



Mikko Paajanen

Koneoppiminen viemäriverkoston kuntotutkimuksen apuna

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

8.5.2023

Tiivistelmä

Tekijä: Mikko Paajanen
Otsikko: Koneoppiminen viemäriverkoston kuntotutkimuksen apuna
Sivumäärä: 43 sivua
Aika: 8.5.2023

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintätekniikka
Ammatillinen pääaine: Ohjelmistotuotanto
Ohjaaja: Lehtori Vesa Ollikainen

Insinööriyöni tavoitteena oli suunnitella ja toteuttaa koneoppiva algoritmi, jonka tehtävänä on tunnistaa kuvasyötteen perusteella viemäriputkesta mahdollisia vuotoja. Työhön sisältyi opetusaineiston kerääminen ja käsittely sekä itse algoritmin suunnittelu ja toteutus. Koneoppimismallin kyvykkyyttä mitattiin kehitystyön aikana ja lopuksi tuloksia arvioitiin.

Työ pohjautui toteuttamaani Big-Flash-hankkeen projektiin, jossa viemäriverkoston kuntotutkimuksen automatisointia kehitettiin laajemmin. Projektin tilaajana oli Underground City Oy.

Insinööriyössä kuvataan prosessi, joka johtaa algoritmin valmistumiseen. Prosessia pohjustetaan esittelemällä käytettyjä menetelmiä ja niiden historiaa sekä tutustumalla yleisemmin viemärien kuntotutkimukseen. Lopuksi arvioin prosessin kulkua ja tuon esiin ajatuksia jatkokehitystä varten.

Avainsanat: koneoppiminen, tekoäly, kuntotutkimus, kuvantunnistus, luokittelu, ohjelmointi

Abstract

Author: Mikko Paajanen
Title: Machine Learning Assisted Inspection of Sewer Systems
Number of Pages: 43 pages
Date: 8 May 2023

Degree: Bachelor of Engineering
Degree Programme: Information and Communications Technology
Professional Major: Software Development
Supervisor: Senior Lecturer Vesa Ollikainen

The aim of the thesis was to design and develop a machine learning algorithm that can recognize potential leaks in sewer pipes from presented images. The work consisted of gathering and processing the training data and building the algorithm. The abilities of the machine learning model were tested during the development and evaluated in the end.

The impetus for this work was established within a sub-project that was made as a part of Big-Flash project. In that work the goal was to develop automatic analysis of sewer inspection on a larger scale. Project was commissioned by Underground City.

The thesis describes a process of making the algorithm. The methods employed and their historical context are presented as well as a general overview of sewer system inspection. Finally, the process is evaluated, and insights of further development are discussed.

Keywords: Machine Learning, Artificial Intelligence, condition survey, image classification, image labelling, programming

Sisällys

1	Johdanto	1
2	Katsaus koneoppimiseen ja konenäköön	2
2.1	Koneoppimisen tutkimuksen historiaa	3
2.2	Konenäköön liittyvät sovellukset	8
2.3	Koneoppimisen tulevaisuudennäkymät	10
3	Viemärien kuntotutkimus	11
3.1	Kuntotutkimuksen menetelmät	12
3.2	Viemäriputkien vikojen luokittelu	13
3.3	Kuntotutkimustulosten analyysin automatisointi	15
4	Toteutettu Big-Flash-hankkeen projekti	16
4.1	Big-Flash-hanke	16
4.2	Projektin lähtökohdat (Underground City Oy)	17
4.3	Big-Flash-projektin tavoitteet	17
5	Koneoppiminen viemäriverkoston kuntotutkimuksen apuna	18
5.1	Työympäristön ja projektin suunnittelu	18
5.2	Projektin toteutus	21
5.2.1	Aineiston käsittely ja luokittelun tekeminen	23
5.2.2	Mallin kehitys	26
5.2.3	Mallin opetus	30
5.3	Koneoppimismallin tuottamien tulosten arviointi	34
5.4	Raportointiohjelman kehittäminen	37
6	Projektin arviointi	37
6.1	Toteutuksen peilaus suunnitelmaan	38
6.2	Hyödyllisyys	39
6.3	Jatkokehitys	40
7	Yhteenveto	41
	Lähteet	42

1 Johdanto

Viemärien kuntokartoitusta tehdään koko ajan maailmanlaajuisesti. Se onkin tärkeää, sillä viemärijärjestelmillä on valtava merkitys yhteiskunnalle. Yksin Suomessa on noin 50 000 kilometriä jätevesiviemäreitä. Infrastruktuuri on pääosin vanhaa, joten se vaatii nyt ja tulevaisuudessa paljon huoltoa. Kuten monella muullakin alalla, ennakoiva huolto olisi huomattavasti taloudellisempaa kuin eskaloituneen tilanteen korjaaminen. Yksi kuntokartoituksen suurista haasteista on sen vähäinen automatisointi. Analysoitavaa materiaalia on paljon ja sitä on kohtuullisen helppo kerätä lisää, mutta toimenpidepäätösten teko edellyttää asiantuntemusta ja manuaalista tarkistusta.

Tässä insinööriyössä perehdytään koneoppimiseen ja tarkemmin sen sovelluskohteena kuvien luokitteluun syvän neuroverkon avulla. Työ käsittelee aihetta viemäriverkoston kuntotutkimuksen näkökulmasta.

Insinööriyön pohjana on toteuttamani Metropolia Ammattikorkeakoulun ja Vantaan kaupungin Big-Flash-hankkeen projekti, jonka tilaajana oli Underground City Oy. Yritys tekee viemärien kuntotutkimusta. Projektin aiheena oli viemäriputkien vikojen luokittelu ja ongelmien vakavuusasteen sekä korjaustoimien kii-reellisyyden määrittely konenäköä apuna käyttäen.

Insinööriyössäni tarkastellaan projektia erityisesti vuotovikoihin erikoistuneen konenäköalgoritmin kehityksen ja saatujen tulosten kannalta. Työn tuloksena valmistui binääriluokitteleva algoritmi, jota voidaan käyttää videotiedostojen automaattiseen analysointiin. Päätavoitteen lisäksi kehitin web-sovelluksen, jolla voidaan liittää kuviin luokittelutietoa. Insinööriyön viidennessä luvussa olen lisäksi kuvannut tapahtuneen prosessin vaiheet.

2 Katsaus koneoppimiseen ja konenäköön

Ennen kuin käsittelen insinööriyön aiheena olevaa projektia, on syytä tutustua hieman tarkemmin aihepiiriin taustoihin ja siihen, mikä projektin on omalta osaltaan mahdollistanut. Tässä luvussa pyyrhdämme pienen matkan tekoälyn kehittämisen vaiheiden muodostaman polun varrella ja poimimme erityisesti koneoppimiseen ja -näköön liittyviä seikkoja lähempään tarkasteluun. Tekoälyn tutkimuskenttä on huimaavan laaja, joten valitettavasti ainoastaan tarkkaan rajattu katsaus on tässä yhteydessä mahdollinen.

Tieteenalan termien käytöstä on vuosien varrella käyty paljon keskustelua, vailla lopullista totuutta. Itse eriytän koneoppimisen tekoälystä siten, että koen tekoälyn olevan (vielä saavuttamaton) kokonaisuus, jossa mekaaninen kone pystyy autonomisesti kehittymään yleisellä tasolla, eikä ainoastaan tehtävässä, johon ihminen on sen mahdollisesti alun perin kehittänyt. Koneoppiminen puolestaan on mielestäni puhdasta laskentatehon hyödyntämistä, ilman tavoitetta "älyn" löytymisestä järjestelmän sisuksista. En kuitenkaan itse pidä termien sekakäyttöä kovinkaan vaarallisena, koska perimmäisenä ajatuksena on puhua menetelmistä, joissa tietokone alkaa vähitellen käsittelemään sille syötettyä dataa siten, että lopputuloksena on uutta tietoa lähdeaineistoon liittyen. Kansantajuisesti kone siis oppii ja älyää, joskin hyvin tarkasti rajatussa kehyksessä.

Oma kiinnostukseni koneoppimiseen sai alkunsa jo ennen insinööriopintojen aloittamista Metropoliasissa. Tiedän, että tutkimusta on tehty jo pitkään, mutta aiheetta noin vuosikymmenen seuranneena voin olla vain hämmästynyt, miten voimakkaassa kiihdytyksessä tällä hetkellä ollaan. Kuten kuitenkin aina kun syntyy tuoreita ideoita ja uusia asioita keksitään ja kehitetään, oli se sitten pyörä, sähkö tai koko kansan hullaannuttava uusi somesovellus, niin kehityksen aikana on mahdotonta tietää missä kohtaa ollaan menossa. Tom Scott [2023] käsittelee aiheetta videossaan: "I tried using AI. It scared me." ja kertoo, että kehityskaarta voi kuvata yksinkertaisella sigmoidikäyrällä, jossa alkukehitys on hidasta, sitten päästään kiivaasti kiihtyvään vaiheeseen ja lopulta kehitys hidastuu, kun kaikki on saatu tehtyä ja huippu on saavutettu. Todennäköisesti

useissa tapauksissa tätä varsinaista päätepistettä ei koskaan saavuteta, tai edes saada tietoon, missä se sijaitsee. Tekoälyn ja koneoppimisen kohdalla tuskin tarvitsee vielä miettiä edes, mikä päätepiste olisi (tietoinen kone kenties?), sillä jaan Scottin näkemyksen, että uskoakseni olemme vielä autuaan tietämättömiä, missä kohdassa sigmoidikäyrää olemme menossa.

2.1 Koneoppimisen tutkimuksen historiaa

Kuvasin luvun johdannossa keksintöjen kehityksen noudattelevan kaavaa hidas alku, kiihtyvät läpimurrot, tasoittuminen ja hiipuminen/täydellisyys. Tämä on tietenkin vain suuri yksinkertaistus, kun todellisuudessa ihmisen elinkaaren aikainen tarkastelu osoittaa pikemminkin kehityksen tapahtuvan sykäyksittäin. Tältä samalta ajanjaksolta voidaan huomata yhtäläisyys esimerkiksi 3D-elokuvien ja tekoälyn kanssa. Molemmille on ollut ominaista pulpahtaa pinnalle aika ajoin ja välillä hukkua syvemmälle pohjamutiin kiinnostuksen laantuessa. Vaikka tekoälyn potentiaali maailman muokkaamisessa on ymmärrettävästi täysin erilainen suhteessa 3D-elokuvaan, niin mikään ei estä kehityksen mahdollista hiipumista, vaikka juuri tällä hetkellä kehitystahti ja -into tuntuvat olevan huipussaan.

Tällä kertaa jätän kylmästi vähäiselle huomiolle noin 70–80 vuotta tekoälyn historiaa, joka alkaa suurin piirtein 1930–40-lukujen taitteessa Alan Turingin toimiessa pioneerina. Reilu vuosikymmen myöhemmin uraa uursi John McCarthy, joka nimetään myös termin ”tekoäly” (”Artificial intelligence”) isäksi [Håkansson & Hartung 2020: 19]. Nostettakoon kuitenkin esille muun muassa kuvantunnistuksen kannalta tärkeinä kehitysaskelina perseptronit (60-luku) ja vastavirta-algoritmi (80-luku). Nykyisen tekoälybuumin moottorina on luonnollisesti tietotekniikan kehityksessä tapahtunut laskentatehon kasvu, joka on mahdollistanut syväoppimiseen perustuvien järjestelmien kehittymisen [Kolari & Kallio 2023: 21].

Aloitetaan aivoista, sillä päidemme sisällä on käynnissä yksi ihmeellisimmistä tuntemistamme tietojärjestelmistä. Se on myös alun perin ollut innoittamassa tekoälytutkijoita etsimään ratkaisuja vastaavanlaisen mekaanisen menetelmän

toteuttamiseksi. Nykyisen ymmärryksemme mukaan aivojen toiminnan voi tiivistää neuroneihin ja synapseihin. Neuronit prosessoivat aivojen vastaanottamaa tietoa ja muodostavat valtavan tiedonkäsittely-yksiköiden verkoston. Kunkin neuronin tehtävänä on tarkkailla tiettyä asiaa ja sen havaittuaan lähettää synapsinsa kautta taajuussignaali toisille neuroneille. [Alpaydin 2021: 108.]

Perseptroni on 60-luvulla kehitetty aivojen toimintaa yksinkertaistetusti simuloiva malli [Rosenblatt 1962: vi]. Perseptronin ajatus on luoda keinotekoisien neuronien ja synapsien verkko. Verkko kehittyi monikerroksiseksi ja sen ensimmäisen kerroksen neuronit toimivat aistimina. Kun näille aistinneuroneille syötettiin arvoja, ne välittivät ne säädettyjen painotusten mukaisesti verkossa eteenpäin. Arvojen virrattua läpi verkon, ne lopulta päätyivät tulokseen, jonka perusteella voitiin toteuttaa tai olla toteuttamatta verkon käyttötarkoitukseen liitetty asia, esimerkiksi koneistossa tapahtuva liike. [Alpaydin 2021: 110.]

Perseptronin ongelmaksi muodostui kuitenkin erilaisten vaihtoehtojen käsittely. Lineaarisuutensa vuoksi se oli kyvytön XOR-ongelman edessä. Mietitään tilannetta, jossa perseptronissa tapahtuisi kaksi samanaikaista aktivointia, joista molemmat olisivat valideja yksinään, mutta jommankumman ollessa tosi, pitäisi toisen olla epätosi. Tässä tapauksessa, jotta kumpikaan aktivoinneista voisi toteutua, niiden on molempien ylitettävä aktivoinnin kynnyksen. Ongelmaa voitiin paikata monikerroksisen perseptronin rakenteilla, mutta sellaisen opettamiseen ei ollut keksitty keinoa. Kesti 80-luvulle ennen kuin vastavirta-algoritmin kehitys ratkaisi ongelman, ja monikerroksinen perseptroni voitiin opettaa selviytymään tehtävästä. [Alpaydin 2021: 112–113.]

