Tampereen ammattikorkeakoulu

Koneoppimisen algoritmien tehokkuus kuvantunnistuksessa

Jyri Pappinen

OPINNÄYTETYÖ
Toukokuu 2024

Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

Pappinen Jyri
Koneoppimisen algoritmien tehokkuus kuvantunnistuksessa

Opinnäytetyö 30 sivua, joista liitteitä 2 sivua
Toukokuu 2024

Opinnäytetyössä tutustutaan tekoälyn algoritmeihin, jotka ovat kykeneväisiä kuvantunnistukseen ja toteutetaan vertailu, jossa mitataan tehokkuutta kirjaimien tunnistamisessa.

Algoritmeja, jotka soveltuvat kuvantunnistukseen on monia, tutkimisen jälkeen algoritmeiksi valikoituivat Valintapuu, K-lähin naapuri, Tukivektorikone ja konvoluutiohermoverkko, jotka ovat ennestään tuttuja testaajalla, poissulkien Tukivektorikoneet.

Opinnäytetyön tarkoituksena on lisätä tietoisuutta koneopin algoritmeista ja helpottaa valintaa, kun on tarkoitus löytää tehokas koneopin algoritmi kuvantunnistukseen. Opinnäytetyöhön tutustuttuaan lukija tuntee neljä algoritmia, jotka soveltuvat kuvantunnistukseen ja mihin nämä ovat tietyssä tilanteessa kykeneväisiä. Tämä mahdollistaa oikean algoritmin käytön vastaavanlaisissa tehtävissä.

Asiasanat: tekoäly, kuvantunnistus, algoritmi, valintapuu, k-lähin naapuri, tukivektorikone, konvoluutiohermoverkko ja koneoppi

ABSTRACT

Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software Engineering

Pappinen Jyri
The effectiveness of machine learning algorithms in image recognition

Bachelor's thesis 30 pages, appendices 2 pages
May 2024

The thesis explores artificial intelligence algorithms capable of image recognition and implements a comparison to measure efficiency in recognizing handwritten letters.

There are many algorithms suitable for image recognition, and after research, the selected algorithms are Decision Tree, K-Nearest Neighbors, Support Vector Machine, and Convolutional Neural Network, which are already familiar to the tester, excluding Support Vector Machines.

The purpose of the thesis is to raise awareness of machine learning algorithms and facilitate decision-making when seeking an efficient machine learning algorithm for image recognition. After familiarizing themselves with the thesis, readers will understand four algorithms suitable for image recognition and their capabilities in specific situations. This enables the correct use of the algorithm in similar tasks.

Key words: artificial intelligence, image recognition, decision tree, k-nearest neighbors, support vector machines, convolutional neural networks and machine learning

SISÄLLYS

LYHENTEET JA TERMIT

ML	Machine Learning
DT	Decision tree
KNN	K-nearest neighbors
SVM	Support vector machines
CNN	Convolutional neural networks
KLUSTERI	Ryhmä tai rypäs
KERNEL TRICK	Ydin temppu. SVM:n tekniikka, kun asetettava taso ei ole lineaarinen.
NEUROVERKKO	Luonnollisia hermoverkkoja myötäilevä laskentamalli.
K-ARVO	KNN:n arvo määriteltäessä naapurien etäisyyksiä.

1 JOHDANTO

Koneoppiminen on tekoälyn osa-alue, jolla on tarkoitus saada asia tai ominaisuus toimimaan tehokkaammin, joko käyttäjältä saadusta tiedosta tai pohjatiedosta. Opinnäytetyössä käytetään pohjatietoina kuvia kirjaimista ja ohjattua oppimista. Opinnäytetyöllä tutkitaan onko valituista algoritmeista yksi merkittävästi parempi kuin muut, kun halutaan tunnistaa kirjaimia kuvista. Kuvantunnistukseen soveltuvia algoritmeja on paljon ja tarkoitus oli valita neljä. Algoritmeille tehtiin tutkimusta olisivatko nämä sopivia tunnistamaan kirjaimia kuvista. Valintoihin vaikuttivat myös algoritmin tunteminen entuudestaan, joka helpottaisi testausta. Algoritmeiksi valikoitui Päätöspuu (DT), K-Lähimmät naapurit (KNN), Tukivektorikone (SVM) ja Konvoluutiohermoverkko (CNN).

Saatuja tuloksia voidaan hyödyntää valittaessa kuvantunnistus algoritmia projekteihin, joissa on tehtävänä tunnistaa kirjaimia kuvista. Mallia on mahdollista kehittää eteenpäin tunnistamaan esimerkiksi rekisterikilpiä.

Opinnäytetyössä tutustutaan koneoppimisen algoritmeihin ja tutkitaan testin avulla tehokkuutta kuvantunnistuksessa. Algoritmeista jokaisella on omat vahvuutensa koneopin saralla ja algoritmit valikoituivat käytettäväksi, koska kyseiset algoritmit ovat soveltuvia myös kuvasta tunnistamiseen. Lopuksi esitellään lopputulokset ja pohditaan saavutettuja tuloksia.

2 TEKOÄLY

Tässä luvussa avataan tekoälyä yleisellä tasolla, mutta keskittyen opinnäytetyön aiheeseen. Tekoäly tulee englannin kielen sanoista Artificial Intelligence ja lyhennettynä AI.

Tekoäly on paljon käytetty termi erilaisille tietokonejärjestelmille, tämän takia tarkempi määrittely on hankalampaa, kunhan laite väitetysti hyödyntää jotain älykkääksi toiminnaksi kuvailtavaa toimintoa. Yleensä kuitenkin tekoäly viittauksella tarkoitetaan koneita, jotka pystyvät ihmisen kaltaisen toimintaan. (Boucher, 2020)

2.1 Tekoälyn historia

Turingin testi etsii vastausta kysymykseen ”voiko kone ajatella?”. Turingin testin ideana on, että jos ihminen keskustelee kahden muun kanssa joista, toinen on kone ja toinen on ihminen, eikä ihminen pysty keskusteluiden perusteella päättämään kumpi on kumpi, voidaan väittää, että kone ajattelee. (Romula, 2000)

Vuonna 1955 neljä tietekniikka- ja kognititiotieteen asiantuntijaa Claude E. Shannon, Marvin L. Minsky, Nathaniel Rochester ja John McCarthy järjestivät Dartmouth Summer Research Project on Artificial Intelligence -työpajan. Tämä työpaja on usein mainittu tekoäly-termin ensimmäisenä esiintymisenä. Tarkoituksena oli luoda perusta tekoälytutkimukselle ja määritellä, kuinka älykästä käyttäytymistä voitaisiin ohjelmoida koneisiin. (Veisdal, 2019)

2.2 Koneoppi

Koneoppiminen on tekoälyn yksi haara, jossa keskitytään opettamaan konetta tulkitsemaan kuvioita ja ratkomaan ongelmia. Koneoppi pitää sisällään erilaisia algoritmisia tekniikoita hyödyntäviä koneoppimisen tyyppejä. On olemassa neljä koneoppimismallia: valvottu oppiminen, valvomaton, puolivalvottu ja vahvistus. (Klusaitė, 2023)

2.2.1 Valvottu oppinen

Valvottu oppiminen tai ohjattu oppiminen rinnastetaan ihmisen oppimiseen. Oppimismallissa konetta koulutetaan määrätyillä säännöillä ja datakokonaisuuksilla. Oppimismallit sisältävät tietopareja joista toinen on syöttötietoa ja toinen on tulostietoa, joista jälkimmäinen sisältää halutun arvon tai lopputuloksen. (Klusaitė, 2023)

2.2.2 Valvomaton oppiminen

Valvomaton oppimien eroavaisuus valvottuun oppimiseen on, että oppimismalleissa ei ole vastaus avaimia, vaan kone tutkii syöttötietoja ja pyrkii tunnistamaan kuvioita ja korrelaatioita. Valvomatonta oppimista mallinnetaan monesti sen perusteella, kuinka ihmiset tarkkailevat maailmaa. (Klusaitė, 2023)

