## IT Projektfeladat Dokumentáció Objektum felismerő modell

Készítette:  
Csernák Gergely TVJ2T2  
GAMF Mérnökinformatikus BsC

A feladat leírása

Objektum felismerő modell betanítása és alkalmazása járművek vagy emberek detektálására. TensorFlow/Keras Deep Learning modellek megismerése, betanítása módszerének kipróbálása, eredmények összehasonlítása. Kamerával minták gyűjtésének és címkézésének automatizálása.

Használt technológiák

Programozási nyelv

A Python az 1990-es évek elején Guido van Rossum nevéhez fűződik. Guido egy holland programozó volt, aki a Centrum Wiskunde & Informatica kutatóintézetnél dolgozott. A Python eredetileg a ABC nyelv utódjaként született meg. A Python neve Guido kedvenc brit komédia sorozatáról, a Monty Python’s Flying Circus-ról származik.

Az első Python verzió 1991-ben jelent meg, azóta több verzió is kiadásra került. A nyelv kezdetben elsősorban oktatási célokra készült, de gyorsan népszerűvé vált a szoftvertervezők, webfejlesztők és adat tudósok körében is.

A Python nyelv jellemzői közé tartozik az egyszerű és olvasható szintaxis, amelyet a fejlesztők gyakran "pythonic" kódolási stílusnak neveznek. Emellett a Python nyelv támogatja az objektumorientált és funkcionális programozási paradigmákat is.

A Python nyelv különlegessége, hogy nagy mennyiségű modul és könyvtár érhető el hozzá. Ezek a modulok és könyvtárak nagyban segítik a fejlesztőket a gyorsabb és hatékonyabb programozásban. A Python további előnye, hogy cross-platform, vagyis a kód futtatható különböző operációs rendszereken, mint a Windows, a macOS és a Linux.

Ma a Python az egyik legnépszerűbb programozási nyelv a világon, a webfejlesztéstől kezdve az adattudományig sok területen használják.

Kiegészítő csomagok

**Numpy:**

A NumPy (Numerical Python) egy Python csomag, amelyet számos matematikai művelet végzéséhez használnak. A NumPy támogatja a többdimenziós tömbök kezelését, valamint nagy mennyiségű matematikai műveletet, mint például a lineáris algebra, a Fourier-transzformáció és a statisztikai analízis. A NumPy különösen hasznos a tudományos és mérnöki számítások területén.

**Pandas:**

A Pandas egy Python csomag, amelyet adatok elemzéséhez és manipulálásához használnak. A Pandas lehetővé teszi a táblázatok, adatkeretek és sorozatok létrehozását, valamint azok egyszerű és hatékony kezelését. A Pandas különösen hasznos a nagy adathalmazok kezelésében, és többek között támogatja a CSV, Excel, SQL és más formátumok importálását és exportálását.

**Matplotlib:**

A Matplotlib egy Python csomag, amelyet adatok vizualizálásához használnak. A Matplotlib lehetővé teszi a különböző grafikonok, ábrák és diagramok létrehozását, valamint azok testreszabását és finomhangolását. A Matplotlib széles körben használják a tudományos és mérnöki számítások területén, valamint az adatvizualizációban és a gépi tanulásban.

**Opencv-python:**

Az OpenCV (Open Source Computer Vision) egy nyílt forráskódú, cross-platform számítógépes látás és képfeldolgozási könyvtár, amelyet különböző képfeldolgozási feladatokhoz használnak. Az OpenCV-python egy Python csomag, amely lehetővé teszi az OpenCV használatát a Python programokban. Az OpenCV-python támogatja a kamerakezelést, az arcfelismerést, az objektumkövetést, a képek és videók feldolgozását, valamint a gépi tanulás algoritmusok integrálását.