Vastavirta-algoritmin avulla neuroverkkoa voidaan opettaa sen virhettä vähentäen. Algoritmin avulla neuronien vaikutus koko verkon virheeseen on mahdollista arvioida neuroni kerrallaan ja näiden arvioiden mukaan voidaan säätää neuronien kytkentöihin liittyviä painoja siten, että virhe vähenee. [Kelleher 2020: 186.]

Jo alkujaan tutkijoiden haaveissa kimmelsi tekoäly, joka pystyisi suoriutumaan tehtävistä yleisellä tasolla. Tähän pyrittiin kehittämällä järjestelmiä, jotka tekisivät ihmiselle ominaisia älyllisiä tehtäviä. Mutta vaikka neuroverkkoa osattiin nyt jo käytännössä opettaa, ei sen tuottamissa tuloksissa päästy riittävälle tasolle, kun sovelluksia laajennettiin. Näin ollen tutkijayhteisöjen kiinnostus hiipui jälleen tultaessa kohti 90-lukua. [Kolari & Kallio 2023: 21.]

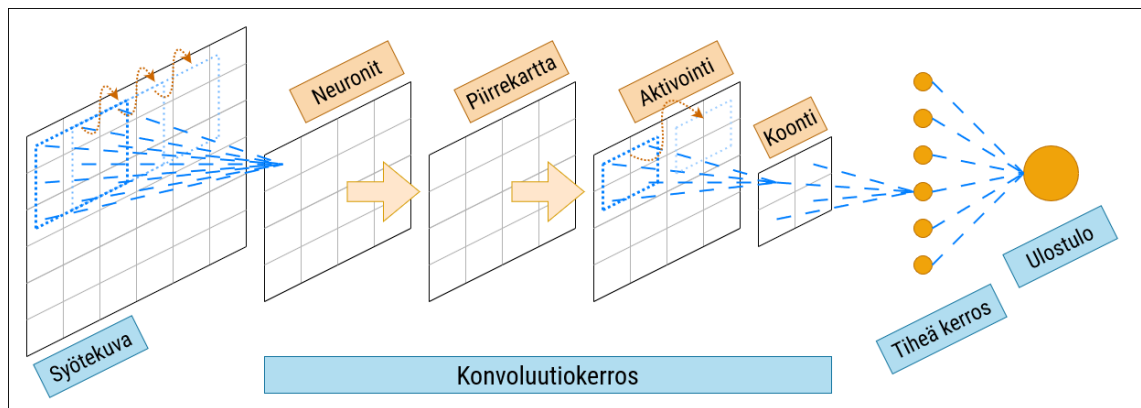
Pitkän historian jälkeen, alalla vallitsevan toiveikkuuden ja toivottomuuden aaltoillen, saavumme lähelle tätä päivää ja tekoälyn kehitysvaiheista kiihkeimpään, syväoppimisen syntyyn. Ihmisen kyky omaksua tietoa on rajallinen. Kun otetaan valtava määrä tietoa, niin ongelmana ei ole ainoastaan tiedon sisäistäminen sellaisenaan, vaan myös sen sisältämien yhteyksien ymmärtäminen. Näitä yhteyksiä päästään tutkimaan syväoppimisen avulla. Syväoppimisen ytimessä on monimutkaisen ja laajan aineiston syöttäminen suuren laskentatehon järjestelmiin. [Kelleher 2020: 11.]

Syväoppimisen aikakaudelle päästiin vuonna 2005. Ensimmäisiä läpimurtoja tekivät Geoffrey Hinton ja hänen kollegansa. Yhtenä ajatuksena oli toteuttaa kolmikerroksinen verkko, jossa syötekerroksen arvot pyritään toistamaan ulostulokerroksessa, vaikka välissä olevassa piilokerroksessa arvot kokevat muutoksia. Tällaisella autoenkoodaajaksi kutsutulla rakenteella neuroverkkoja voitiin alkaa esiopettamaan kerroksittain. Seuraaviin kehitysaskeleihin kuuluivat neuronikytkentöjen painojen parannetut alustusmenetelmät ja uusien aktivaatiofunktioiden käyttö. [Kelleher 2020: 130–138.]

Toistaiseksi kovin harppaus syväoppimisen kehittämisessä koneoppimisen lipulaivana oli 2010-luvun taitteessa yleistynyt grafiikkasuorittimien käyttö. Niiden poikkeuksellinen kyky matriisilaskujen suorittajana oli omiaan kiihdyttämään syvien neuroverkkojen opetusta kokonaan uudelle tasolle. Varsinkin NVIDIA:n CUDA-alusta helpotti grafiikkasuorittimien käyttöä. Kasvaneelle laskentateholle oli myös yhä enemmän tarvetta, koska saatavilla oli yhä enemmän ja enemmän tallennettua tietoa prosessoitavaksi. Näin syntyi syväoppimisen hyvä

kolminaisuus, jonka elementit olivat kiivaasti kehittyneet algoritmit sekä suuri määrä tietoa ja laskentatehoa. [Kelleher 2020: 138–140.]

Kuvantunnistuksen ja konenäön tärkeimpiin kehitysvaiheisiin lukeutuu konvoluutioneuroverkkojen (Kuva 1) kehitys. Tämän tutkimuksen alkuvaiheet sijoittuvat jo 80-luvulle. Konvoluutioneuroverkot, kuten nimikin kuvaa, ovat neuroverkkojen yksi syväoppiva arkkitehtuuri. Konvoluutioverkon perusajatuksena on siihen syötetyn kuvan visuaalisten piirteiden omaksuminen vaiheittain siten, että varhaisemmissa kerroksissa verkko oppii yksityiskohtia ja etenee sen jälkeen yleisempiin piirteisiin. [Kelleher 2020: 145.]



Kuva 1. Yhden kerroksen konvoluutioverkko

Konvoluutioneuroverkon rakenne koostuu syöte-, konvoluutio- ja ulostulokerroksesta. Syötekerros kuvantunnistusta ajatellen on itse kuva, eli sen pikselien numeeriset arvot. Konvoluutiokerroksella on monta tehtävää, ja se on itsessäänkin monikerroksinen. Sen neuronien kerros on kytketty syötteeseen tarkoituksenaan etsiä paikallisia ominaisuuksia. Kukin neuron peittää samankokoisen alueen syötteessä ja laskenta (painojen muutos) etenee järjestyksessä tulosten summautuessa paikallisiin naapureihin. Vierekkäisten neuronien peittämät alueet menevät limittäin säädettävän päällekkäisyyden verran. Kun kaikkien neuronien kytkentöjen painot on laskettu, saadaan tuloksena piirrekartta, johon seuraavaksi sovelletaan aktivaatiofunktioita. Tämän jälkeen suoritetaan koonti, jonka voidaan käytännössä ajatella piirrekartan skaalauksena. Yleisin toteutus tapa on järjestelmällisesti valita piirrekartasta alue, jonka isoin arvo otetaan

ulostuloksi. Konvoluutiokerroksesta voidaan edetä uuteen konvoluutiokerrokseen tai lopulta päätyä tiheään kerroksen kautta ulostulokerrokseen. [Kelleher 2020: 148–152.]

Edellä kuvatun rakenteen keskeinen osa on piirrekartta. Piirrekarttoja on käytännössä useita, sillä niitä muodostuu sen mukaisesti, kuinka monta suodatinta (painomatriisia) konvoluutiokerroksen käyttöön on määritelty. Opetuksen myötä piirrekartat alkavat hahmottamaan syötteen ominaisuuksia, kuten reunoja, viivoja ja kulmia. Kun konvoluutiokerroksissa edetään ja koonnin avulla tarkastellaan syötettä laajemmin, alkavat piirrekartatkin havaitsemaan monimutkaisempia piirteitä esimerkiksi muotoja ja tekstuureita tai jopa kuvassa esiintyviä esineitä. [Baeldung 2023.]

Konvoluutioverkon toimintatapa on merkittävä myös siitä syystä, että piirteiden sijainnilla syötteessä ei ole merkitystä. Koko kuva käydään järjestelmällisesti läpi ja opitut piirteet havainnoidaan piirrekarttoihin. Sitä myötä arvot siirtyvät lopulta ulostulokerroksen käsiteltäväksi. Ulostulokerroksena käytetty ns. tiheä kerros on perinteinen täysin kytketty standardikerros, jonka neuronit oppivat yhdistelemään eri piirrekartoilta saatua tietoa ainutlaatuisin tavoin, mikä mahdollistaa alkuperäisen syötteen tulkitsemisen. [Kelleher 2020: 152–153.]

Kerrosten ja painojen määrän tarpeen kuvailu on haastavaa, sillä vuosien saatossa erilaisilla neuroverkkoarkkitehtuureilla on saatu hyviä tuloksia. Skaalaa voi auttaa hahmottamaan esimerkit tunnetuista konvoluutioneuroverkoista kuten AlexNet (2012) ja ResNet (2015). Näistä AlexNet sisältää viisi konvoluutiokerrosta ja kolme tiheää kerrosta. Sen sijaan erilaisia kytkentämenetelmiä esitelleessä ResNet:ssä on peräti 152 kerrosta. [Kelleher 2020: 153–154.]

On syytä huomioida, että suuri määrä dataa ei vielä sellaisenaan takaa toimivia koneoppimiskäytöksiä. Tiedosta itsestään tarvitaan myös tietoa. Vaikka rakentaisimme kuinka hienon ja monimutkaisen neuroverkon, on sen opettamiseen käytettävän tiedon oltava oikeellista. Eräs tärkeä suuraineistoihin liittyvä kehityskäsi oli ImageNet-kokoelman luominen. Se on yli 14 miljoonaa manuaalisesti

luokiteltua kuvaa sisältävä aineistomammutti. Lisäksi miljoonassa sen kuvista on kuvassa itsessään luokiteltu erikseen sen sisältämät objektit. Jos kuvia luokittelee yhden sekunnissa, niin tehtävää suorittaessa ympäröivä vuorokausia täytyisi 162 kappaletta! ImageNet on ollut tiedeyhteisölle tärkeä elementti ja edistänyt kuvien luokitteluun tarkoitettujen algoritmien kehitystä, koska suurten aineistojen luokittelu on kustannuksiltaan kallista. [Kelleher 2020: 205–207.]

2.2 Konenäköön liittyvät sovellukset

Edellisessä alaluvussa pääsimme tekoälytutkimuksen historiaa tarkastelemalla tutustumaan konvoluutioverkkoihin ja kuvantunnistukseen ja -luokitteluongelman ratkaisumenetelmiin. Seuraavaksi käymme läpi joitakin sovelluskohteita, joihin edellä mainitut menetelmät ovat päteviä.

Viittasin jo aiemmin konenäköön, jolla tarkoitetaan koneen kykyä tehdä havain- toja ja antaa niille tulkintoja. Konenäköä hyödynnetään muun muassa robotii- kassa ja lääketieteessä. Yleisestikin viime vuosina esille ovat nousseet esimer- kiksi autonomiset autot, joissa lukuisten muiden sensorien joukossa kuvantun- nistuksella on oma roolinsa. Siinä missä aikaisemmin ajatus itsenäisistä au- toista nostatti ensimmäisenä mieleen turvallisuushuolia, on nykyään päästy ti- lanteeseen, jossa automatisoitu liikenne nähdään nimenomaan luomassa tur- vallisempaa ympäristöä. Toistaiseksi teknologia ei vielä mahdollista riittävän luotettavaa itseohjautumista, mutta näyttäisi siltä, että sillä tiellä kuitenkin ol- laan. [Kolari & Kallio 2023: 82–87.]

Lääketieteessä konenäköalgoritmit toimivat erinomaisena apuna muun muassa erilaisten kuvausten diagnosointiin. Vuosien varrella on kerätty mittava määrä lääkärien hyvin analysoimaa materiaalia, joita nyt voidaan käyttää opetusaineis- toina neuroverkkojen koulutuksessa. Kun nämä numeronmurskaimet sitten käynnistetään, niin jää ihmiselle enää tarkkailijan rooli, kun kone ylittää ammatti- laiset sekä nopeudessa että tarkkuudessa. [Kolari & Kallio 2023: 94.]

Teollisuudessa konenäkösovelluksia on otettu käyttöön esimerkiksi tekemään laadunvalvontaa tuotteiden valmistuksessa. Syväoppivan neuroverkon kyvykyys voittaa erottelukyvyllään perinteisen kuvien keskinäisen vertailun ja osaa sekä ottaa relevantit asiat huomioon että olla välittämättä seikoista, joilla ei ole merkitystä laatuun. [Kolari & Kallio 2023: 102–103.]

Eryteisesti omaan työhöni liittyen oli mielenkiintoista tutustua tanskalaiseen vuonna 2021 julkaistuun Joakim Bruslund Haurumin ja Thomas B. Moeslundin Sewer-ML-projektiin, jossa konenäköä käytettiin nimenomaan viemärien kunto-tutkimuksessa. Projektissa käytiin läpi erilaisia sekä yleisiä kuvantunnistusmenetelmiä että viemärien vikoihin erikoistuneita malleja. Eri mallien testauksessa käytettiin yhtenäisiä toimenpiteitä.

Haurumin ja Moeslundin [2021: 1] aineisto koostui yli 1,3 miljoonasta kuvasta, jotka olivat kuvankaappauksia videoista, joita tanskalaiset vesihuolto-yhtiöt olivat ottaneet vuosina 2011–2019. Aineiston kuvat oli jaettu 18 eri luokkaan, joista 17 oli varsinaisia vikaluokkia. Luokilla oli omat painoarvonsa niiden vaikuttavuuden mukaan. Vertailuun valittujen mallien alkuperäiset aineistot olivat määriltään huomattavasti pienempiä. Sewer-ML-projektin aineistossa oli panostettu myös luokkien esiintymien tasaisuuteen. Noin puolet aineistosta oli niin sanotusti virheetöntä ja lopuissa esiintyi keskimäärin 1,5 erilaista vikatapausta.