2.2.3 Puolivalvottu oppiminen

Puolivalvottu oppiminen on kahden aikaisemmin mainitun mallin yhdistelmä, jossa datan joukkoon, jossa ei tulostietoja, lisätään pieni määrä tietopareja, joissa on syöttötieto ja tulostieto. Tällöin oppimisnopeus ja tarkkuus paranevat. (Klusaitė, 2023)

2.2.4 Vahvistus oppiminen

Vahvistetussa oppimisessa oppiminen tapahtuu vuorovaikutteisessa järjestelmässä kokemuksen ja virheen kautta. Vahvistavassa oppimismallissa määritellään sallitut säännöt ja mahdolliset lopputulokset. Vahvistettu oppiminen muistuttaa peliä, jossa koneen täytyy itse löytää ratkaisu. (Klusaitė, 2023)

2.3 Algoritmi

Algoritmi on yksityiskohtainen kuvaus tai ohje, kuinka tehtävä tulee suorittaa tai tehdä. Modernissa kielessä usein algoritmeilla viitataan oppiviin algoritmeihin. (Lampinen, 2022)

2.4 Konenäkö

Konenäkö tuottaa kuvaan pohjautuvaa uutta tietoa, jonka esitetään typistetyssä muodossa. Yksinkertaisimmillaan konenäöllä tehdään kyllä/ei-tulkinta kuvan sisällöstä. Monimutkaisempi tulkinta on kuvan kategorisointi. Oikein ilmaisten kuvasta tehty johtopäätös on aina kokoelma lukuja. (Sandelin, 2020)

2.4.1 Kuvantunnistus

Kuvantunnistus on tekoälyn typpi, jossa ohjelmisto on kykeneväinen tunnistamaan kuvista esimerkiksi paikkoja, esineitä, eläimiä, ihmisiä tai tekstiä. Kuvantunnistus koostuu neljästä päätekniikasta, joita ovat luokittelu, merkintä, objektin tunnistus ja segmentointi. Luokittelussa pyritään tunnistamaan luokka, johon tietty kuva sopii. Merkinnässä pyritään merkitsemään objekteja kuvista, joita voi olla samassa kuvassa useampia. Objektin tunnistuksessa pyritään tunnistamaan asia tai esine kuvasta, jonka jälkeen tunnistus tarkentuu tiettyyn objektiin kuvasta. Segmentoinnissa kuvan yksittäinen elementti pyritään lokalisoimaan tarkimpaan pikseliin asti. (Meltwater, 2022)

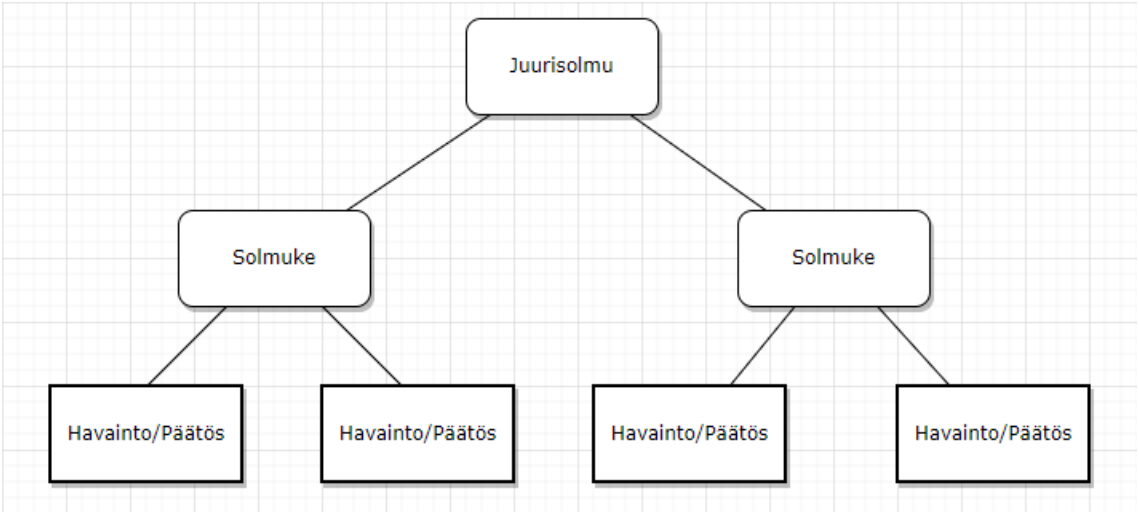
3 ALGORITMIT

Algoritmien valinnassa on kiinnitettävä huomiota algoritmien toimintatapoihin, jotta tarjolla olisi testaamisen kannalta erilaisia ja näin ollen eroavaisuuksia syntyisi. Tässä luvussa on tarkoitus käydä hiukan syvemmin läpi algoritmeja ja kertoa näiden toiminnoista, luokittelu kyvyistä ja tavoista oppia, joilla voidaan mahdollisesti myös selittää saatuja tuloksia.

Algoritmien valinnassa hyödynnettiin keskutelubottia, jonka suosituksista algoritmeja aloitettiin tukimaan mahdollisina kandidaatteina. Onko algoritmi oikeasti soveltuva kuvantunnistamiseen, onko algoritmi sopiva tunnistamaan kirjaimia kuvista, olivat kysymyksiä, joiden perusteella algoritmejä tarkasteltiin. Valintoihin vaikutti myös se, että osa algoritmeista oli testaajalle entuudestaan tuttuja tämä helpottaa algoritmin käyttöä.

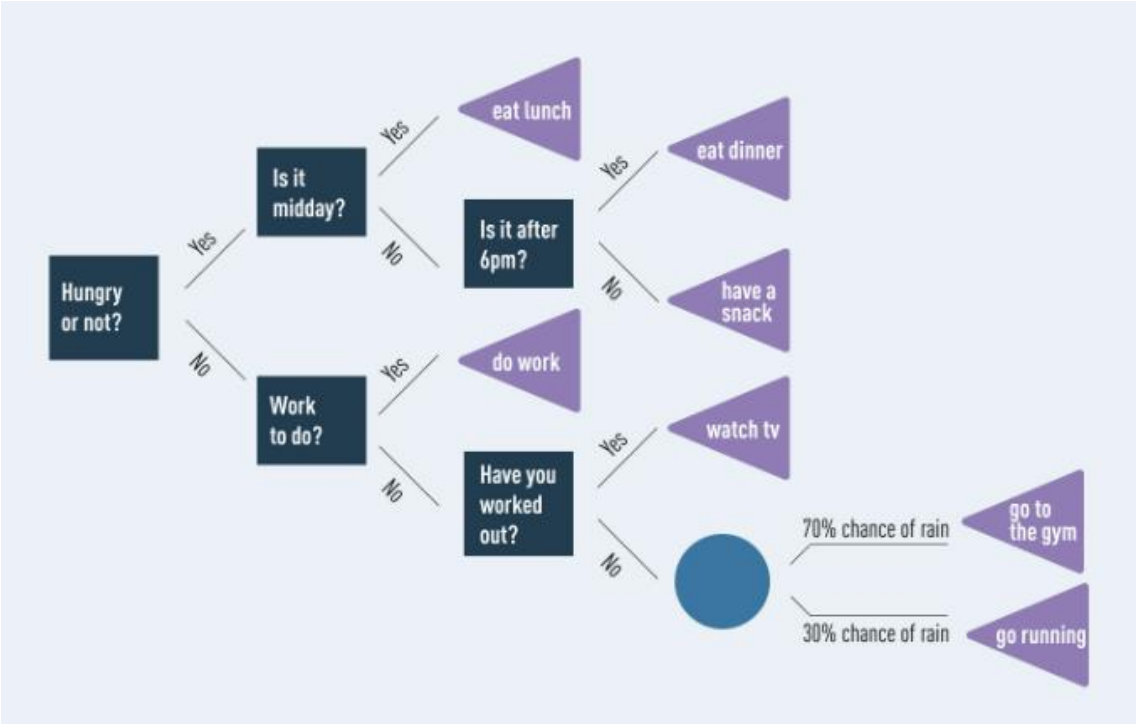
3.1 Päätöspuu (DT)

Päätöspuu on ohjatun oppimisen algoritmi, joka soveltuu sekä regressio- että luokittelutehtäviin. Päätöspuu luo päätöshaaroja ja näiden avulla pyrkii jakamaan dataa oikeisiin lohkoihin. Kuva 1 esittää miten päätöspuu oppii ja kategorisoi dataa.

