


# Tekoäly tuo tarkkuutta matkailuyrityksen kävijädataan

29.12.2022

## Johdanto

Etelä-Pohjanmaan alueella on useita matkailualan yrityksiä, joiden liiketoimintamalli ainakin osaltaan perustuu paikan päällä käyviin ja ostoksia tekeviin vierailijoihin. Koska kävijämäärillä on selkeä yhteys liikevaihtoon ja kannattavuuteen, on yrityksillä halua seurata niiden kehittymistä esimerkiksi erilaisten markkinointitoimenpiteisiin liittyen. Kuitenkin varsin yleinen piirre yrityksissä on, että matkailuliiketoiminta on kehittynyt jonkin muun yritystoiminnan kylkeen, joten se on mittasuhteiltaan maltillista ja siten kustannusrakenteeltaan tarkkaa. Niinpä kävijämäärän seuranta on usein toteutettu esimerkiksi silmämääräisten arvioiden tai kassan myyntikuittien määrän mukaan.

TehoData-hankkeen pilottina haluttiin tarkastella mahdollisuutta toteuttaa kustannustehokas kävijäseuranta, joka tuottaa päiväkohtaisen raportin paikalla olleiden ihmisten lukumäärästä ja jota on helppo käyttää. Tarvemäärittelyn myötä totesimme liikkuvien kohteiden videoseurannan olevan tässä tapauksessa tarkoituksenmukaisin perusratkaisu, esimerkiksi valokennolaskureiden tai älypuhelimien seurannan sijaan. Tarkoitukseen tuotettuja kaupallisia ratkaisuja löytyy markkinoilta runsaasti, mutta kustannusvaikutuksen minimoimisen mielessä halusimme tarkastella, saisiko ongelman ratkaistua myös avoimen lähdekoodin pohjalta. Niinpä pilotin tarvemäärittelyksi asettui:

- (Lähes) nollakustanteisuus.
- Automaattinen tietyllä alueella päivän aikana vierailleiden ihmisten laskenta.
- Ilman että koko aluetta voidaan “nähdä” yhdellä kertaa (esimerkiksi rakennusten asettelun vuoksi).
- Järjestelmästä tulee saada raportit esimerkiksi sähköpostiin tai web-käyttöliittymän kautta CSV-muodossa.

- järjestelmä toimii itsekseen ilman että henkilökunnan tarvitsee puuttua sen toimintaan.

# OpenCV videokuvan käsittelyn pohjana

Erilaisista toteutusmahdollisuuksista lähempään tarkasteluun nousi OpenCV-konenäkökirjasto. Kirjasto sisältää muutamia valmiita perinteisempään kuvamanipulaatioon perustuvia kohteenseurannan algoritmeja. Halusimme kuitenkin tunnistaa seurattavat kohteet (ihmiset) videokuvasta automaattisesti, joten päätimme kokeilla syväoppimiseen perustuvien kuvantunnistusneuroverkkojen käyttöä. Eräs kohteiden tunnistamiseen hyvin soveltuva avoimen lähdekoodin neuroverkkoneuroverkkokehys (framework), jota OpenCV tukee on Darknet, jolle rakennettu neuroverkkomalli YOLOv4 on saatavilla valmiiksi COCO-aineistoon koulutettuna eli se soveltuu myös ihmisen erottamiseen kuvan muista elementeistä. Lukuisista internetissä tarjolla olevista esimerkeistä jatkokehittelyn kohteeksi valikoitui TechVidvan.com-sivuston esittelemä mallipohja (TechVidvan, 2021), jossa keskeiset ominaisuudet kuten OpenCV:n hyödyntäminen live-videon kohteiden tunnistamiseen ja alkeelliseen seuraamiseen siten, että asetetun viivan yli liikkuvat kohteet lasketaan molempiin suuntiin ja erikseen olivat valmiiksi toteutettuna.

Koodia muokattiin käyttötarkoituksen mukaisesti siten, että sovellus luo havainnoistaan tilaston esimerkiksi vuorokauden välein ja tuo tiedot, tilastonhaku- ja etsintäalueen säädön toiminnot sekä live-kuvan Flask-pohjaiseen web-käyttöliittymään. Näin sovellus on helposti käytettävissä yhtenä paikallisena asennuksena. Sovellus on ladattavissa GitHubista (Ylimäki, 2022).