Projektissa kehitettiin oma pisteytys tulosten vertailemiseksi. Yleisten kuvantunnistusmallien kesken vertailu oli kohtuullisen tasaista, mutta erikoituneista malleista nousi yksi selkeästi muiden yläpuolelle. Tutkimusryhmä huomasi mallien alkavan ylioppimaan ja epäili syyksi lähdemateriaalin samankaltaisuutta. Parhaiten pisteytetyn mallin ansioksi esitettiin, että se pystyi hyvin erottamaan mahdollisen vian, vaikkei välttämättä pystynyt varsinaisen vian tarkkaan luokitteluun. [Haurum & Moeslund 2021: 6–7.]

Tutkimusryhmä yhdisti parhaimmat pisteet saaneen mallin muiden mallien kanssa kaksivaiheiseksi järjestelmäksi ja havaitsi, että voittajamalli yhdistettynä

yleiseen kuvantunnistukseen tarkoitetun TResNet-L-mallin kanssa tuotti parhaimmat pisteet. [Haurum & Moeslund 2021: 7–8.]

Valitettavasti kokonaisuudessaan tuloksissa oli vielä paljon parantamisen varaa. Heikkoja tuloksia perusteltiin muun muassa luokkien ja esiintymien samankaltaisuudella ja tiettyjen vikojen visuaalisuuden monipuolisella vaihtelulla. Ryhmän ehdotus menetelmän parantamiseen liittyi ensisijaisesti kriittisimpien vikahavaintojen luokittelutarkkuuden nostamiseen. [Haurum & Moeslund 2021: 8.]

2.3 Koneoppimisen tulevaisuudennäkymät

Koneoppimisen tulevaisuutta tehdään koko ajan. Varsinkin generatiivisten tekoälyalgoritmien kehittyminen on ollut niin nopeaa, että useinkaan ei tarvitse puhua viime vuosien kehityksestä, vaan alkaa tuntua jo siltä, että uusia yllättäviä tuloksia tulee viikoittain. Esimerkiksi tätä insinööriyttä viimeistellessäni Károly Zsolnai-Fehér [2023] esitteli videossaan NVIDIA:n kehittämää tekoälyä, joka kykenee luomaan jopa viisi minuuttia pitkiä videoita teksti- tai kuvasyötteestä. Tämä vain vähän aikaa sen jälkeen, kun maailmalle on esitelty kuvia generoivia algoritmeja kuten OpenAI:n Dall·E 2 heinäkuussa 2022.

Tieteellisissä yhteisöissä syväoppimisen menetelmille on tilausta laajasti erilaisissa tutkimusaloilla. Tästäkin seuraa positiivista kehitystä, kun vastaavasti tietojenkäsittelytieteen puolella huomataan eri alojen läpi ulottuva tarve yhä kyvykkäimmille algoritmeille ja menetelmille niiden luomiseksi. [Kelleher 2020: 205.]

Suuraineistojen potentiaali on valtava ja käytettävissä olevan tiedon määrän jatkuva kasvu asettaa luokitteluongelman, sillä kustannusten mittakaava on myös ennenkokematon. Nähtäväksi jää esimerkiksi, onko ohjaamattoman opetuksen menetelmille mahdollisuuksia nykyistä laajemmin. Niillä saattaisi pystyä luomaan synteettisiä aineistoja ohjatun oppimisen avuksi. Toinen luokitteluongelman helpottaja on siirto-oppiminen, jossa neuroverkon pohjakerrosten opetus on tehty etukäteen ja erikoistuva osa verkkoa koulutetaan ikään kuin jo kokeenneen koneoppimismallin päälle. [Kelleher 2020: 207–209.]

Kuvantunnistuksessa konvoluutioverkon käytön saattaa tulevaisuudessa korvata piirteiden oppimisen lisäksi niiden sijainnin havaitseva kapselineuroverkko [Kelleher 2020: 210] tai vaikkapa Transformer-pohjainen ratkaisu, jolla nähdään potentiaalia ja jonka tulokset perinteisiin malleihin verrattuna ovat olleet kilpailukykyisiä [Winastwan 2023]. Ennustaminen on vaikeata, mutta ainakin kehitystä tehdään tällä hetkellä jälleen vailla taantuman uhkaa näköpiirissä.

3 Viemärien kuntotutkimus

Olemme Suomessa onnellisessa asemassa ja vesihuolto on pääosin toimivaa. Kuten vesilaitosyhdistys asian ilmaisee:

Vesihuolto on näkymätön, mutta samalla välttämätön palvelu.
[Vesilaitosyhdistys 2023]

Viemäriverkoston tarkoituksena on johtaa jätevesi kuluttajalta jätevedenpuhdistamolle. Jätevesiviemäriverkoston laajuus suomessa on noin 50 000 kilometriä, eli sen voisi hyvinkin kiertää maapallon ympäri. Vesihuoltojärjestelmää ylläpitävät vesihuoltolaitokset ja niiden kuntaomistajat.

Viemäriverkoston laskennallinen uusiutumisaika nykyisellä saneeraustahdilla on yli 200 vuotta. Vesilaitosyhdistyksen mukaan saneeraamisen tarve olisi jopa kolminkertainen. Yhdistys painottaa, että putkistojen kuntoa ei voi tarkkailla vain niiden ikää katsomalla. [Vesilaitosyhdistys 2023.]

Helsingin Seudun Ympäristöpalvelut -kuntayhtymän alueella viemäriverkon keski-ikä on noin 36 vuotta, mutta keski-ikä vaihtelee eri alueilla ovat suuria. Putken tyyppin mukaan sen käyttöikä asettuu 50–80 vuoden välille. Vesilaitosyhdistys suosittaa säännöllisiä kuntotutkimuksia, mutta osalle vanhoja putkiosuusia ei ole välttämättä koskaan tehty tarkastusta. [Lampola 2018: 13–14.]

Vesilaitosyhdistys on listannut uusien toimintamallien kehittämisen yhtenä selvänä kehittämiskohteena. Uudet toimintamallit voisivat liittyä nimenomaan kuntotutkimusten tulosten analysointiin. [Lampola 2018: 12.]

3.1 Kuntotutkimuksen menetelmät

Viemärien kuntotutkimukseen on mahdollista käyttää useita erilaisia menetelmiä. Tietyn alueen verkoston tarkastelu voidaan aloittaa verkon yleisellä luokittelulla. Tätä voidaan toteuttaa kriittisyysluokittelun, zoom-kuvausten (Kuva 2) ja data-analyysin sekä muiden tilastollisten menetelmien avulla. Verkon sisään kulkeutuvia vuotovesiä voidaan paikantaa virtaamamittauksin. Yksityiskohtaista tietoa verkosta saadaan kerättyä kuvauslaitteistolla, joka ajetaan putkiston läpi. Tässä menetelmässä voidaan lisäksi käyttää laserskannausta, jolloin saadaan tarkempaa rakenteellista tietoa. Maan pinnalta käsin on mahdollista tehdä tutkimusta kaikuluotauksen avulla. Joillekin putkityypeille soveltuvat myös akustiset ja sähköiset kuntotutkimusmenetelmät. Näitä voidaan käyttää muun muassa vuotokohtien paikantamiseen. [Lampola 2018: 32–33.]



Kuva 2. Zoom-kuvauksen tekoa ja kuvauslaite. Kuvankaappauksia. [Underground City 2023]

Projektissani käytetty aineisto on edellä mainitun zoom-kuvaustekniikan tuottama videokuvaa. Zoom-tekniikalla tarkoitetaan kuvauksia, jossa kamera laskeaan viemärikaivoon esimerkiksi tankoon kiinnitettynä. Kuvauksen ajan kamera on paikallaan kaivossa, mutta laitteiston käyttäjä zoomaa kuvaa putken sisään. Menetelmä sopii suorien putkiosuuksien kuvaamiseen ja kerrallaan tutkittava

etäisyys on 80–100 metriin asti. Kuvaukset toteutetaan normaalisti samalle putkiosuudelle sen molemmissa päissä olevan kaivon kautta. [Lampola 2018: 54.]

Kuvauksia tehdessä laitteistoa siirretään koko ajan kaivosta seuraavaan ja siten saadaan aineistoa alueen koko viemäriverkosta. Putkiosuudesta yhdestä suunnasta kuvatun videon kesto on tyypillisesti alle minuutin.

Zoom-tekniikan etuna on nopeus, sillä aineistoa saadaan kymmenkertaisesti läpiajettavaan kuvaukseen verrattuna. Toisaalta huonona puolena on se, että joissakin tapauksissa putkiosuudesta ei saada joko hyvää tai ollenkaan kokonaiskuvaa, jos osuus esimerkiksi mutkittelee, on painaumilla tai täyttöaste on suuri. Menetelmällä ei saada juurikaan tietoa rakenteellisesta kunnosta kuin vain, jos siinä on selvästi nähtäviä ongelmia. [Lampola 2018: 54.]

3.2 Viemäriputkien vikojen luokittelu

Vikojen luokitteluun on kehitetty Suomessa ja ulkomailla erilaisia malleja. Malleissa on yleensä eritelty vika ja sen aste. Vesilaitosyhdistys listaa oppaassaan seuraavat tyypilliset viat: Juuret, rasva, halkeamat, sisä- ja ulkopuolinen korrosio, vuotavuus, liitosten vuotavuus, vuotavat tonttijohdot, virheellinen asennus, liittymien virheelliset asennukset, muodonmuutos ja muut viat. [Lampola 2018: 25.]

Tässä projektissa käytetty luokittelu (Taulukko 1) on asiakkaan toimittama ja vielä mahdollinen muutoksille. Joidenkin luokkien kohdalla voi esimerkiksi muodostua ongelmaksi saada riittävä määrä opetusaineistoa. Toisaalta myös käytettävän järjestelmän ja raportoinnin asettamat vaatimukset voivat myöhemmässä vaiheessa aiheuttaa muutoksia luokitteluun. Joka tapauksessa aineistoa luokiteltaessa viat on jaettu toiminnallisiin, rakenteellisiin ja vuotovikoihin. Sen lisäksi kuville merkitään putken materiaali ja täyttöaste.

Taulukko 1. Viemäriputken kuntoon liittyviä vikoja

Toiminnalliset viat	Vuotoviat	Rakenteelliset viat
rasva	vuoto saumassa	syöpymät
irtokertymä	vuoto halkeaman vuoksi	halkeamat
juuret	juuret	
putkirikko		
viettokaltevuus		
muodonmuutos/ kaartuminen		
vieras esine tai este		

Tilaaajan nykyisessä järjestelmässä videoita ei varsinaisesti luokitella, vaan niihin liittyy tekstikenttä, johon kirjoitetaan vapaamuotoinen kuvaus putkiosuuden tilasta ja mahdollisista ongelmista.

Vikojen kirjo (Kuva 3) on laaja. Lisäksi samatkin viat voivat poiketa ulkonäöllisesti toisistaan. Kuvien tulkinta voi olla harjaantuneellekin käsittelijälle vaikeaa ja tutkimukset ovat osoittaneet tulkinnoissa tulevan herkästi virheitä [Lampola 2018: 43–45].



Kuva 3. Erilaisia vikoja. Ylhäällä vasemmalla vuoto saumassa. Ylhäällä oikealla syöpymää. Alhaalla vasemmalla putki kaartaa. Alhaalla oikealla viettokaltevuus.

Luokittelun tarkkuuden parantamiseksi on ehdotettu muun muassa luokitusjärjestelmien yksinkertaistamista, kuvien lisäystä raportteihin ja eri tutkimustapojen tulosten yhdistelyä. [Lampola 2018: 45.]

3.3 Kuntotutkimustulosten analyysin automatisointi

Kuntotutkimuksen tulosten automaattinen tai tietokoneavusteinen analysointi on selvästi kiinnostava aihe. Vesihuoltoyhdistys listaa useita tutkimusprojekteja eri puolilta maailmaa. Myös Suomessa on ollut esimerkiksi HSY:n ÄlykäsVesi -hanke (2015–2017), jonka loppuraportissa kerrotaan: ”- - viemärikuvausten tulkinnan automatisointi konenäkösovelluksella on mahdollista.”

Vesihuoltoyhdistys pitää tutkimustulosten analysointia nykytilassaan epäluotettavana, mutta sen automatisointi voisi tuoda analyysin laatuun parannuksia. Suuren osan virheistä on tutkittu johtuvan inhimillisistä tekijöistä. Niihin vaikuttavat lisäksi tulkintamenetelmien vaihtelevuus. [Lampola 2018: 50–51.]

Tiedossa olevien tutkimusten tuloksissa on esimerkiksi toivottu parannusta visuaalisen tulkinnan prosessiin yksiselitteisemmän luokittelun ja runsaasti opetusaineistoa sisältävien kirjastojen muodossa. Tutkijoiden kokeilemiin menetelmiin on kuulunut erilaisia algoritmisen kuvantulkinnan menetelmiä, mukaan lukien neuroverkkoja. Myös tällä alalla tutkimusta on aikaisemmin hidastanut tekniset vaatimukset prosessointitehon, muistitarpeen ja tiedonsiirron osalta. [Lampola 2018: 50–51.]

Kuntotutkimuksen avuksi on kehitetty myös matemaattisia malleja. Saneeraus- ja korjaustoimenpiteitä ennakoivat putkien hajoamismallit voivat olla toimivia, kunhan ne saadaan kalibroitua riittävällä määrällä dataa. Hajoamismallien hyödyntäminen on kuitenkin haastavaa, koska viemäriverkoston todellinen kunto selviää niillä vain epäsuorasti. Näillä malleilla onkin mahdollisesti enemmän arvoa omaisuudenhallinnassa ja kustannuslaskennassa. [Lampola 2018: 51.]

4 Toteutettu Big-Flash-hankkeen projekti

Insinööriyön pohjana on toteuttamani projekti, jossa kehitettiin menetelmää viemäristä kuvatun videon pohjalta tehtävään automaattiseen vian tunnistukseen. Projekti oli osa Metropolian ja Vantaan kaupungin yhteistä Big-Flash-hanketta. Se oli jatkoa aikaisemmalle projektille, jonka toteuttajana toimin myös.