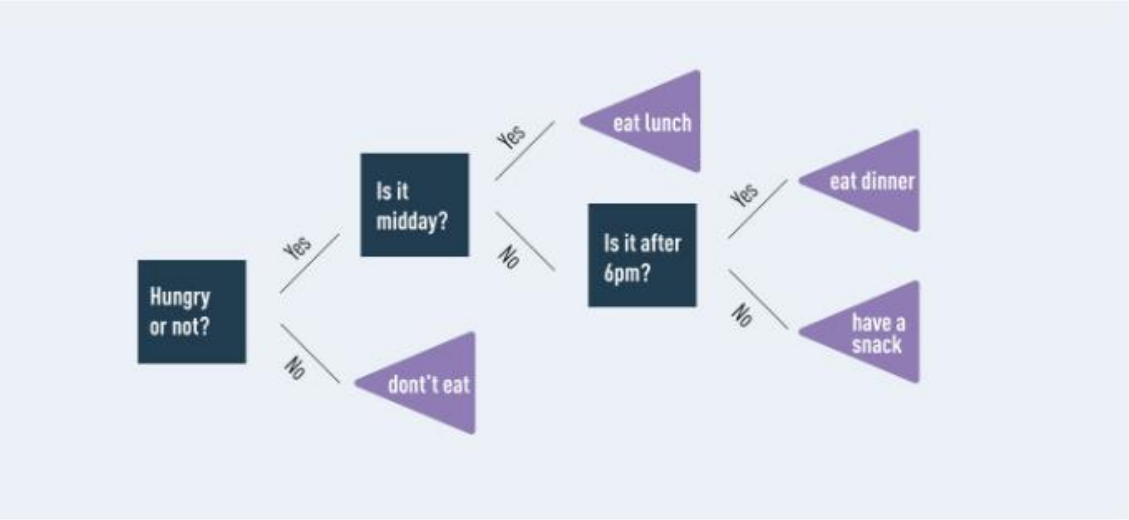


KUVA 1. Päätöspuu. Päätöspuun kuvaaja yksinkertaistettuna.

Päätöspuun suurimpia etuja on sen helppo ymmärrettävyys ja visualisointi. Päätöspuun opetusdata ei myöskään vaadi kovin paljoa datan esivalmistelua, mikä helpottaa ja nopeuttaa prosessia. Päätöspuun heikkouksia on, että puusta saattaa tulla liian kompleksinen, kun datapisteillä on paljon ominaisuuksia, jotenka luokkien ja datan määrä on oltava oikeassa suhteessa. Valintapuun haittapuolena voi myös mainita oppimisen puolueellisuuden, jos datassa on luokkia, jotka hallitsevat enemmistöä koko datapaketista. Käytetyssä datajoukossa jokaista luokkaa kohden on yhtä monta datapistettä. Kuva 2 esittää käytännön läheisempää mallia päätöspuun toiminnasta. Kuvia 2 ja 3 vertaillessa on huomattava ero. Kuva 2 edustaa dataa, jota ei ole esikäsitelty, kun datalla on paljon ominaisuuksia, tulee päätöspuustakin hyvin nopeasti monimutkainen. Tällöin on mahdollisuus, että keskitytään alkuongelman kannalta epäolennaiseen tietoon ja opetusaika kasvaa ja lopputulos ei ole halutun kaltainen. Kuva 3 on selkeämpi kuvaus tilanteesta, koska data on esikäsitelty ja näin on myös pysytty alkuperäisen ongelman ratkaisussa.



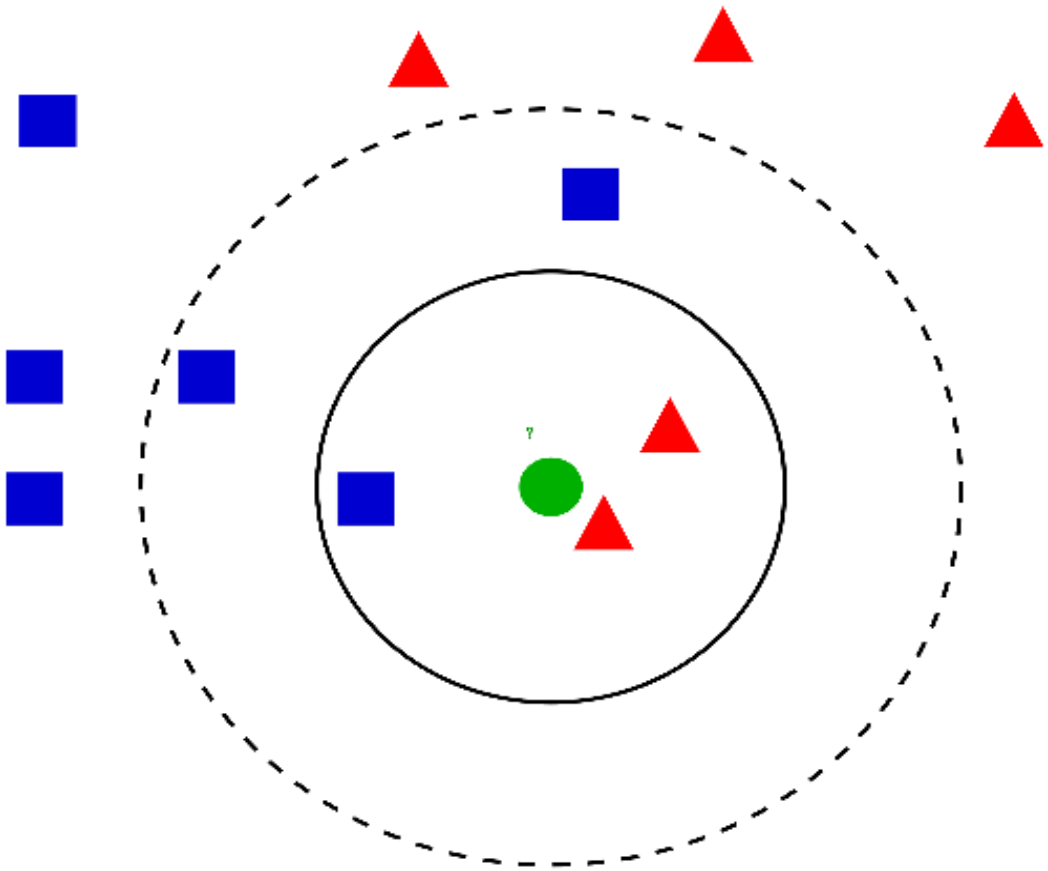
KUVA 2. Päätöspuun suurella määrällä ominaisuuksia (Career Foundry)



KUVA 3. Päätöspuun esimerkki siistitymmällä datalla. (Career Foundry)

3.2 K-Lähin naapuri (KNN)

KNN:n on päätöspuiden tapaan myös ohjatun oppimisen algoritmi, jonka vahvuudet ovat regressio- ja luokittelutehtävissä.