Kävijälaskurin toiminta perustuu peräkkäisistä kuvista tunnistettujen ihmishahmojen seuraamiseen (tracing). Kyseessä on varsin haastava tehtävä, sillä ihmiset eivät kulje turistikohteissa selkeissä muodostelmissa ja suoraviivaisilla reiteillä vaan jäävät usein toistensa (tai rakennuksen osien) peittoon. Tässä pilotissa sovellukseen luotiin erittäin yksinkertainen seuranta-algoritmi, joka perustuu oletukseen siitä, että seurattava kohde löytyy läheltä edellisen kuvan sijaintiaan. Sovellus laskee kulkijoita kahteen suuntaan siten, että kohteen on ylitettävä viiva joka on kuvan suhteen joko vaaka- tai pystysuunnassa ja sijaitsee säädettävän etäisyyden päässä “keskiviivasta” – jonka paikka on myöskin säädettävissä kuvan sisällä.

# Jetson Nano korttikone ajoalustana

YOLOv4 on suhteellisen raskas neuroverkko. Vaikka sovellusta on mahdollista ajaa “normaalilla” suorittimella, suoritus tehostuu huomattavasti, mikäli käytettävissä on rinnakkaislaskentaan kykenevä grafiikkaprosessori. Darknet-kehys tukee Nvidian CUDA-rinnakkaisajoalustaa, joten päätimme kokeilla sovelluksen ajamista Jetson Nano -korttitietokoneella. Nopeuden ohella Jetsonin etu oli korttikoneen kompakti fyysinen koko: 3D-tulostetun suojakuoren ja pienen web-kameran kanssa kokonaisuuden ulkomitat ovat 11 * 8 * 6 cm3 mikä tekee siitä helposti sijoitettavan ja siirreltävän, katso kuva 1.



Kuva 1. Jetson Nano ja LTE-kortti 3d-tulostetussa kuoressa. Laitteeseen on liitetty erillinen USB-kamera sekä LTE-kortin antenni  (kuva: Tommi Ylimäki).

Jetson Nanon Developer kit (Nvidia, i.a.) asennus ja käyttöönotto vaati muutamia iteraatiokierroksia. Nvidian tarjoama JetPack SDK-paketin (Nvidia Jetpack, i.a.) työskentelyhetkellä tarjoama tuorein versio 4.6.1 ei jostain syystä toiminut lainkaan, mutta vanhempi versio 4.6 sai koneen käyntiin ongelmitta. OpenCV:n CUDA-tuettua versiota ei ollut suoraan saatavilla Debian-pakettijakeluna, joten se oli käännettävä lähdekoodista (Q-Engineering, 2022). Tässä auttoi valmis skripti1. Järjestelmän kykyä toipua sovelluksen kaatumisen tai esimerkiksi kameran USB-liittimen irtoamisen kaltaisista virhetilanteista parannettiin tekemällä sovelluksesta automaattikäynnisteinen systemd-palvelu (Archlinux.org, 2022) paitsi laitteen käynnistyessä, myös virhetilanteiden jälkeen.

# Mobiili-internetyhteys ja virtuaalilähiverkko ZeroTier

Jetson Nano -korttikoneessa ei ole sisäänrakennettua WLAN-moduulia. Jotta sovellusta on mahdollista käyttää web-käyttöliittymällä, täytyy käyttäjän koneelta olla vähintään paikallisverkkoyhteys korttikoneelle. Jos korttikone voidaan kytkeä ethernet-kaapelilla sisäverkkoon, sovelluksen käyttöliittymään pääsee selaimella suoraan. Koska LTE-pohjainen internet-yhteys toisi järjestelmän käyttöön sekä riippumattomuuden paikallisverkosta että langattoman sijoittelun vapauden, hankimme erillisen Jetson Nanon kanssa toimivan

Waveshare sim7600g LTE-moduulin (Waveshare.com, i.a.). Asennusohjeiden (Stearns, 2022) avulla internet-yhteys saatiin toimimaan ja Jetsoniin asennettiin ZeroTier-niminen VPN-ohjelmisto (ZeroTier, i.a.), jonka avulla on helppoa luoda virtuaalisia lähiverkkoja eri fyysisissä verkoissa olevien laitteiden välille. Näin syntyi kokonaisuus, jossa korttikone toimii paikallisena web-palvelimena omassa mobiilidata-verkossaan, johon saa yhteyden muissa fyysisissä verkoissa olevista koneista (tai esimerkiksi kännyköistä) joihin on asennettu ja säädetty sama ZeroTier-virtuaalipaikallisverkko.