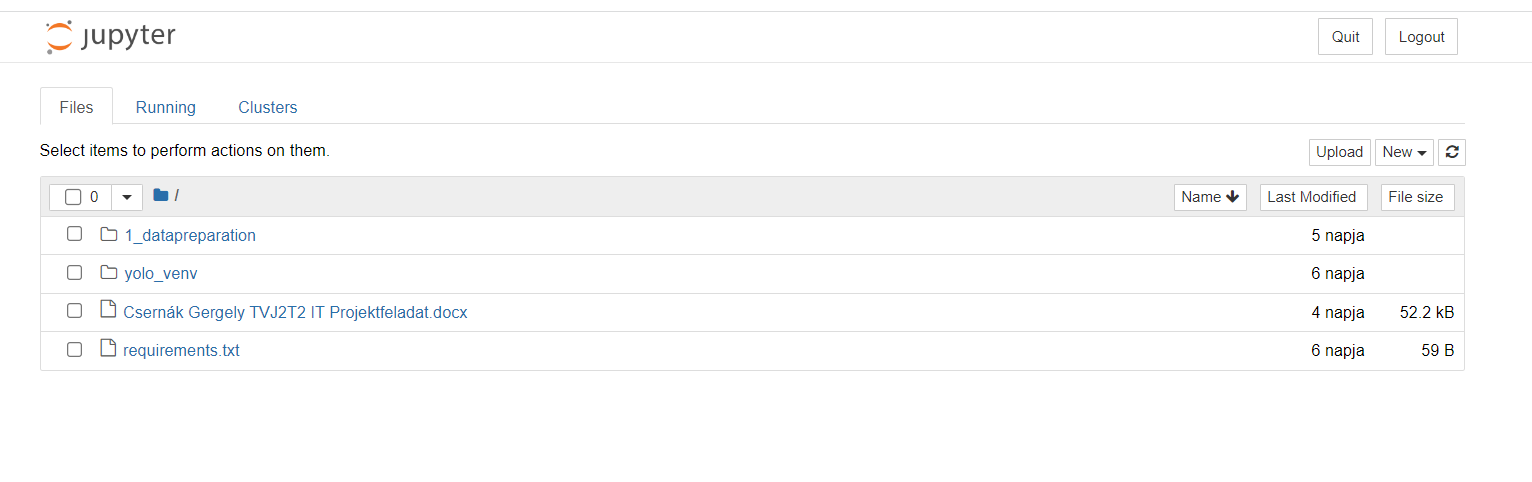
**Jupyter:**

A Jupyter egy interaktív környezet, amely lehetővé teszi a kód, a szöveg és a grafikák kombinálását egy notebook-ban. A Jupyter notebookok különösen hasznosak a tudományos és mérnöki számítások területén, mivel lehetővé teszik a kód és az eredmények megosztását, az interaktív adatvizualizációt és a dokumentációk.

**CV2:**

Az OpenCV (cv2) egy népszerű nyílt forráskódú képfeldolgozási könyvtár, amelyet a Python programozási nyelvhez készítettek. A cv2 lehetővé teszi a képek és videók feldolgozását, manipulálását, elemzését és vizualizációját.

Az OpenCV Python könyvtár rengeteg funkciót és metódust kínál a képfeldolgozási feladatokhoz, például képek beolvasása és megjelenítése, képek és videók manipulálása (pl. méretezés, vágás, elforgatás), képek szűrése és javítása (pl. élek detektálása, zajcsökkentés), objektumok detektálása és követése, arcfelismerés, képek és videók összevonása, geometriai transzformációk (pl. átméretezés, forgatás, torzítás) végrehajtása, és még sok más.



LabelImg – Kiegészítő eszköz

A LabelImg egy Python alapú eszköz, amely lehetővé teszi a képek címkézését. A képek címkézése azért fontos, mert számos gépi tanulási feladat, például az objektumfelismerés vagy az objektumkövetés alapját képezi. A LabelImg segítségével könnyen lehet címkézni az objektumokat a képen, majd az így címkézett adatokat fel lehet használni a gépi tanulási modell kiképzéséhez.

A LabelImg egyszerűen használható grafikus felülettel rendelkezik, amely lehetővé teszi a képek megnyitását, a kívánt objektumok kijelölését és a címkézésüket. Az elkészült címkék a kép mellett kerülnek elmentésre, és számos formátumban (például XML, JSON) exportálhatók, amelyeket a gépi tanulási modell kiképzésekor lehet felhasználni.

A LabelImg a PyQT alapú felhasználói felülettel és a Python Imaging Library (PIL) képfeldolgozási könyvtárral van összekapcsolva. A LabelImg-t a számítógépre kell telepíteni, majd azt követően már használható a képek címkézésére. A LabelImg alkalmas többek között az objektumok, arcok és szövegek címkézésére.

A képen szöveg, fa, út, kültéri látható

Automatikusan generált leírás

YOLO – Kiegészítő eszköz

A YOLO (You Only Look Once) egy olyan neurális hálózat alapú objektumfelismerő rendszer, amely képes a képen található objektumok pontos lokalizálására és osztályozására. A YOLO alapvetően három részből áll: a bemeneti képek előfeldolgozásából, a neurális hálózatból és a végrehajtásból.