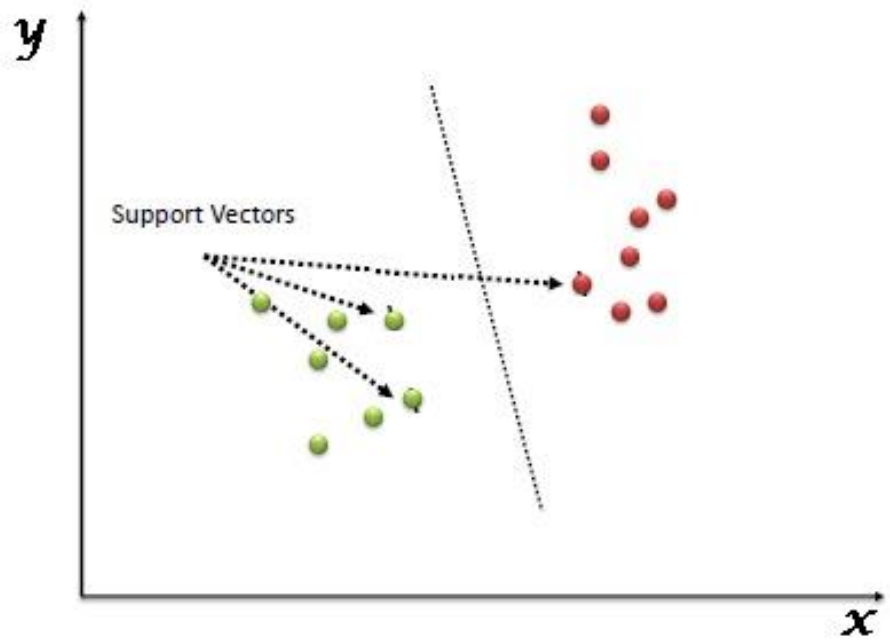
KUVA 4. K-Lähin naapurin yksinkertainen luonnos (Analytics Vidhya)

KNN:n tapa on jakaa datapisteitä ryhmiin, joita kutsutaan klustereiksi. Jakaminen tapahtuu mielivaltaisesti algoritmin mielestä parhaimmalla tavalla. Kun uusi datapiste annetaan, pyrkii malli tutkimaan datapisteen ympäristöä ja määrittämään mihin klusteriin datapiste kuuluu. Mallia luodessa on hyvin tärkeää, että K-arvo on määritetty oikein, koska liian pieni K-arvo voi ryhmien rajoilla tarkoittaa, että datapiste päätyy väärään klusteriin, kun lähimmät oikeat naapurit olisivat toisessa dataryhmässä. Toisaalta liian suuri K-arvo aiheuttaa, että naapureita on liian monta ja näin ollen tuloksien tarkkuus heikentyy.

KNN:n etuja ovat yksinkertaisuus, skaalattavuus ja tehokkuus suurillakin datapaketeilla. KNN toimii myös hyvin epälineaarisella datalla, koska KNN:lla ei ole oletuksia datan suhteen. KNN:n haittapuolia on mallin hidas ennustusnopeus suuremmilla datapaketeilla. Toinen heikkopuoli on, että KNN on ”laiska oppija”, koska se ei opettele datasta mitään kaavoja tai kuvioita, vaan pyrkii muistamaan datan ja tämän avulla luokittelee uuden datapisteen. KNN:lla on myös hankaluuksia käsitellä useampiulotteista dataa.

3.3 Tukivektorikone (SVM)

SVM on myös ohjatun oppimisen algoritmi, jonka vahvuudet ovat luokittelussa ja kuvioiden tunnistamisessa.



KUVA 5. Yksinkertainen kuva Tukiverkkokoneesta (Analytics Vidhya)

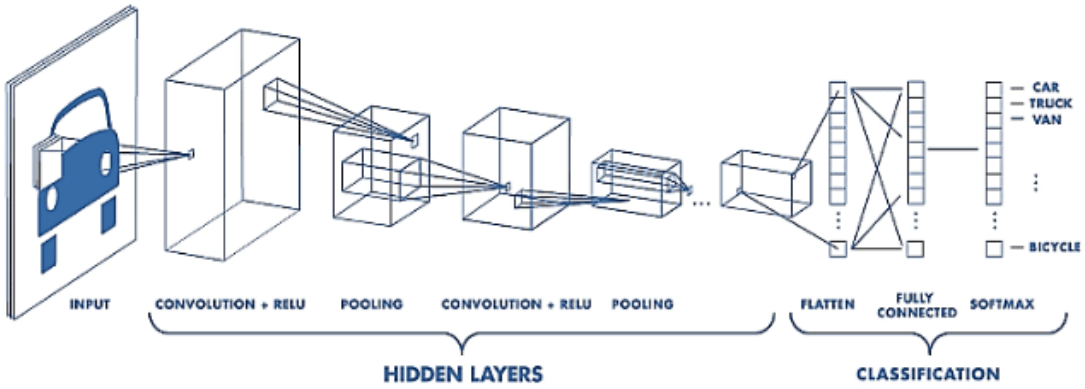
SVM:n sovittaa datajoukkojen väliin tason niin että tukivektorien mukaan asetettuihin marginaalitasoihin välimatka on yhtä suuri, eikä marginaalitasojen välissä ole yhtäkään datapistettä, kuten kuva 5 näyttää. Todellisuudessa datajoukko on harvoin täysin lineaarisesti erotettava, tällöin tukivektorikone käyttää joko pehmeää marginaalia tai ydin temppua *(kernel trick).* Pehmeän marginaalin tapauksessa taso on epälineaarinen ja tämä hyväksyy muutamien väärin luokiteltujen datapisteiden olemisen väärissä klustereissa. Tällöin pyritään tasapainottamaan tarkkuus ja marginaali, niin että virheitä olisi mahdollisimman vähän ja marginaali mahdollisimman suuri. Kernel trick puolestaan käyttää joukon ominaisuuksia. Ydin temppu muuntaa data joukon korkeampaan ulottuvuuteen ja pyrkii tällöin löytämään lineaarisen tason.

SVM:n toimii hyvin, kun datapisteiden väleillä on selviä eroja ja käytössä on moni ulotteisia tiloja. SVM:n ei ole hyvä isojen datapakettien kanssa. Kun SVM algoritmia käytetään suurten datapakettien kanssa, kasvaa koulutusaika merkittävästi. Myös optimaalisten asetuksien löytämien voi olla hankalaa.

3.4 Konvoluutiohermoverkko (CNN)

Hermoverkkoa lyhyesti määriteltäessä voisi kuvailla ihmisen aivojen inspiroimaksi laskennalliseksi malliksi. Konvoluutiohermoverkko on neuroverkkojen alakategoria, johonka kuuluu konvoluutiokerros.

Konvoluutiokerroksen avulla pystytään tunnistamaan kuvista ominaisuuksia, kuten kirkkaus, väri tai kuvioita. Konvoluutiohermoverkko on suunniteltu käsittelemään dataa, joka on taulukkomaista. (Goodfellow, Bengio ja Courville 2016).



KUVA 6. Konvoluutiohermoverkko, CNN (Achitecture of CNN)

Konvoluution tarjoama taso verrattuna normaaliin neuroverkkoon tulee hyödylliseksi kuvista tunnistamisessa, kun konvoluution opetusdatassa on ainakin yksi kuva, jossa on haluttu kohde esimerkiksi auto, jossakin kohtaa kuvaa. Normaalien neuroverkkojen kohdalla tilanne on erilainen ja jotta malli onnistuisi tunnistamaan kuvan, jossa auto on keskellä, on opetusdatassa oltava kuva, jossa auto on juuri keskellä ja samankokoisena. Testaamiseen käytettävä data on kokoelma yksinkertaisia kirjaimia, jotenka CNN:n pitäisi pystyä suoriutuman testeistä kiitettävästi.

Konvoluutiohermoverkkojen etuja on, että ne pystyvät tunnistustehtävissä saavuttamaan hyvin suuren tarkkuuden ja kestää hyvin alkudatan vääristymiä, joten datan esikäsittelyn tarve vähenee. Konvoluutioverkkojen heikkouksia on tarve suurelle laskentateholle, jolloin järjestelmä, jolla mallia opetetaan pitää olla huomattavasti tehokkaampi muiden mallien tarpeeseen verrattuna. Konvoluutioverkot vaativat myös paljon opetusdataa, jotta voidaan saavuttaa suuri tarkkuus. Tämä voi olla tietyissä tapauksissa hankalaa saada tai tuottaa. Heikkoutena mainitsemisen arvoista on, että käyttäjällä voi tulla helposti vaikeuksia valita oikea määrä kerroksia tai kerroksien tiheyksiä hermoverkkoon.

