie-  
man, mutta toistaiseksi tulokset ovat vielä kaukana täysin automaattisesta ratkaisun löydöstä ja menetelmän käyttäjän on korjattava ratkaisuja joissain määrin. Menetelmälle esitellään työssä useita eri jatkotutkimussuuntauksia. Menetelmä saattaa nähdä hyötykäyttöä myös sellaisenaan, jos se koulutetaan oikealla tavalla.

Asiasanat: signaalinkäsittely, syväoppimismalli, automatisointi, Python, mallintaminen, murtoviiva, paikkatieto-ohjelmisto, kalliorakojen tunnistus.

# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>3</b>
<b>2</b>	<b>Ongelman kuvaus</b>	<b>3</b>
2.1	Aineisto . . . . .	3
2.2	Kirjallisuutta . . . . .	4
2.3	Tavoitteet . . . . .	5
2.4	Menetelmät . . . . .	5
2.4.1	Signaalinkäsittely . . . . .	5
2.4.2	Koneoppiminen . . . . .	6
<b>3</b>	<b>Datan käsittely</b>	<b>6</b>
3.1	Lähdeaineiston esikäsittely . . . . .	7
3.1.1	QGIS-ohjelmisto . . . . .	8
3.2	Metriikat . . . . .	9
3.2.1	Rakoverkosto ja binäärimatriisi . . . . .	10
3.2.2	Ennusteen vertailu lähdematriisiin . . . . .	11
<b>4</b>	<b>Signaalinkäsittely</b>	<b>14</b>
4.1	Ridge-tunnistus . . . . .	14
4.1.1	1D-tapaus . . . . .	14
4.1.2	2D-tapaus . . . . .	17
<b>5</b>	<b>Koneoppiminen</b>	<b>17</b>
5.1	Toimintaperiaate . . . . .	17
5.2	Koulutus- ja testiaineisto . . . . .	19
5.3	U-Net . . . . .	19
<b>6</b>	<b>Lopullinen menetelmä</b>	<b>21</b>
6.1	Alkukuvan käsittely . . . . .	23
6.2	Ennusteen tekeminen U-Netillä . . . . .	23
6.3	Ridge-tunnistuksen soveltaminen U-Netin ennusteeseen . . . . .	23
6.4	Rakoviivojen luominen Ridge-tunnistuksen ennusteesta . . . . .	25
6.5	Rakoviivojen yhdistäminen . . . . .	26

6.5.1	Yksityiskohtaisemmin . . . . .	27
6.6	Rakoviivojen koordinaattien muuntaminen . . . . .	31
<b>7</b>	<b>Tulosten tulkinta</b>	<b>31</b>
<b>8</b>	<b>Jatkotutkimus</b>	<b>37</b>
	<b>Lähteet</b>	<b>37</b>

# 1 Johdanto

Kallioissa esiintyvät raot ovat seurauksia kallion sisäisistä ja ulkoisista rasituksista. Rakoja, joilla on samoja ominaisuuksia, kuten suunta ja täyte, kutsutaan rakojoukoksi. Rakojoukot muodostavat rakoverkoston, joita hyödynnetään mallintaessa kallioperän neste- ja hiilivetyvaroja [1] ja hydrotermisen lämmön [2] ja saasteen siirtymistä [3]. Lisäksi pohjavesivirtaukset riippuvat vahvasti rakoverkostoista [4].

Jotta rakoverkosta pystyy analysoimaan, se joudutaan ensin muuntamaan oikeanlaiseksi dataksi. Eräs tapa tehdä tämä on taltioida kalliosta korkearesoluutioinen kuva ja piirtää jollain ohjelmistolla rakojen päälle vektorikartta. Yksittäisessä alueessa saattaa kuitenkin olla jopa satoja rakoja, ja analysoitavia alueita saattaa olla useita, joten alueen kartoittamiseen saattaa uppoutua runsaasti aikaa, ennen kuin sitä pystyy edes analysoimaan. Tämän tutkielman tavoitteena on luoda kaksi menetelmää, jotka pyrkivät löytämään raot edellämainituista korkearesoluutioisista kuvista, ja vertailla niiden tuottamia tuloksia.

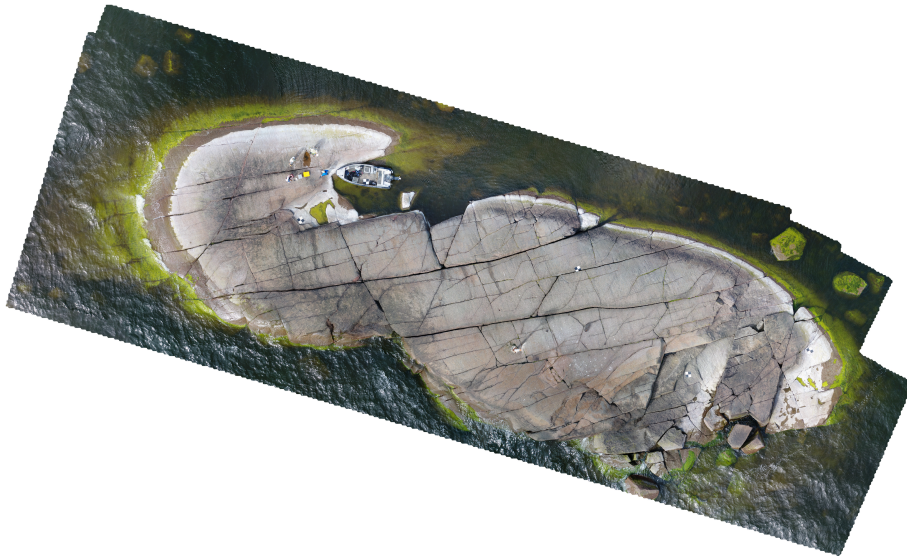
Ensimmäinen menetelmä pyrkii analysoimaan kuvaa signaalinkäsittelyn avulla, kun toinen taas hyödyntää koneoppimista. Koneoppimisen tuottamat tulokset ovat nostaneet päätään 2010-luvulla, ja, jotta sen tulokset pääsevät tässä työssä paremmin esille, on mielekästä antaa sille kontekstia näyttämällä millaisia tuloksia voidaan saada ilman sitä. Toisaalta eräs mielenkiinnon kohde on signaalinkäsittelyn tulokset itsessään, sillä koneoppiminen vaatii toimiaakseen runsaasti dataa, joka on tarpeeksi samanlaista ennustettavan datan kanssa. Mikäli tällaista dataa ei ole saatavilla tarpeeksi, voi olla mielekkäämpää käyttää signaalinkäsittelyn menetelmiä.

## 2 Ongelman kuvaus

### 2.1 Aineisto

Lähdeaineisto on kahdessa eri formaatissa: TIFF-tiedostoina ja Shapefileina. Yksittäinen TIFF-tiedosto sisältää yhden kuvan kalliosta, jonka raot halu-

taan löytää. Kuva on muodostettu ottamalla useita kuvia samalta korkeudelta dronella ja yhdistämällä nämä yhdeksi isoksi kuvaksi. Näitä kuvia on yhteensä kahdeksan kappaletta, ja niiden koot vaihtelevat väleillä  $7228 \times 10619$  ja  $53668 \times 46910$ . Shapefilet sisältävät alueen koordinaattitiedot ja sisältävät käsinmääritellyt koordinaatit tulkittavalle alueelle ja sen raoille. Näitä koordinaatteja ei luonnollisesti voi käyttää hyväksi rakoja löydettyessä, sillä nämä ovat juuri ne koordinaatit, jotka halutaan löytää alkuperäisestä kuvasta. Niitä kuitenkin käytetään ohjelmaa kehittäessä vertailemalla ohjelman tuloksia näihin käsin piirrettyihin koordinaatteihin, joita pidetään oikeina ratkaisuuina. Tästä vertailusta kerrotaan lisää myöhemmin metriikka-osiossa.



Kuva 1: Yksi kuva sisältää koko analysoitavan alueen.

## 2.2 Kirjallisuutta

Tämän työn ongelman kanssa samankaltaisia ongelmia ovat esimerkiksi rakojen löytäminen betonitasoista [8] ja kivistä [9]. Tämän työn ongelma eroaa näistä siinä mielessä, että mielenkiinnonkohteena ei ole raon täydellinen löytäminen, vaan sen mallintaminen murtoviivana. Tämän ongelman ratkaisemisen kannalta ei ole siis tärkeää tietää, mitkä kuvan pikselit esittävät rakoja

ja mitkä eivät, vaan valita muutama piste, jotka luokitellaan eri joukkoihin omiksi murtoviivoiksi siten, että näitä murtoviivoja voi sitten analysoida geologiassa tyypillisillä topologisilla menetelmillä. Tämän lisäksi aineistona saadut kuvat sisältävät runsaasti häiriötekijöitä, jotka vaikeuttavat kuvan analysointia, joten myös tästä syystä ongelmaan on kehitettävä erillinen ratkaisu.

## 2.3 Tavoitteet

Tässä tutkielmassa tavoitteena on luoda ohjelma, joka huomattavasti nopeuttaa rakojen luontiprosessia. Tavoitteena ei ole luoda täysin itsenäiseen työskentelyyn kykenevää ohjelmaa, vaan ohjelman tulosten on aina hyvä tarkastaa ammattilaisen toimesta. Ohjelma pyrkii löytämään mahdollisimman paljon rakoja sillä kustannuksella, että se saattaa löytää rakoja, joita ei ole oikeasti olemassa. Tämä preferenssi johtuu siitä, että ohjelman lopputulosta tarkasteltaessa on helpompaa poistaa ylimääräisiä rakoja kuin lisätä uusia. Jatkossa tässä työssä *rako* tarkoittaa fyysistä rakoja, joka yritetään havaita. Vastaavasti rakoverkostolla tarkoitetaan näiden edellämainittujen rakojen joukkoa kyseisessä tulkinta-alueessa.

## 2.4 Menetelmät

Kutakin menetelmää kehittäessä ollaan ensisijaisesti käytetty aluetta, joka näkyy kuvassa 1. Tämä alue on ihanteellinen siinä mielessä, että siinä on vähän häiriötekijöitä kuten vettä, kasvillisuutta, outoja varjoja, vaikeita pinnanmuotoja tai muita objekteja kuin analysoitavaa kalliota. Menetelmät on ensin asetettu toimimaan käyttämällä tätä aluetta, minkä jälkeen menetelmää yritetään sovittaa muuhun testiaineistoon säätämällä parametreja.

### 2.4.1 Signaalinkäsittely

Signaalinkäsittelyssä ideana on käsitellä yksittäistä kuvaa sellaisenaan ja muut analysoitavat kuvat eivät vaikuta lopputulokseen. Tässä työssä tavoitteena ei ole luoda omaa algoritmia kuvan analysoimiseen, vaan soveltaa so-

pivaa menetelmää aineistoon ja asettaa menetelmän parametrit ongelmaan sopiviksi. Erityisesti tässä työssä sovelletaan Ridge Detection -metodia [5], joka esitellään tarkemmin luvussa 3.

### 2.4.2 Koneoppiminen

Koneoppimisessa ideana on syöttää ohjelmalle ensin harjoitusdataa ja harjoitusdatan ratkaisut. Ohjelma sitten muokkaa itse parametrejään vastaamaan harjoitusaineiston tuloksia mahdollisimman hyvin ja mahdollisesti toistaa tämän operaation useita kertoja. Tämän jälkeen ohjelmaa testataan täysin erillisellä aineistolla kuin millä se koulutettiin. On yleistä, että erilaisia koneoppimisen metodeja yhdistellään, minkä takia niitä esitellään työssä tarkemmin myöhemmin. Tutkielmaan valittu U-Net [7] on ohjattuun koneoppimiseen kuuluva menetelmä, eli malli pyrkii ennustamaan, mihin luokkaan tai luokkiin sille annettu syöte kuuluu. Tässä työssä syötteenä on kuva, jonka jokaiselle pikselille malli antaa arvon väliltä  $[0, 1]$ , joka kuvastaa mallin ennustamaa todennäköisyyttä, että pikseli kuuluu rakoon.

