Harjoitus 4 – Google Colab – konenäkömallin kouluttaminen

Tehtävänanto pähkinänkuoressa:

Tee Google Colab-notebook –ratkaisu, joka pystyy kouluttamaan konenäkömallin (neuroverkon) omalla kuva-aineistollasi, sekä testaamaan koulutetun mallin toimintaa uusilla esimerkkikuvilla.

Voit rakentaa neuroverkon valmiista pohjasta kokonaan, tai käyttää style transfer –tekniikkaa (suositeltu).

Tuota neuroverkolle oma kuva-aineisto, esim. porojen ja hirvien kuvista (kaksi luokkaa). Myös muut kuva-aineistot ovat ok! Jos haluat lisää luokkia, voit hakea myös peuran kuvia. Tee kuvien lataamiselle erillinen notebook, ja tallenna tuotettu kuva-aineisto Google Driveen.

Voit rakentaa aineiston helposti lataamalla hakusanoilla (esim. **"reindeer", "elk" tai "moose".**) valokuvia Google Colabista käsin.

Ks. esimerkki Moodlesta: miten kuvia ladataan Google Colabissa kuvahausta esim. Google Driveen.

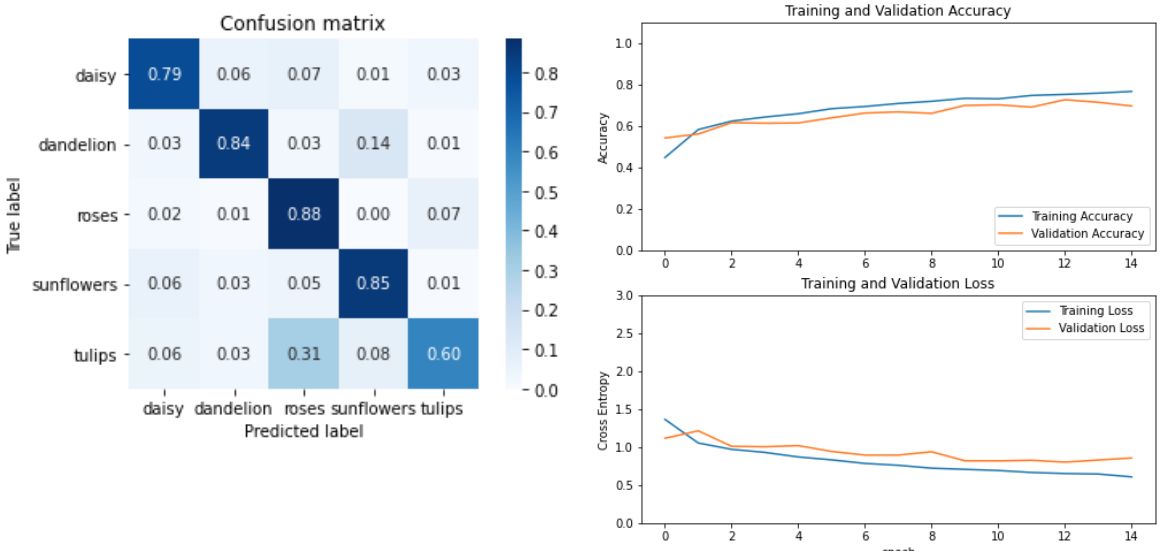
Koko aineisto pitää olla omassa kansiossaan, ja jokaisella tunnistettavalle asialle pitää olla oma alikansio. Esim. images –kansiossa tulee olla hirvi- ja poro-kansiot.

Harjoituksen Google Colab-notebookista pitää löytyä seuraavat ominaisuudet:

- Itse tuotetun aineiston lataaminen ja käsittely
- Neuroverkon rakentaminen
- Neuroverkon kouluttaminen
- Virhemetriikka (training/val accuracy ja loss, confusion matrix, muu statistiikka)
- Testikoodi, joka yrittää tunnistaa testiaineiston ulkopuolisia kuvia 5-10kpl kerralla, ja tulostaa jokaisen kuvan todennäköisimmän luokan prosenttilukuna

Mallin tavoitteena on tunnistaa vähintään 60 - 70% valokuvista. Huomaa, että tarkkuusprosenttiin vaikuttaa suuresti aineiston laatu, neuroverkon soveltuvuus aineistoon sekä myös tunnistettavien asioiden muoto. (koska on helpompaa tunnistaa kuvasta rekka-auto kuin pieni kivi).

