

KUVA-ANNOTAATTORI

Projektisuunnitelma, 21.02.2022

Rajala Joni

778549, Kolmas vuosi kauppatieteet

1. Yleiskuvaus ja vaikeustaso

Ohjelmisto, jonka avulla voidaan rakentaa konvoluutioverkkoa kouluttava tietojoukko. Ohjelmiston avulla voit annotoida kuvia, selata jo annotoituja kuvia, sekä ohjelmisto tarjoaa valmiiksi luodun PyTorch datasetin annotoiduista kuvista.

Yhdestä kuvasta voidaan annotoida monia eri esineitä omiin luokkiinsa. Annotaatiot SQL tiedostoon kuvan nimen sekä annotaation luokan kanssa. Yksi rivi SQL taulussa sisältää siis: id (primary key), image, annotation (muodossa begin_x, begin_y, end_x, end_y), object.

Tästä SQL tiedostosta ohjelma myös lukee annotaatiot sekä rakentaa datasetin tarvittaessa. Vaikeustaso vaikea.

Koska ohjelma on melko helppo ja nopea toteuttaa, tulen keskittymään eritoten sen testaukseen.

2. Käyttötapauskuvaus ja käyttöliittymän luonnos

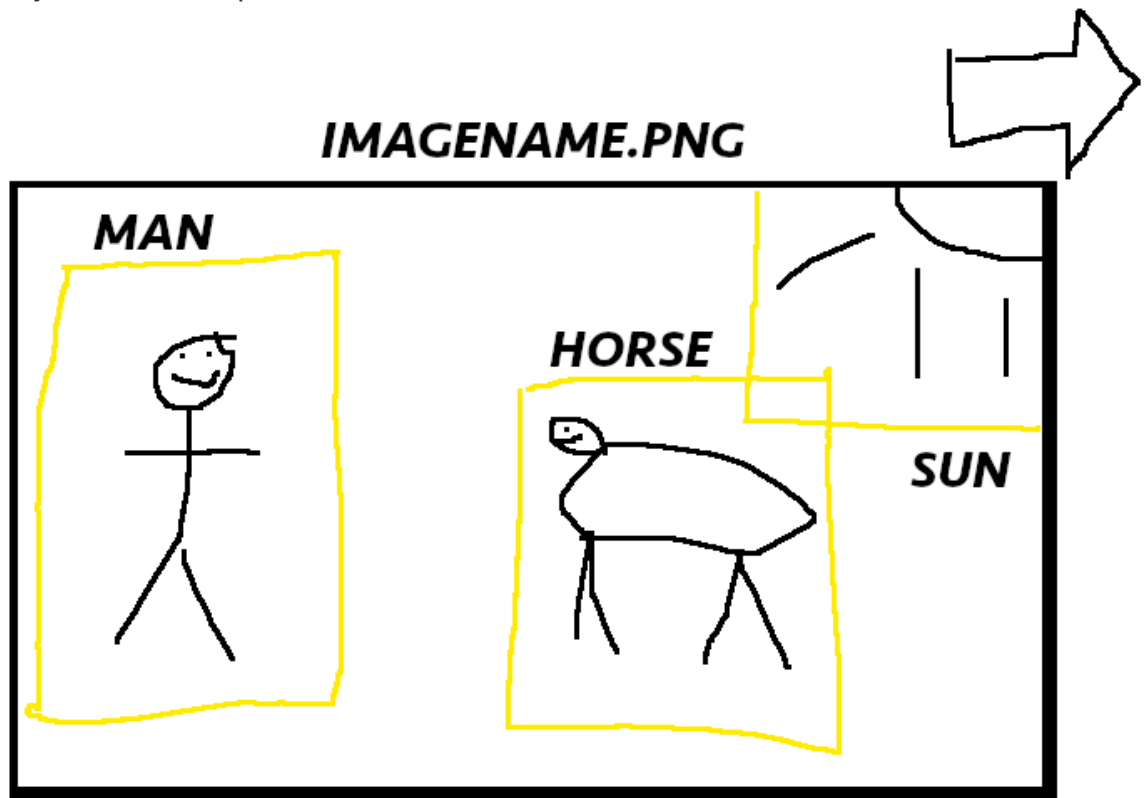
Ohjelman avautuessa käyttäjä voi valita haluaako hän annotoida kuvia vai selata valmiiksi annotoituja kuvia.