4 OPETTAMINEN JA TESTAAMINEN

Tässä luvussa kerrotaan miten testaamisella pyritään selvittämään mallien toiminta ja tehokkuus kuvantunnistamisessa eri kokoisilla datapaketeilla. Tarkkailun kohteina on mallin rakennuksen sujuminen, mallin opetuksen kesto ja mallin saavuttama tarkkuus.

4.1 Mallin rakennus

Algoritmien testaaminen aloitettiin tuottamalla jokaisesta algoritmista toimiva koneopin malli. Tämä mahdollistaa kriteerien määrittämisen toimivalle mallille. Kriteerinä on, että opetettu malli on kykeneväinen tunnistamaan omasta opetusdatastaan satunnaisesti poimitut kirjaimet ja mallin opetus kestää alle vuorokauden. Opetusdatan kuvat ovat kooltaan 32x32 pikseliä ja kuvat pidetään alkuperäisen kokoisina, jotta opetusaika ei kasvaisi kohtuuttoman pitkäksi, jota kuvien muuntaminen tuottaa. Kuville tehtiin värin muutos. Kuvat olivat alun perin mustalla taustalla olevia valkoisia kirjaimia, nämä värit käännettiin toisin päin koska tunnistetavat kirjaimet ovat valkoisella taustalla olevia mustia kirjaimia. Lopuksi data muunnetaan moniulotteisesta yksiulotteiseksi, jotta voidaan mahdollistaa suurien datapakettien käyttäminen säästämällä muistia. Jokaisesta mallista tulostuu lopuksi tarkkuus ja kulunut aika. Kulunut aika oletettavasti kasvaa, kun datamäärä kasvaa ja tarkkuus kasvaa, koska opettamiseen on enemmän aineistoa käytettävissä. Mallien rakentamisessa hyödynnettiin tekoälyä esimerkiksi datan käsittelyssä ja valmistelussa, mutta kaikki asetukset algoritmeille testaaja tuotti itse.

4.2 Testauksen vaiheet

Testaaminen ja mallin opetus toteutetaan kolmessa vaiheessa. Jokaisen vaiheen opetusmateriaali on leikattu samasta materiaalista, joka sisältää noin 390 000 kuvaa kirjaimista, jotka ovat isoja ja pieniä. Opetusmateriaalin kasvaessa, myös materiaalin likaisuus kasvaa. Tällöin on myös mahdollista testata kuinka malli toimii silmin havaittavien ”virheellisten” kuvien kanssa ja kuinka tämä likaisuus vaikuttaa lopputulokseen. Ensimmäinen vaihe on opettaa malli 780 kuvalla, joka on 30 kuvaa per luokka. Toisessa vaiheessa malli opetetaan 26 000 kuvalla, jolloin

jokaiselle luokalle on 1000 kuvaa. Kolmannessa vaiheessa kuvien määrä on kasvanut 130 000 kuvaan, jolloin jokaisessa luokassa on 5000 kuvaa. Jokaisessa vaiheessa malli testataan samalla testausmateriaalilla, jotta pystytään havainnoimaan kehitys ja kuinka hyvin malli on kykeneväinen suoriutuman rajoitetulla opetusmateriaalilla. Mitä suurempia kuvapaketit ovat, sitä enemmän mallin toteuttavalta koneelta vaaditaan tehoa. Ainoa ero ensimmäisen vaiheen ja muiden vaiheiden välillä on jakosuhde. Ensimmäisen vaiheessa data jaetaan 90/10 suhteella opetus- ja testausdataan, muissa vaiheissa käytetään 80/20 suhdetta. Tämä koska muuten ensimmäisessä vaiheessa opetusdata jäisi pieneksi.

4.3 Testausmateriaali

Mallin testauksessa käytetyt kirjaimet ovat käsin kirjoitettuja ja tämän jälkeen skannattuja. Mallin testauksessa käytettyjä kirjaimia ei ole käytetty mallin opettamisessa, jotta on pystytty havainnoimaan tilanne, jossa käytetty materiaali ei ole entuudestaan tunnettu. Kuvassa 7 on kaksi versiota kirjaimesta A. Vasemmanpuoleinen on opetusdatasta ja oikeanpuoleinen on testausdatasta. Kuvasta 7 voi huomata, että vasemmanpuoleinen kirjain on paljon virallisemman näköinen, kuin taas testaamiseen tarkoitettu käsin kirjoitettu kirjain, joka on paljon ”luontaisemman” näköinen. Tämä tuottaa mallille haasteita, joka ei varsinaisesti näe kirjainta, vaan kuva muuntuu numeroiksi ja tällöin kaksi silmin havaittavaa samaa kirjainta voivat olla aivan erilaisia numeraalisesti tarkasteltuna.

KUVA 7. Esimerkkikuvat opetus- ja testausdatasta.

Testauksessa on käytössä jokaisesta 26 englannin kielen kirjaimesta käsin kirjoitettu malli. Suomen kielessä kirjaimia on 29, mutta opetusmateriaalin löytämisen helpottamiseksi malleilla käytetään englannin kielisiä kirjaimia. Kuvasta 8 voi huomata, että jokainen kirjain on silmin ymmärrettävä, mutta opetusdatassa kirjaimet ovat tarkkoja esimerkiksi symmetrisyys ja koko. Käsin kirjoitetuissa kirjaimissa on enemmän ”virheitä”. Kirjaimet eivät ole symmetrisiä, mittasuhteet eivät ole tasaisia ja kirjaimen paksuus voi vaihdella.

4.4 Mallin testaus

Testaus on hyvin suoraviivaista. Ensimmäiseksi malli lataa testattavat kuvat, jotka tässä tapauksessa ovat kuvasta 8 irrotetut 26 isoa englannin kielen kirjainta. Suomen kielessä on 29 kirjainta. Testauksessa käytetään englannin kielen aakkosia, koska opetusmateriaalin löytäminen on helpompaa. Tämän jälkeen malli muuntaa kuvat samankokoisiksi kuin opetuksessa käytetyt kuvat ovat. Kun kuvat on käsitelty malli alkaa tuottamaan ennustuksia. Ohjelma tulostaa ensin kirjaimen, joka oikeasti on kyseessä. Oikea kirjain saadaan kuvan nimestä, koska kuvat on nimetty oikealla kirjaimella. Oikean kirjaimen viereen ohjelma tulostaa mallin ennustaman kirjaimen, jos nämä kaksi kirjainta ovat samat lisätään listaan yksi oikein, koska näiden oikein merkittyjen avulla saa helposti laskettua oikein ennustettujen määrän. Lopuksi malli tulostaa prosentti määrän oikein ennustetuista kirjaimista ja ennustukseen kuluneen ajan.

KUVA 8. Kaikki testidatan kirjaimet.