## 3 Datan käsittely

Yleisesti ottaen, dataa on hyvä käsitellä mahdollisimman yksinkertaisessa tai ymmärrettävässä muodossa. Näin ollen on täysin luonnollista muuntaa saatua dataa yksinkertaisemmaksi tai poimia pelkästään mielenkiintoiset osuudet datasta. Tässä luvussa esitellään keskeisimpiä työssä esiintyviä datatyyppejä ja niihin liittyviä käsitteitä. Tutkielman menetelmä on toteutettu Pythonin versiolla 3.7.7. Käytetyt modulit ovat:

- geopandas
- ridge-detection
- numpy
- matplotlib
- PIL

- skimage
- scipy

### 3.1 Lähdeaineiston esikäsittely

Jotta lähdeaineistoa voidaan käsitellä, se on järkevää ensin muuntaa johonkin matemaattiseen muotoon. Koska kyseessä on kuvia, luonnollista on muodostaa niistä matriiseja. Vaikka lähdeaineisto on värikuvina, mustavalkokuvat ovat riittäviä valituille menetelmille. Tässä työssä  $\mathbb{N} = 0, 1, 2, \dots$  ja  $\mathbb{N}_1 = 1, 2, 3, \dots$ . Jos kuvan koko on  $n \times m$ , jossa  $n, m \in \mathbb{N}_1$ , niin se pyritään muuntamaan matriisiksi  $A \in \mathbb{N}^{n \times m}$ , jossa jokainen alkio kuuluu välille  $[0, 255]$ . Näin saatua matriisia kutsutaan *kuvamatriisiksi*. Jos kuvamatriisin alkio saa arvon 0, se tarkoittaa, että sitä vastaava pikseli on täysin musta. Vastaavasti alkion arvo 255 tarkoittaa pikselin olevan täysin valkoinen. Jatkossa merkintä  $n_a$  tarkoittaa kaikkia luonnollisia lukuja välillä  $[0, a]$ . Huomionarvoista on, että koska kuva muunnetaan matriisimuotoon, kuvan ylimmäinen rivi pikseleitä on matriisin ensimmäinen rivi pikseleitä. Tällöin matriisin rivinumeron kasvaessa, kuvan  $y$ -koordinaatti laskee. Kuva ja matriisi jakavat silti saman  $x$ -akselin, eli kuvan origo  $(0, 0)$  on sitä vastaavassa matriisissa  $(n - 1, 0)$ .

Nyt, kun lähdeaineiston kuvat ovat saatu haluttuun muotoon, täytyy vielä muuntaa lähdeaineistosta saadut koordinaattitiedot matemaattiseen muotoon. Luonnollisin valinta on yhdistää jokainen koordinaatti mahdolliseen kuvamatriisin alkion sijaintiin. Toisinsanoen täytyy tehdä koordinaattimuunnos. Tätä varten täytyy jotenkin varmistaa, että kukin koordinaatti kuvautuu oikein kuvamatriisiin. Lähdeaineistossa on koordinaattitiedot käsinpiirretyille raoille vertailua varten, mutta myös koordinaatit, jotka muodostavat monikulmion tulkittavan alueen ympärille. QGIS-ohjelmistolla, josta kerrotaan enemmän luvussa 3.1.1, pystyy yhdistämään kuvan ja tähän monikulmioon kuuluvat koordinaatit. Tämän jälkeen ohjelmiston avulla pystyy muodostamaan alkuperäisestä TIFF-kuvasta rajatun PNG-kuvan siten, että rajaus on pienin mahdollinen suorakulmio, joka sisältää annetun monikulmion. Tämä rajaus on havainnollistettu kuvassa 2. Määritellään tässä työssä käytettävä

*kuvamatriisin koordinaattimuunnos* käyttäen näin saatua kuvamatriisia ja suorakulmiota koordinaattitietoineen.



Kuva 2: Punainen monikulmio vastaa käsinmääritettyä tulkinta-aluetta. Tämän ympärille on piirretty mustalla pienin mahdollinen suorakulmio, jonka koordinaatteja käytetään koordinaattimuunnoksissa, ja jonka sisäänjäävästä alueesta luodaan käsiteltävä kuvamatriisi.

**Määritelmä 3.1** (Kuvamatriisin koordinaattimuunnos). Olkoon  $A \in \mathbb{N}^{n \times m}$  kuvamatriisi, ja  $[x_0, x_1] \times [y_0, y_1]$  tätä vastaava suorakulmio karttakoordinaatistojärjestelmässä  $K$ . Tällöin koordinaattimuunnos pisteelle  $(x, y) \in K$  kuvamatriisiin  $A$  on

$$\begin{aligned} x' &= \lfloor (m-1) \frac{x-x_0}{x_1-x_0} \rfloor \in [0, m] \\ y' &= \lfloor (n-1) \frac{y-y_0}{y_1-y_0} \rfloor \in [0, n]. \end{aligned} \quad (1)$$

Voidaan myös asettaa jokainen kuvamatriisin  $A$  alkio  $a_{ij}$  vastaamaan koordinaatistojärjestelmän  $K$  pistettä:

$$\begin{aligned} x' &= (x_1 - x_0) \frac{j}{m} + x_0 \in [x_0, x_1] \\ y' &= (y_1 - y_0) \frac{n-i}{n} + y_0 \in [y_0, y_1]. \end{aligned} \quad (2)$$

▽

### 3.1.1 QGIS-ohjelmisto

QGIS [10] on vapaa paikkatieto-ohjelmisto, joka sisältää mahdollisuuden tiedon selaamiseen, muokkaamiseen ja analysointiin. Tässä työssä sitä erityisesti

käytetään alkuperäisten kuvien kompressointiin ja rajaukseen sekä lopputulosten tarkasteluun. Ohjelmistoa voi myös käyttää esimerkiksi maanpinnan topologisten muotojen analysoinnissa. Tässä työssä on käytetty QGIS:n versiota 3.12.1.

## 3.2 Metriikat

Menetelmästä riippumatta on välttämätöntä, että ohjelman onnistuvuutta pystytään mittaamaan jollain sopivalla metriikalla. Kuva-analyysia varten on olemassa useita eri metriikkoja, kuten *Dice similarity coefficient* (DSC) [6], joka määritellään seuraavasti:

**Määritelmä 3.2** (DSC). Olkoon  $X$  ja  $Y$  diskreettejä joukkoja. Tällöin

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}. \quad (3)$$

▽

Jos joukot  $X$  ja  $Y$  olisivat koordinaatteja kuvamatriisissa, joista toinen sisältäisi koordinaatit ennustettaville pisteille ja toinen käsinpiirretyille, tämä olisi hyvä metriikka ohjelmalle. Ongelmana kuitenkin on, että lähdemateriaali on joukko koordinaatteja, jotka kuvaavat kunkin raon janan pätkinä, ja pelkästään janan päätepisteet ovat annettu. Tämä ei kuitenkaan sovellu kuvaamaan rakoja täydellisesti todellisuudessa, sillä vaikka rako voitaisiin ilmaista janojen avulla, sillä on myös paksuus, joka saattaa vaihdella suuresti eri rakojen välillä, kuten nähdään kuvassa 3.

DSC:n arvot vaihtelevat välillä  $[0,1]$ . Jos DSC saa arvon 1, vertailtavat alueet ovat identtiset ja arvolla 0 täysin erilliset. Kuvan resoluutiosta riippuen ohjelman ehdottama ratkaisu tietyn raon päätepisteille voi erota malliratkaisusta muutamalla pikselillä, ja päätepisteet saattavat silti kuulua rakoon, eli ratkaisu on tarpeeksi tarkka. DSC kuitenkin tulkitsee yhdenkin pikselin eron jäävän ulos leikkauksesta (kts. määritelmä 3.2), ja näin ollen DSC:n antama arvo laskee, vaikka ennuste olisi ollut täysin kelvollinen.



(a)



(b)

Kuva 3: Kuvassa (a) kaksi kapeampaa rakoja kohtaavat yhden paksumman raon. Kuvassa (b) pystyy paremmin tarkastelemaan rakojen kokoeroja.

### 3.2.1 Rakoverkosto ja binäärimatriisi

Eräs tapa lähestyä tätä ongelmaa on arvioida keskimääräistä raon paksuutta ja muodostaa annetuista rakoverkoston päätepisteistä binäärimatriisi. Tässä työssä määritellään  $\mathbb{R} = (-\infty, \infty)$ . Määritellään ensin *rakoviiva*, joka on raon matemaattinen ilmaisu:

**Määritelmä 3.3** (Rakoviiva). Olkoon  $J$  rakoviiva ja  $\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbb{R}^2$  siihen kuuluvien janojen päätepisteet,  $n \geq 1$  ja  $a_i \neq a_j$ , kaikille  $i \neq j$ . Nyt rakoviiva koostuu murtoviivasta:

$$J = \{ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = c\mathbf{a}_i + (1 - c)\mathbf{a}_{i+1}, 0 \leq i \leq n - 1, i \in \mathbb{N}, c \in [0, 1] \}. \quad (4)$$

Rakoviivan päätepisteet ovat pisteet  $\mathbf{a}_0$  ja  $\mathbf{a}_n$ . ▽

Tyypillisesti yksi rakoviiva sisältää yhdestä kymmeneen janaa. Määritellään seuraavaksi *rakoviivaverkosto*, joka koostuu kuvan koosta riippuen kymmenistä, aina satoihin rakoviivoihin asti.

**Määritelmä 3.4** (Rakoviivaverkosto). Olkoon rakoviivaverkoston  $J$  kuuluvat rakoviivat  $J_0, J_1, \dots, J_r$ ,  $r \geq 0$ . Tällöin rakoviivaverkosto  $J$  on

$$J = \bigcup_{0 \leq i \leq r} J_i. \quad (5)$$

▽

**Määritelmä 3.5** (Rakoviivaverkoston binäärimatriisi). Olkoon  $J$  rakoviivaverkosto. Ensin rakoviivaverkostolle tehdään koordinaattimuunnos. Olkoon  $x_{min}$  koordinaatti avaruudessa  $\mathbb{R}^2$  siten, että  $x_{min_1}$  on pienin  $x$ -koordinaatti rakoviivaverkostossa  $J$ , ja  $x_{min_2}$  pienin  $y$ -koordinaatti, eli

$$x_{min} = \{(x_{min_1}, x_{min_2}) \in \mathbb{R}^2 \mid \begin{aligned} x_{min_1} &= \min\{x_1 \in \mathbb{R} \mid (x_1, x_2) \in J\}, \\ x_{min_2} &= \min\{x_2 \in \mathbb{R} \mid (x_1, x_2) \in J\}. \end{aligned} \} \quad (6)$$

Vastaavasti määritellään

$$x_{max} = \{(x_{max_1}, x_{max_2}) \in \mathbb{R}^2 \mid \begin{aligned} x_{max_1} &= \max\{x_1 \in \mathbb{R} \mid (x_1, x_2) \in J\}, \\ x_{max_2} &= \max\{x_2 \in \mathbb{R} \mid (x_1, x_2) \in J\}. \end{aligned} \} \quad (7)$$

Nyt, koordinaattimuunnos määrittelee funktion  $\mathbf{x} = (x_1, x_2) \rightarrow \mathbf{x}' = (x'_1, x'_2)$ :

$$\begin{aligned} x'_1 &= (n-1) \frac{x_1 - x_{min_1}}{x_{max_1} - x_{min_1}} \\ x'_2 &= (m-1) \frac{x_2 - x_{min_2}}{x_{max_2} - x_{min_2}}, \end{aligned} \quad (8)$$

jossa  $n$  ja  $m$  vastaavat halutun binäärimatriisin ulottuvuuksia. Binäärimatriisi  $B$  voidaan nyt kirjoittaa seuraavasti:

Olkoon  $p \in \mathbb{R}, p > 0$  rakojen keskiarvoinen paksuus, ja  $b_{ij}$  binäärimatriisin  $B$  alkio rivissä  $i$  ja sarakkeessa  $j$ .

$$b_{ij} = \begin{cases} 1, & \text{jos } \exists \mathbf{x}' \in J' : x'_1 - p \leq j \leq x'_1 + p \text{ ja } x'_2 - p \leq i \leq x'_2 + p, \\ 0, & \text{muulloin.} \end{cases} \quad (9)$$

▽

Jatkossa merkintä  $B^J(p)$  tarkoittaa rakoviivaverkoston  $J$  binäärimatriisia  $B$  raon keskimääräisellä paksuudella  $p$  ja tästä käytetään nimitystä *lähdematriisi*.