Joni's sick annotation program

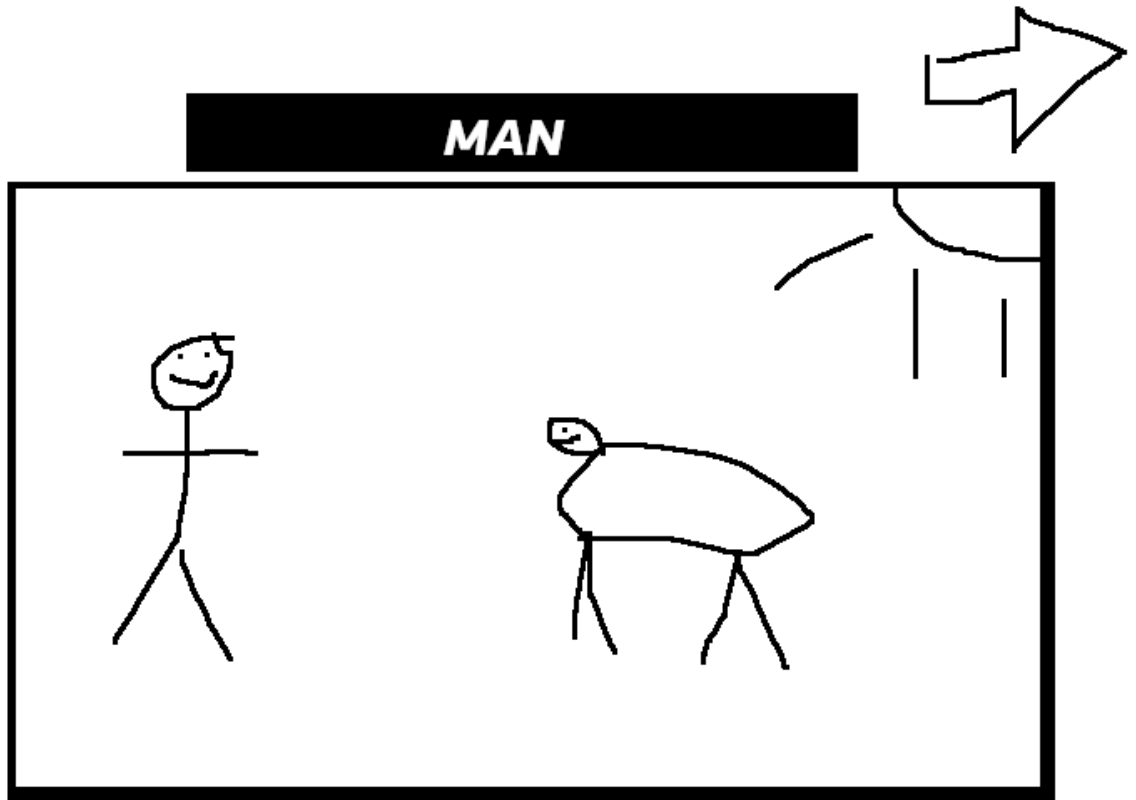


Annotoituja kuvia selatessa kuvat vain avautuvat ruudulle annotaatiot

piirrettynä. Nuolesta painaessa tulee ruudulle uusi kuva.



Jos haluaa annotoida kuvia, tulee kuvat ensin siirtää images kansioon, josta ohjelma osaa lukea ne. Kuvat avautuvat ruudulle, jossa on tekstikenttä, johon käyttäjä kirjoittaa ensin annotoitavan objektin luokan ja sen jälkeen hiirellä piirtää neliön objektin ympärille. Kun kaikki annotaatiot ovat piirrettynä, käyttäjä painaa nuolta, jonka jälkeen annotaatiot tallentuvat ja ruudulle ilmestyy uusi kuva.



3. Ohjelman rakennesuunnitelma

GUI luokka toimii ohjelman MainWindowina, johon eri widgetit lisätään. Se pitää sisällään muuttujat ja funktiot, joita kaikki widgetit käyttävät. Tallettaa itseensä aina näytettävän widgetin, jota esimerkiksi Menu luokka muuttaa, kun käyttäjä valitsee, haluaako hän selata vai annotoida kuvia.