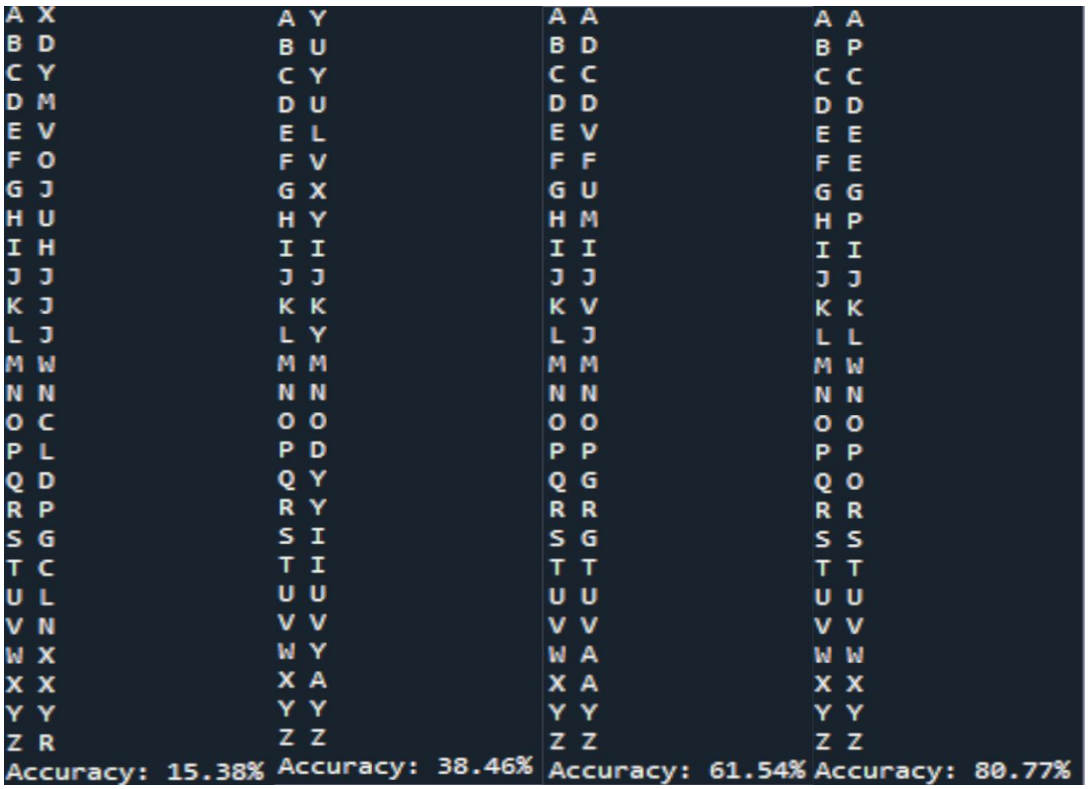
Kyseisessä testauksessa mallit opetetaan käyttäen kahdeksanytimistä AMD Ryzen 7 5800X prosessoria ja keskusmuistia käytettävissä oli 32 gigatavua.

5 TULOKSET

Luvussa on tarkoitus käsitellä saadut tulokset luokkakohtaisesti. Tuloksista on esitelty taulukkomuodossa mallin tarkkuus, testaukset opetusdatalla sekä käsin kirjoitetuilla, kuinka kauan mallin opetus kesti ja kuinka kauan mallin ennustus kesti. Kuvista 9, 10 ja 11 pystyy havainnoimaan mitkä kirjaimet mallit saivat oikein ja mitkä väärin.

5.1 Pieni kuvamäärä

Ensimmäisessä testauksessa pienellä datamäärällä oli hyvin paljon vaihtelua. Kuten kuvasta 9 voi huomata, että vaihtelu tapahtuu reilusta 15 prosentista vajaaseen 81 prosenttiin. Vahvimmin tehtävästä suoriutui konvoluutiohermoverkko, joka niukalla materiaalilla onnistui saamaan oikein 21 kappaletta, seuraavaksi tuli tukivektorikoneet, joka sai oikein 16 kappaletta, toiseksi viimeiseksi tuli K-lähin naapuri, joka onnistui saamaan oikein 10 kappaletta ja viimeisenä tuli päätöspuu, joka sai vain neljä kappaletta oikein.



KUVA 9. Mallien tulokset käsin kirjoitetuilla kirjaimilla. Vasemmalta oikealle DT, KNN, SVM ja CNN

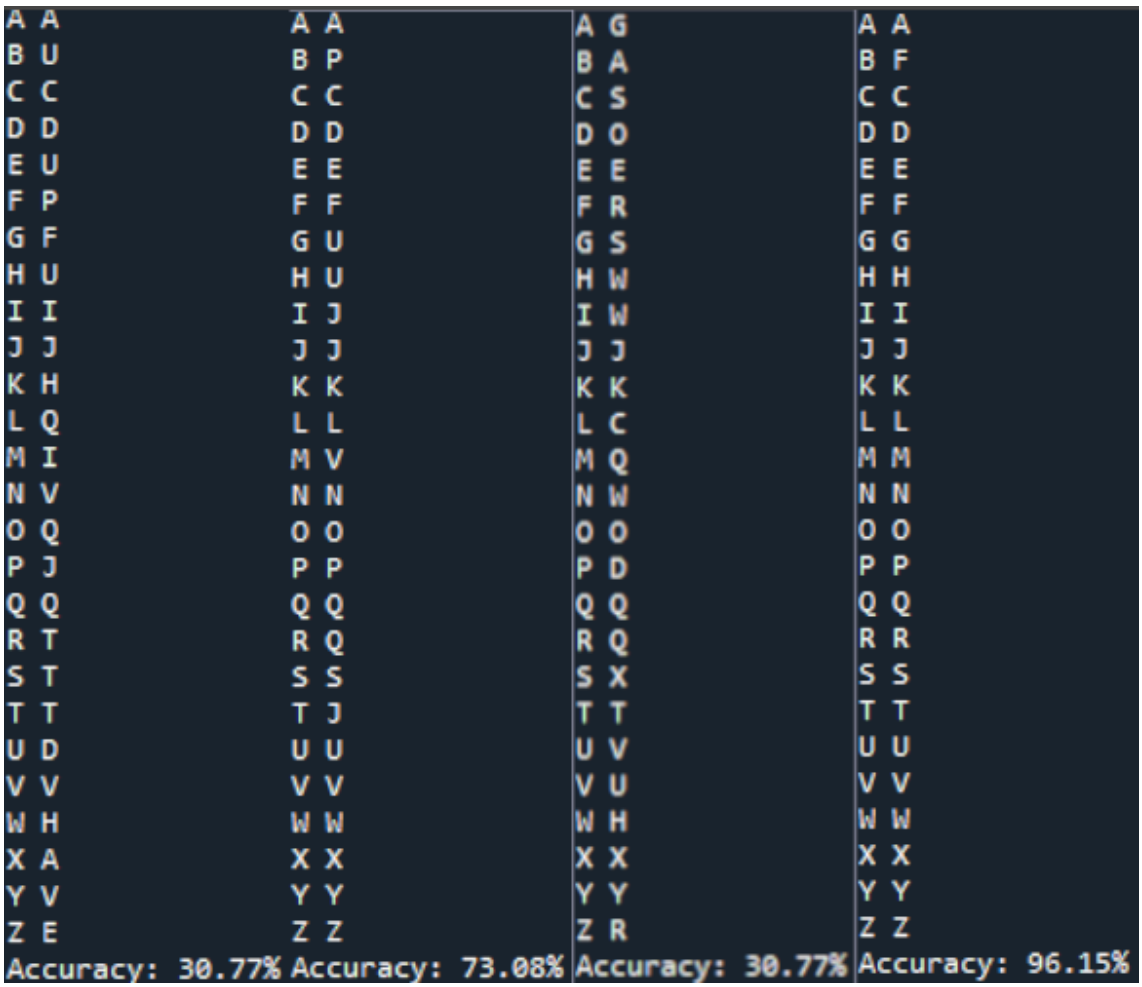
Taulukossa 1 on vielä merkattu tarkempia asioita malleista. Suurimpia eroja mallien välillä on opetusajoissa, vaikka K-lähin naapuri on nopein sisäistämään datan on ennustaminen hitaampaa. Käytännön tasolla eroa ei huomaa näin pienillä datapaketeilla, mutta isompiin paketteihin siirryttäessä erot kasvavat. Tukivektorin suoritus on myös mielenkiintoinen, vaikka testauksessa eroa konvoluutiohermoverkkoihin on 19,23 % on mallin opetuksessa saatu tarkkuus on sama kuin konvoluutiohermoverkolla, mutta opetusaika on vain murto-osa verrattuna konvoluutiohermoverkon opetusaikaan. Päätöspuun heikkoa tulosta voi selittää datan pieni määrä suhteessa luokkien määrään.