### 3.2.2 Ennusteen vertailu lähdematriisiin

*Ennustematriisilla* tarkoitetaan mitä tahansa binäärimatriisia  $A \in [0, 1]^{n \times m}$ , joka pyrkii hahmottamaan, missä raot sijaitsevat kuvassa. Ennustematriisilla  $A_v \in \{0, 1\}^{n \times m}$  tarkoitetaan binäärimatriisia, jonka alkiolle  $a_{ij}$  pätee

$$a_{ij} = \begin{cases} 1, & \text{jos pikselin } (i, j) \text{ ennustetaan kuuluvan rakoviivaan,} \\ 0, & \text{muulloin.} \end{cases} \quad (10)$$

Tätä matriisia tullaan kutsumaan *viivamatriisiksi*.

Määritellään binäärimatriiseille  $A, B$ , jotka ovat samaa muotoa, *unioni, leikkaus ja mahtavuus*:

**Määritelmä 3.6** (Unioni). Olkoon  $A, B \in \{0, 1\}^{n \times m}$  binäärimatriiseja. Silloin matriisien  $A$  ja  $B$  unioni määritellään:

$$A \cup B = \begin{cases} 1, & \text{jos alkio } a_{ij} = 1 \text{ tai alkio } b_{ij} = 1, \\ 0, & \text{muulloin.} \end{cases} \quad (11)$$

▽

**Määritelmä 3.7** (Leikkaus). Olkoon  $A, B \in \{0, 1\}^{n \times m}$  binäärimatriiseja. Silloin matriisien  $A$  ja  $B$  leikkaus määritellään:

$$A \cap B = \begin{cases} 1, & \text{jos alkio } a_{ij} = 1 \text{ ja alkio } b_{ij} = 1, \\ 0, & \text{muulloin.} \end{cases} \quad (12)$$

▽

**Määritelmä 3.8** (Mahtavuus). Olkoon  $B \in \{0, 1\}^{n \times m}$  binäärimatriisi ja  $b_{ij}$  sen alkio. Silloin matriisin mahtavuus määritellään:

$$|B| = \sum_{0 \leq i \leq n, 0 \leq j \leq m} b_{ij}, \quad (13)$$

eli niiden pikseleiden lukumäärä, jotka tulkitaan kuuluvan rakoon.

▽

Määritellään kummallekin ennustematriisille *ennustemetriikka*, joka kuvaa ennusteen onnistuvuutta antamalla ennustematriisille  $A$  arvon väliltä  $[0, 1]$  siten, että suuremmat arvot kuvaavat paremmin onnistunutta ennustetta lähdematriisiin  $B$  nähden.

**Määritelmä 3.9** (Ennustemetriikka viivamatriisille). Olkoon  $B^J(p) \in \mathbb{R}^{n \times m}$  lähdematriisi ja  $A_v \in \mathbb{R}^{n \times m}$  viivamatriisi. Tällöin ennustemetriikka  $M_v$  viivamatriisille  $A_v$  lähdematriisilla  $B^J(p)$  määritellään

$$M_v = p \frac{|A_v \cap B^J(p)|}{|B^J(p)|}. \quad (14)$$

▽

Esitettyjen metriikoiden lisäksi tutkitaan rakoviivojen lukumäärien, pituuksien, kulmien ja eri pituisten rakoviivojen kulmien tilastollisia jakauksia ennustettujen ja todellisten rakoviivojen välillä. Tilastollisen tarkastelun tuloksista ei voida suoraan päätellä, kuinka hyvin rakoviivojen mallintaminen on onnistunut, mutta ne voivat vahvistaa viivamatriisien ennustemetriikan kelpoisuutta ja antaa mahdollisesti lisätietoa, missä mallintaminen on onnistunut paremmin ja missä huonommin. Tilastollisen tarkastelun yhteydessä rakoviivat yksinkertaistetaan janoiksi, joiden päätepisteet ovat sitä vastaavan rakoviivan päätepisteet. Tällä tavoin on loogista ajatella tilastollisen tarkastelun kontekstissa, että pituus on näin saadun janan pituus ja kulma  $\alpha$  voidaan ilmaista kulmakertoimen  $k \in (-\infty, \infty]$  funktiona  $\alpha : (-\infty, \infty] \rightarrow (-\frac{\pi}{2}, \frac{\pi}{2}]$ ,

$$\alpha(k) = \begin{cases} \tan^{-1}(k), & \text{jos } k \in \mathbb{R} \\ \frac{\pi}{2}, & \text{muulloin.} \end{cases} \quad (15)$$

Kulmakertoimen  $k$  tulkitaan saavan arvon  $\infty$  jos ja vain jos edellämainitun janan päätepisteiden  $x$ -koordinaatit ovat samat. Seuraavaksi esitellään tilastolliset tarkastelumenetelmät tarkemmin.

Olkoon  $J_a$  alkuperäinen rakoviivaverkosto ja  $J_p$  ennustettu rakoviivaverkosto. Muodostetaan joukot  $A$  ja  $P$  jotka sisältävät alkuperäisten ja ennustettujen rakoviivojen tutkittavat ominaisuudet. Tutkitaan ensin joukkojen normaaliisuutta Shapiro-Wilk-testin avulla. Jos ainakin toisen joukon  $p$ -arvo on alle 0.05, tutkitaan joukkojen eroja Kruskal-Wallis-testin avulla, muutoin Studentin  $t$ -testin avulla.

Kruskal-Wallis  $H$ -testiin tulosta (eli  $H$ -arvo) varten jokaiselle havainnolle jokaisessa vertailtavassa joukossa annetaan *sija*, joka riippuu havainnon arvosta.  $H$ -arvo saadaan kaavasta

$$H = (N - 1) \frac{\sum_{i=1}^g n_i (\bar{r}_i - \bar{r})^2}{\sum_{i=1}^g \sum_{j=1}^{n_i} (r_{ij} - \bar{r})^2}, \quad (16)$$

jossa

- $g$  on vertailtavien joukkojen lukumäärä

- $n_i$  on havaintojen määrä joukossa  $i$
- $r_{ij}$  on joukon  $i$  havainnon  $j$  sija
- $N$  on kaikkien joukkojen yhteenlaskettu havaintojen määrä
- $\bar{r}_i$  on joukon  $i$  havaintojen sijojen keskiarvo
- $\bar{r} = \frac{N+1}{2}$  on kaikkien havaintojen sijojen keskiarvo.

Tätä  $H$ -arvoa vertaillaan johonkin kriittiseen arvoon. Jos  $H$ -arvo on suurempi kuin tämä kriittinen arvo, nollahypoteesi joukkojen samanlaisuudesta voidaan hylätä. Tätä kriittistä arvoa arvioidaan tässä työssä  $\chi^2$ -jakaumalla. Kulmia tarkastellessa on tärkeää huomata, että funktion (15) arvojoukon supremum ja infimum kuvastavat itseasiassa samansuuntaisia rakoviivoja. Tämä ilmiö täytyy ottaa huomioon kulmia tarkastellessa.

## 4 Signaalinkäsittely

Signaalinkäsittelyssä ideana on käsitellä annettua kuvadataa suoraan erilaisilla menetelmillä. Toisin kuin esimerkiksi koneoppimisessa, signaalinkäsittelymenetelmää ei tarvitse kouluttaa ja sen tulokset eivät riipu sille aikaisemmin syötetystä aineistosta. Mahdollisia esikäsittelyitä ja datamuutoksia lukuunottamatta se käsittelee annettua dataa sellaisenaan. Tässä luvussa esitellään työssä esiintyvien signaalinkäsittelymenetelmien toimintaperiaatteita.

### 4.1 Ridge-tunnistus

Tässä työssä keskeisin signaalinkäsittelymenetelmä on Ridge-tunnistus [5]. Ridge-tunnistus valittiin, koska se etsii samanaikaisesti viivoja jokaiselle kulle ja ottaa huomioon, että kuvan kontrasti saattaa vaihdella raon eri puolilla.

#### 4.1.1 1D-tapaus

Määritellään ensin rakofunktio yksiulotteisessa tapauksessa.

**Määritelmä 4.1** (Rakofunktio). Olkoon  $w > 0$ ,  $a \in [0, 1]$  kuvan suhteellinen kontrasti toisella puolella rakoja, ja raon leveys  $2w$ . Nyt rako voidaan määritellä funktiona  $f : \mathbb{R} \rightarrow [0, 1]$ ,

$$f(x) = \begin{cases} 1, & \text{jos } x \in [-w, w], \\ a, & \text{jos } x > w, \\ 0, & \text{muulloin.} \end{cases} \quad (17)$$

▽

Jos kuvassa raot ovat tummempia pikseleitä, ja näin ollen saavat pienempiä arvoja, niin analysoidaan rakofunktiota  $u(x) = 1 - f(x)$ . Lisäksi rakofunktiota voi skaalata skalaarilla  $h \in \mathbb{R}$  ja tutkia näin saatua funktiota  $v(x) = hf(x)$ . Rakofunktiota tarkastellessa tutkitaan kuitenkin funktioiden  $f$  ja  $g$  konvoluution  $r(x, \sigma, w, h) = (f * g_\sigma)(x)$  käyttäytymistä, jossa  $g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$  on normaalijakauman tiheysfunktio odotusarvolla 0 ja keskihajonnalla  $\sigma$ , joka kuvaa rakojen leveyksien keskihajontaa. Määritellään ensin konvoluutio:

**Määritelmä 4.2** (Konvoluutio). Olkoon  $f$  ja  $g$  integroituvia funktioita. Tällöin niiden *konvoluutio* määritellään

$$(f * g)(x) = \int_{-\infty}^{\infty} f(x-t)g(t)dt. \quad (18)$$

Jos  $f$  ja  $g$  ovat diskreettejä, niin niiden konvoluutio määritellään

$$(f * g)(n) = \sum_{k=-\infty}^{\infty} f(k)g(n-k). \quad (19)$$

▽

Nyt voidaan tarkastella sileän funktion  $r$  ensimmäistä ja toista derivaattaa:

$$r(x, \sigma, w, h) = g_\sigma(x) * f(x) = h(\phi_\sigma(x+w) - \phi_\sigma(x-w)), \quad (20)$$

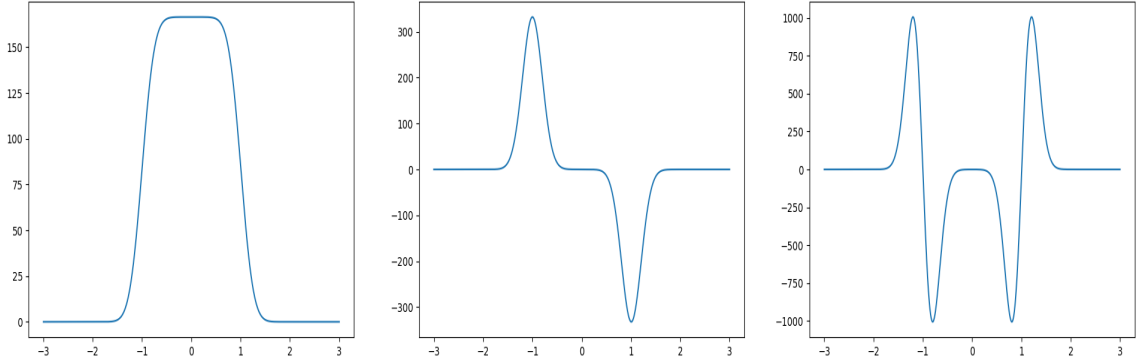
$$r'(x, \sigma, w, h) = g'_\sigma(x) * f(x) = h(g_\sigma(x+w) - g_\sigma(x-w)), \quad (21)$$

$$r''(x, \sigma, w, h) = g''_\sigma(x) * f(x) = h(g'_\sigma(x+w) - g'_\sigma(x-w)), \quad (22)$$

jossa

$$\phi_\sigma(x) = \int_{-\infty}^x e^{-\frac{t^2}{2\sigma^2}} dt. \quad (23)$$

Raon reunat voidaan nyt löytää joko funktion  $r'(x, \sigma, w, h)$  ääriarvoista tai funktion  $r''(x, \sigma, w, h)$  nollakohdista. Tämä havainnollistetaan kuvassa 4.