A YOLO neurális hálózata egy konvolúciós neurális hálózat, amely a képet több, egymásra következő rétegben dolgozza fel, és minden réteg kiszámítja a képen található objektumok koordinátáit és az objektumok osztályát. A YOLO-nak csak egyszer kell végrehajtani a teljes képen, és az eredményeket azonnal megkapjuk, ezért nevezték el "You Only Look Once"-nak.

A YOLO betanítása az ún. objektumkövetési adathalmazokkal történik. Ez azt jelenti, hogy minden kép adatait előre meghatározott formában kell tárolni, és minden objektumhoz meg kell adni a helyzetét és az osztályát. Ezeket az adatokat általában XML fájlokban tárolják.

A YOLO betanítása lépésről lépésre történik. Először létre kell hozni egy modellt, majd betölteni a betanító adathalmazt. Ezután a modellt be kell állítani, és az eddig tanult adatok alapján be kell állítani az optimális paramétereket. Ezután történik maga a betanítás, amely során a modell az adathalmazon keresztül megtanulja az objektumok helyzetét és osztályát.

A YOLO betanítása több tíz, vagy akár száz óráig is tarthat, attól függően, hogy milyen nagy és összetett az adathalmaz. Azonban, ha a betanítás befejeződött, a modell képes lesz a képen található objektumok helyzetének és osztályának pontos meghatározására. Ezután a modell készen áll a valódi adatokon történő tesztelésre és alkalmazásra.

A képen diagram látható

Automatikusan generált leírás

A betanítás során alkalmazott fájltípusok

XML

Az XML (Extensible Markup Language) egy szöveg alapú jelölőnyelv, amelynek fő célja az információ átvitele és tárolása. Az XML fájlok egy nagyon elterjedt formátum, amely gyakran használatos különböző adatok leírására és megosztására.

Az XML fájlok felépítése hierarchikus struktúrára épül, amely a dokumentum "fa" szerkezetében jelenik meg. Az XML fájlokban lévő adatokat elemekként és attribútumokként nevezzük, amelyeknek saját nevük és értékük van. Az elemeknek lehetnek alárendelt vagy testvér elemek, amelyek megadják a fájl szerkezetét és a tartalmát.

Az XML fájlokhoz általában egy XML sémát is rendelünk, amely leírja az adatok struktúráját és típusait. Ez segít abban, hogy az adatokat pontosan értelmezni tudjuk és az adatokat az XML sémában meghatározott formátumba alakítsuk.

Az XML fájlokban lévő adatok széles körben használják az adatcsere és adatintegráció területén, például a webes szolgáltatásokban és az adatbázis rendszerekben. Az XML-t számos nyelv, keretrendszer és programozási nyelv támogatja, így az adatok egyszerűen elérhetők és felhasználhatók a különböző platformokon.

TXT

A TXT fájlok (vagyis a plain text fájlok) olyan szöveges állományok, amelyek tartalmukat egyszerű karakterek sorozataként tárolják. A TXT fájlok bármilyen szöveges adatot tartalmazhatnak, például szöveges üzeneteket, számokat, adatbázis-kimeneteket stb.

A gépi tanulásban a TXT fájlok gyakran adathalmazok forrásaként szolgálnak, amelyekben minden sornak egy-egy adatsorra kell esnie. Ezeket az adatsorokat később feldolgozzuk, és betanítjuk velük a modelleket.

Például, ha egy adathalmazban az árakat szeretnénk betanítani egy regressziós modell segítségével, akkor az adathalmazunk minden sora lehetne egy árérték (pl. 10.99, 15.99, stb.) egy-egy termék esetében.

PNG

A PNG (Portable Network Graphics) egy formátum a digitális képek tárolására és megjelenítésére, amelyet az áttetszőség és az alfa-csatornák támogatása jellemzi. Az alapja a bitmap (raszteres) grafikus formátum, azaz egy képet egy előre definiált számú képpont, azaz pixel alkot.

A PNG formátum támogatja az ős-színeket (RGB), az áttetszőséget (RGBA), az indexelt színeket (paletta módban), a szürkeárnyalatos képeket és az alfa-csatornákat (transzparencia), ami azt jelenti, hogy a kép minden pixeléhez rendelhetünk egy átlátszósági értéket.

YAML

Az YAML (Yet Another Markup Language) egy ember- és gépközeli szöveges adatformátum, amely az XML és a JSON alternatívájaként terjedt el az utóbbi években. Az YAML fájlok nagyon hasonlítanak a JSON-hoz, azonban szabadabban kezelik a whitespace karaktereket és könnyebben olvashatóak. Az YAML fájlok nagyon sok helyen használatosak, például konfigurációs fájlokban, adatbetöltésre és adatkimutatásra, de akár adatcsere formátumként is használhatók.