Nimi	Tarkkuus	Testaus opetusdatalla	Opetusaika (s)	Testaus käsin kirjoitetuilla	Testausaika (s)
DT	50,00 %	100 %	0.573	15,38 %	0.008
KNN	52,56 %	100 %	0.013	38,46 %	0.150
SVM	73,72 %	100 %	0.240	61,54 %	0.017
CNN	73,72 %	100 %	8.286	80,77 %	1.660

TAULUKKO 1. Mittaustulokset pienellä datamäärällä

5.2 Keskisuuri kuvamäärä

Toisessa testauksessa datamäärä oli kasvanut 26 000 kuvaan ja vaikutukset alkoivat jo näkyä tuloksissa. Mittaamisen huomio kiinnittyi jo algoritmien esittelyssä mainittuun SVM:n mahdolliseen heikkoon suoritukseen suuremmalla datapaketilla. Tukivektorikoneiden tulos tippui alle puoleen ensimmäisestä testistä, joka on sama tulos päätöspuun kanssa ja opetusaika kasvoi yli kolminkertaiseksi seuraavaksi hitaimpaan verrattuna. Mutta myös merkittäviä parannuksia oli joukossa. Esimerkiksi päätöspuut tuplasi tuloksensa, tätä voidaan selittää datamäärän merkittävällä kasvulla, mutta luokkien määrän pysymistä samana. Datamäärän kasvaminen lisää datan likaisuutta, joka hankaloittaa algoritmien toimintaa.



KUVA 10 Mallien tulokset käsin kirjoitetuilla kirjaimilla. Vasemmalta oikealle DT, KNN, SVM ja CNN

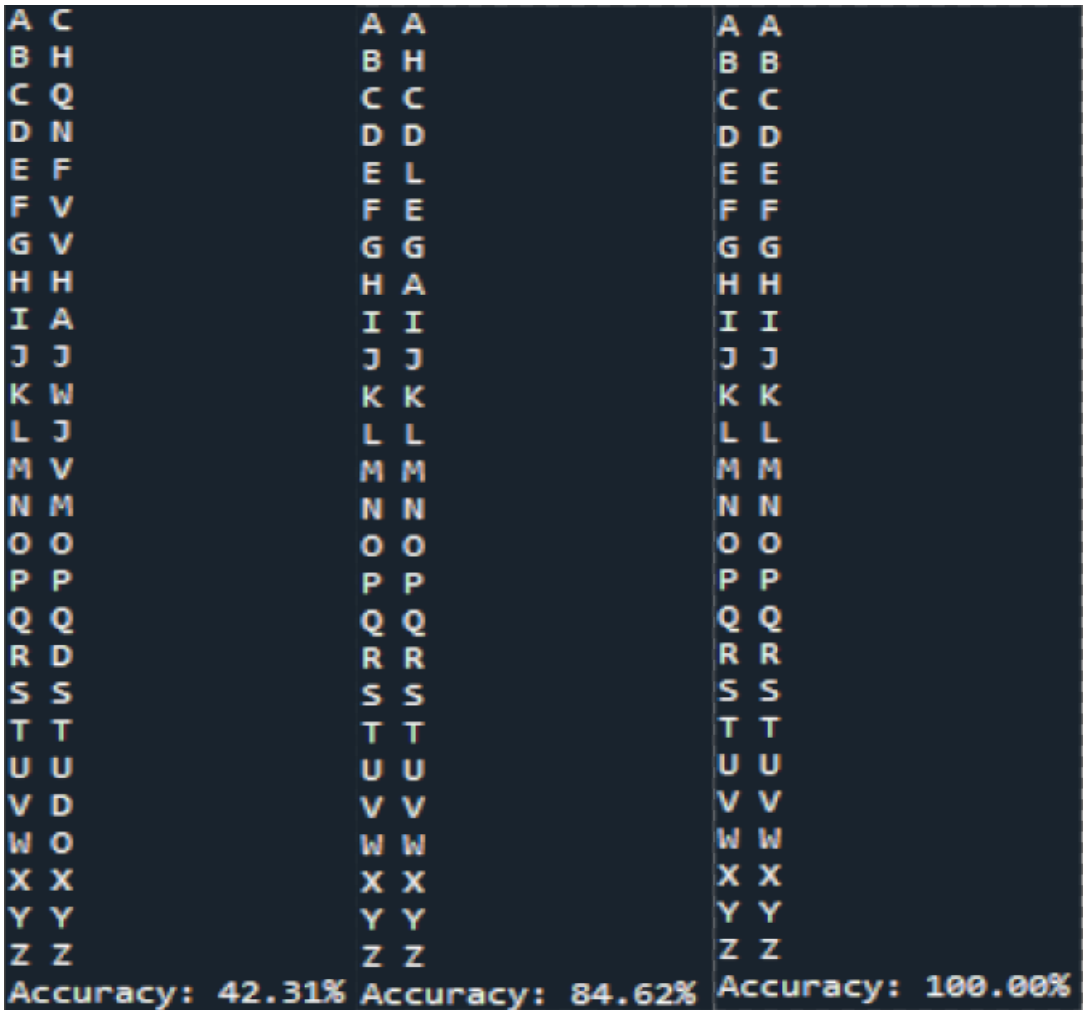
Testaukseen kuluneita aikoja vertaillessa muiden osalta kuin KNN:n kohdalla ei ole merkittävää muutosta. KNN:n testaukseen kulunutta aikaa voidaan selittää mallin tavasta muistaa data.

Nimi	Tarkkuus	Testaus opetusdatalla	Opetusaika (s)	Testaus käsin kirjoitetuilla	Testausaika (s)
DT	53,00 %	100 %	28.580	30,77 %	0.009
KNN	63,46 %	100 %	3.141	73,08 %	4.690
SVM	59,37 %	100 %	323.975	30,77 %	0.333
CNN	79,00 %	100 %	94.932	96,15 %	1.444

TAULUKKO 2. Mittaustulokset keskisuurella datamäärällä

5.3 Suuri kuvamäärä

Kolmannessa testauksessa datanmäärä oli viisinkertaistunut toisesta testauksesta ja tuloksissa oli jo huomattavia eroja. Vaihtelua tapahtui reilun 42 % ja 100 % välillä. Ensimmäisenä huomiona on, että kaikki mallit eivät suoriutuneet annetussa ajassa opetuksesta. Tukivektorikone ei onnistunut opettamaan mallia annetun vuorokauden rajoissa. Mallin annettiin oppia noin kaksi ja puoli vuorokautta nähdäksemme olisiko malli mahdollista opettaa pidemmällä ajalla, mutta tämä ei onnistunut, jotenka opetus lopetettiin kesken. Tämä oli jo algoritmin esittelykappaleessa mainittu mahdollisuus. Kuvista 9, 10 ja 11 vertaillessa kirjaimet E ja F ovat konvoluutiohermoverkkoa lukuun ottamatta hankalia tunnistaa. Suurin onnistuja oli konvoluutiohermoverkko, joka onnistui saamaan kaikki käsin kirjoitetut kirjaimet oikein, toisaalta tulos ei ollut kaukana keskisuurella datamäärällä, kun vain yksi meni väärin.



KUVA 11 Mallien tulokset käsin kirjoitetuilla kirjaimilla. Vasemmalta oikealle DT, KNN ja CNN

Taulukosta 3 voi havaita KNN:n opetusajan, joka on hiukan alta puolitoista minuuttia, joka on merkittävästi pienempi kuin muilla algoritmeilla. Toisaalta ennustaminen mallilla on taas pisin, joka oli aikaisempien testien perusteella odotettavissa. Tarkkuudet jokaisella mallilla on kasvanut pois lukien SVM. Tästä voi päätellä, että datan lisäämisellä on vaikutusta, kun halutaan malli, joka on kykeneväinen toimimaan datan kanssa, joka ei ole mallille entuudestaan tuttu.

Nimi	Tarkkuus	Testaus opetusdatalla	Opetusaika (s)	Testaus käsin kirjoitetuilla	Testausaika (s)
DT	62,75 %	100 %	180.653	42,31 %	0.009
KNN	71,76 %	100 %	88.522	84,62 %	23.403
SVM	-	-	-	-	-
CNN	85,15 %	100 %	277.274	100,00 %	1.446

TAULUKKO 3. Mittaustulokset suurella datamäärällä

6 POHDINTA

Tässä luvussa tarkoituksena on pohtia saavutettuja tuloksia, mitenkä työtä voisi jatkaa ja mitä mahdollisesti tehtäisiin toisin. Toteutetussa testauksessa selvästi parhaiten suoriutui konvoluutiohermoverkko, joka saavutti testauksessa kaikki käsin kirjoitetut kirjaimet oikein.