Kuva 4: Kuvassa havainnollistettu vasemmalta oikealle funktioiden  $r$ ,  $r'$  ja  $r''$  arvoja  $x$ -akselilla, kun  $\sigma = 0.2$ ,  $w = 1$  ja  $h = 1$ .

Aineistona on kuitenkin kuvia, joten niissä esiintyvät raot ovat pikseleitä, toisin sanoen siis diskreettejä. Tässä tapauksessa rako määritellään lukujo-  
nona  $f_n$  ja varsinainen rakofunktio  $f(x) = f_n$ , kun  $x \in (n - \frac{1}{2}, n + \frac{1}{2}]$ . Tällöin konvoluutiossa käytetty maski  $g_\sigma$  saa myös diskreetin muodon:

$$g_{n,\sigma} = \phi_\sigma(n + \frac{1}{2}) - \phi_\sigma(n - \frac{1}{2}) \quad (24)$$

$$g'_{n,\sigma} = g_\sigma(n + \frac{1}{2}) - g_\sigma(n - \frac{1}{2}) \quad (25)$$

$$g''_{n,\sigma} = g'_{n,\sigma}(n + \frac{1}{2}) - g'_{n,\sigma}(n - \frac{1}{2}). \quad (26)$$

Diskreetissä tapauksessa rakojen reunojen sijaintia selvittäessä tutkitaan lukujonon  $f_n$  muodostaman funktion toisen asteen Taylorin polynomia. Olkoon  $r, r', r''$  funktion arvioidut derivaatat kuvan pisteessä  $n$ , jotka on saatu ottamalla konvoluutio kuvan pisteistä funktioiden  $g_n, g'_n, g''_n$  kanssa. Tällöin Taylorin polynomi  $p$  on

$$p(x) = r + r'x + \frac{1}{2}r''x^2. \quad (27)$$

Raon sijainti on tällöin  $x = -\frac{r'}{r''} = \hat{x}$ . Piste  $n$  tulkitaan kuuluvan rakoön, jos näin saadulle ratkaisulle pätee  $\hat{x} \in [-\frac{1}{2}, \frac{1}{2}]$  ja toinen derivaatta  $r''$  on suurempaa kuin käyttäjän asettama raja-arvo.

#### 4.1.2 2D-tapaus

Kaksiulotteisessa tapauksessa raot voidaan ajatella viivoina  $s(t)$ , joilla on suunta, joka on kohtisuoraan niitä vastaavan yksiulotteisen raon  $f$  kanssa. Näin ollen, yksiulotteisen raon menetelmä pystytään yleistämään kaksiulotteiseen tapaukseen sillä lisäehdolla, että jokaiselle kuvan pikselille on laskettava sen suunta käyttäen konvoluutiofunktion  $r$  osittaisderivaattoja  $r_x, r_y, r_{xx}, r_{xy}$  ja  $r_{yy}$ , ja tutkimalla niiden Hessian matriisia

$$H(x, y) = \begin{pmatrix} r_{xx} & r_{xy} \\ r_{xy} & r_{yy} \end{pmatrix}. \quad (28)$$

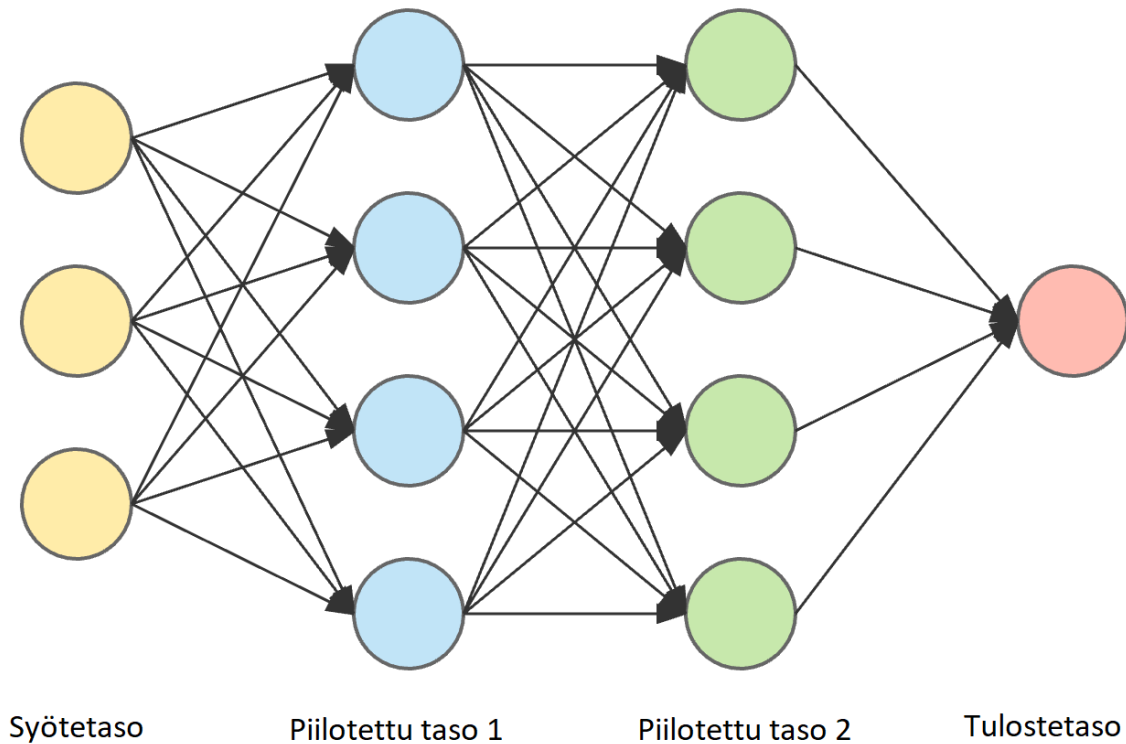
## 5 Koneoppiminen

Signaalinkäsittelyn lisäksi tämän työn menetelmässä käytetään myös koneoppimista. Toisin kuin signaalinkäsittelyssä, koneoppiminen mahdollistaa yleisten häiriötekijöiden, kuten varjojen, kasvien ja veden, karsinnan. Ideaalisti luotu koneoppimismalli karsisi pois kaikki häiriötekijät ja suuresti korostaisi kuvan rakoja. Seuraavaksi esitellään hieman yleisemmin ohjatun koneoppimisen toimintaperiaatteita ja sen jälkeen työhön valitun mallin ominaisuuksia.

### 5.1 Toimintaperiaate

Ohjatun yksisuuntaisen koneoppimismallin toimintaperiaate perustuu tasoihin ja niissä sijaitseviin soluihin. Mallin ensimmäinen taso kutsutaan syötetasoksi ja viimeistä tulostetasoksi. Näiden välissä olevia tasoja kutsutaan piilotetuiksi tasoiksi, ja näitä saattaa olla useita. Jokainen taso taas sisältää useita soluja. Jokainen solu sisältää analysoitavaa dataa. Tätä dataa muutetaan malliin määritetyllä tavalla kyseisen solun sen hetkisten painoarvojen avulla, ja näin saatu uusi data syötetään muille soluille. Soluissa olevia

painoarvoja pyritään optimoimaan, kun mallia vielä koulutetaan. Yksisuuntaisessa koneoppimismallissa dataa syötetään vain "eteenpäin", eli pelkäs-  
tään soluille, jotka sijaitsevat jossain myöhemmässä tasossa. Kuvassa 5 esi-  
tetty kaavio havainnollistaa tätä prosessia. Jokainen solu siis vastaanottaa



Kuva 5: Kuvassa eräs yksisuuntainen koneoppimismalli, jossa ympyrät ovat soluja ja samanväriset ympyrät muodostavat yhden tason. Nuolet kuvaavat datan kulkureitin.

arvoja muilta soluilta. Olkoon vastaanotetut arvot  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^m$  ja näitä vastaavat painoarvot  $w_1, w_2, \dots, w_n \in \mathbb{R}$ . Tällöin solun antama arvo on  $\sum_{i=1}^n \mathbf{x}_i w_i$ , joka annetaan syötteeksi solun *aktivointifunktiolle*. Aktivointifunktio yleensä syöttää jokaisen syötteenä saaman vektorin tai matriisin alkion jollekin toiselle funktiolle, jonka tarkoituksena on rajoittaa mahdollisia arvoja. Tyypillisimpiä käytettyjä funktiota ovat logistinen funktio, *tanh*

ja alhaalta rajoitettu *max* -funktio, eli järjestyksessä:

$$f(x) = \frac{1}{1 + e^{-x}},$$
$$f(x) = \tanh(x),$$
$$f(x) = \max(0, x).$$

On myös täysin mahdollista, että aktivointifunktio on identiteettifunktio  $f(x) = x$ .

## 5.2 Koulutus- ja testiaineisto

Ennen kuin mallia aletaan kouluttamaan, aineisto jaetaan kahteen osaan: koulutus- ja testiaineistoon. Koulutusaineiston avulla malli koulutetaan syöttämällä sille aineistoa ja oikeita vastauksia, joihin verrata. Malli korjaa itseään sille määritetyllä tavalla aina ennen seuraavan koulutusaineistoryypään syöttämistä. Kun malli on koulutettu, erilliselle testiaineistolle tehdään ennuste, jonka tuloksia tarkastellaan. Tällä tavoin vältetään ylisovittamiselta ja saadaan realistisempi käsitys siitä, millaisia tulokset oikeasti ovat kun mallia vielä kehitetään. Tämän työn menetelmää kehittäessä kuva 1 jaettiin erillisiin  $256 \times 256$  osiin, joista kolmasosa käytettiin testiaineistona ja loput kuuluivat koulutusaineistoon. Kuvan osiinjakaminen käydään läpi tarkemmin luvussa 6.1, jossa myös kerrotaan, miten toimitaan kun kuvan dimensiot eivät ole luvun 256 monikertoja. Osasta näistä koulutusaineiston kuvista myös luotiin lisäaineistoa esimerkiksi kääntämällä, venyttämällä ja lisäämällä kohinaa, ja näin saatiin keinotekoisesti nostettua koulutusaineiston määrää. Työn lopussa suoritettiin myös validointi, jolloin malli koulutetaan kokonaan eri kuville kuin mille ennuste tehdään, jotta nähdään, miten hyvin malli pystyy ennustamaan todellisuudessa.

## 5.3 U-Net

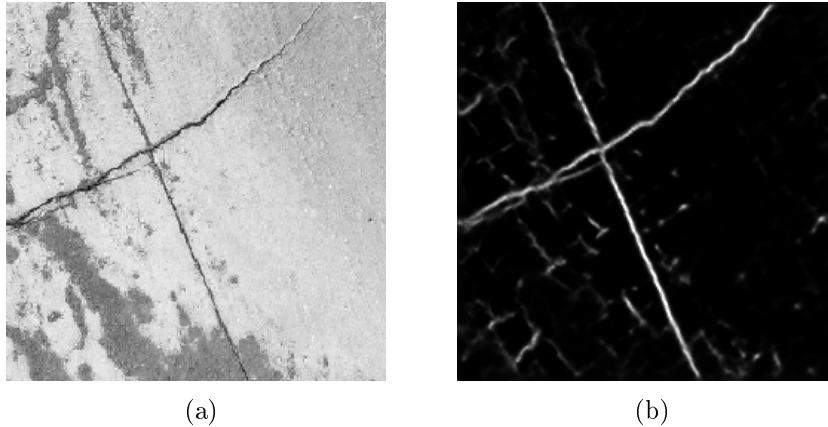
Tähän työhön valittu neuroverkkomalli U-Net on konvoluutioneuroverkko, joka eroaa monikerroksisesta perseptroniverkosta siten, että mallissa esiintyvät kerrokset eivät ole täysin yhdistyneitä seuraavaan kerrokseen. Täl-

lä tavoin konvoluutioneuroverkko mahdollistaa yksinkertaisempien muotojen, kuten viivojen, havaitsemisen. U-Net saa koulutusvaiheessa syötteen useamman  $256 \times 256$  kuvaparin. Yhden mustavalkokuvan, joka on siis matriisi  $n_{255}^{256 \times 256}$  ja yhden binäärimatriisin, jonka alkiot arvolla 1 tulisi tulkita kuuluvan rakoon. Malli tuottaa ennusteen, joka esitellään kuvassa 6 ja vertailee tätä ennustetta annettuun binäärimatriisiin laskemalla sille määritetyn häviöfunktion avulla jonkin suureen. Tätä suureta pyritään minimoimaan. Tässä työssä häviöfunktioiksi on valittu logaritminen häviöfunktio:

**Määritelmä 5.1** (Logaritminen häviöfunktio). Olkoon ennustettavien pisteiden määrä  $N$ . Olkoon  $y_i$  pisteen  $i$  oikea arvo  $y_i \in 0, 1$  ja  $p(y_i)$  ennustettu todennäköisyys, että  $y_i = 1$ . Tällöin Logaritminen häviöfunktio on

$$H(p) = -\frac{1}{N} \sum_{i=1}^N y_i \ln(p(y_i)) + (1 - y_i) \ln(1 - p(y_i)). \quad (29)$$

▽



Kuva 6: Vasemmalla olevasta  $256 \times 256$  kuvasta luodaan U-Netin avulla ennuste (oikealla).