4.1 Big-Flash-hanke

Big-Flash-hankkeen tavoitteena on saada yrityksistä innostumaan uusien teknologioiden kokeiluun ja hyödyntämiseen. Hanke on kaksivuotinen ja alkoi 1.8.2021. Hankkeen kokonaisprojektimäärä on 122. Projekteissa opitaan ja sovelletaan nousevia teknologioita kuten:

- kobotiikka (yhteistyörobotit)
- autonominen robotiikka
- IoT + 5G (esineiden internet)
- koneoppiminen ja tekoäly.

Suurimmassa osassa projekteista tavoitteena on tehdä demo, eli pyritään luomaan yritykselle näyte jonkin edellä mainitun teknologian hyödyistä. Osasta projekteja syntyy suurempia jatkoprojekteja ja muutama viedään vielä isomaksi kokonaisuudeksi. Eri vaiheiden projektit on nimetty Suvituuli, Salama ja Iso Myrsky -liittein. [Big-Flash 2023.]

Pienemmät projektit ovat yksittäisten opiskelijoiden tai pienten opiskelijaryhmien tekemiä ja kelpaavat tyypillisesti innovaatioprojekteiksi. Keskisuuressa projektissa, kuten omani, tekijä voi olla loppuvaiheen opiskelija ja yhdistää projektiin insinööriyönsä tekemisen. Osassa projekteja opiskelija palkataan Metropoliaan projektityöntekijäksi. Isoimmat projektit voivat kattaa suurempaa yritys- ja oppilaitosyhteistyötä ja tarkoituksena on viedä toteutukset tuotantoasteelle.

Big-Flash-hanke on saanut REACT-EU-rahoitusta. Siihen osallistuminen on yrityksille lähtökohtaisesti maksutonta. [Big-Flash 2023.]

4.2 Projektin lähtökohdat (Underground City Oy)

Itse osallistuin Salamavaiheen projektiin, joka oli nurmijärveläisen Underground City Oy:n tilaama. Projekti oli jatkoa kesällä 2022 toteutettuun suvituuliprojektiin, jonka lopputuloksena syntyi yksinkertaiseen binääriluokitteluun kykenevä algoritmi ja sen ympärille rakennettu demosovellus.

Molempien projektien lähtökohtana oli yrityksen mielenkiinto viemäriputkien kuntoarvioinnin automatisointiin. Jos lähdemateriaalia pystyttäisiin tulkitsemaan tietokoneavusteisesti vaikkapa osittainkin, niin se voisi tuoda aikasäästöä tällä hetkellä manuaalisesti tapahtuvaan menetelmään. Yritys on tehnyt putkistojen kuntokartoitusta jo usean vuoden ajan ja sen hallintaan on kehitetty sovellus, jolla muun muassa tallennetaan viemäreistä zoom-kuvaustekniikalla otettuja videoita ja niihin liittyviä kuntoarviointeja.

Yrityksellä oli tiedossa eri kuvaustekniikoihin liittyviä vastaavia projekteja maailmalta, ja tavoitteena oli kokeilla koneoppimisen toimivuutta heidän omien menetelmiensä tukena.

4.3 Big-Flash-projektin tavoitteet

Ennen projektin tavoitteiden asettamista tarkasteltiin aikaisemmasta työstä saatuja tuloksia ja tutkittiin muita mahdollisia samankaltaisia projekteja. Oli selvää, että ollakseen käytännöllinen lopputuloksen tulisi pystyä avustamaan käyttäjänsä videoiden tarkistusprosessissa, eikä se saisi olla ylimääräisenä työkaluna hidastamassa työntekoa. Kuitenkaan projektin puitteissa ei pystyttäisi integroimaan automatiikkaa yrityksen olemassa olevaan järjestelmään. Tavoitteena oli siten luoda toimiva itsenäinen ratkaisu, jonka pohjalta yritys voi arvioida millaiseen suuntaan sitä voitaisiin jatkokehittää. Onnistuessaan kehitetyllä uudella työkalulla voitaisiin ohjata myös olemassa olevan järjestelmän kehitystä siihen suuntaan, että järjestelmien integrointi olisi mahdollista.

Alustavasti lopputuotteen suunniteltiin voivan löytää kaikkia yleisiä putkistoista löytyviä vikatilanteita, mutta muun muassa tarpeeksi laadukkaan luokittelun vaatiman ajan vuoksi otettiin ensisijaiseksi tavoitteeksi vuotovikojen tunnistaminen ja niiden raportointi. Tämän lisäksi pidettiin tärkeänä kehittää ja toteuttaa koneoppimismallia koko ajan sillä oletuksella, että se tulee olemaan osa isompaa kokonaisuutta, jossa vikoja tunnistetaan joko tiettyihin vikoihin erikoistuneilla algoritmeilla tai moniluokitteluun kykenevällä järjestelmällä.

5 Koneoppiminen viemäriverkoston kuntotutkimuksen apuna

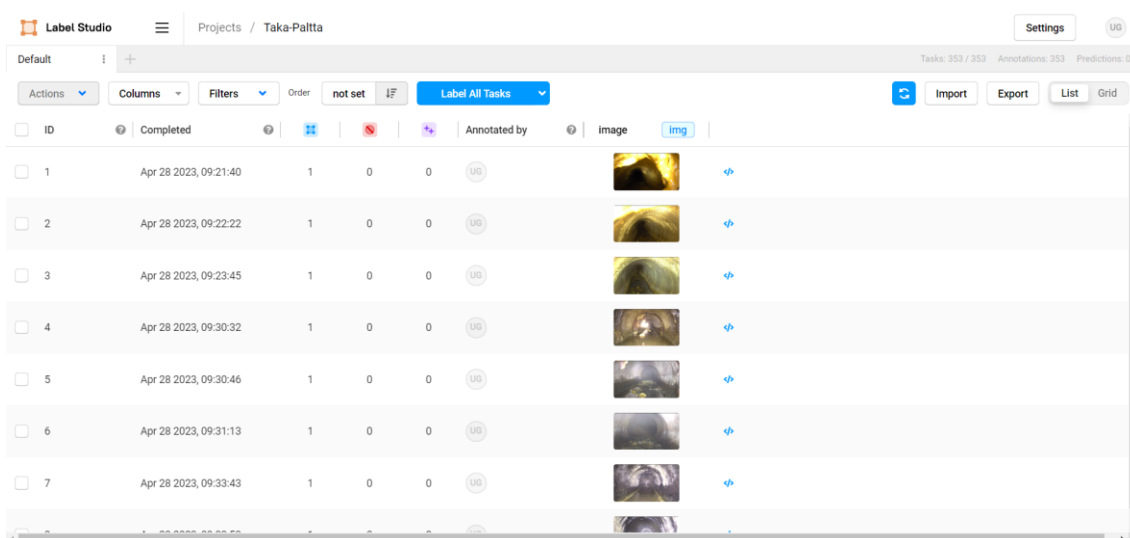
Tässä luvussa esittelen insinööriyön pohjana toteutetun projektin vaiheet ja tulovaisuuden näkymät. Projektia tehtiin neljän kuukauden ajan helmikuusta toukokuuhun 2023. Insinööriyön tiukemman aikataulun vuoksi en pysty kuvaamaan projektin vaiheita kokonaisuudessaan, mutta käyn läpi sekä tehdyt vaiheet, että asiat, jotka vielä toteutuvat lisäksi. Aikataulujen pieni yhteensopimattomuus on pyritty ottamaan huomioon projektin vaiheissa siten, että kokonaisuus on silti hyvin hahmotettavissa ja lopputuloksen toiminnallisuus on saatu selvitettyä tätä insinööriyötä ajatellen. Insinööriyön aikataulu ei myöskään ollut vaikuttamassa projektin etenemisen kannalta merkittäviin ratkaisuihin.

5.1 Työympäristön ja projektin suunnittelu

Projektin toteuttamiseksi tarvittiin erilaisia työympäristöjä. Suunnittelun tukena ja dokumentointiin käytettiin perinteisiä toimistosovelluksia. Data toimitettiin ja säilöttiin Google Drive -palvelun kautta. Opetusdatan pilkkomiseen videoista kehitettiin oma Python-skripti. Myös muita auttavia skriptejä koodattiin Pythonilla. Opetusdatan luokittelua tehtiin verkossa ja sitä varten pystytettiin ensin Label Studio -ohjelma Google Cloud virtuaalikoneelle, mutta lisäksi kehitettiin myös oma web-sovellus, jonka vakaus oli parempi ja käyttö suoraviivaisempaa.

Label Studio [2023] (Kuva 4) valikoitui nopeasti projektin alkuvaiheessa käytettäväksi ohjelmaksi luokittelun tekemiseen. Ohjelma tarjosi kaiken tarvittavan ja enemmänkin. Esimerkiksi oman luokittelunäkymän rakentaminen on helppoa ja

käyttöliittymä on suoraan tarkoitettu web-sivuksi. Ongelmia kohdattiin kuitenkin saada Label Studio toimimaan vakaasti verkossa. Tämä johtui osittain siitä, että Big-Flashissa oli rinnakkaisprojektina samaan aikaan Google Cloud [2023] -palveluiden käyttöönottoon liittyvä projekti ja sen ollessa vielä varhaisessa vaiheessa ei ollut täysin selvää, mitä Googlen palvelua olisi optimaalisinta käyttää tämä projektin kohdalla. Niinpä päädyttiin melko pitkällisesti useiden yritysten ja erehdysten kautta ylläpitämään Label Studiota http-protokollan yli virtuaalikooneella.

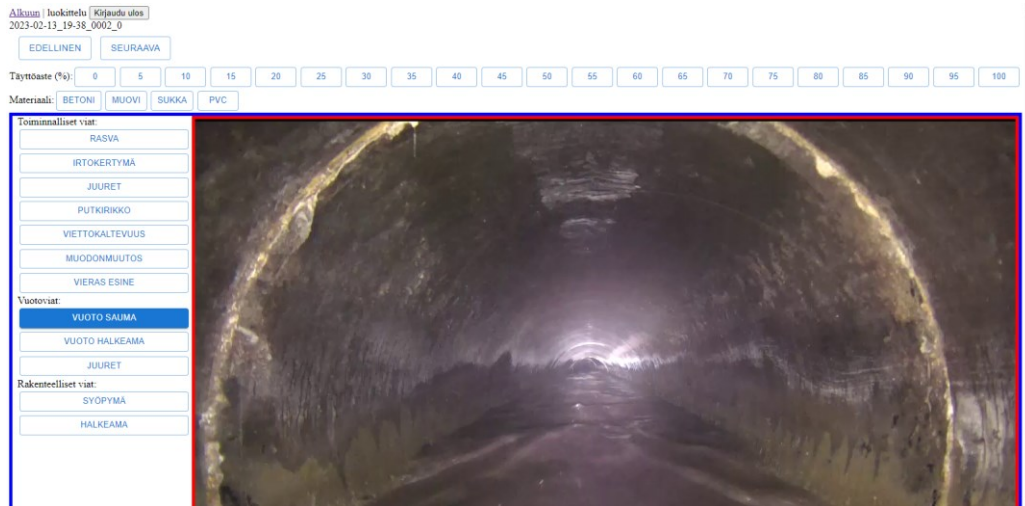


Kuva 4. Label Studio -luokitteluohjelman projektinhallinnan näkymä

Label Studion asennuksen ja konfiguroinnin sekä yllättävien suunnitelmamuu-
tosten aiheuttaessa viivästyksiä aloin vaihtoehtoisena järjestelmänä kehittä-
mään omaan MERN-pinoon pohjautuvaa web-sovellusta. Sana MERN koostuu
siinä käytettävien ohjelmistokehitysratkaisujen alkukirjaimista. Tietokantana on
MongoDB [2023], taustajärjestelmän tukiympäristönä puolestaan Express
[2023], selaimessa näkyvän osan kehityskirjastona käytetään Reactia [2023] ja
ajoympäristönä toimii Node.js [2023].

Kehittämäni sovelluksen tietokanta on äärimmäisen pelkistetty. Tietokanta pitää
sisällään vain merkkijonomuotoisen tiedon kuvan nimestä ja taulukon kuvaan
liittyvistä luokista. Käyttöliittymän yksinkertainen visuaalinen ilme tukee

luokittelutehtävää. Kunkin kuvan kohdalla on selkeästi havaittavissa, mitä luokkia siihen liittyy, ja luokkien lisääminen ja poistaminen käy nopeasti painikkeilla (Kuva 5).



Kuva 5. Kuvien luokitteluun kehitetyn sovelluksen päänäkymä

Myöskään oman sovelluksen kehitys ei onnistunut täysin vailla ongelmia. Kriittisin vaihe osui aivan loppumetreille, kun sovellus ei toiminutkaan täysin moitteetta Fly.io [2023] -verkkosisäntöpalveluun siirrettynä. Kehityksen aikaan siirtäminen oli testattu, mutta puutteellisesti, joten ikävänä yllätyksenä kehitysympäristössä mainiosti toiminut sovellus rikkoutuikin sitten tuotantoversiona.

Loppujen lopuksi sekä Label Studio että oma sovellus (vaalikevään teeman mukaisesti nimetty "Luokittelukone") saatiin käyttökuntoon suunnilleen samanaikaisesti Label Studion viedessä voiton kuitenkin siinä määrin, että ensimmäinen erä luokiteltavia kuvia ladattiin ja luokiteltiin sen avulla.

Ohjelmointia tehtiin Visual Studio Code [2023] -tekstieditorilla. Edellä kuvatun web-sovelluksen kehittämisen lisäksi koodaamiseen ja koneoppimismallin kehitykseen käytettiin Pythonia [2023].

Muutamissa kohdissa projektia oli tarvetta pienille koodinpalasille esimerkiksi ai-
neiston ja niihin liittyvien tiedostojen hallinnoimisessa. Päätin muutamassa

tällaisessa tapauksessa yrittää hyödyntää ChatGPT:n apua. Omakohtaiset kokemukset siitä olivat vielä vähäisiä, mutta teknologia on ollut siinä määrin nousussa viime aikoina, että rohkenin kokeilla sitä työkaluna muiden joukossa.

