

Tässä demonstraatiodokumentissa kuvataan TensorFlow-kirjaston käyttämistä kuvien luokittelussa. TensorFlow on Googlen kehittämä avoimen lähdekoodin tekoäly- ja koneoppimiskirjasto, joka on kenen tahansa ladattavissa Internetistä. Kuvatussa demonstraatioissa käytetään TensorFlow'ta Python-ohjelmointikielellä. Kuvadatanä käytetään Schlagenhaufin (2021) kuvakokoelmaa kunnollisista ja kuluneista kuularuuveista.

Ohjelmointikieli: Python. Saatavilla: <https://www.python.org/>

Kehitysympäristö: Visual Studio Code. Saatavilla: <https://code.visualstudio.com/>

Kirjasto: TensorFlow. Saatavilla: <https://www.tensorflow.org/install>

Kuvadata saatavilla: <https://bwdatatdiss.kit.edu/dataset/323>

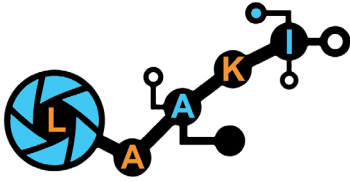
Koska TensorFlow on ohjelmointikirjasto, sen käyttö edellyttää ainakin jonkinlaista ohjelmointitaitoa. Tässä demonstraatioissa kirjastoa käytettiin asennettuna omalle tietokoneelle, joten lisäksi tarvittiin luonnollisesti tulkki tai kääntäjä käytettävälle ohjelmointikielelle sekä kehitysympäristö eli jonkinlainen editori, jolla koodia kirjoitetaan. TensorFlow'ta voi käyttää myös Googlen Colab-pilvipalvelussa, jolloin mitään ei tarvitse asentaa omalle koneelle (<https://colab.research.google.com/>). Colab-palvelua on mahdollista käyttää ilmaiseksi, mutta tällöin käytössä oleva laskentateho on rajattu.

TensorFlow'ssa käytettävä neuroverkko määritellään kerros kerrokselta. Perusymmärtämys neuroverkon rakenteesta siis vaaditaan. Internet on kuitenkin pullollaan esimerkkejä, joita voi vapaasti kopioida ja kokeilla. Erittäin hyviä valmiita esimerkkejä on myös kirjassa Zhang ym. (2021). Joka tapauksessa kirjaston käyttö vaatii jonkinlaista vihkiytymistä aiheeseen ja ohjelmointiymmärrystä.

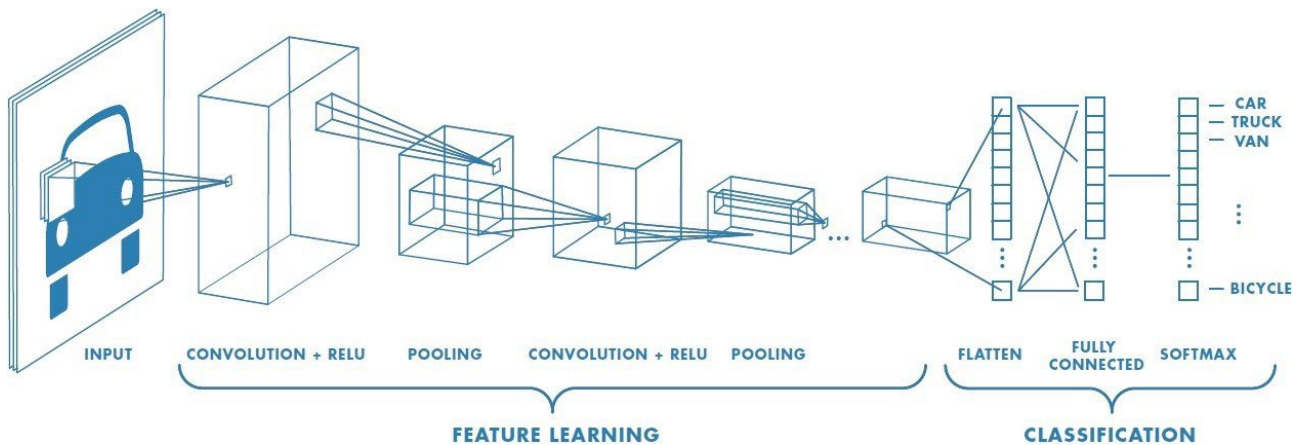
Neuroverkon opettaminen vaatii yleensä paljon kuvadataa. Tässä käytetyssä kuva-arkistossa on yhteensä 21 835 kuvaa, joista 11 075 esitti kunnossa olevia kuularuuveja ja loput kuluneita. Opetusta ja valmiin neuroverkon testaamista varten kuvadata jaettiin kolmeen joukkoon: 16 631 kuvaa käytettiin suoraan neuroverkon opetuksessa ja painojen etsimisessä, 4157 kuvaa käytettiin opetuksen aikaisessa validoinnissa eli opetussyökin välisessä testaamisessa ja valmiin verkon lopputestaukseen jätettiin 1047 kuvaa. Näitä 1047 kuvaa ei siis käytetty millään lailla hyödyksi opetuksessa.