Varsinainen ennustamisprosessi, jossa syötematriisista luodaan ennustematriisi, on erittäin monimutkainen ja sisältää monta askelta. Nämä askeleet on esitelty tarkemmin kuvassa 7. Esitellään kaksi tyypillisintä ennustamisprosessin askelta seuraavaksi: *matriisikonvoluutio* ja *max-pool-funktio*.

**Määritelmä 5.2** (Matriisikonvoluutio). Olkoon  $A \in \mathbb{R}^{n \times m}$  ja  $B \in \mathbb{R}^{p \times q}$

matriiseja, ja  $p < n, q < m$ . Tällöin matriisien välinen konvoluutio on:

$$B * A = C \in \mathbb{R}^{(n-p+1) \times (m-q+1)}, \quad (30)$$

jossa matriisin  $C$  alkio  $c_{ij}$  on sen matriisin  $A$  alimatriisin pistetulo matriisin  $B$  kanssa, jonka ensimmäinen rivi on matriisin  $A$  rivillä  $i$ , ensimmäinen sarake matriisin  $A$  sarakkeella  $j$  ja on muotoa  $\mathbb{R}^{p \times q}$ , olettaen, että tällainen alimatriisi on olemassa. Toisinsanoen:

$$c_{ij} = \sum_{k=0}^{p-1} \sum_{t=0}^{q-1} a_{i+k, j+t} b_{k,t}, \text{ kun } i \leq n-p+1 \text{ ja } j \leq m-q+1, \quad (31)$$

jossa siis  $a_{ij}$  ja  $b_{ij}$  ovat vastaavasti matriisien  $A$  ja  $B$  alkioita rivillä  $i$  ja sarakkeella  $j$ . Matriisin  $B$  alkioita ovat optimoitavia painoarvoja, ja siksi matriisia kutsutaan nimillä *painomatriisi* tai *maski*.  $\nabla$

**Määritelmä 5.3** (Max-pool). Olkoon  $A \in \mathbb{R}^{n \times m}$ ,  $b, c \in \mathbb{N}_1$  siten, että  $b$  jakaa luvun  $n$  ja  $c$  jakaa luvun  $m$ . Tällöin Max-pool-funktio  $f^{b \times c}$  on määritelty arvolla  $A$ , ja

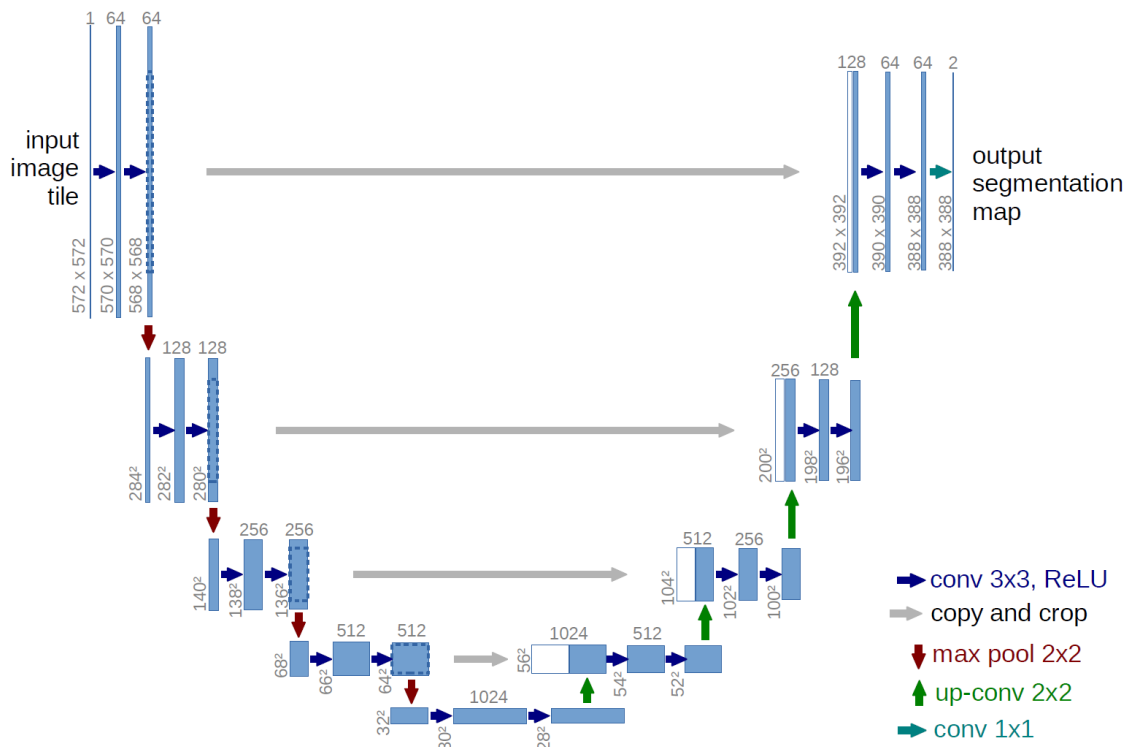
$$f^{b \times c}(A) = C \in \mathbb{R}^{\frac{n}{b} \times \frac{m}{c}}, \quad (32)$$

jossa matriisin  $C$  alkio  $c_{ij} = \max\{a_{kt} \mid i \leq k \leq i+b-1, j \leq t \leq j+c-1\}$ , jossa  $a_{kt}$  on matriisin  $A$  alkio.  $\nabla$

## 6 Lopullinen menetelmä

Lopullisessa menetelmässä alkuperäinen TIFF-kuva on muunnettu PNG-kuvaksi, samalla pienentäen resoluutiota. Tätä PNG-kuvaa kutsutaan jatkossa *alkukuvaksi* ja TIFF-kuvaa joko *lähdeaineiston kuvaksi* tai *skaalaamattomaksi kuvaksi*. Alkukuva ja sitä vastaava .shp-tiedosto, joka sisältää koordinaattitiedot analysoitavasta alueesta (kts. luku 3.1), syötetään menetelmälle, minkä jälkeen siinä on kuusi pääosiota:

1. Alkukuvan käsittely
2. Ennusteen tekeminen U-Netillä



Kuva 7: Kuvassa U-Netin ennustamisprosessi [11], johon erilaiset matriisin muutosfunktiot on merkattu erivärisillä nuolilla. Konvoluutioissa saatetaan käyttää useaa eri painomatriisia, luoden useampia konvoluutioita.

3. Ridge-tunnistuksen soveltaminen U-Netin ennusteeseen
4. Rakoviivojen luominen Ridge-tunnistuksen ennusteesta
5. Rakoviivojen yhdistäminen
6. Rakoviivojen koordinaattien muuntaminen.

Menetelmän pääosiot käydään läpi yksityiskohtaisemmin tämän luvun aliluvuissa. Menetelmällä, ja eritoten sen tuottamalla ratkaisulla, on tiettyjä kriteereitä, joiden täytyy toteutua. Nämä kriteerit ovat:

1. Menetelmän täytyy nopeuttaa rakoviivojen luontiprosessia.
2. Menetelmän tuottamien rakoviivojen täytyy olla avoimia murtoviivoja.

3. Kunkin rakoviivan päätepisteiden täytyy olla uniikkeja kaikkien rakoviivojen päätepisteiden joukossa, eli rakoviiva ei saa alkaa tai päättyä pisteestä, mistä toinen rakoviiva alkaa tai päättyy.

## 6.1 Alkukuvan käsittely

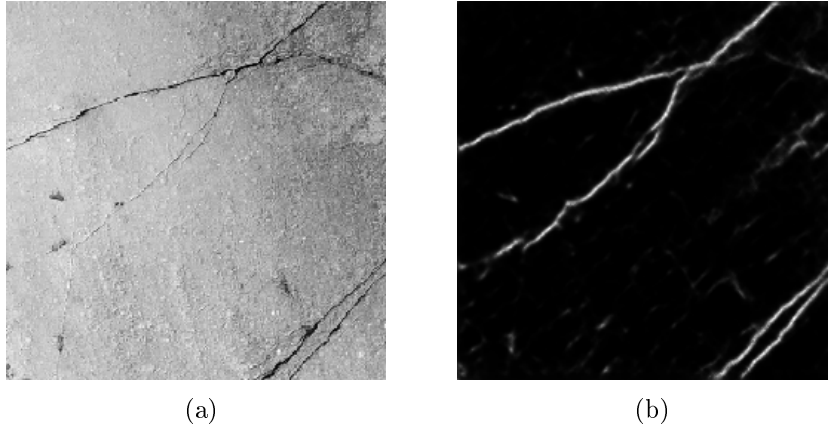
Alkukuva avataan ohjelmassa ja käännetään mustavalkokuvaksi, eli  $n_{256}^{p \times q}$  matriisiksi. Tämän lisäksi menetelmä tallentaa alueen koordinaattitiedot ja alkukuvan rivien ja sarakkeiden määrän muistiin. Erityisesti ohjelma tallentaa määritelmässä 3.1 esiintyvät luvut  $n$ , matriisin rivien määrä,  $m$ , matriisin sarakkeiden määrä,  $x_0$ , pienin  $x$ -koordinaatti koordinaattijärjestelmässä,  $x_1$ , suurin  $x$ -koordinaatti,  $y_0$ , pienin  $y$ -koordinaatti ja  $y_1$ , suurin  $y$ -koordinaatti. Tämän jälkeen alkukuva jaetaan  $256 \times 256$  kokoisiksi kuviksi siten, että jos alkukuva ei ole leveydeltään tai korkeudeltaan jaollinen luvulla 256, niin sitä kasvatetaan oikealta tai alhaalta lisäämällä perään mustia pikseleitä, kunnes tämä ehto toteutuu. Kutsutaan näitä kuvia (alkukuvan) *alikuviiksi*.

## 6.2 Ennusteen tekeminen U-Netillä

Jokaista alikuvaa ensin analysoidaan seuraavasti: jos 95% tai enemmän kuvan pikseleistä ovat täysin mustia (pikselin arvo 0), niin kuva ei todennäköisesti sisällä mitään merkittäviä rakoja, ja sitä ei siis analysoida enempää. Tällaisia kuvia saattaa ilmaantua esimerkiksi analysoitavan alueen muodon vuoksi. Tämä on nähtävissä kuvan 2 mustan suorakulmion vasemmassa alaja oikeassa yläkulmassa. Tämän jälkeen kuva syötetään U-Netille, joka antaa tuloksena  $[0, 1]^{256 \times 256}$  matriisin, joka havainnollistetaan kuvassa 8.