ChatGPT on tekoälyjärjestelmä, jota kehittää Microsoftin rahoittama OpenAI-tutkimuslaboratorio. Sen opetusaineisto kattaa tietoa vain vuoteen 2021 asti. Järjestelmä toimii keskustelevana koneena. Käyttäjän vuorovaikutus, esitetyt kysymykset ja annetut ohjeet johtavat vastauksiin, jotka kannattaa vielä tarkistaa faktojen sekoittuessa välillä koneen hallusinoimaan fiktion. [Kolari 2023: 43–44; OpenAI 2023.]

Omalla kohdallani ChatGPT:n tuottama koodi oli pieniä muokkauksia lukuun ottamatta hyvin toimivaa. On tärkeä huomata, että saadaksesen todellisen hyödyn ChatGPT:n ja vastaavien tekoälysovellusten avulla käyttäjän tulee tietää, mitä tahtoo ja olla kykenevä tulkitsemaan saamaansa koodia. Sanoisin, että näissä muutamassa kohdassa saavutin jonkin verran säästöä työajassa.

Luvussa 2.2.1 käsittelin tanskalaisen Sewer-ML-projektin vaiheita ja tuloksia. Oman projektin suunnitteluvaiheessa testasin projektin malleja myös itsenäisesti, koska projektin koodi ja aineisto ovat ladattavissa. Projektin kloonaaminen omaan työympäristöön onnistui kohtuullisen vaivattomasti. Mallien testaaminen jäi lopulta kuitenkin vähäiseksi, sillä projektin pääpaino siirtyi oman mallin kehittämiseen muun muassa siksi, että mallin arkkitehtuuri ja aineiston luokittelu tulisi olemaan omanlaisensa. Sewer-ML-projektiin tutustuminen antoi kuitenkin hyviä ideoita ja siinä olisi varmasti jatkotutkimuksen paikka.

5.2 Projektin toteutus

Projektin onnistumisen kannalta kolme tärkeintä vaihetta olivat täsmällisen luokittelun, tarkan koneoppimismallin ja käytännöllisen raportointityökalun tekeminen. Kaikkia voitiin suunnitella ja edistää koko projektin ajan, mutta luonnollisesti eteneminen tapahtui pääasiassa vaiheittain osuudesta toiseen. Niinpä viivästykset alkupäässä täytyi kiritä projektille suunnitellun aikataulun puitteissa.

Kaikki alkoi kuitenkin suunnittelulla ja aikataulutuksella. Hyvän lopputuloksen kannalta on tärkeää tehdä huolellinen suunnittelu. Kaikkea ei luonnollisestikaan voi ottaa huomioon, ja tiukasti aikataulutetussa projektityössä erilaiset yllätykset voivat aiheuttaa merkittäviä viivästyksiä eri vaiheiden välillä. Itse aloitin projekti-suunnitelman hahmottelun miettimällä, mitkä ovat tämän projektin päävaiheet. Vaiheita tuli seitsemän ja niiden lisäksi kahdeksantena osana läpi koko projektin kestävä dokumentointi.

Näille seitsemälle vaiheelle arvioin käytettävän minimiajan, jonka jälkeen aloin sovittelemaan niitä aikatauluun eräänlaisen vesiputousmallin mukaisesti. Riippuen vaiheesta sijoitin niitä limittäin sen mukaan, miten tiesin niiden olevan tehtävissä yhtäaikaisesti. Kun kokonaisuus hahmottui paremmin, niin kasvatin aikataulun puitteissa eri vaiheille varattuja ajanjaksoja pidemmiksi. Jokaiselle vaiheelle määrittyi päätösajankohta, joskin tietyillä vaiheilla oli mahdollisuus ja jopa tarve olla jatkettavissa myös myöhemmin projektin edetessä. Kaikkien vaiheiden alkuun lisäsin mahdollisuuden pieneen liukumaan, jos hommat etenisivät odotettua ripeämmin.

Aikataulu (Kuva 6) oli tärkeässä roolissa läpi projektin ja sen avulla tiesin varoa kriittisten vaiheiden lipsumista yli niille varatun ajan. Suhteellisen pienen käytettävissä olleen kokonaisajan vuoksi täysin ongelmitta ei kuitenkaan selvitty, vaan osa vaiheista kesti huomattavasti suunniteltua kauemmin tai siirtyi jonkin verran omalta paikaltaan häiriten muiden vaiheiden suorittamista. Kuvassa näkyvä aikataulu muuttui vielä, kun projektin päätös siirrettiin toukokuun loppuun.

	vko 4	5	6	7	8	9	10	11	12	13	14	15	16	17
	23.-29.1.	30.1.-5.2.	6.-12.2.	13.-19.2.	20.-26.2.	27.2.-5.3.	6.-12.3.	13.-19.3.	20.-26.3.	27.3.-2.4.	3.-9.4.	10.-16.4.	17.-23.4.	24.-30.4.
Tehtävät:														
Projektin käynnistys ja suunnittelu														
Työympäristön kuntoonlaitto														
Datan kanssa työskentely														
Koneoppimismallin suunnittelu														
Koneoppimismallin toteutus														
Työkalun/tulosraportin kehitys														
Tulosten arviointi														
Projektin dokumentointi														
Tarkistuspisteet:														
Tilannekatsaus														
Projektin luovutus														

Kuva 6. Projektin aikataulun visualisointi

Suurimmaksi haasteeksi osoittautui opetusaineiston luokittelu. Tämä ei sinänsä tullut yllätyksenä, sillä se on tyypillisesti suuritöinen ja tarkkuutta vaativa vaihe. Arvioin etukäteen, että luokittelua tullaan tekemään läpi koko projektin ja että alkuun pääsemiseksi riittäisi hieman vaatimattomampikin aineisto. Kuitenkin koska luokittelu oli varsinkin alussa sidottu pitkälti siihen, että saan yrityksen kautta asiantuntija-apua, niin viivästykset sekä omalta ja Big-Flash-tahon että asiakkaan puolelta, tarkoittivat pian sitä, että edes suppeampaa aineistoa ei ollut helposti saatavilla.

Ensimmäisenä valmistuneena luokitteluna oli itseni kokoama aineisto, johon merkitsin pääasiassa putken materiaalin ja täyttöasteen. Näiden kuvien pohjalta aloin suunnittelemaan ensimmäisiä koneoppimismalleja, joille annoin tehtäväksi oppia tunnistamaan putken täyttöasteen.

Erilaisten koneoppimismallien suunnitteluun ja kehitykseen ei tämän projektin puitteissa varauduttu kovinkaan suurella aikamäärällä. Tämä tuli väistämättä tarkoittamaan sitä, että jos jokin malli tai arkkitehtuuri havaittiin toimivaksi siinä määrin, että sillä saatiin tuloksia aikaiseksi, niin sen jälkeen parantelu ja laajempi vertailu erilaisten vaihtoehtojen osalta jäisi jatkokehityskohteeksi. Tässä kohtaa huomionarvoista on myös se, että jatkokehitystä oli tulossa vielä välittömästi projektin jatkuessa vielä tämän insinööriyön päättymisen jälkeen.

Lopputuotteen osalta tärkeintä oli saada selkeä raportti algoritmin löydöksistä. Raportti olisi myös keskeinen malli muiden erikoistuneiden algoritmien ja mahdollisesti moniluokittelevan algoritmin tulosten esittämisessä.

5.2.1 Aineiston käsittely ja luokittelun tekeminen

Henkilökohtaisesti halusin panostaa paljon tähän esikäsittelyvaiheeseen, sillä se oli yksi puutteellisimmista asioista kesäprojektin osalta. Kesäprojektissa ongelmaa yksinkertaistettiin niin paljon, että käytännön hyödyt jäivät olemattomiksi. Tämä toisaalta oli täysin suunniteltua ja sallittua sen projektin osalta.

Salamavaiheen projektin tarkoitus oli tuottaa käytettävä lopputulos ja tiesin jo lähtötilanteessa, että täsmällinen opetusaineisto on tärkeässä asemassa.

Käytetyn menetelmän vuoksi koneoppimismallia varten syöteaineisto (viemärien videokuvaukset) täytyi muuntaa kuviksi. Mallia opetetaan antamalla sille mahdollisimman laaja ja paikkansapitävä kokoelma kuvia, joihin on liitetty luokka, johon kyseinen kuva kuuluu. Voidaan ajatella, että malli opettelee yhdistämään tarkastelemassaan kuvassa olevia piirteitä kuvaan liitettyihin luokkiin.

Kuvien keräämiseen videoista tehtiin yksinkertainen Python-ohjelma, jota ajettiin suoraan editorista. Ohjelman parametrejä muutettiin suoraan ohjelman koodia muuttamalla. Parametrejä olivat

- lähde- ja kohdekansio
- kuvankaappausten tiheys
- lopetusruutu.

Lähdekansion kaikki videot käsiteltiin automaattisesti. Käsittelysilmuksien sisällä oli toinen silmukka, jossa videon kuvaruutu kirjoitettiin omaksi tiedostoksi kohdekansioon, jos kyseessä oli valitun tiheyden mukainen kohta. Kun saavutettiin lopetusruutu, niin sisempi silmukka lopetettiin ja jatkettiin seuraavalla videolla, kunnes kaikki videot oli käyty läpi. Myöhemmässä vaiheessa lisäsin ohjelmaan mahdollisuuden rajata lähdekansioista valittavat tiedostot yksinkertaisen taulukkomuuttujan avulla. Tämä oli käytännöllistä esimerkiksi tilanteessa, jossa luokittelun perusteella halusin valita vain tietyt videot prosessoitaviksi.

Lopullinen tarkoitus oli kerätä kaikista videoista kaikki kuvaruudut, mutta luokittelutyön alussa oli hyödyllistä kerätä useista videoista muutamia kuvia. Käytetyllä laitteistolla esimerkiksi sadasta videosta 3–4 ruudun kirjoittaminen omaksi tiedostoksi kesti noin 10 minuuttia. Näin ollen myös ajankäytöllisesti oli järkevämpää aluksi prosessoida videot suppeasti. Myöhemmin totesin myös, että liian tiheä ruutujen kerääminen aiheuttaisi aivan liikaa ylimääräistä liian samankaltaista dataa. Sen vuoksi pääosa videoista pilkottiin siten, että niistä otettiin joka viides ruutu. Videoiden ruudunpäivitysnopeus oli 15 ruutua sekunnissa.

Luokkia voi olla useita ja kuvaan voi liittää useita luokkia. Luokittelutyössä on hyvä huomioida, että riippumatta lopullisesta tai senhetkisestä käyttötarkoituksesta luokittelun voi tehdä kattavammin kuin on tarpeen. Luokittelu tapahtuu manuaalisesti. Luokittelija katsoo kuvia kuva kerrallaan ja määrittää luokat, joihin kuvan tapaus kuuluu. Prosessi vie aikaa, joten jos on mahdollista suunnitella kattavasti erilaiset käyttötapaukset, niin se kannattaa tehdä, jottei samoihin kuviin tarvitse välttämättä palata myöhemmin.

Esimerkiksi tämän projektin tapauksessa kuvista kiinnostivat viat, putken materiaali ja täyttöaste. Luokittelutyön alkaessa ei kuitenkaan ollut selvää minkälaisilla malleilla erilaisia luokitteluja oli määrä tehdä. Jos olisi päätetty, että luokitellaan ensisijaisesti vain viat, niin projektin myöhemmässä vaiheessa olisi käytännössä jouduttu käymään koko materiaali uudelleen läpi putken materiaalin ja täyttöasteen osalta. Toisaalta, jos projekti lopulta rajautuisi vain vikoihin, niin ajallinen haitta ei olisi kovinkaan suuri, vaikka luokittelun yhteydessä tehtäisiin ns. turhaa työtä.

Hyvän luokittelun tekeminen ei toki aineiston mukaan ole triviaali tehtävä. Siinä missä lähes tulkoon kuka tahansa voi tehdä luokittelua esimerkiksi kissojen ja koirien välillä, vaatiikin viemäriputkesta otetun kuvan tulkitseminen enemmän harjaantuneisuutta. Maallikolle voi olla helppo tunnistaa putkeen tunkeutuneet juuret ja esimerkkikuvien perusteella hän voi mahdollisesti tunnistaa myös putkeen kertyneet rasvan tai vierasesineet, mutta vaikkapa syöpymien tai halkeamien havainnointi voi olla haastavampaa. Henkilökohtaisena kokemuksena yllätyin, miltä esimerkiksi vuoto saattoi näyttää. Ruskea patti putkiston saumassa ei vastannutkaan ajatusta tipoittain lirisevästä vesivanasta.

Luokittelun laadukkuuden vuoksi prosessissa otettiin mukaan asiakas, jolla oli paras tietämys ja kyky luokitteluun. Työtaakkaa pyrittiin kuitenkin jakamaan siten, että mahdollisimman alkuvaiheessa saataisiin useita eri vikatapauksia luokiteltua ja näin ollen arvokasta mallikuva-aineistoa amatöörien avuksi.

5.2.2 Mallin kehitys

Projektin lopullisessa rajauksessa päädyttiin kehittämään vuotovikoja tunnistava algoritmi. Mallista tuli kolmikerroksinen konvoluutioneuroverkko, joka antaa arvon välillä 0...1 sen mukaan miten todennäköisenä se arvioi kuvassa näkyvän kohdan sisältävän vuodon. Malli ohjelmoitiin Pythonilla Tensorflow'lla Keras-kirjastoa käyttäen.

Tensorflow on Googlen kehittämä kokonaisvaltainen alusta koneoppimisjärjestelmien kehittämiseen. Se on laaja kokoelma työkaluja ja kirjastoja. Lisäksi Tensorflow'n verkkosivusto tarjoaa paljon esimerkkejä ja oppimateriaalia erilaisten koneoppimismallien tekoon. [Tensorflow 2023.]