Tärkeintä on, että koulutetun mallin metriikka-arvot ovat oikeansuuntaisia (training_accuracy, val_accuracy, training_loss ja val_loss) sekä precision, recall, accuracy ja f1-score



Kuvan lähde:
https://colab.research.google.com/github/tensorflow/hub/blob/master/examples/colab/image_feature_vector.ipynb

Suoraviivaisin versio toteuttaa tämä harjoitus on noudattaa tätä notebookia:

https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/images/classification.ipynb

Tämä Colab-pohja on erittäin hyvä lähtökohta harjoitukselle, mutta se ei välttämättä taivu kovin helposti monipuolisempiin tarpeisiin, kuten **transfer learning**in käyttämiseen. Huomaa, että tämä esimerkki laatii neuroverkon alusta asti itse, jolloin tarkkuus yleensä on 60-80% välillä.

Lataa tämän jälkeen oma aineisto haluamistasi asioista (esim. hirvi, poro, peura). Konenäön opettamista varten tarvitaan tyypillisesti 1000 kuvaa per kategoria. Kuvia on hyvä olla myös monenlaisia, monesta asennosta, monenlaisissa keliolosuhteissa ja myös kuvien tarkkuus saa vaihdella. Tämän harjoituksen osalta on riittävää kerätä 400-500 valokuvaa per kategoria, pääsemme sillä hyvin alkuun.

Muista siivota aineistosta turhat kuvat pois (esim. kuvat leluista). Voit siivota aineistoa joko käsin, tai kehittää hakusanaa, jolla kuvia haetaan. Voit jättää huomiotta hakutuloksesta sanoja käyttämällä – merkkiä,
esim. ***"reindeer –toy –cartoon"***, eli tässä tapauksessa kuvista otetaan lelut ja piirretyt porokuvat pois. Poista myös duplikaatit aineistosta, sillä emme tarvitse useita kopioita samasta valokuvasta.

Lopputuloksena tulee syntyä neuroverkko, joka pystyy valokuvasta tunnistamaan vähintään kahta eri asiaa, esim. poroja ja hirviä.

Jos löydät itsellesi paremmin soveltuvan Google Colab –pohjan neuroverkon kehittämistä varten, voit käyttää myös sitä!

Kirjaa lopulta kaikki kokemuksesi oppimispäiväkirjaan! Mitä mieltä olet neuroverkkojen rakentamisesta ja käyttämisestä kuvantunnistuksessa? Mikä on yllättävää, mikä haastavaa, mitä mahdollisuuksia neuroverkot mielestäsi tuovat?

Aiheeseen liittyviä linkkejä ja muita notebookeja:

https://colab.research.google.com/github/google/eng-edu/blob/master/ml/pc/exercises/image_classification_part1.ipynb

https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/classification.ipynb

https://colab.research.google.com/github/tensorflow/hub/blob/master/examples/colab/image_feature_vector.ipynb

https://www.tensorflow.org/tutorials/images/transfer_learning

https://thedatafrog.com/en/articles/image-recognition-transfer-learning/

https://colab.research.google.com/github/kylemath/ml4a-guides/blob/master/notebooks/transfer-learning.ipynb#scrollTo=vIwMY_ZXYoax