## 6.3 Ridge-tunnistuksen soveltaminen U-Netin ennusteseen

Luvussa 4 esitellylle Ridge-menetelmälle annetaan syötteenä U-Netin tuottama ennustematriisi. Ridge-tunnistuksen pitäisi olla vähemmän altis häiriötekijöille, kun sille syötetään U-Netin tuottama ennuste alkuperäisen alikuvan

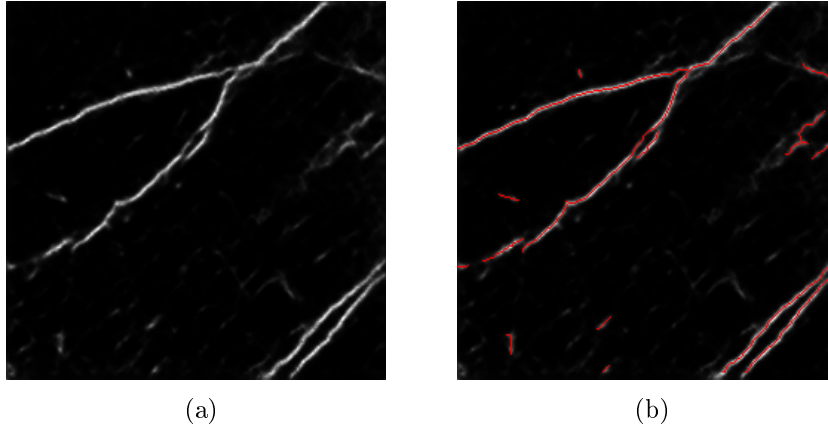


Kuva 8: Vasemmalla oleva  $256 \times 256$  kuva on vain osa analysoitavaa kuvaa. Oikealla nähdään U-Netin tuottama ennuste.

sijasta. Tämän lisäksi Ridge-tunnistus myös jakaa löydetyt rakoviivojen pisteet omiin joukkoihin siten, että yksi tällainen joukko muodostaa rakoviivan matriisissa. Ridge-tunnistuksen tuloksessa on kuitenkin kaksi ongelmaa:

- Ridge-tunnistuksen määrittelemä rakoviiva on joukko  $(x, y)$ -koordinaatteja, joista jokainen on alkio matriisissa. Näiden rakoviivojen jatkoanalyysi on erittäin raskasta, kun yksi rakoviiva sisältää jopa satoja pisteitä ja näitä rakoviivoja itsessään voi olla satoja. Ridge-tunnistuksen tulokset ovat havainnollistettu kuvassa 9.
- Rako saattaa ohentua tai täysin kadota kallioperän pinnalta, jolloin joko U-Netin antama ennuste tai Ridge-tunnistuksen antama tulos saattaa tulkita raon kahdeksi erilliseksi rakoviivaksi.

Näistä syistä johtuen, vaikka Ridge-tunnistuksen tulokset ovat tarkkoja, ne täytyy muuntaa toiseen muotoon. Kun rakoviivoja määritellään käsin piirtämällä, se tehdään määrittelemällä rakoviivaan kuuluvien janojen päätepisteet. On siis luonnollista yrittää löytää Ridge-tunnistuksen tuottamasta ratkaisusta kullekin rakoviivalle sellaiset pisteet  $\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_n$ , että nämä pisteet vastaavat määritelmässä 3.3 esiteltyjä janojen päätepisteitä, ja tämän jälkeen yhdistellä rakoviivoja toisiinsa käyttäen hyväksi näitä janoja.



Kuva 9: Vasemmalla nähdään Ridge-tunnistuksen syöte ja oikealla punaisella merkattu Ridge-tunnistuksen tulokset.

## 6.4 Rakoviivojen luominen Ridge-tunnistuksen ennusteesta

Tässä aliluvussa esitellään, miten yksi Ridge-tunnistuksen tuottaman tuloksen sisältämä rakoviiva käsitellään, jotta se voidaan määritellä janojen päätepisteiden avulla sen sijasta, että se määriteltäisiin kaikkien rakoviivaan kuuluvien pikselien joukkona. Ridge-tunnistuksen tuloksen tuottama pistejoukko on järjestetty siten, että ensimmäinen  $(x, y)$ -koordinaatti on jokin rakoviivan päätepiste ja seuraava koordinaatti on aina jokin edellisen koordinaatin ympäröivistä pisteistä. Koordinaatin  $(x, y)$  ympäröivät pisteet on tässä tapauksessa joukko

$$\{(x', y') \in \mathbb{N}^2 \mid |x - x'| \leq 1 \text{ ja } |y - y'| \leq 1\}. \quad (33)$$

Tätä järjestystä käytetään myös myöhemmin, kun puhutaan Ridge-tunnistuksen tuloksen rakoviivojen kvantiiliväleistä. Yksinkertaisimmillaan rakoviiva voidaan määritellä Ridge-tunnistuksen tuottaman tuloksen ensimmäisen ja viimeisen koordinaatin avulla. Suuressa osassa tapauksista nämä rakoviivat kaartuvat, jolloin tämä jana ei kuvasta rakoviivaa lainkaan. Yksi askel kaksiulotteisessa Ridge-tunnistuksessa on kulmien tarkastelu Hessen matriisiin avulla. Nämä kulmat ovat siis laskettu jokaiselle pikselille, ja niitä tarkastelemalla rakoviivat voidaan jakaa jananpätkiksi. Esitellään ensin *rakoviivan*

*pisteiden kvantiilivalinta* ja sitten jananpätkien muodostamisprosessi.

**Määritelmä 6.1** (Rakoviivan pisteiden kvantiilivalinta). Olkoon  $X$  Ridge-tunnistuksen tuloksen erään rakoviivan järjestetty pistejoukko,  $n$  siinä olevien pisteiden lukumäärä ja  $q \in \mathbb{N}$  valittu kvantiili,  $q \leq 3$ . Valitaan joukon  $X$  ensimmäisen kvantiilivälin viimeinen ja viimeisen kvantiilivälin ensimmäinen piste ja merkitään näitä  $(x_0, y_0)$  ja  $(x_1, y_1)$  vastaavasti. Piirretään pisteiden kautta suora ja loitonnetaan niitä toisistaan, kunnes

$$\max\{|x_0 - x_1|, |y_0 - y_1|\} = n. \quad (34)$$

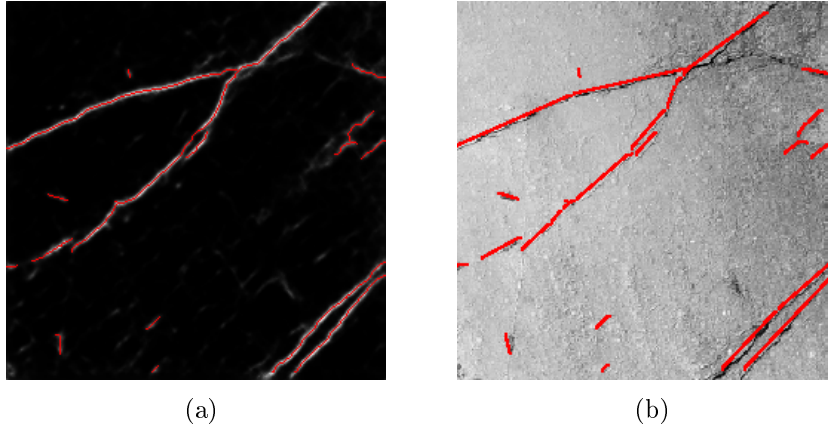
Näin saatu koordinaattipari on rakoviivan pisteiden kvantiilivalinnan tulos.

▽

Nyt, jos rakoviiva sisältää alle 20 pikseliä, tätä vastaava jana on rakoviivan pisteiden kvantiilivalinnan tuottamien pisteiden määrittelevä jana  $q$  arvolla 3. Jos rakoviiva sisältää yli 80 pikseliä, niin rakoviiva jaetaan kahtia ja jatketaan prosessia näille osittaisille rakoviivoille. Jos pikselien määrälle  $n$  taas pätee  $20 \leq n \leq 80$ , niin lasketaan pikselien kulmien keskihajonta  $\sigma$ . Jos  $\sigma < 0.2$ , niin valitaan tälle jana, jonka päätepisteet ovat rakoviivan pisteiden kvantiilivalinnan tulokset  $q$  arvolla 3. Muulloin katkaistaan rakoviiva keskeltä ja aloitetaan alusta näille osittaisille rakoviivoille. Tämän askeleen vaikutus voidaan nähdä kuvassa 10.

## 6.5 Rakoviivojen yhdistäminen

Kun rakoviivat ovat luotu, niitä pyritään yhdistämään toisiinsa ensin kunkin alikuvan sisällä. Käytännössä yhdistämisellä tarkoitetaan, että jokin rakoviivan päätepiste yhdistetään toisen rakoviivan päätepisteeseen janalla. Yhdistämisen pääperiaatteena on tutkia kahta eri rakoviivaa kerrallaan. Näiden kahden rakoviivan suhteellinen sijainti ja kulma määrittelevät kuuluuko rakoviivat yhdistää vai ei. Yhdistämiselle ei ole olemassa mitään selkeitä rajoja, kuinka suuri kulma kahden rakoviivan välillä on, tai kuinka etäällä toisistaan rakoviivat ovat, jotta ne voidaan tulkita samaksi raoksi. Rakoviivojen yhdistäminen käsin on parhaimmillaankin erittäin subjektiivista ja yksi työn menetelmän eduista on sen tuoma objektiivisuus. Näin ollen yhdistämismetodi,



Kuva 10: Ridge-tunnistuksen tulokset (vasemmalla) on muutettu rakoviivoiksi (oikealla).

joka esitellään myöhemmin tässä aliluvussa, seuraa vain joitakin yleisimpiä kriteereitä, joiden täytyy toteutua, jotta kaksi rakoviivaa voidaan yhdistää ja näin siis kuvastavat samaa rakoja. Yhdistämismetodin peruseriaatteen on siis vertailla kahta rakoviivaa keskenään. Yksityiskohtaisemmin metodi vertailee kahden rakoviivan ensimmäisiä ja viimeisiä janoja keskenään, paitsi jos rakoviiva muodostuu vain yhdestä janasta, jolloin vertailu tapahtuu vain siihen. Menetelmässä ei vertailla kaikkiin janoihin, sillä rako saattaa kaartua, jolloin raon sen hetkinen suunta saattaa näkyä vain rakoviivan päädyimmäsissä janoissa.

### 6.5.1 Yksityiskohtaisemmin

Menetelmä käy läpi kaikki rakoverkoston rakoviivat läpi ja vertailee niiden päädyissä olevia janoja jokaisen muun rakoviivan päädyissä olevien janojen kanssa. Jos nämä janat täyttävät tietyt kriteerit, niiden välinen kulma ja etäisyys parametrisoidaan yhdeksi arvoksi  $p > 0$ . Esitellään seuraavaksi *yhdistysparametrisaatio* ja tämän jälkeen yhdistämiseen vaaditut kriteerit.

**Määritelmä 6.2** (Yhdistysparametrisaatio). Olkoon  $AB$  ja  $CD$  janoja  $\mathbb{R}^2$ :ssa ja  $y_1, y_2$  niitä vastaavat suorat, eli  $A, B \in y_1$  ja  $C, D \in y_2$ . Jos suorat leikkaavat, valitaan leikkauskulmista itseisarvoltaan pienin, ja merkitään tätä

arvolla  $\alpha$ . Jos suorat eivät leikkaa, niin  $\alpha = 0$ . Lisäksi valitaan sen janan pituus  $d$ , joka yhdistää nämä kaksi janaa toisiinsa päätepisteistä, eli

$$d = \min\{|AC|, |AD|, |BC|, |BD|\}. \quad (35)$$

Nyt Yhdistysparametrisaatio on funktio  $P : [0, \frac{\pi}{2}] \times [0, \infty) \rightarrow [0, 1]$ ,

$$P(\alpha, d) = \sqrt{\frac{(1 - f(\alpha))^2 + (1 - g(d))^2}{2}}, \quad (36)$$

jossa  $f : [0, \frac{\pi}{2}] \rightarrow [0, 1]$  on kulman muunnosfunktio,

$$f(\alpha) = \begin{cases} 1 - \frac{3}{\pi}\alpha, & \text{jos } 0 \leq \alpha \leq \frac{\pi}{6}, \\ \frac{1}{2}, & \text{jos } \frac{\pi}{6} < \alpha < \frac{\pi}{3}, \\ 1 - \min\{1, \frac{9\alpha - 3\pi}{2\pi} + \frac{1}{2}\}, & \text{muulloin.} \end{cases} \quad (37)$$

ja  $g : [0, \infty) \rightarrow [0, 1]$  on etäisyyden muunnosfunktio,

$$g(d) = 1 - \frac{1}{1 + e^{\frac{-d}{42} - 4}}. \quad (38)$$

▽

Määritelmässä valitut funktiot  $f$  ja  $g$  ovat luotu vastaamaan periaatteita, joiden mukaan rakoviivoja yhdistellään käsin. Nämä periaatteet ovat:

- Pienemmät janojen väliset kulmat yhdistetään todennäköisemmin kuin isompien. Tämä kuitenkin tapahtuu epälineaarisesti siten, että kulman ollessa hyvin lähellä nollaa, yhdistämistodennäköisyys kasvaa suuresti. Kulmien ollessa lähellä arvoa  $\frac{\pi}{2}$  yhdistäminen on hyvin epätodennäköistä. Kulmien ollessa näiden kahden ääripään välissä, kulma ei vaikuta vahvasti suuntaan tai toiseen yhdistämisen kannalta.
- Janat, jotka ovat lähempänä toisiaan yhdistetään todennäköisemmin. Todennäköisyyden tulee laskea huomattavasti tietyn etäisyyden jälkeen, sillä isommat löytämättömät välit rakoviivojen välillä tarkoittaa todennäköisemmin vain sattumaa, että raot ovat samansuuntaiset, eikä sitä, että rako on suurimmaksi osaksi piilossa.