Itse kävijälaskuri ei käytä internet-yhteyttä, sillä se tallentaa kävijädatan CSV-muodossa korttikoneen muistikortille. Datan lukeminen on kuitenkin huomattavasti helpompaa, jos koneeseen ei tarvitse päästä fyysisesti käsiksi, etenkin jos kone kameroineen asennetaan johonkin vaikeakulkuiseen paikkaan. Tietosuojasyistä kameran ottamia kuvia (videota) ei tallenneta missään muodossa.

# Havainnot ja suositukset

1. Järjestelmää päästiin testaamaan käytännössä pilottiyrityksen pääsisäänkäynnin tuntumassa olevalla terassilla. Järjestelmä itsessään toimi riittävän luotettavasti: kahden kuukauden koekäyttöjakson aikana laitteen internet-yhteys katkesi tuntemattomaksi jääneestä syystä kerran, mutta senkin katkoksen aikana itse kävijälaskuri pyöri moitteitta. OpenCV-kirjaston Darknet-kehys vaikutti aiheuttavan Python-sovelluksen satunnaista, joskin harvinaista, kaatumista, mutta käyttöjärjestelmätasolla toteutettu sovelluksen uudelleenkäynnistys-toiminto palautti sovelluksen ajoon luotettavasti. Päiväkohtainen koontitilasto ja toisaalta kaikki havainnot aikaleimoineen tallentava raakamuotoinen tilasto syntyivät levylle ja ne oli helppo ladata omalle koneelle web-käyttöliittymän kautta.

Aikataulutettu koontiraportti (esimerkiksi kerran vuorokaudessa):

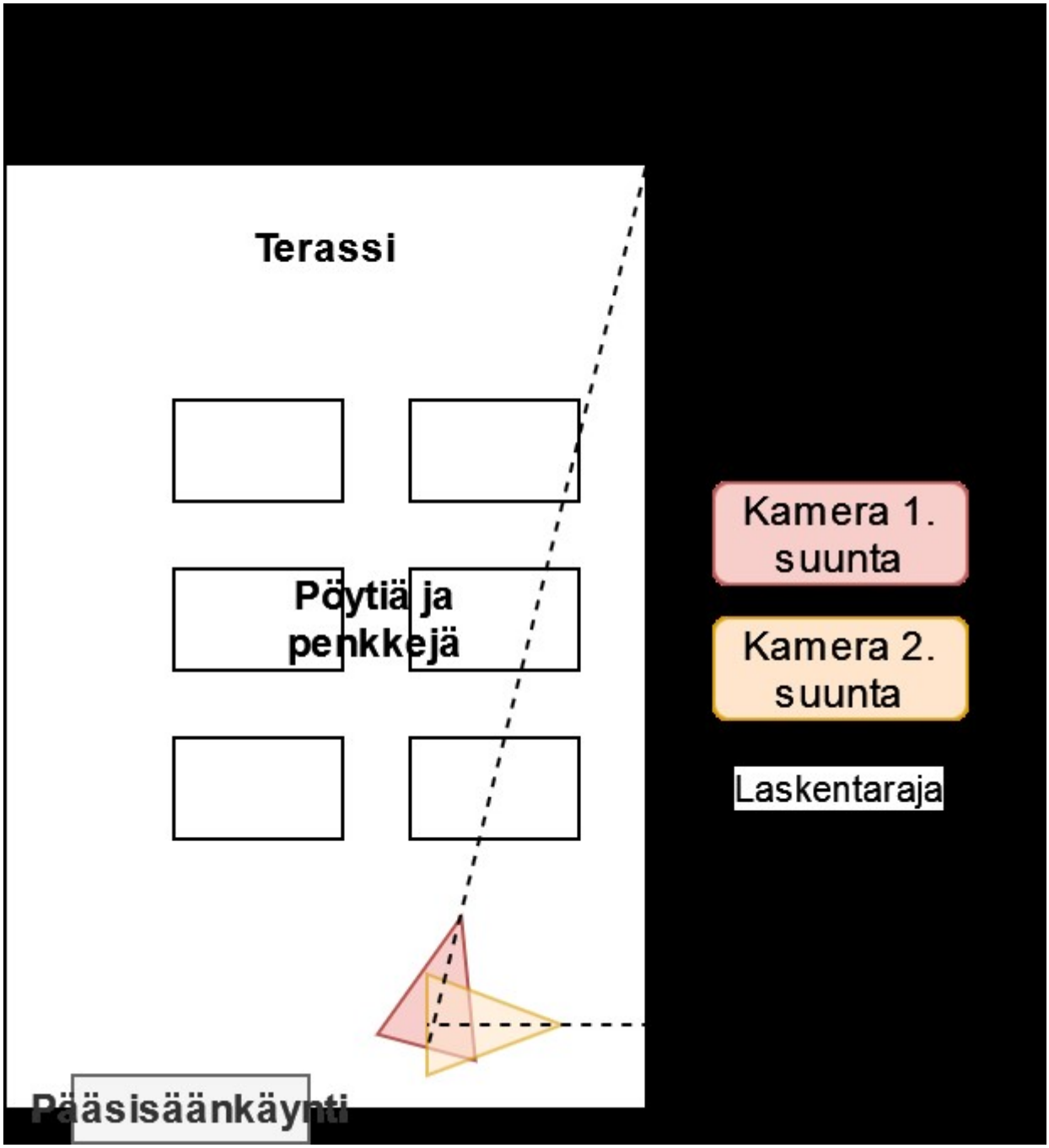
**report time,2022-06-03,17-47-27**
**Direction,person,car,truck**
**Ulos,23,2,1**
**Sisaan,20,3,2**
**2022-06-03,17-58-15,person,Ulos**