Az YAML fájlok alapstruktúrája három fő részből áll:

* Kulcs-érték párok: Egy kulcs-érték pár a fájlban egy kulcsot és az azt követő értéket jelöli meg, amely lehet egyszerű érték (pl. szám, string), lista vagy akár egy objektum.
* Listák: A listák az egyszerű értékek, objektumok vagy akár más listák sorozatait jelentik. A listák elemeit kötőjellel kezdjük a sor elején, majd egy szóköz után folytatjuk a listaelemet.
* Objektumok: Az objektumok összetett értékekből állnak, amelyek kulcs-érték párokként jelennek meg. Az objektumok elemeit kettős kacsacsőrök közé írjuk, majd a kulcs-érték párokat kettősponttal választjuk el.

A betanítás folyamata

01\_extract\_object\_info\_from\_xml.ipynb

Adatok előkészítése

Első lépésként a betanításra szánt adatokat elő kell készíteni. Ez magában foglalja a képek és azok címkézéseit, amelyek tartalmazzák az objektumok pozícióját és típusát.

Adatok felosztása

A betanító adatokat általában felosztják tanító- és validációs adatokra. A tanító adatokat használjuk a modell betanításához, míg a validációs adatokat a modell teljesítményének ellenőrzésére.

yolo\_training.ipynb

YOLOv5 telepítése

A YOLOv5 telepítése a betanítás előtt szükséges. Ehhez a pip segítségével telepíthetjük.

Konfiguráció

A YOLOv5 betanítása során konfigurálnunk kell a modellt. A konfigurációs fájlok adatait általában a következőképpen adhatjuk meg:

### data.yaml

Ez a fájl tartalmazza a betanító és validációs adatok útvonalát, valamint az objektum típusokat és azok számát.

### train.yaml

Ez a fájl tartalmazza a betanítási beállításokat, például az epochok és a batch méretét.

Modell betanítása

A modell betanításához a következő parancsot kell futtatni:

*python train.py --data path/to/data.yaml --cfg path/to/train.yaml --weights yolov5s.pt*

Ez a parancs elindítja a betanítást, és a modell paraméterei az epochok előrehaladtával frissülnek.

Modell kiértékelése

A betanítás után érdemes kiértékelni a modell teljesítményét a validációs adatokon. A kiértékeléshez használható a következő parancs:

*python test.py --weights path/to/best/weights.pt --data path/to/data.yaml --task val --img-size 640*

Modell finomhangolása

Ha a modell teljesítménye nem megfelelő, akkor finomhangolást lehet végezni. Ehhez a betanítás előrehaladtával csökkenteni kell a tanulási arányt.

A képen szöveg látható

Automatikusan generált leírás

A képen asztal látható

Automatikusan generált leírásA képen szöveg, diagram, sor, Diagram látható

Automatikusan generált leírás

A képen kollázs, ruházat, képernyőkép, Fotómontázs látható

Automatikusan generált leírás

Valós idejű objektum detektálás

yolo\_predictions.py

Az osztály inicializálásakor az alábbi műveleteket végzi el:

* Betölti a YAML fájlt, amely tartalmazza a modell osztályneveit és a címkéket.
* Betölti a YOLO modellt ONNX formátumban.
* Beállítja a preferált OpenCV backend-et és célplatformot.

Az **predictions** metódus segítségével előrejelzéseket készít a megadott képre. A következő lépések vannak a metódusban:

* A bemeneti képet négyzet alakúvá alakítja, hogy megfeleljen a modell által elvárt méretnek.
* Az átalakított képet átalakítja blob formátumra a YOLO modell számára.
* Végrehajtja a detektálást a YOLO modell segítségével, és kap egy predikciós eredményt.
* A predikciós eredményeket szűri az elhinni kívánt bizonyossági küszöbök alapján.
* Végrehajtja a nem maximális elnyomást (NMS) az átfedő detektációk csökkentése érdekében.
* Kirajzolja a határoló dobozokat és a címkéket a képen az eredmények megjelenítéséhez.

Az **onnx\_model** és **data\_yaml** paraméterekkel inicializálva ez a kódrészlet lehetővé teszi a YOLO objektumdetektálás végrehajtását az adott modellre és adatfájlra. A **predictions** metódus segítségével előrejelzéseket készít a képre és visszaadja a képet, amelyen a detektált objektumokat a határoló dobozokkal és címkékkel jelölte.