Yksi keskeisimmistä jatkotutkimuksen aiheista olisi juuri tämän yhdistysparametrisaation optimointi. Jatkotutkimusaiheista kerrotaan lisää luvussa 8.

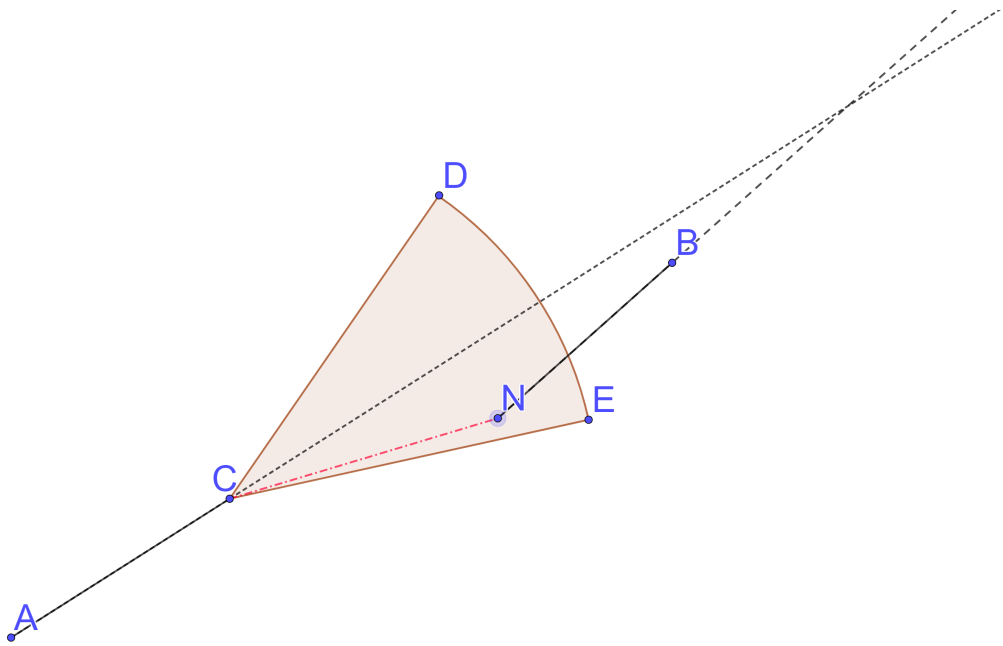
Merkitään kahta vertailtavaa janaa  $AC$  ja  $NB$  ja niitä yhdistävä jana on  $CN$ . Jotta yhdistysparametri lasketaan, kahden vertailtavan janan on toteutettava seuraavat kriteerit:

1. Rakoviivojen yhdistys täytyy myös olla rakoviiva, eli murtoviiva.
2. Janojen välinen kulma täytyy olla alle kulmakynnyksen  $\alpha_h$  tai jos janan  $CN$  pituus on alle pituuden  $d_a$ , niin janojen välinen kulma täytyy olla alle  $\alpha_a > \alpha_h$ .

Näiden ehtojen lisäksi on kaksi lisäehtoa, joista janan  $CN$  on toteutettava ainakin toinen. Ensimmäistä kriteeriä kutsutaan *pallokriteeriksi*, ja jotta se täyttyy, janan  $CN$  pituus täytyy olla alle  $d_h$ . Toista kriteeriä kutsutaan *sektorikriteeriksi*, ja se edellyttää, että janan  $CN$  pituus on enintään  $d_s$  ja ainakin yksi seuraavista ehdoista on voimassa:

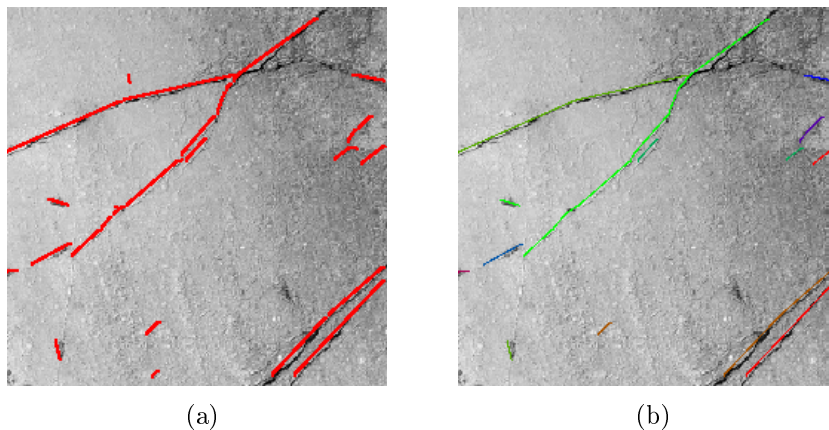
1. Janan  $CN$  kulma janan  $AC$  ja janan  $NB$  kanssa täytyy olla alle  $2\alpha_h$ .
2. Jos janojen  $AC$  ja  $NB$  välinen kulma on alle  $\frac{\alpha_h}{2}$  niin janan  $CN$  kulma janan  $AC$  ja janan  $NB$  kanssa täytyy olla alle  $3\alpha_h$  ja janan  $CN$  pituus täytyy olla alle  $\frac{d_h}{2}$ .
3. Yhdistävän janan  $CN$  kulma janan  $AC$  tai  $NB$  kanssa täytyy olla alle  $\alpha_h + \frac{\pi}{36}$  ja pisteiden  $AC$  kautta kulkevan puolisuoran täytyy leikata pisteiden  $NB$  kautta kulkevan puolisuoran. Tämä ehto on hahmotettu kuvassa 11.

Yhdistämisessä käytetään parametreja:  $\alpha_h = \frac{\pi}{18}$ ,  $d_h = 5$ ,  $d_s = 32$ ,  $\alpha_a = \frac{5\pi}{36}$ . Parametrien etäisyyksien arvot ovat annettu pikseleissä. Kun kaikki alikuvien rakoviivaverkostot on luotu, alikuvat ja rakoviivaverkostot yhdistetään takaisin yhdeksi kokonaisuudeksi. Rakoviivaverkoston yhdistäminen tapahtuu yksinkertaisesti ensin siirtämällä jokaista yhdistettävän alikuvan rakoviivaverkoston janaa yhtä paljon ja sitten ottamalla unionin kaikista yhdistettyjen alikuvien rakoviivaverkoston yli. Näiden janojen siirron suuruus ja suunta riippuu alikuvan alkuperäisestä sijainnista alkukuvassa. Koska alkukuva jaettiin  $256 \times 256$  kokosiin alikuviin, jokaisella alikuvalla on olemassa



Kuva 11: Kuvassa jana  $CN$  sijaitsee kokonaan sektorilla  $CED$ , eli janat  $AC$  ja  $NB$  ovat tarpeeksi lähellä toisiaan. Tämän lisäksi janojen katkoviivoilla merkatut puolisuorat leikkaavat toisensa, joten sektorikriteerin kolmas ehto täyttyy.

sijainti alkukuvassa: rivi- ja sarakenumero. Näin ollen, jos alikuva sijaitsee alkukuvan rivillä  $n$  ja sarakkeella  $k$  (ensimmäinen rivi/sarake on sarake numero 1), kukin siinä alikuvassa sijaitseva janan piste  $(x, y) \in \mathbb{N}^2$  saa arvon  $(x+256(k-1), y+256(n-1))$ . Tässä täytyy huomioida, että vaikka pisteen sijainti voidaan tulkita koordinaattina, se sisältää kuitenkin vain sen sijainnin matriisissa. Tämän takia vaikka alikuvan rivin sijainti nousee, eli mennään ”alaspäin” kuvassa, niin pisteen koordinaatin arvo nousee, koska myös piste sisältää vain rivin tiedot. Kun kaikki alikuvat ja niiden rakoverkostot ovat yhdistetty ja siirretty, suoritetaan yksi viimeinen rakoviivojen yhdistäminen isommalla sektoripituusparametrilla  $d_s = 64$ .



Kuva 12: Alkuperäiset löydetyt rakoviivat (vasemmalla) ovat pyritty yhdistämään tämän alikuvan sisällä. Kuvassa (oikealla) eri rakoviivat ovat väritetty erilaisilla väreillä, jotta on helpompi tarkastella, mitkä rakoviivat kuuluvat yhteen.

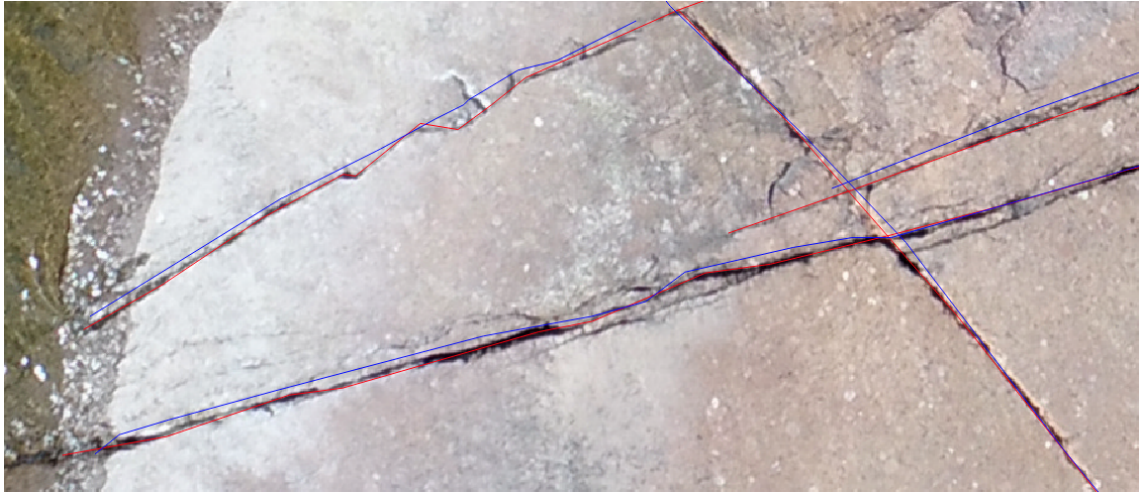
## 6.6 Rakoviivojen koordinaattien muuntaminen

Nyt jokaisen rakoviivan jokainen piste on vain sen koordinaattitiedot alkukuvasa. Nämä pisteet voidaan siis helposti muuntaa takaisin haluttuun koordinaattijärjestelmään käyttämällä määritelmää 3.1.

## 7 Tulosten tulkinta

Tulosten tarkasteluun on valittu lähdematriisit  $B^J(1)$ ,  $B^J(3)$  ja  $B^J(5)$  ja ennustemetriikaksi 3.9. Rakojen keskimääräisen paksuuden kasvattaminen tämän jälkeen ei nosta viivamatriisin ennustemetriikkaa merkittävästi. Päinvastoin keskimääräisen paksuuden kasvattaminen saattaa antaa vääränlaista tietoa. Metriikkaa voi vääristää siten, että ennusteena toimiva yhden pikselin paksuinen rakoviiva voi mennä viistoon lähteenä toimivan rakoviivan sisällä, peittäen enemmän pikseleitä ja nostaen tulosta, vaikka kyseessä on vain yksi viiva.

Malli koulutettiin kuudella aineistona saatuna kuvan avulla ja ennusteet ja tulokset ovat nähtävissä taulukossa 1. Yksi kuva jätettiin pois, koska kuvan koko ja alueen rehevyisyys erosi suuresti muusta aineistosta, eikä siis ollut



Kuva 13: Kuvassa siniset viivat kuvaavat ennusteita ja punaiset ovat käsinpiirrettyjä oikeita ratkaisuja. Voidaan havaita, että selkeästi havaittavat raot menetelmä tunnistaa paremmin kuin kuvassa 14 olevat vaikeasti erottuvat raot.

järkevää kouluttaa mallia tämän kuvan avulla saatika yrittää tulkita tämän alueen ennusteen tuloksista mitään, jos sitä ei ole käytetty kouluttamisessa.