Raakatiedot yksittäisistä havainnoista aikaleimoineen:

**2022-06-03,17-55-16,person,Ulos**
**2022-06-03,17-55-19,person,Sisaan**
**2022-06-03,17-55-20,person,Ulos**
**2022-06-03,17-55-23,person,Sisaan**
**2022-06-03,17-56-05,person,Ulos**
**2022-06-03,17-56-07,person,Sisaan**
**2022-06-03,17-56-09,person,Ulos**
**2022-06-03,17-56-48,person,Ulos**
**2022-06-03,17-56-51,person,Sisaan**
**2022-06-03,17-56-53,person,Ulos**
**2022-06-03,17-58-13,person,Sisaan**

Jo sen määritteleminen on vaikeaa, minkä kohdan ylittänyt henkilö lasketaan “sisään” ja taas minkä

“ulos”. Sovelluksen rajoituksista johtuen näiden alueiden raja on kameran kuvaan nähden joko pysty- tai vaakasuora viiva (säädettävällä marginaalilla), mikä johtaa siihen, että kameran on oltava mahdollisimman suorassa kuvattavaan alueeseen nähden. Enimmäkseen käytetyn seuranta-algoritmin alkeellisuudesta johtuen kameran asettelulla ja laskentaviivan sijoittelulla kuvan sisällä on erittäin iso merkitys laskurin tulosten luotettavuudelle. Pilottikohteessa asetimme kameran kuvaamaan ensin pääsisäänkäynnin terassin suuntaan (kuvio 1), mikä aiheutti välillä piikkejä laskuriin: sopivasti laskentaviivan kahdella puolella istuneet henkilöt sekoittuivat peräkkäisten kuvien välillä toinen toisiinsa. Ongelma ratkesi kääntämällä kamera suuntaan, jossa ihmiset liikkuvat mahdollisimman usein suoraviivaisesti eivätkä yleensä pysähtele.