02\_predictions.ipynb

Ez a kódrészlet lehetővé teszi a videó feldolgozását képkockánkként YOLO objektumdetektálással, és megjeleníti a detektált objektumokat az eredeti videóban.

* Inicializál egy videó fájl (video.mp4) megnyitására egy **cv2.VideoCapture** objektummal.
* Egy végtelen ciklusban olvasza a videóból a képkockákat.
* Ellenőrzi, hogy sikeres volt-e a képkocka olvasása. Ha nem, akkor kilép a ciklusból.
* A **yolo.predictions** metódust használva előrejelzéseket készít a képkockára.
* Az előrejelzéseket megjeleníti az ablakban a **cv2.imshow** segítségével.
* Vár egy billentyűlenyomásra 1 ezredmásodpercig. Ha az Esc (27) billentyűt nyomják le, akkor kilép a ciklusból.
* Bezárja az ablakot a **cv2.destroyAllWindows** függvénnyel.
* Felszabadítja a videóforrás erőforrásait a **cap.release** függvénnyel.

Webalkalmazás az elvégzett feladathoz

Streamlit bemutatása

Streamlit egy nyílt forráskódú Python keretrendszer, amely lehetővé teszi az adatok interaktív és gyors vizualizációját és alkalmazások készítését. Segítségével könnyedén létrehozhat és megoszthat adatvizualizációs és adatelemző alkalmazásokat egyetlen Python fájlban.

A Streamlit egy egyszerű és deklaratív API-t nyújt, amely lehetővé teszi a fejlesztők számára a vizuális elemek, például gombok, csúszkák, grafikonok, táblázatok és térképek hozzáadását az alkalmazáshoz. Az alkalmazásokat valós időben frissítik, így a felhasználók azonnal láthatják a változásokat.

A Streamlit könnyen telepíthető a pip segítségével, és egyszerűen futtatható a parancssorból a streamlit run parancs segítségével.

Alkalmazás felépítése – Főoldal

A főoldalon pár mondatban leírtam a webalkalmazás működését, amelynél felsoroltam mind a 20 csoportot (tárgyak, állatok stb.) amelyeket képes detektálni az alkalmazás.

Az oldalhoz tartozó kódrészleteket a Home.py tartalmazza.

A képen szöveg, képernyőkép, szoftver, Weblap látható

Automatikusan generált leírás

Alkalmazás felépítése – Képfeldolgozás

Az említett oldalon tetszőleges képeket lehet feltölteni az oldalra, amely képes kielemezni a megadott fényképeket.

Az oldalhoz tartozó kódrészleteket a 1\_YOLO\_for\_image.py tartalmazza.

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Automatikusan generált leírás

Alkalmazás felépítése – Videó feldolgozás

Lehetőség van élő videóképet is detektálni az alkalmazással, amelynél a kamera eszköz kijelölése után elindíthatjuk a folyamatot.

Az oldalhoz tartozó kódrészleteket a 2\_YOLO\_webrtc.py tartalmazza.

A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Automatikusan generált leírás

[A feladat leírása 2](#_Toc134826875)

[Használt technológiák 2](#_Toc134826876)

[Programozási nyelv 2](#_Toc134826877)

[Kiegészítő csomagok 3](#_Toc134826878)

[LabelImg – Kiegészítő eszköz 4](#_Toc134826879)

[YOLO – Kiegészítő eszköz 5](#_Toc134826880)

[A betanítás során alkalmazott fájltípusok 6](#_Toc134826881)

[XML 6](#_Toc134826882)

[TXT 7](#_Toc134826883)

[PNG 7](#_Toc134826884)

[YAML 7](#_Toc134826885)

[A betanítás folyamata 8](#_Toc134826886)

[Adatok előkészítése 8](#_Toc134826887)

[Adatok felosztása 8](#_Toc134826888)

[YOLOv5 telepítése 8](#_Toc134826889)

[Konfiguráció 8](#_Toc134826890)

[data.yaml 8](#_Toc134826891)

[train.yaml 9](#_Toc134826892)

[Modell betanítása 9](#_Toc134826893)

[Modell kiértékelése 9](#_Toc134826894)

[Modell finomhangolása 9](#_Toc134826895)

[Valós idejű objektum detektálás 11](#_Toc134826896)

[Webalkalmazás az elvégzett feladathoz 13](#_Toc134826897)

[Streamlit bemutatása 13](#_Toc134826898)

[Alkalmazás felépítése – Főoldal 13](#_Toc134826899)

[Alkalmazás felépítése – Képfeldolgozás 14](#_Toc134826900)

[Alkalmazás felépítése – Videó feldolgozás 14](#_Toc134826901)