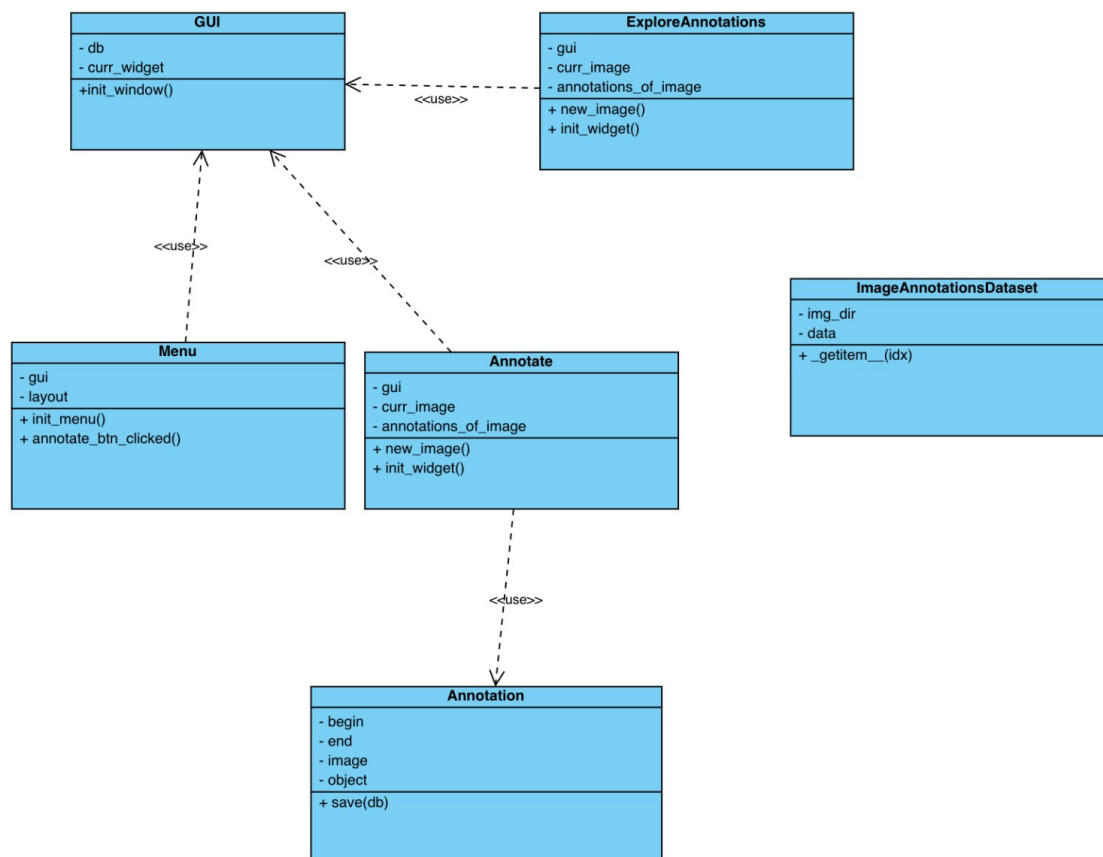
Menu luokka on widget, joka avautuu, kun ohjelma käynnistetään. Sen avulla käyttäjä voi suunnistaa joko tutkimaan annotoituja kuvia tai annotoimaan lisää kuvia. Sisältää funktiot napeille, joilla käyttäjä suunnistaa ohjelmiston sisällä ja funktion, jolla widget alustetaan.

Annotate luokka on widget, jonka avulla käyttäjä voi annotoida kuvia. Annotaatioista se luo Annotation luokan olioita. Sisältää funktiot, joilla luokka hoitaa annotaatioiden piirtämisen sekä funktiot napeille, joilla vaihdetaan kuvaa, aloitetaan alusta ja palataan menuun.

ExploreAnnotations luokka on widget, jonka avulla käyttäjä voi selata jo annotoituja kuvia. Sisältää funktion, jolla kuva ja annotaatiot piirretään ja funktioita napeille.

Annotation luokan avulla voidaan mallintaa ja tallentaa annotaatioita. Sisältää vain funktion, jonka avulla kyseinen annotaatio voidaan tallentaa databaseen.

ImageAnnotationsDataset luokka tarjoaa käyttäjälle mahdollisuuden importata luomansa annotoidut kuvat datasettinä omaan käyttötarkoitukseensa. Se ei siis sisälly graafiseen käyttöliittymään. Sen rakenne noudattaa PyTorchin Custom Datasetin rakennetta.



Ohjelman olisi voinut rakentaa myös niin, että jokaiselle ohjelman eri osalle ei ole omaa kokonaista sivua vaan ohjelma olisi koostunut monesta pienestä palasesta, joita vaihdetaan ruudulle riippuen mitä käyttäjä haluaa tehdä. Mielestäni oli kuitenkin selkeintä, että jokaiselle omalla osalla on oma luokka, joka kuvaa koko osan toimintaa.