Kuva	$p = 1$	$p = 3$	$p = 5$
A	0.123	0.300	0.370
B	0.102	0.264	0.335
C	0.120	0.289	0.364
D	0.151	0.341	0.426
E	0.074	0.199	0.271
F	0.119	0.280	0.341
G	0.189	0.436	0.516
H	0.176	0.393	0.458

Taulukko 1: Taulukossa on nähtävillä ennustemetriikat viivamatriiseille eri lähdematriisin arvioiduilla keskivertaisilla rakojen paksuuksilla  $p$ . Kuvia A ja B ei käytetty mallin koulutuksessa.



Kuva 14: Punaisella merkatut viivat ovat käsin piirretyt oikeat ratkaisut, kun taas sinisellä piirretyt ovat menetelmän tuottama ratkaisu. Kuvasta voi helposti nähdä, miksi se ei sovellu hyvin aivan kaikkien rakojen löytämiseen.

Tuloksista voi päätellä, että menetelmä ei onnistu mallintamaan rakoja täydellisesti. Menetelmän antamaa ratkaisua tarkastellessa, se onnistuu löytämään ja yhdistämään kaikkein selkeimpiä tapauksia kohtalaisesti, kun taas ihmissilmällä kuvista vaikeasti havaittavat raot jäävät suurilta osin tai täysin tunnistamatta. Tämän voi havaita paremmin kuvissa 13 ja 14. Maininnanarvoista on se, että lähdeaineistossa lukumäärältään ja pituudeltaan valtaosa raoista muodostuu juuri näistä vaikeasti havaittavista raoista. On mahdollista, että jos mallin kouluttaisi korkearesoluutioisimmille kuville, malli oppisi paremmin hahmottamaan nämä vaikeammin havaittavat viivat. Tässä työssä käytettyjen kuvien resoluutiot oltiin kompressoitu pieneksi, jotta tarvittava muistin määrä pienenee ja näin menetelmä toimisi tyypillisellä laitteistolla. Menetelmästä saanee huomattavasti käyttökelpoisemman, jos sille syötetään koulutusaineistona esimerkkiratkaisuja, joissa on valittu ainoastaan kaikkein

selkeimmät tapaukset. Menetelmän pitäisi tällöin löytää tällöin vähemmän rakoja, mutta ne raot, joita menetelmä pyrkii mallintamaan ovat tarkempia. Tällöin yhdistämiseen käytettyjä parametreja tulisi relaksoida.

Lisäksi taulukoissa 2 ja 3 on esitelty tuloksia käyttäen tilastollisia menetelmiä. Taulukossa 2 on tutkittu ennustettujen rakoviivojen ja käsin piirrettyjen rakoviivojen välisiä eroja lukumäärän ja pituuksien avulla. Tutkimalla löydettyjen rakoviivojen ja oikeiden rakoviivojen lukumäärää ja pituuksien keskimääräistä eroa, nämä tulokset tukevat taulukon 1 tuloksia näyttämällä, että rakojen havaitseminen kuvasta E onnistui erittäin kehnosti, kun taas kuva G onnistui huomattavasti paremmin.

Kuva	Oikea lkm.	Ennuste lkm.	H-arvo
A	520	1186	296.395
B	240	464	169.269
C	1218	2717	620.211
D	234	688	189.794
E	256	3102	607.513
F	708	1106	384.476
G	919	894	72.900
H	2244	5218	1203.435

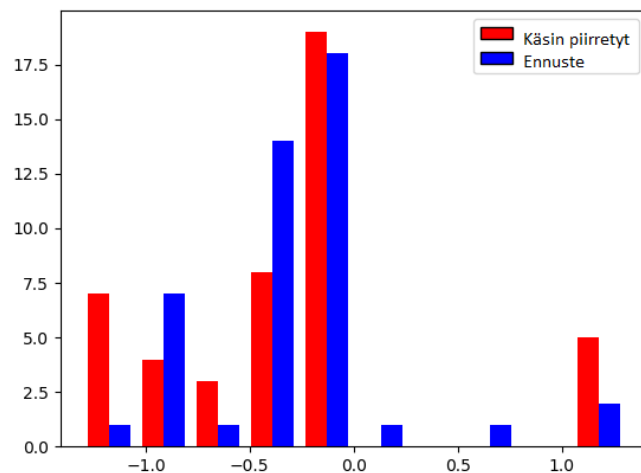
Taulukko 2: Rakojen pituuksien jakaumat eivät noudata normaalijakaumaa millään kuvalla Shapiro-Wilk -testin mukaan ( $p$ -arvo  $< 0.001$ ). Taulukossa esitetty käsinpiirrettyjen rakoviivojen lukumäärä, ennustettujen rakoviivojen lukumäärä sekä näiden rakoviivojen pituuksien välinen Kruskal-Wallis -testin H-arvo. Kruskal-Wallis testien  $p$ -arvot ovat jokaiselle kuvalle myös selvästi alle 0.001, eli ennustettujen ja käsin määritellyiden rakoviivojen pituuksien välillä on eroja.

Vielä lopuksi on hyvä tarkastella rakoviivojen kulmia eri rakoviivojen pituusluokissa. Tarkastellaan siis luvussa 3.2.2 esiteltyllä tilastollisella menetelmällä kaikkia rakoviivojen kulmia, sekä pituudeltaan kvantiiliväleihin  $[0, 0.05]$ ,  $[0.05, 0.95]$  ja  $[0.95, 1]$  kuuluvia rakoviivojen kulmia. Koska rakoviivojen

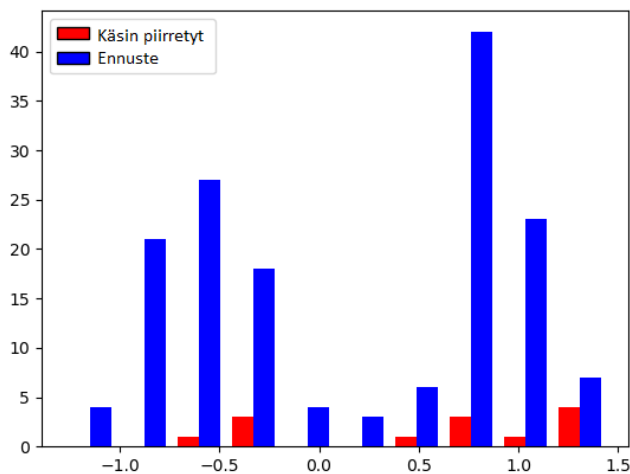
Kuva	Kaikki	0-0.05	0.05-0.95	0.95-1
A	0.000	0.000	0.000	0.119
B	0.000	0.000	0.000	0.481
C	0.000	0.226	0.000	0.003
D	0.040	0.131	0.027	0.751
E	0.000	0.054	0.000	0.022
F	0.000	0.027	0.000	0.349
G	0.007	0.656	0.005	0.787
H	0.000	0.482	0.000	0.711

Taulukko 3: Taulukossa on esitetty Kruskal-Wallis testien p-arvot oikeiden ja ennustettujen rakoviivojen kulmille eri osajoukoille. Pienet p-arvot viittaavat jakaumien erilaisuuteen. Vasemmalta oikealla osajoukot ovat: kaikki rakoviivat, pituudeltaan kvantiiliväliin  $[0, 0.05]$  kuuluvat, kvantiiliväliin  $[0.05, 0.95]$  kuuluvat ja  $[0.95, 1]$  kuuluvat.

vat, joiden kulmat ovat lähellä arvoa  $\frac{\pi}{2}$  tai  $-\frac{\pi}{2}$  ovat lähes samansuuntaisia, voidaan tehdä muunnos tilastollista tarkastelua varten siten, että jos rakoviivan kulmalle  $\alpha$  pätee  $\alpha \in (-\frac{\pi}{2}, -\frac{\pi}{2} + \frac{\pi}{18}]$ , niin tarkasteltavaan joukkoon lisätään kulman  $\alpha$  sijasta arvo  $-\alpha$ . Tämä väli valittiin, koska 10 astetta eli  $\frac{\pi}{18}$  radiaania valittiin yleiseksi yhdistämisparametrin arvoksi, jolloin siis tulkittiin rakoviivojen kulmien olevan tarpeeksi lähellä toisiaan. Näin saatujen joukkojen tilastollisten menetelmien tulokset ovat nähtävillä taulukossa 3. Taulukosta voi havaita, että suurten rakojen p-arvot kasvavat huomattavasti kuvilla, jotka aikaisemmissa taulukoissakin tulkittiin ennusteen onnistuvan hyvin. Histogrammeissa 15 ja 16 on nähtävissä taulukossa 3 käytetyt jakaumat kuville G ja E vastaavasti.



Kuva 15: Histogrammissa esiintyy käsin piirrettyjen ja ennustettujen rako-  
viivojen lukumäärät pitkille rakoviivoille. Ennusteen ja oikeiden arvojen voi  
havaita jakautuvan jokseenkin samalla tavalla kuvassa G.



Kuva 16: Histogrammista esiintyy käsin piirrettyjen ja ennustettujen rakovi-  
viivojen lukumäärät pitkille rakoviivoille. Ennuste käyttäytyy hyvin eri tavalla  
kuin käsin piirretty ratkaisu kuvassa E.

## 8 Jatkotutkimus

Menetelmää kehittäessä huomattiin joitakin parempia metodeja, joita ei pystytty toteuttamaan erinäisistä syistä. Jatkotutkimusehdotukset ovat:

1. Yhdistysparametrisaation optimointi.
  - Määritelmässä 6.2 esitetty Yhdistysparametrisaatio ottaa syötteen kahden vertailtavan janan kulman ja niiden päätepisteitä yhdistävistä janoista lyhimmän pituuden, ja palauttaa arvon väliltä  $[0, 1]$ , joka kuvastaa todennäköisyyttä, että rakoja ei tule yhdistää. Tämän parametrisaation parantaminen siten, että janojen yhdistäminen hoituisi pelkästään tämän funktion avulla nopeutaisi ja selventäisi koodia.
  - Parannusehdotuksiin kuuluu muun muassa yhdistävän janan kulman yhdistettävien janojen välillä lisääminen yhdistysparametrisaatiofunktion syötteenä ja funktiossa käytettyjen muunnosfunktioiden optimointi.
2. Menetelmän tulisi tarkastaa, että mikään rakoviiva ei leikkaa toista rakoviivaa kahdesti.
3. Menetelmää voisi nopeuttaa hyödyntämällä moniytimistä prosessointia.
4. Menetelmä voisi jakaa alkukuvan alikuviin siten, että alikuvat risteävät toistensa kanssa hieman.

## Lähteet

- [1] Mohaghegh, S.D., 2013. Reservoir modeling of shale formations. *Journal of Natural Gas Science and Engineering* 12, 22-33.
- [2] Puresh, K., Narasimhan, T., 1982. A practical method for modeling fluid and heat flow in fractured porous. *Sixth SPE Symposium on Reservoir Simulation*, 42.

- [3] Bear, J., 1993. Modeling Flow and Contaminant Transport in Fractured Rocks. *Flow and Contaminant Transport in Fractured Rock*. Woodhead Publishing Limited, 1-37.
- [4] Neuman, S.P., 2005. Trends, prospects and challenges in quantifying flow and transport through fractured rocks. *Hydrogeology Journal* 13, 124-147.
- [5] Steger, C., 1998. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2), 113–125.
- [6] Sørensen, T., 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Kongelige Danske Videnskabernes Selskab*. 5(4), 1-34.
- [7] Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation *MICCAI*. 9351, 234-241.
- [8] Yamaguchi, T., Hashimoto, S., 2010. Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications*, 21, 797-809.
- [9] Beijing, X., Ai, D., Yang, Y., 2019. Crack Detection and Evolution Law for Rock Mass under SHPB Impact Tests, *Hindawi, Shock and Vibration*. 2019.
- [10] QGIS, <https://qgis.org/en/site/>, 30.7.2020.
- [11] University of Freiburg, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/>, 1.7.2020.