6.1 Datan käsittely ja algoritmit

Datan käsittelyssä päädyttiin valintaan olla käsittelemättä dataa ollenkaan, jotta yksikään algoritmi ei saa etua, että data olisi käsitelty vahingossa jollekin tietylle algoritmille optimaalisempaan tilaan. Datassa oli myös likaisuutta, joka kuvastaa hyvin reaalimaailmaa, jossa kaikki ei ole aina optimaalista tai täydellistä. Algoritmien tuottaminen oli verrattain yksinkertaista, mutta optimaalisten asetusten löytäminen tuottaa työtä ja tässä olisi ollut työn kohdalla parannettavaa. Varsinkin konvoluutiohermoverkon kohdalla haasteena oli löytää toimiva verkon rakenne. Tulokset voisivat olla hyvin erilaisia erilaisilla asetuksilla, joko parempia tai huonompia.

6.2 Tulokset

Tuloksien kannalta on harmillista, ettei tukivektorikone pystynyt saavuttamaan mitään tulosta suurella datamäärällä, vaikka jotain tämän kaltaista oli odotettavissa jo teoriaan perehdyttäessä. Mielenkiintoista oli huomata tuloksissa selkeitä muutoksia pelkällä opetusdatan lisäämisellä. Tämä ei toisaalta tule täysin uutena asiana, koska modernit keskustelubotit esimerkiksi paljon huomiota saanut ChatGPT on opetettu huomattavilla määrillä tietoa, jota on kerätty jopa useiden vuosien ajan. Tuloksia ei voi liiaksi yleistää, koska testauksessa käytettiin vain yhdenlaista materiaalia. Tälläiset tulokset luovat hyvän pohjan jatkaa tutkimista.

6.3 Jatkaminen

Testaamista voisi vielä jatkaa kasvattamalla opetusdatan määrää entisestään, tällöin voitaisiin saada myös mahdollisesti tarkkuutta suuremmaksi. Testausta

voisi jatkaa pienten kirjainten osalta tai lisätä kolme suomen kielen puuttuvaa kirjainta. Toisaalta voisi testata muita algoritmeja, jotka soveltuvat kuvantunnistukseen. Kyseisiä malleja voisi jatkojalostaa tunnistamaan tekstiä tai vaikka johdannossa mainittuja rekisterikilpiä. Myös dataa voisi esikäsitellä ja testata uudelleen, jolloin todennäköisesti saataisiin parempia tuloksia, jo pienemmällä data määrällä. Vaihtoehtoisesti voisi luoda mallit tunnistamaan pelkästään isoja tai pieniä kirjaimia, jolloin data olisi yksinkertaisempaa ja helpompi opettaa mallille.

LÄHTEET

Thomas Lin. 2020. Alphabet Characters Fonts Dataset, Accessed 15.02.2024. https://www.kaggle.com/datasets/thomasqazwsxedc/alphabet-characters-fonts-dataset

Mikä on päätöspuu? Unite.ai. https://www.unite.ai/fi/what-is-a-decision-tree/. Luettu 20.02.2024

Mikä on KNN (K-Nearest Neighbors)? Unite.ai. https://www.unite.ai/fi/mik%C3%A4-on-k-l%C3%A4himm%C3%A4t-naapurit/. Luettu 20.02.2024

Mitä tukivektorikoneet ovat? Unite.ai. https://www.unite.ai/fi/mit%C3%A4-ovat-tukivektorikoneet/. Luettu 20.02.2024

Mitä ovat CNN:t (Convolutional Neural Networks)? Unite.ai. https://www.unite.ai/fi/what-are-convolutional-neural-networks/. Luettu 20.02.2024

What is decision tree? ibm.com. https://www.ibm.com/topics/decision-trees. Luettu 24.02.2024

Helsingin Yliopisto, Edistyneet neuroverkkomenetelmät. course.elementsofai.com. https://course.elementsofai.com/fi/5/3. Luettu 27.02.2024

Goodfellow, Ian, Yoshua Bengio ja Aaron Courville. 2016. Deep Learning. The MIT Press. ISBN: 0262035618, 9780262035613. Luettu 27.02.2024

Career Foundry. KUVA 2. https://careerfoundry.com/en/blog/data-analytics/what-is-a-decision-tree/. Luettu 27.02.2024

Career Foundry. KUVA 3. https://careerfoundry.com/en/blog/data-analytics/what-is-a-decision-tree/. Luettu 27.02.2024

Analytic Vidhya. KUVA 4. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/. Luettu 27.02.2024

Analytic Vidhya. KUVA 5. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/. Luettu 27.02.2024

Deciccion tree. scikit-learn.org. https://scikit-learn.org/stable/modules/tree.html. Luettu 27.02.2024

Boucher, P. (2020). Artificial intelligence: How does it work, why does it matter, and what can we do about it? In Panel for the Future of Science and Technology. https://www.europarl.europa.eu/RegData/etudes/STUD/2020/641547/EPRS_STU(2020)641547_EN.pdf. Luettu 27.03.2024

Veisdal, J. (2019). The Birthplace of AI. https://medium.com/cantors-paradise/thebirthplace-of-ai-9ab7d4e5fb00. Luettu 28.03.2024

Sami Romula. (2000). Alan Turing. https://www.cs.helsinki.fi/u/kerola/tkhist/k2000/alustukset/turing/turing.html. Luettu 28.03.2024

Laura Klusaitė. (2023) Mitä on koneoppiminen? https://nordvpn.com/fi/blog/koneoppiminen/. Luettu 28.03.2024

Airi Lampinen. (2022). Algoritmit suodattavat ja suosittelevat. https://www.mll.fi/vanhemmille/tietoa-lapsiperheen-elamasta/hyvinvointia-digiajassa/algoritmit-suodattavat-ja-suosittelevat/. Luettu 28.03.2024

Jan-Erik Sandelin. (2020). Mitä konenäkö on? https://blogit.lab.fi/health/2020/04/09/mita-konenako-on/. Luettu 28.3.2024

Kuvantunnistus: mitä se on ja miten se toimii, meltwater.com. https://www.meltwater.com/fi/blog/mita-on-kuvantunnistus-ja-miten-se-toimii. Luettu 30.03.2024

Achitecture of CNN. KUVA 6. https://www.researchgate.net/figure/Architecture-of-CNN16_fig1_350093267. Luettu 15.4.2024

LIITTEET

Liite 1. Mittauspöytäkirja

Datan määrä 26 luokkaa A-Z 30 kuvaa per luokka					
Nimi	Tarkkuus	Testaus tarkkuus opetusdatalla	Testaus tarkkuus	Opetus aika (sekunteja)	Testaus aika
DT	0,50000000000000	100,00	15,38	0.5729999542236328	0.008136
KNN	0,52564102564103	100,00	38,46	0.013000011444091797	0.150139
SVM	0,73717948717949	100,00	61,54	0.23951029777526855	0.017018
CNN	0,73717945814133	100,00	80,77	8.28600001335144	1.659480
Datan määrä 26 luokkaa A-Z 1000 kuvaa per luokka					
DT	0,53000000000000	100,00	30,77	28.579509496688843	0.008507
KNN	0,63461538461538	100,00	73,08	3.140878438949585	4.691229
SVM	0,59365384615385	100,00	30,77	323.9746150970459	0.333031
CNN	0,79000000000000	100,00	96,15	94.932448387146	1.443982
Datan määrä 26 luokkaa A-Z 5000 kuvaa per luokka					
DT	0,62746153846154	100,00	42,31	180.6532392501831	0.008504
KNN	0,71757692307692	100,00	84,62	88.52202296257019	23.40342
SVM	nan	nan	nan	nan	nan
CNN	0,85149997472763	100,00	100,00	277.2743287086487	1.446105

Liite 2 Koodit ja mallit

https://github.com/Jypaa/MachineLearningModels