4. Tietorakenteet

Annotaatioiden tallennus toteutetaan siis SQL tietokantaan. Alustavien suunnitelmien mukaan tietokanta tulee sisältämään vain yhden taulukon

”Annotations”. Tallettaminen voitaisiin hoitaa myös csv tiedoston avulla, mutta SQL tietokannan käyttö on tuo mukavaa harjoitusta sekä se antaa mahdollisuuden tulevaisuudessa tehdä tietokannasta monimutkaisemman. Esimerkiksi kuville voitaisiin tehdä oma taulukko, jos haluisimme tallentaa enemmän tietoa kuvista ja sitten annotations taulukon image sarake voisi olla foreign key.

Ennen tallennusta tietokantaan, annotaatiot tallennetaan listaan. Listaa käytetään myös tallettamaan kaikki kuvat, kun halutaan selata niiden annotaatioita.

5. Tiedostot ja tiedostoformaatit

Kuten jo mainittu annotaatiot tallennetaan SQL tietokantaan. Tämän lisäksi ohjelma käyttää kuvatiedostoja. Ohjelma tukee yleisimpiä kuvatiedostomuotoja, mutta niiden on käytettävä värimuotona RGB:tä, jotta PyTorch pystyy lukemaan ne.

6. Algoritmit

Koska ohjelma on teknisesti melko simppeli ei se sisällä erikoisia algoritmeja tai matemaattisia laskutoimituksia. Ainoastaan Annotaation-luokan save methodi käyttää sorttaus algoritmiä, selvittääkseen annotaation vasemman yläkulman ja oikean alakulman.

7. Testaussuunnitelma

Tärkeimpänä asiana ohjelmassa tulee testata sitä, että minään hetkenä vääränlaisia annotaatioita ei tallenneta tietokantaan tai vääränlaista dataa ei lueta tietokannasta. Tätä tulee testata kuvia annotoidessa, annotoituja kuvia katsoessa sekä datasettiä rakennettaessa. Koska annotaatio rakentuu kuvasta, annotaation koordinaateista ja annotoiduista objektista, tulee kaikki nämä osat varmentaa oikeanlaisiksi. Esimerkiksi koordinaattien tulee olla nollan ja kuvan koon välissä, kuvan tulee olla oikeassa tiedostomuodossa ja objektin tulee tekstiä eli se ei voi olla tyhjä. Ennen objektin tallennusta on myös tärkeää tarkistaa, ettei se sisällä komentoja, jotka mahdollistaisivat SQL Injektion.

Nämä ongelma eivät luonnollisesti saa kaataa ohjelmaa vaan ne tulee käsitellä oikealla tavalla. Tallettaessa annotaatioita, vääränlaiset annotaatiot tulee olla tallentamatta ja lukiessa annotaatioita vääränlaiset annotaatiot tulee jättää huomiotta. Koska itse ohjelmisto on melko simppeli tulen keskittymään eritoten ohjelmiston testaukseen.

8. Kirjastot ja muut työkalut

PyQt5

- Graafisen käyttöliittymän luonti

OS

- Kuvatiedostojen hakeminen

SQLite3

- SQL tietokannan luonti ja sen kanssa kommunikointi

Torchvision, PyTorch ja Pandas

- Datasetin luonnissa käytettävät kirjastot

9. Aikataulu

Ensin luon GUI ja Annotate luokat. Lähes valmiin Annotate luokan pohjalta pystyn sitten helposti luomaan ExploreAnnotations ja Annotation luokat. Lopuksi luon ImageAnnotationDataset luokan.

Kun luokat ovat lähes valmiit alan rakentamaan testejä. Kun luokat ovat lähes valmiit, on testejä helppo tehdä, mutta luokkien rakenteen muuttaminen on kuitenkin vielä helppoa, jos testeissä ilmenee ongelmia.

15.2 – 21.2 Välisenä aikana sain luokat lähes valmiiksi

- Aikaa kuluin noin 7 tuntia

21.2 - 28.2 välisenä aikana luon ohjelmalle testit ja varmistan, että ohjelma toimii halutulla tavalla.

- Oletettu kesto noin 10 tuntia

28.2 - 7.3 välisenä aikana viimeistelen ohjelmiston lisäämällä kommentteja ja parantamalla koodin luettavuutta.

- Oletettu kesto noin 5 tuntia

10. Kirjallisuusviitteet ja linkit

Tärkeimpänä lähteenä Qt:n oma dokumentaatio.

- <https://doc.qt.io/qt-5.15/>
- <https://doc.qt.io/qtforpython/>

SQLite3 käyttöön

- <https://www.sqlitetutorial.net/sqlite-python/creating-database/>

Datasetin luomiseen

- <https://pytorch.org/docs/stable/data.html#torch.utils.data.Dataset>
- https://pytorch.org/tutorials/beginner/basics/data_tutorial.html

Lisäksi StackOverFlow artikkeleita eri toteutusteknisiin asioihin