Keras on ohjelmointirajapinta ja kirjasto, jota käytetään Tensorflow'n ohjaamiseen. Sen avulla koneoppimismallien rakentaminen suoraviivaistuu. Keras sisältää tietomallit esimerkiksi seuraaville ydintoiminnoille: mallit, mallien kerrokset ja mallien opetukseen, tarkasteluun sekä arviointiin tarkoitetut funktiot. [Keras 2023.]

Koneoppimismallin rakennuksen kanssa alkuun pääseminen on nykyisillä työkaluilla jopa yllättävän yksinkertaista. Lisäksi moni asia määräytyy käytettävän laitteiston asettamista rajoituksista ja/tai mallin opetukseen käytettävissä olevasta ajasta. Projektin tarkoituksena ei ollut etsiä ja vertailla erilaisia malleja, vaan pitäytyä perinteisissä ratkaisuissa, jotta tulosten saaminen ja tulkitseminen olisi yksinkertaisempaa. Tästä syystä mallin arkkitehtuuriset valinnat oli mahdollista lyödä lukkoon nopealla aikataululla ja kehityksen painopiste oli opetusaineiston vaatimukseen vastaamisessa.

Käytetyssä mallissa kuvista saatuun syötedataan tehdään ensin keinotekoista vaihtelua. Toimenpiteen tarkoituksena on laajentaa opetusaineistoa tekemällä kuville pieniä muutoksia, kuten kuvan kääntö (vaakasuunnassa), pyöritys ja zoomaus. Koska muutokset tapahtuvat sattumanvaraisesti, voi siten muokattuja kuvia pitää tietystä mielessä uusina näytteinä opetusjoukossa. Tämän jälkeen kuvapikselien arvot väliltä 0–255 skaalataan välille 0–1. Jokaiseen kolmesta

konvoluutiokerroksesta on liitetty satunnaista tuloksen huomiotta jättämistä (karsintaa), jonka tarkoitus on hillitä mahdollista ylioppimista. Ylioppimisella tarkoitetaan sitä, että malli oppii kuvista niin paljon piirteitä, että algoritmi kykenee tunnistamaan sille syötetyn opetusaineiston lähes kauttaaltaan, muttei pysty yleistämään oppimaansa uuden aineiston tulkintaan [Håkansson & Hartung 2020: 163]. Konvoluutiokerrosten aktivointifunktiona on ReLU, joka on epälineaarinen ja nopea käyttää. Konvoluutiokerrosten jälkeen on kaksi tiheää kerrosta (joiden välissä vielä yksi karsinta), joista jälkimmäinen antaa ulostuloarvon.

Konvoluutiomallin rakennuksessa tärkeitä muuttujia ovat

- opetusjoukon koko
- virhefunktio
- optimointifunktio ja sen oppimisnopeus
- aktivointifunktiot
- kuvan resoluutio
- opetus- ja validointiaineiston jakauma
- konvoluutiokerrosten filttorien määrä ja koko
- konvoluutiokerrosten määrä
- karsinta.

Opetusjoukon koko kertoo sen, kuinka suurta määrää opetukseen käytettävistä näytteistä (kuvista) käsitellään, ennen kuin neuronien kytkentöjen painoja muutetaan. Isompi joukon koko vaikuttaa siihen, että oppimiseen on käytössä kerrallaan monipuolisempi läpileikkaus aineistosta. Toisaalta pienemmällä joukolla on mahdollista saada oppiminen nopeammaksi.

Neuronikytkentöjen painojen muutokseen puolestaan vaikuttaa valittu virhefunktio, joka tarkkailee neuroverkossa tapahtuvaa virhettä odotettuun tulokseen nähden [Håkansson & Hartung 2020: 226]. Optimointifunktion tehtävänä on saada aikaan siirtymä kohti parempia tuloksia, eli vähentää virhettä säädetyllä nopeudella. Pyrkimyksenä on luonnollisesti päästä nopeasti parempaan lopputulokseen varoen kuitenkin samalla, etteivät korjausliikkeet ole liian suuria [Håkansson & Hartung 2020: 228]. Aktivointifunktio puolestaan määrittelee tavan,

jolla neuroneilta seuraavaan kerrokseen siirtyvät arvo epälinearisoidaan [Keller 2020: 222]. Selkeämmin ilmaistuna se toimii mittarina neuronin vaikutuksesta, eli antaako kyseinen neuroni signaalia eteenpäin.

Kuvan resoluution valinta on yksi seikoista, joka määräytyy pitkälti käytettävissä olevien laitteistoresurssien mukaan. Kun digitaalinen kuva toimii syöteenä ja verkon laskentaoperaatioiden määrä johtuu kuvan pikselien määrästä, on helppo ymmärtää, että nykymittapuulla jopa vaatimattomat resoluutiot kasvattavat laskennan tarvetta nopeasti. Mallin kehitysvaiheessa käytössä oli muutamia vuosia vanha kuluttajatasen (grafiikkasuorittimella varustettu) kannettava tietokone. Valitsemalla syötekuvien (Kuva 7) resoluutioksi 256x256 pikseliä käyttäessäni verkossa oli noin 59 miljoonaa opetusparametriä. Kaksinkertaisen kokoisella kuvalla opetettavien parametrien määrä olisi kasvanut 250 miljoonaan. Siltikin kyseessä olisi vielä voimakas skaalaus alkuperäisestä kuvasta, jonka resoluutio on 1280x720 ja joka tuottaisi lähes miljardi parametriä opetettavaksi.



Kuva 7. Opetusaineiston kuvia ja tieto niiden luokasta

Skaalauksen myötä kuvasuhteet muuttuvat, mutta sillä ei ole kuvantunnistuksessa oleellista merkitystä. 1:1-suhteesta on jopa apua tilanteissa, joissa algoritmille syötettävä aineisto saattaa sisältää esimerkiksi pysty- ja vaakakuvia, mutta algoritmi käsittelee vain yhtä resoluutiota.

Opetusta ennen aineisto jaetaan opetus- ja validointiaineistoihin. Käytännössä opetusosa on aineistoa, jonka perusteella algoritmi oppii piirteitä ja validointiosa puolestaan toimii varmistustehtävässä. Toisin sanoen opetusvaiheessa olevan algoritmin kyvykkyyttä tarkkaillaan aineistolla, jonka piirteitä se ei pyri oppimaan, mutta jonka tunnistaminen hyvin tai huonosti auttaa oppimisprosessissa. Jakauma valitaan perinteisesti siten, että suurempi osa (esimerkiksi 80 %) näytteistä on opetusaineistoa.

Konvoluutiokerrosten rakennetta ja toimintaa kuvattiin aliluvussa 2.1. Kuten aikaisemmin mainitsin, projektissa ei ollut tarkoitus tehdä erityistä vertailua eri arkkitehtuurien välillä, joten muun muassa konvoluutiokerrosten määrään ja rakenteeseen valittiin toteutustapa yleisesti tunnetuista ratkaisuksista. Kun sitten osoittautui, että valitulla rakenteella oli havaittavissa kyky oppia syötetystä aineistosta, ei projektin tässä vaiheessa tutkittu muita vaihtoehtoisia mahdollisuuksia.

Koneoppimismallin kehittämiseen liittyy siis paljon muuttujia. Vaikka totesinkin, että alkuun pääseminen on nykyisillä menetelmillä jopa helppoa, niin se ei poisulje sitä, etteikö tutkittavaa ja kokeiltavaa riittäisi myös valtavasti. John D. Kelleher [2020: 92–93] nostaa esiin joukon kysymyksiä liittyen näihin perustavanlaatuisiin muuttujiin, joita edellä yritin selventää. Kysymyksiä esimerkiksi neuroverkossa tarvittavista kerroksista tai käytettävistä aktivointifunktioista. Niihin kysymyksiin varmasti monet koneoppimisen kanssa aloittavat ovat kaivanneet yleispäteviä vastauksia, mutta käytännössä niihin ei sellaisia löydy. Sen sijaan hän kirjoittaa, että kokeellinen työ on usein merkittävä osa syväoppivaa verkkoa luotaessa ja että arvot muuttujille voivat löytyä niin sääntöjä noudattamalla, kuin myös yrityksen ja erehdyksen kautta tai joskus jopa tekijänsä intuitiolla.

5.2.3 Mallin opetus

Mallin valintaan liittyneiden testauksien jälkeen ensimmäinen vuotovikoja sisältänyt aineisto koostui yhteensä 1291 kuvasta. Näistä 650 kuvassa ei ollut selvästi näkyvää vuotoa ja 641 kuvassa vuodon aiheuttamat jäljet olivat havaittavissa. Kuvat kerätiin 26 eri videosta, joka on verrattaen pieni määrä, mutta jolla uskottiin, että päästään alkuun.

Vuotoviat valittiin ensimmäiseksi kehityskohteeksi muun muassa sen vuoksi, että niiden merkitys jatkotoimenpiteiden määräytymisessä on usein suuri. Yrityksen kanssa käydyn keskustelun pohjalta huomattiin, että vakavimmat tapaukset otetaan tarkempaan käsittelyyn käytännössä jo kuvauksia suoritettaessa ja että vähäisempien tapauksien kohdalla priorisointi yleensä johtaa siihen, että nykytilanteessa ne jäävät isompien vikojen varjoon. Sen vuoksi automaattisen analysoinnin suurin kiinnostus rajautuu niin sanotusti keskivaiheen ongelmiin, joita ei todennäköisesti huomata kuvauksia tehdessä, mutta joilla on suurempi painoarvo toimenpidesuosituksen tekemisessä. Tällaisiin vikoihin kuuluvat vuotojen lisäksi esimerkiksi muodonmuutokset, viettokaltevuudet ja putkiosuuk-sien kaartumiset. Ammattilaisen avulla tehdyissä luokitteluissa löytyi hyviä esimerkkejä vuotovioista ja niiden tulkinta oli kohtuullisen helppo omaksua. Nämä seikat ohjasivat juuri vuotojen valintaa algoritmin ensimmäiseksi tehtäväksi.

Opetusaineiston vähäisyyttä yritettiin kompensoida valitsemalla molempiin luokkiin kuuluvia kuvia tarkoituksella samoista videoista, jotta niiden yleisilme olisi yhtäläinen ja nimenomaan vuodoilla olisi mahdollisuus erottua. Oletus oli, että jos kuvat valittaisiin eri videoista, saattaisi algoritmi helpommin harhaantua etsimään muita piirteitä vuotokuvissa esiintyvistä putkista. Joukkoon mahtui tapauksia, joissa samassa videossa oli suuri määrä jompaankumpaan luokkaan kuuluvia tapauksia ja vähemmistö toisia. Pintapuolisen tarkastelun perusteella kokonaisuudessa oli kuitenkin paljon erilaisia putkia ja niissä esiintyneiden vuotojen kirjo oli melko hyvä. Ensimmäisellä aineistolla oli kuitenkin ensisijaisesti tarkoitus päästä testaamaan valitun mallin toimivuutta.

Koska algoritmin oppimisvaiheessa käytetään aineistoa, jonka kuuluvuus jompaankumpaan luokkaan on tiedossa, saadaan tietoa luokittelun onnistumisesta niin sanotun konfuusiomatriisiin (Kuva 8) muodossa. Se koostuu neljästä laskennallisesta arvosta: todelliset positiiviset ja negatiiviset sekä virheelliset positiiviset ja negatiiviset. Todellisella positiivisella tarkoitetaan näytettä, joka kuuluu luokitella, tässä tapauksessa, vuodoksi ja myös luokitellaan niin. Jos luokittelu meni väärin, niin kyseessä olisi virheellinen negatiivinen. [Håkansson & Hartung 2020: 181.]

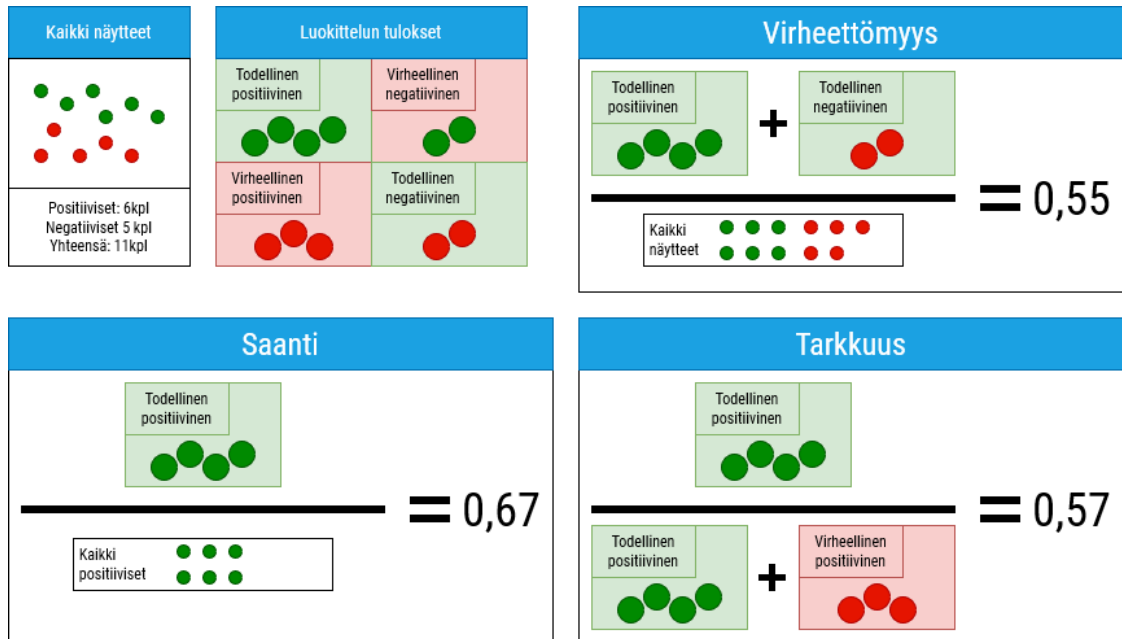
	Luokiteltu positiiviseksi	Luokiteltu negatiiviseksi
Positiiviset näytteet	Todellinen positiivinen	Virheellinen negatiivinen
Negatiiviset näytteet	Virheellinen positiivinen	Todellinen negatiivinen

Kuva 8. Konfuusiomatriisi

Mallin kyvykkyyttä ja oppimisen edistymistä tarkasteltiin seuraavilla arvoilla:

- virheettömyys
- tarkkuus
- saanti.