Välttämättä näin suurta kuvamassaa ei tarvita, mutta neuroverkon opetus onnistuu sitä paremmin, mitä enemmän kuvia on. Esimerkiksi Amazon Lookout for Visionin opetukseen käytetty kuvadata ei ole riittävä TensorFlow'n avulla rakennetulle neuroverkolle, sillä virhetilanteista otettuja kuvia oli vain muutama kymmentä.

Tässä demonstraatioissa käytettiin lisäksi niin sanottua datan lisäämistä. Neuroverkon ensimmäinen kerros siis kääntelee ja zoomaa käytettäviä kuvia, jolloin opetuskuviin määrää saadaan keinotekoisesti lisättyä. Muuten käytössä ollut neuroverkko oli hyvin klassinen konvoluutioneuroverkko, ja se rakentui seuraavista kerroksista: viisi konvoluutiokerrosta, joita kutakin seurasi max pool -kerros, yksi dropout-kerros, kaksi täysin yhdistettyä kerrosta ja ulostulokerros. Konvoluutiokerroksia käytetään etsimään kuvasta piirteitä, max pool -kerroksia yksinkertaistamaan laskuja ja dropout-kerrosta poistamaan satunnaisesti joitain osia



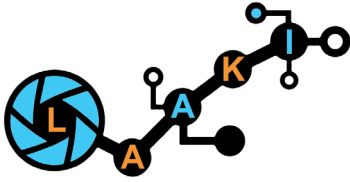
neuroverkosta toiminnasta ja näin estämästä ns. ylioppimista. Lopun täysin yhdistetyt kerrokset ovat samanlaisia kuin ns. perinteisessä neuroverkossa, ja niiden kautta tehdään lopullinen luokittelu. Kuva 1 esittää periaatekuvan konvoluutioneuroverkon toiminnasta. Kuva 2 näyttää demonstraatioissa käytetyn neuroverkon määrittelyn.



Kuva 1. Tyypillisen konvoluutioneuroverkon rakenteen periaate (Saha 2018).

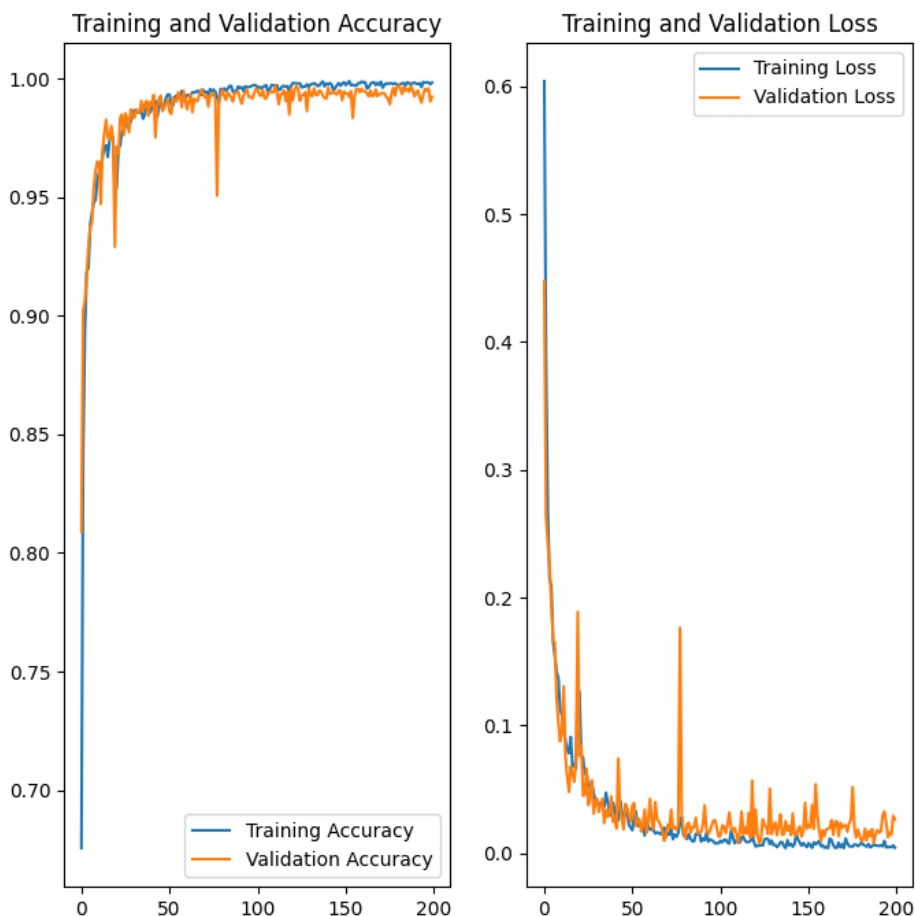
```
model = Sequential([
    data_augmentation,
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(128, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(256, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(64, activation='relu'),
    layers.Dense(num_classes, activation='softmax')
])
```

Kuva 2. Demonstraatioissa käytetyn konvoluutioneuroverkon määrittely TensorFlow:ssa.