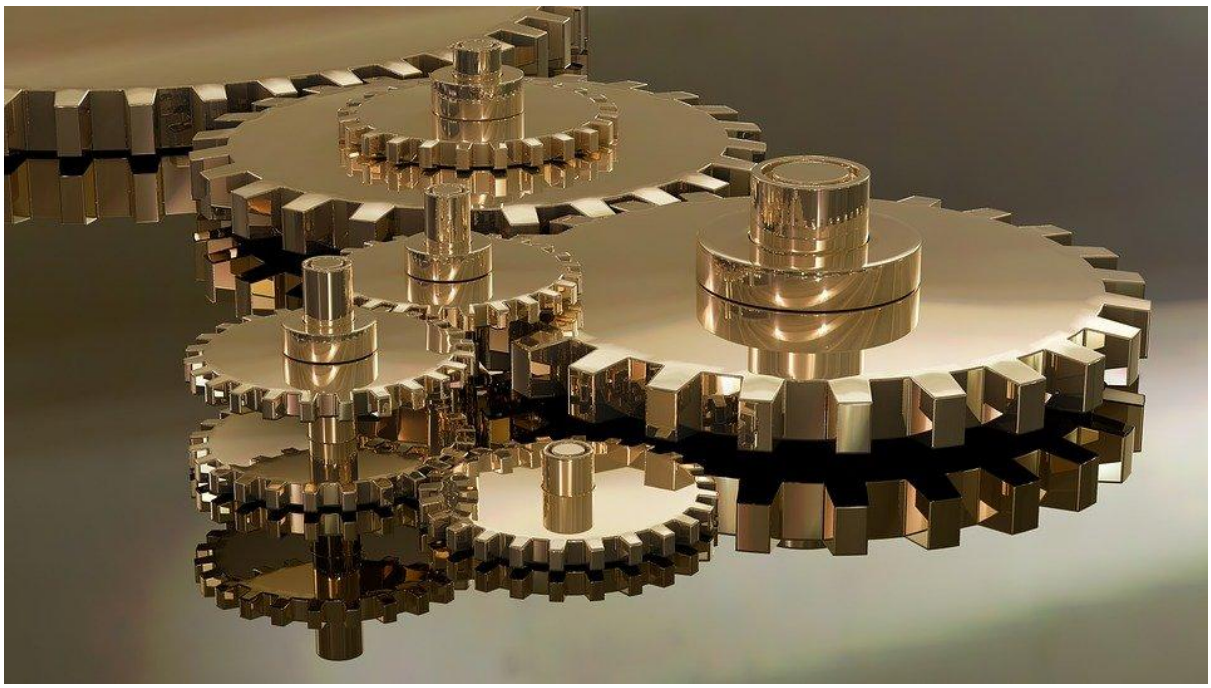
…tai sitten voi googlettaa lisää esimerkkejä sanoilla **"Google Colab Image Classification"** tai **"Google Colab Keras Transfer Learning"** ☺

Lisätehtäviä:

- Muokkaa kuva-aineiston hakuskriptiä siten, että se osaa poistaa myös duplikaattikuvia automaattisesti (todennäköisesti kaikkien poistaminen ei onnistu helposti)

- **Pyri optimoimaan neuroverkkosi toimintaa siten, että saat mallin tarkkuuden paremmaksi**

    - Yleisiä toimenpiteitä tarkkuuden parantamiseksi:
        - data augmentation
        - neuroverkon osien muuttaminen (internetistä esimerkkejä, joita on käytetty kuvantunnistuksessa)
        - loss-funktioiden / optimizereiden säätäminen
        - aineiston kasvattaminen ja/tai parantaminen
        - transfer learningin käyttäminen



- Käytä transfer learningia, ja kokeile haluamiasi neuroverkkopohjamalleja, neuroverkon säätöjä, aineiston käsittelymenetelmiä, optimizer-funktioita, loss-funktioita. Tämä on hyvin vapaamuotoinen lisätehtävä, ja kenelläkään on tuskin mahdollista kokeilla kaikkea mahdollista. Valitse mieleisesi kokeiltavat aiheet!

  Ks. myös osa "fine-tuning" tästä tutorialista:
  https://www.tensorflow.org/tutorials/images/transfer_learning

  - **Yleisiä neuroverkkomalleja:**
    - VGG16 ja VGG19
    - MobileNet
    - Inception
    - Xception
    - ResNet (raskas kouluttaa)

  - **Yleisiä optimizer –funktioita**
    - Adam
    - Adamax
    - SGD
    - RMSprop

  - **Yleisiä loss-funkioita (muitakin on)**
    - categorical cross entropy
    - hinge
    - KL divergence

  Moodlesta löytyy myös muutama loss-funktio / optimizer-esimerkki, joita voit kokeilla kukka-aineistolle! Muista myös Dropout, EarlyStop ja ReduceLROnPlateu –tekniikat!