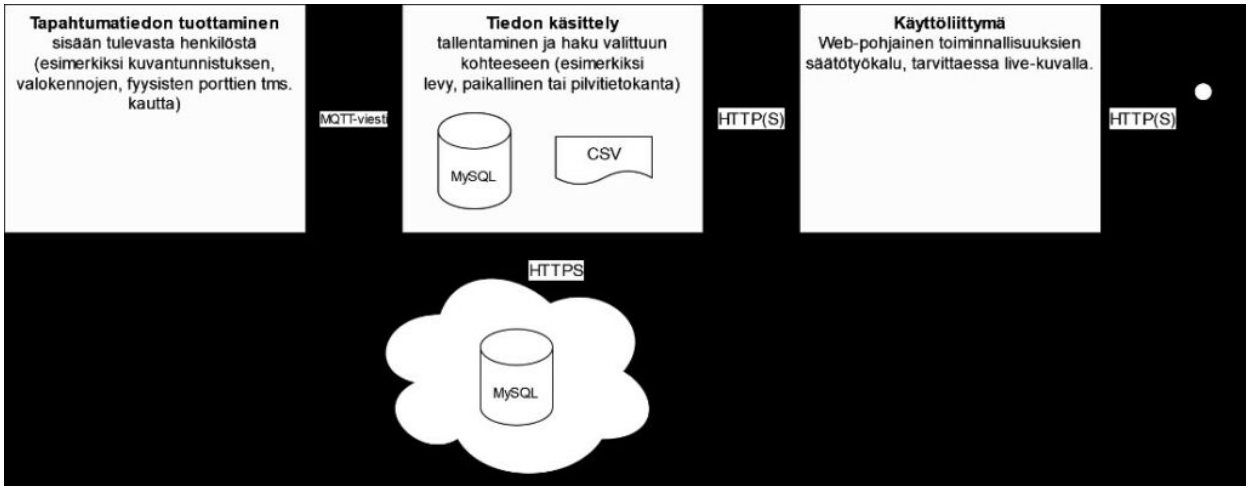

Kuvio 1. Pilottikohteessa kokeillut kameran suunnat.

2. Pilotissa hyödynnetty valmiiksi koulutettu neuroverkko tunnistaa MS COCO-aineiston mukaisesti 80

erilaista asiaa kuten ihmisiä, autoja, kuorma-autoja ja monia eläimiä. Kuitenkaan se ei tunnista ihmistä missä tahansa suunnassa ja asennossa vaan ihmisen tulee olla kuvaan nähden pystyssä. Tästä johtuen kameraa ei voitu asettaa kuvaamaan esimerkiksi suoraan ylhäältä alas, mikä olisi ollut virheellisten laskentojen minimoimisen kannalta hyödyllistä.

3. Käytetty neuroverkko on varsin massiivinen, joten se on toisaalta kiitettävän tarkka ja toisaalta tämän käyttötapauksen kannalta haitallisen hidas. Vaikka Jetson Nanossa on korttikoneeksi suorituskykyinen rinnakkaislaskentasuoritin, YOLOv4:n kevyempää lite-mallia käyttäessäkin videon virkistystaajuudeksi jäi noin 7-8 kuvaa sekunnissa, mikä on kohteen seurannan kannalta hyvin matala luku. Käsitykseksi jäi, että pelkästään ihmisen tunnistamiseen käytettävillä neuroverkoilla suorituskyky voisi olla keveyden vuoksi parempi, mutta pilotin puitteissa asiaan ei päästy perehtymään.

4. Sovellus rakennettiin monoliittisen mallipohjan päälle, kertyneiden kokemusten pohjalta koodille kannattaisi suunnitella eheämpi rakenne. Parannettavaa olisi sekä sovelluksen modulaarisuudessa että kohteenseurannan algoritmissa: algoritmin olisi kyettävä hyödyntämään tietoa seurattavan kohteen liikkeestä ja graafisista ominaisuuksista (kuten vaatteiden väristä). Kuvassa hetkeksi muiden asioiden taakse jäävä ihminen ei tällä algoritmilla todennäköisesti yhdisty oikeaan seurattavaan kohteeseen tullessaan taas esille. Pilotin aikana kehittyi ajatus modulaarisesta arkkitehtuurista, josta karkea havainnekuva kuviossa 2: kokonaisuus olisi järkevää jakaa osiin, joista yksi huolehtii varsinaisesta tiedon tuottamisesta, toinen sen käsittelystä ja kolmas näyttämisestä käyttäjälle.




Kuvio 2. Model-view-controller-mallin mukainen modulaarinen arkkitehtuuri.

5. Käytännön asennuksen kannalta toimivaksi malliksi osoittautui se, että laitteella on (tarvittaessa) langaton verkkoyhteys, mieluusti ilman tarvetta yrityksen paikallisverkkoon liittämistä. Pilotin tapauksessa LTE-pohjainen internetyhteys virtuaalipaikallisverkolla osoittautui toimivaksi ratkaisuksi.