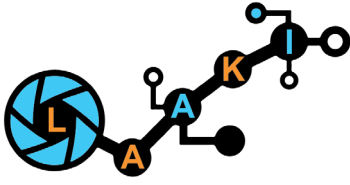


Neuroverkon opettamista ja testaamista varten kuvadata jaettiin kansioon opetusdata, jossa oli alikansiot OK ja NG kunnollisista ja rikkinäisistä kuularuuveista otetuille kuville, ja kansioon testidata, jossa oli myös vastaavat alikansiot OK ja NG. Opetusdatakansiossa oli 20 788 kuvaa ja testidatakansiossa 1047 kuvaa. Opetusdatakansion sisältämien kuvien jako opetus- ja validointikuviin tehtiin TensorFlow'n ohjelmakoodissa suhteella 80 % – 20 %.

Neuroverkkoa opetettiin 200 epookkia eli opetussykliä. Opetus vei noin kahdeksan tuntia (Intel i7-7700HQ CPU @ 2.80GHz, 16 Gt RAM). Kuva 3 näyttää opetustarkkuuden ja validointitarkkuuden kehityksen epookkien aikana. Lopullinen neuroverkko saatiin epookilla 181, ja sen opetustarkkuus oli 99,74 % ja validointitarkkuus 99,71 %. Kun neuroverkkoa käytettiin testidatan kuville, sai se kaikki 1047 kuvaa luokiteltua oikein. Kuva 4 esittelee luokittelutuloksia testidatan kuville.

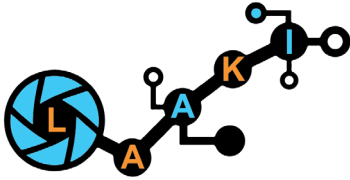


Kuva 3. Mallin tarkkuuden ja häviön kehitys.



<p>Ennustus: NG Oikea: NG</p>	<p>Ennustus: NG Oikea: NG</p>	<p>Ennustus: OK Oikea: OK</p>	<p>Ennustus: OK Oikea: OK</p>
<p>Ennustus: OK Oikea: OK</p>	<p>Ennustus: OK Oikea: OK</p>	<p>Ennustus: NG Oikea: NG</p>	<p>Ennustus: OK Oikea: OK</p>
<p>Ennustus: OK Oikea: OK</p>	<p>Ennustus: NG Oikea: NG</p>	<p>Ennustus: NG Oikea: NG</p>	<p>Ennustus: OK Oikea: OK</p>
<p>Ennustus: OK Oikea: OK</p>	<p>Ennustus: OK Oikea: OK</p>	<p>Ennustus: OK Oikea: OK</p>	<p>Ennustus: OK Oikea: OK</p>
<p>Ennustus: OK Oikea: OK</p>	<p>Ennustus: NG Oikea: NG</p>	<p>Ennustus: NG Oikea: NG</p>	<p>Ennustus: NG Oikea: NG</p>





SeAMK

SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Kuten nähdään, TensorFlow'lla voi tehokkaasti opettaa neuroverkkomallin luokittelemaan kuvadataa. Kirjaston vahvuuksia ovat sen ilmaisuus sekä mahdollisuus säätää mallin parametreja täysin vapaasti. Huonoina puolina on toki se, että kirjaston käyttäminen vaatii ymmärrystä ohjelmoinnista sekä neuroverkkojen rakenteesta. Hankkeessa tehtyjen kokeiden mukaan kuvadataa tarvitaan melko paljon, että opetustuloksesta saadaan hyvä. Kun kirjastoa kokeiltiin noin sadalla kuvalla, jotka kuuluivat neljään eri luokkaan, opetetun neuroverkon tarkkuus oli vain 80 %. Jos kuvat olivat kovin monimutkaisia, saatettiin päätyä vain 30 % tarkkuuteenkin. On toki mahdollista, että eri neuroverkkorakenteella päädyttäisiin hieman parempaan tulokseen. Tässä dokumentissa esitellyn lisäksi kokeiltiin suosittua AlexNet-verkkoa (Krizhevsky ym. 2017), mutta tulokset pysyivät pienillä kuvamäärillä jokseenkin samoina.

Lähteet

Krizhevsky, A., Sutskever, I. & Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 60 (6): 84–90. doi:10.1145/3065386

Saha, S. 15.12.2018. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way [Blogikirjoitus]. *Towardsdatascience*. [Viitattu: 3.9.2021]. Saatavilla: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Schlagenhauf, T. 2021 Ball Screw Drive Surface Defect Dataset for Classification. [Sähköinen tietoaaineisto]. *Karlsruher Institut für Technologie*. [Viitattu: 3.9.2021]. Saatavilla: <https://bwdatadiss.kit.edu/dataset/323>, doi: 10.5445/IR/1000133819

Zhang, A., Lipton, Z. C., Li, M. & Smola, A. J. 2021. Dive into Deep Learning. [Verkkokirja], arXiv preprint. [Viitattu: 3.9.2021]. Saatavilla: <https://d2l.ai/>, doi: arXiv:2106.11342



Euroopan unioni
Euroopan aluekehitysrahasto

Vipuvoimaa
EU:lta
2014–2020



ETELÄ-POHJANMAAN LIITTO
Regional Council of South Ostrobothnia