Arvot saadaan laskettua konfuusiomatriisin tulosten perusteella (Kuva 9). Virheettömyys kuvaa sitä, miten suuri osuus todellisia positiivisia ja negatiivisia koko joukosta pystytään havaitsemaan. Tarkkuus on todellisten positiivisten osuus kaikista positiivisiksi luokitelluista, ja saanti kertoo todellisten positiivisten osuuden kaikista positiivisista.

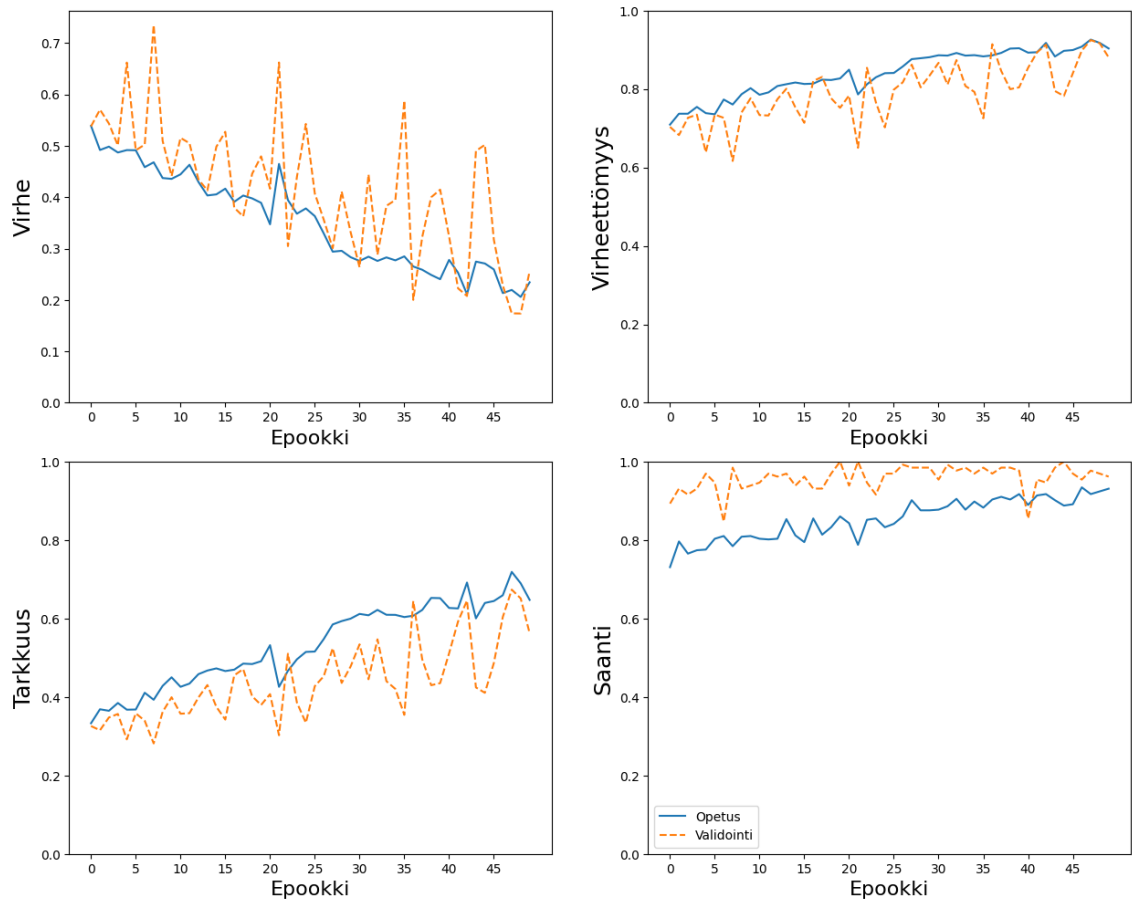


Kuva 9. Konfuusiomatriisin avulla voidaan laskea mm. virheettömyys, tarkkuus ja saanti

Edellä mainittujen arvojen lisäksi seurataan opetukseen liittyvää virhettä ja sen vähenemistä. Virhearvo määräytyy optimointifunktion tuottaman pisteytyksen mukaisesti [Håkansson & Hartung 2020: 226].

Ennen ensimmäistä opetusajoa ei voinut olla varma aineiston riittävydestä tai mallin kyvystä oppia. Opetus tapahtuu erissä, joita kutsutaan epookeiksi. Yksi epookki tarkoittaa tilannetta, jossa koko opetusaineisto on käyty läpi ja algoritmin arvot on muutettu tulosten mukaisesti. Tein ensimmäiset testit vaiheittain siten, että aloitin 20 epookilla, sitten tallensin mallin ja jatkoin vielä toisen kerran 20 epookilla. Koska tulokset näyttivät paranevan, tallensin mallin vielä kertaalleen ja ajoin opetusta lopuksi 40 epookin verran. Jokaisen vaiheen jälkeen kuvansin edistymisen tarkasteltujen arvojen kaaviona (Kuva 10).

Opetus ja validointi



Kuva 10. Esimerkki opetuksen seurantaan tulostetusta kaaviosta

Tämän jälkeen tein testejä yksittäisillä kuvilla, jotka oli otettu opetus- ja validointiaineiston ulkopuolelta. Myös niiden testien tulokset antoivat viitteitä algoritmin kyvystä oppia.

Koska prosessi antoi viitteitä toimivuudesta, parantelin testaamiseen liittyvää koodia, jotta testaaminen olisi suoraviivaisempaa ja tulokset olisivat helpommin tarkasteltavissa. Tässä vaiheessa testiaineistoon kuului 300 kuvaa, joista vuotokuvia oli 57 ja vuodottomia 243.

Ensimmäinen kolmessa osassa suoritettu opetusajo kesti kokonaisuudessaan noin 45 minuuttia. Tiedossa oli, että opetettavan aineiston määrän kasvaessa prosessi tulee hidastumaan. Siitä huolimatta uskalsin kasvattaa syötteen

resoluutiota ja pienentää algoritmin oppimisnopeutta. Muutokset olivat sopivia ja opetusajojen kestot pysyivät maltillisina. Viides ajo kesti pisimpään, 5 tuntia ja 15 minuuttia. Siinä oli myös eniten epookkeja (50 + 50 + 25). Yhden epookin kesto oli suurin piirtein kolminkertaistunut ensimmäiseen opetusajoon verrattuna.

Mallin opetus perustui siis pääasiassa datasettien muunteluun (Taulukko 2). Tämä voi tuntua käänteiseltä lähestymistavalta, mutta oli perusteltua, koska aineiston luokittelua tehtiin koko ajan ohessa ja kiinnostusta oli myös siihen miinäläisiä vaatimuksia algoritmi aineistolle asettaa. Kolmen ensimmäisen opetuskeran jälkeen toin opetusmateriaaliin tarkoituksellista epätasapainoa. Siten oli mahdollista kasvattaa aineiston kokoa merkittävästi. Aineiston epätasaisuus kerrotaan mallille asettamalla luokkakohtaiset painotukset.

Taulukko 2. Opetusaineistojen koko. Luokkakohtainen jakauma sekä opetus- ja validointijakauma.

Käytetty opetusajoissa	Vuodot	Ei vuotoa	Kaikki	Opetukseen	Validointiin
1 ja 2	641	650	1291	1033	258
3	1024	1021	2045	1636	409
4	1021	2068	3089	2472	617
5	711	3608	4319	3456	863

Opetusajoja kertyi kaikkiaan viisi. Niiden pohjalta päästään tekemään jatkokehitystä ja mahdollisesti luomaan vastaavia malleja ja opetusaineistoja muille tutkittaville vioille.

5.3 Koneoppimismallin tuottamien tulosten arviointi

Koneoppimismallien tuottamien tulosten tulkinta ja vertailu voi olla hankalaa etenkin alkupään vaiheiden osalta. Erityisesti tähän vaikutti se, että tässä

projektissa muutokset tapahtuivat aineistossa. Oli kuitenkin mahdollista havaita, että oppimista tapahtui.

Merkittävin mallin kyvykkyyden mittari on kuitenkin sen kyky luokitella oikein sille entuudestaan tuntemattomia näytteitä. Tulosten tulkinnan kanssa pieni ongelma syntyi siitä, että projektin alussa ei tehty yksittäistä kattavaa testiaineistoa, jota olisi käytetty eri opetuskertojen jälkeen. Tämä johtui pääasiassa siitä, että perinteisessä mallin kehityksessä erilaisia algoritmeja kehitetään opetus- ja validointiaineistojen tuottamien tulosten perusteella ja vasta lopuksi niissä parhaiten suoriutunut malli tarkistetaan varsinaisella testiaineistolla. Projektissa puolestaan testiaineisto kehittyi samalla muun prosessin mukana ja toimi eräänlaisena pikatarkastuksena siitä, että tuloksista olisi käytännön hyötyä. Näin ollen tulosten arvioinnissa pääpaino on opetuksen aikaisten mittarien seuraamisessa.

Jokaisessa opetusajossa algoritmin suorituskkyky validoinnin mittareilla kasvoi hyviin lukemiin. Testiaineistot sen sijaan osoittivat, että ylioppimisen mahdollisuus on todennäköistä. Opetuksen jälkeisistä tuloksista (Taulukko 3) on nähtävissä, että opetuksen jatkuessa tarpeeksi pitkään mittarien arvot ovat erittäin hyviä. Käytännössä tilanne on tällöin kuitenkin se, että algoritmi on oppinut kuvista niin yksityiskohtaisia piirteitä, että se osaa tunnistaa jo lähes koko syötetyn aineiston.

Taulukko 3. Opetuksen mittarit viimeisen opetusjoukon jälkeen

Opetusajo	Virheettömyys	Tarkkuus	Saanti
2	0,95	0,95	0,96
3	0,93	0,93	0,93
4	0,97	0,94	0,97
5	0,95	0,78	0,95

Vaikka validointijoukko (Taulukko 4) sinänsä on sattumanvaraisesti erotettu opetukseen käytetyistä näytteistä, on se kuitenkin lähtökohtaisesti samasta aineistosta. Näin ollen ylioppimisen voi nähdä myös näistä tuloksista.

Taulukko 4. Validoinnin mittarit viimeisen opetusjoukon jälkeen

Opetusajo	Virheettömyys	Tarkkuus	Saanti
2	0,95	0,92	0,97
3	0,95	0,93	0,97
4	0,92	0,81	0,99
5	0,94	0,74	0,97

Karu todellisuus algoritmin tekemässä testiaineistojen tulkinnessa kertoo tarpeesta saada luokiteltua runsaasti lisää kuvia, jotta malli osaisi paremmin yleistää oppimaansa. Toisaalta testitulokset (Taulukko 5) myös tukevat sitä, että laadukkaammalla aineistolla nähdään selkeitä parannuksia.

Taulukko 5. Testauksen mittarit opetuksen jälkeen

Opetusajo	Virheettömyys	Tarkkuus	Saanti
2	0,55	0,29	0,93
3	0,79	0,06	0,67
4	0,85	0,73	0,79
5	0,73	0,79	0,37

Tarkasteltaessa tuloksia kokonaisvaltaisemmin voidaan tehdä muutamia päätelmiä ja arvioida tilanteen jatkumahdollisuuksia. Testaustuloksissa virheettömyys ja tarkkuus pääasiassa paranevat isompien ja tarkemmin valikoitujen opetusaineistojen myötä. Saannissa vaihtelu on suurempaa, mutta siihen voi vaikuttaa esimerkiksi se, mihin suuntaan algoritmi alkaa kehittymään todellisten positiivisten valinnassa. Kun ajatellaan projektin tilannetta, niin positiiviset tapaukset (vuodot) ovat niitä, joita varsinkin alkuvaiheessa halutaan saada mahdollisimman paljon havaittua ja samalla voidaan hyväksyä isompi määrä virheellisiä positiivisia. Sen vuoksi viidennen opetusajon saanti ei ole hyvä tulos, vaan siinä on selvä parantamisen kohde.

On myös yhä huomioitava, että testauksessa ei käytetty säännöllistä kuva-aineistoa, joten eri ajojien keskinäinen vertailu täytyy suorittaa, kun pidetään tämä seikka mielessä. Kun aineistoa saadaan luokiteltua enemmän ja testiin käytettävä osuus laajenee, voidaan palata tarkastamaan näiden aikaisemmin koulutettujen mallien kykyä. Todennäköisempää kuitenkin on, että siinä vaiheessa on jo myös uusia parempia opetusajoja suoritettu.

Saadut tulokset kertovat mahdollisuuksista, mutta ensisijaisesti parantamisen tarpeesta. Nykyisessä tilassaan malli luultavasti aiheuttaisi vielä enemmän työtä sen sijaan, että se helpottaisi työntekoa. Parannuskeinoja kuitenkin on paljon, eivätkä ne rajoitu ainoastaan isomman opetusaineiston tekemiseen, vaan luonnollisesti myös käytetyn mallin kehittämisessä on vielä paljon potentiaalia.

5.4 Raportointiohjelman kehittäminen

Big-Flash-projektin tavoitteissa on saada algoritmin tuottamista tuloksista selkeä raportti. Algoritmin nykyisessä kehitysvaiheessa tuloksia saadaan kuitenkin vain opetusprosessin yhteyteen liitetyillä mittareilla ja testiaineiston käsittelyyn tarkoitettulla koodiosuudella. Suunnitteilla on ohjelma, joka tarkkailisi tiettyä kansiota, jonne kuvatut videot viedään. Tämän jälkeen ohjelma tekisi videoiden pilkkomisen kuviksi ja syöttäisi ne algoritmille. Tuloksissa voitaisiin huomioida sekä videokohtaiset arviot että nostaa esille tiettyjä algoritmin vakavimmin arvioimia tapauksia.