6. Tuotettu sovellus toimii tarvittaessa esimerkiksi kannettavalla tietokoneella eli se on varsin helposti siirrettävissä kunkin yrityksen käytettävissä oleville laitteistoille. Käytännössä suorituskyky jää heikoksi ilman rinnakkaislaskentaan soveltuvia näytönohjaimia, samoin pelkkä kannettavan tietokoneen web-kamera voi olla kuvanlaadultaan ja sijoiteltavuudeltaan liian heikko käytännön tarpeisiin. Kehitystarpeena olisikin käyttötapauksen kannalta YOLOv4:ää tehokkaamman algoritmin valitseminen siten että sovellus pyörisi riittävän jouhevasti heikomman suorituskyvyn laitteissa tai vaikka

älypuhelimissa.

# Johtopäätökset

Pilotissa haluttiin testata “lähes nollakustanteisen” kävijäseurantatyökalun luomista alueen pienille matkailuyrityksille näiden kävijädatan tuottamiseksi. Kehitetty sovellus ja sen ympärillä laajempi järjestelmäkokonaisuus suoriutuu tehtävästä olosuhteista riippuen ainakin välttävästi tai jopa hyvin, mutta verrattuna valmiiksi markkinoilla olevien tuotteiden hinta- ja kyvykkyystasoon kokonaisuus ei ole riittävän kustannustehokas. Pilotissa käytetty korttikone maksaa noin 200 euroa ja mikäli kameran ja nettiliittymän joutuu hankkimaan erikseen, kokonaisuus nousee jo halvimpien valmiiden tuotteiden luokkaan. Toisaalta, jos erillinen web-kamera on jo olemassa ja jokin riittävän tehokas kannettava tietokone sovitettavissa tähän käyttöön, voi kehitettyä sovellusta ajamalla saada kävijälaskurin nollan euron lisäkuluilla.

Artikkeli on valmisteltu osana Datasta ketteryyttä ja uutta liiketoimintaa Etelä-Pohjanmaan pk-yrityksiin (TehoData) -hanketta (Seinäjoen ammattikorkeakoulu (SeAMK), i.a.). Samalla haluamme kiittää hankkeen ja tämän artikkelin rahoittamisesta Keski-Suomen ELY-keskusta. Hanke rahoitetaan REACT-EU-välineen määrärahoista osana unionin covid-19-pandemian johdosta toteuttamia toimia.

Lue lisää hankkeen verkkosivulta.**Tommi Ylimäki**
lehtori
SeAMK, TKI, Digitalisaatio ja älykkäät teknologiat

**Juha-Matti Arola**
projektipäällikkö
SeAMK, TKI, Digitalisaatio ja älykkäät teknologiat

Lähteet:

Archlinux.org. 13.12.2022. Archlinux systemd (Suomi). https://wiki.archlinux.org/title/Systemd_(Suomi)

Nvidia Jetpack. (i.a). Jetpack. https://developer.nvidia.com/embedded/jetpack

Nvidia. (i.a). Jetson Nano Developer Kit. https://developer.nvidia.com/embedded/jetson-nano-developer-kit

Pjreddie. (i.a.). Github Darknet. https://github.com/pjreddie/darknet

Q-Engineering. 13.10.2022. Install OpenCV 4.5 on Jetson Nano.
https://qengineering.eu/install-opencv-4.5-on-jetson-nano.html

Seinäjoen ammattikorkeakoulu (SeAMK). (i.a.). TehoData.
https://projektit.app-staging.seamk.hion.dev/TehoData/

Stearns, P. D. 13.12.2022. Github. https://github.com/phillipdavidstearns/simcom_wwan-setup

TechVidvan. 1.10.2021. TechVidvan.
https://techvidvan.com/tutorials/opencv-vehicle-detection-classification-counting/#

Waveshare.com. (i.a). Sim7600g for Jetson Nano. https://www.waveshare.com/sim7600g-h-4g-for-jetson-nano.htm

Ylimäki, T. 1.6.2022. Vehicle Count. https://github.com/pontta/vehicle_count

Zerotier. (i.a). ZeroTier – Global area networking. https://www.zerotier.com/