6 Projektin arviointi

Projektin ja insinööriyön rajaukset olivat aikataulusyistä hiukan erilaiset. Insinööriyössäni olen keskittynyt projektin osalta sen toteutuneisiin vaiheisiin. Pyrin tarkastelemaan tässä luvussa projektia kokonaisuutena ja esimerkiksi jatkokehityksen osalta näkökulma on koko projektin päättymishetkeltä.

6.1 Toteutuksen peilaus suunnitelmaan

Mielestäni projektisuunnitelman laatiminen onnistui suurimmaksi osaksi hyvin. Kaikki oleelliset seikat oli pystytty ottamaan huomioon suunnitelmassa ja aikataulut oli selkeästi laadittu. Aikataulussa oli myös pyritty ottamaan huomioon mahdolliset yllätykset ja viivästykset. Projektin tavoitteet vastasivat melko hyvin niille varattua aikaa. Suunnittelua auttoi selvästi aikaisemmin toteutettu projekti ja siitä saatu kokemus. Suunnittelussa otin myös huomioon yhtä aikaa tehtävän insinööriyön aiheuttaman aikavaatimuksen. Toisaalta projektin etenemiseen täydellä varmuudella insinööriyön aikataulua ajatellen en ollut suunnitellut riittävän hyvin ja lopussa jouduin tekemään joitakin kompromisseja ja rajauksen tarkennuksia.

Suunnitelmaa tehdessä arvioin selkeästi väärin sen, kuinka suuressa roolissa ulkopuoliset tekijät voivat olla alkuvaiheen prosessissa. Osittain kyse oli virhearviosta, mutta toisaalta kyse oli myös siitä, että luokitteluympäristön pystyttämävaiheessa sallin itseni poiketa alkuperäisestä aikataulusta. Pienet viivästykset aiheuttivat isoa haittaa, kun oletukseni niiden merkityksestä osoittautuivatkin lopulta vähätteleviksi. Jälkeenpäin koen, että suunnitelmaa tehdessä olisi täytynyt varautua viivytyksiin aikataulun väljyyden lisäksi joillakin selkeillä konkreettisimmilla toimilla, jos jokin vaihe ei sujuisi odotetulla tavalla. Tämä antaisi suunnitelmallisen hätäulostien ongelmakohdissa.

Projektin lopputuloksen osalta viivästykset eivät kuitenkaan aiheuttaneet samanlaista aikataulupainetta, kuin miten tilanne saattoi olla insinööriyön osalta. Oikeastaan insinööriyön asettama aikataulu loi hyvän rytmityksen luokitteluaineiston vähitellen tapahtuvan valmistumisen ja algoritmien kehityksen väliseen aikaan. Ilman painetta saada insinööriyölle yhtenäinen kokonaisuus, olisi ollut mahdollista, että koneoppimismallia ei olisi lähdetty kehittämään toteutuneella tavalla pienten iteratiivisten askelien muodossa, vaan kiusaus olisi voinut olla suuri yrittää saavuttaa valmis analyysijärjestelmä yhdellä moniluokittelyllä algoritmilla.

Ei ole silti epäilystäkään, etteikö myöhään käynnistynyt luokittelu olisi vaikuttanut silti joihinkin ratkaisuihin negatiivisesti ja tavoitteistakin jouduttaneen tinkimään projektin loppuvaiheissa.

Koneoppimisprojektissa toteutui paljon niitä asioita, joita siihen oli kaavailtakin. Moni asia tuli itselleni jo kertauksena, mutta samalla huomasin, että kertaukselle oli tilausta. Vähiten yllätti, kuinka paljon aineistotyötä tällaiseen projektiin liittyy. Se ei ole mielenkiintoisinta, mutta se on hinta, joka on maksettava, jos haluaa tehdä jotain uutta sen sijaan, että toistaa valmiiksi kaluttuja oppikirja-esimerkkejä.

Asioista, joita tekisin toisin, nousee kirkkaimpana se, että projektista olisi voinut tehdä paremmin esiteltävän. Varsinkin oltuani osana Big-Flash-hanketta on minulta useaan otteeseen toivottu, että olisi jotain visuaalista esitettävää. Toki sellaisen järjestäminen on haastavaa, kun suuri osa työstä on suunnittelua, koodin kirjoittamista ja aineiston luokittelemista, mutta mahdollista se silti olisi. Onneksi loppua kohden näytettävääkin alkaa syntyä tulosten muodossa kuin itsestään.

6.2 Hyödyllisyys

Projektilla oli arvoa yritykselle. Aihe on kiinnostanut heitä jo pidemmän aikaa ja oli mukava huomata, että projektin edetessä kiinnostus ei laantunut, vaan tuloksista haluttiin olla tietoisia ja tietynlainen jatkokehityspöhinä oli koko ajan läsnä. Lopputulos on vielä tuntemattoman matkan päässä varsinaiseen tuotantokäyttöön liittämistä, mutta käytetyillä menetelmillä on selvästi mahdollisuus tuottaa hyviä tuloksia.

Toteutettu vuotoja tunnistava koneoppimismalli on hyvä ensiaskel kokonaisvaltaisemmalle järjestelmälle. Vuotoviat ovat yksi kriittisistä asioista kuntotutkimusta tehtäessä. Mallin toistaminen muiden vikojen osalta tulee pääpiirteissään onnistumaan melko suoraan. Eri viat tulevat toki antamaan erilaisia haasteita, mutta niihin on nyt jo huomattavasti helpompaa varautua, kun ensimmäistä algoritmia on saatu kehitettyä.

Projektin hyödyllisenä sivutuotteena oli myös luokitteluun kehitettyjen menetelmien käyttöönotto Label Studion ja oman sovelluksen muodossa. Nämä molemmat järjestelmät ja kokemukset niiden käyttöönotosta ja käytöstä tekevät jatkokehityksestä suoraviivaisempaa. Nykytilanteessa vikojen luokittelu on tapahtunut sanallisin kirjauksin järjestelmään, mutta sen hyödyntäminen koneoppimismallien opetusaineiston kokoamisessa on haastavaa. Vaikka prosessissa on nyt kaksi rinnakkaista järjestelmää, on tämä hyvä suunta laadukkaamman datan keräämiselle jatkossa.

6.3 Jatkokehitys

Kehitetylle järjestelmälle riittää runsaasti jatkokehityskohteita. Näitä ovat muun muassa luokittelutulosten tarkentaminen, kattavampi tulosten raportointi ja yrityksen järjestelmään integrointi.

Projektissa kehitetty malli on joukko yksittäisiä algoritmeja, jotka pystyvät havaitsemaan vikoja, joihin ne ovat erikoistuneet. Pelkästään vian havaitseminen ei kuitenkaan monessa tapauksessa kerro jatkoselvityksen saatikka korjaustoimenpiteiden kiireellisyydestä, vaan tarve tarkemmalle tulokselle on olemassa. Pidemmälle kehitettynä ja ”kenttätestattuna” algoritmit voisivat yhteistyössä antaa suoraan ehdotuksen toimenpideaikataulusta, esimerkiksi kokonaisen kaupunkialueen kuvausten tulosten perusteella.

Tarkemmat ehdotukset ja tehtävien priorisointi edellyttäisi havainnollisempaa raportointia. Algoritmien tuloksista tulisi koota yhtenäinen selvitys, jossa yksittäisten videoiden pohjalta tehdyt havainnot eivät olisi merkittäviä, vaan voitaisiin saada kokonaisvaltaisempi kuva laajemman kuvausotoksen pohjalta luotuna.

Parhaimmillaan koneoppimisavusteinen analyysi ja yrityksen nykyinen olemassa oleva zoom-kuvausten tallennus- ja määritysjärjestelmä voitaisiin yhdistää ja automaattinen raportointi tapahtuisi osana videoiden lataamista järjestelmään.

7 Yhteenveto

Viemärien kuntotutkimuksella on valtava merkitys yhteiskunnan ja kuntien nykyhetkessä ja tulevaisuudessa. Kotimaisia toimijoita ei ole liiaksi, ja infrastruktuurin ikärakenne osoittaa ongelmien vertauskuvallisen jäteäallon saapuvan pian. Aineiston kerääminen ja tiedon tallentaminen tuottaa ongelmista paljon tietoa, mutta ilman kunnollista suunnitelmaa ja priorisointia ennakoiva kunnostus voi olla myöhäistä, ja maksettava lasku saattaa muodostua kalliimmaksi.

Projektia toteuttaessani olen päässyt omin silmin näkemään kauhistuttavassa kunnossa olevia viemäriputkia. Toisaalta tämä on tarkoittanut mittavia mahdollisuuksia kattavan ja monipuolisen opetusaineiston kokoamiseen, mutta se on myös ollut omiaan avaamaan silmiä tilanteen vakavuudelle. Toivon, että kehittämäni järjestelmä on vasta hyvä alku alan toiminnan muutokselle, jossa manuaalista tarkistamista päästäisiin vähentämään ja sekä ennakointiin että korjaustoiimiin panostaminen tulisi saavutettavammaksi.

Insinööriyön tavoitteena oli luoda koneoppimisalgoritmi viemäriputkien vuotovikojen havainnointiin ja samalla kuvata siihen johtanutta prosessia, jotta jatkokehitystyö muiden kuntotutkimuksessa etsittävien vikojen analysoimiseksi olisi mahdollista. Kehittämälläni mallilla on mahdollista tunnistaa vuotovikoja, ja olen osoittanut, että opetusaineiston kasvaessa ja sen laadun parantuessa vuotojen tunnistamiseen liittyvä epävarmuus vähenee. Projektin oheisansiona kehitin kuvien luokitteluun käytettävän web-sovelluksen. Sovellus on helposti muokattavissa vastaamaan luokittelun mahdollisesti vaihtuvia tarpeita.

Insinööriyössäni kuvattu prosessi koneoppimisprojektin käynnistämiseksi ja läpiviemiseksi on hyvä ohjenuora jatkokehitystä ja uusia projekteja ajatellen. Se antaa tärkeää tietoa projektityöhön liittyvistä riskeistä ja vaatimuksista, sekä oivalluksia asioiden toteuttamiseen. Koneoppiminen on vielä suhteellisen tuore alue, jossa pioneerihenkisyys on yhä arvossaan, vaikka markkinat tarjoavat jo valmiitakin ratkaisuja. Tekemäni työ toimikoon innoittajana kaikille koneoppimisesta kiinnostuneille.

Lähteet

Alpaydin, E. 2021. Koneoppiminen. Helsinki: Terra Cognita Oy.

Baeldung. 2023. Verkkoaineisto. What is the Purpose of a Feature Map in a Convolutional Neural Network. <<https://www.baeldung.com/cs/cnn-feature-map>>. Luettu 6.5.2023.

Big-Flash. 2023. Verkkoaineisto. <<https://bigflash.metropolia.fi/hanke/>>. Luettu 6.5.2023.

Express. 2023. Verkkoaineisto. <expressjs.com>. Luettu 6.5.2023.

Fly.io. 2023. Verkkoaineisto. <fly.io>. Luettu 6.5.2023.

Google Cloud. 2023. Verkkoaineisto. <cloud.google.com>. Luettu 6.5.2023.

Haurum, J.B., Moeslund, T.B. 2021. Sewer-ML: A Multi-Label Sewer Defect Classification Dataset and Benchmark.

Håkansson, A., Hartung, R.L. 2020. Artificial Intelligence Concepts, Areas, Techniques and Applications. Lund: Studentlitteratur AB.

Kelleher, J.D. 2020. Syväoppiminen. Helsinki: Terra Cognita Oy.

Keras. 2023. Verkkoaineisto. <github.com/keras-team/keras>. Luettu 6.5.2023.

Kolari, J., Kallio, A. 2023. Tekoäly 1·2·3 Matkaopas tulevaisuuteen. Jyväskylä: Docendo.

Label Studio. 2023. Verkkoaineisto. <labelstud.io>. Luettu 6.5.2023.

Lampola, T., Kuikka, S. 2018. Viemäreiden kuntotutkimusopas. Helsinki: Vesilaitosyhdistys.

MongoDB. 2023. Verkkoaineisto. <www.mongodb.com>. Luettu 6.5.2023.

Node.js. 2023. Verkkoaineisto. <nodejs.org>. Luettu 6.5.2023.

OpenAI. 2023. Verkkoaineisto. <openai.com/blog/chatgpt>. Luettu 6.5.2023.

Python. 2023. Verkkoaineisto. <www.python.org>. Luettu 6.5.2023.

React. 2023. Verkkoaineisto. <react.dev>. Luettu 6.5.2023.

Rosenblatt, F. 1962. Principles of Neurodynamics. Washington DC: Spartan Books.

Scott, T. 2023. Verkkoaineisto. I tried using AI. It scared me. <<https://www.youtube.com/watch?v=jPhJbKBuNnA>>. Luettu 6.5.2023.

Tensorflow. 2023. Verkkoaineisto. <www.tensorflow.org>. Luettu 6.5.2023.

Underground City. 2023. Verkkoaineisto. UC Presentation (ENG). <www.youtube.com/watch?v=HOTABVSBJwI>. Luettu 6.5.2023.

Vesilaitosyhdistys. 2023. Verkkoaineisto. <www.vvy.fi/vesihuolto/mita-vesihuolto-on/>. Luettu 6.5.2023.

Visual Studio Code. 2023. Verkkoaineisto. <code.visualstudio.com>. Luettu 6.5.2023.

Winastwan, R. 2023. Verkkoaineisto. Image Classification with Vision Transformer. <towardsdatascience.com/image-classification-with-vision-transformer-8bfde8e541d4>. Luettu 7.5.2023.

Zsolnai-Fehér, K. 2023. Verkkoaineisto. NVIDIA's New Video AI: Game Changer!. <<https://www.youtube.com/watch?v=3A3OuTdsPEk>>. Luettu 7.5.2023.