Pythonda rasmdagi obyektlarni aniqlaymiz - Deep Machine Learning

Python - bu eng istiqbolli tillardan biri bo'lib, bu sizga sun'iy intellektni hayotga tadbiq qilish imkonini beradi. Maqolada biz **Python** va **ImageAI**dan foydalanib obyektni tanib olishni yaratamiz.

Kompyuterlar va dasturlashning kelajagida asosiy o'ringa ega bo'lgan yo'nalishlardan biri bu - **Computer vision ("kompyuter ko'rishi")**dir . Bu kompyuterning rasmning mohiyatini tanib olish va aniqlash qobiliyatiga erishishini bildiradi. Bu sun'iy intellektning eng muhim sohasi bo'lib, o'z ichiga bir nechta harakatlarni: fotosurat tarkibini aniqlab olish, obyektni ko'rish va uning tasnifi yoki turiga (oilasiga) aniqlik kiritish oladi. **Rasmdagi obyektlarni izlash,** ehtimol, **"kompyuter ko'rishi"**ning eng muhim sohasidir.

Rasmlardagi narsalar yoki tirik mavjudotlar ta'rifi quyidagi sohalarda faol qo'llaniladi:

- **Avtomobil qidirish;**
- **Odamlarni aniqlash tizimi;**
- **Piyodalarning qidirish va sonini hisoblash;**
- **Xavfsizlik tizimini mustahkamlash;**
- **Uchuvchisiz transport vositalarini yaratish va boshqalar.**

Bugungi kunda qo'llanish yo'nalishiga qarab ob'ektlarni topish uchun ko'plab usullarni ishlab chiqish mumkin bo'ldi. Ushbu sohada, IT texnologiyalaridan foydalanishning boshqa sohalarida bo'lgani kabi, ko'p narsa dasturchiga bog'liq. Bu ijodkorlik uchun juda yaxshi vosita bo'lib, uning yordamida **" ijod mahsuli "** o'z ongigaga ega bo'lishi mumkin. Dastur intellektidan qanday foydalanish foydalanuvchininhning ijodiy fikrlashiga bog'liq.

Natija

Texnologiyalar olamida sun'iy intelekt tushunchasida haqiqatan ham inqilob qildi. Kelajakda bu **R-CNN** , **Fast-RCNN** , **Faster-RCNN** , **RetinaNet** usullar uchun asos bo'ldi . Ular orasida yuqori aniqlikdagi, tezkor usullar - bu SSD va YOLO. Chuqur o'rganishga asoslangan yuqoridagi algoritmlardan foydalanish uchun siz matematikada chuqur bilimga ega bo'lishingiz kerak.

Boshlaymiz

Bugun ishlatadigan kutubxonamiz - bu ImageAIbo'lib, u ham Pythonda yozilgan.

Pythonni o'rnatish

Python bu yerda ajralmas hisoblanadi. Siz shunchaki o'rnatuvchi faylni rasmiy saytdan yuklab olishingiz kerak va o'rnatish jarayonini amalga oshiramiz.

Kerakli modullarni o'rnatamiz

Endi `pip paket mejjeri`orqali foydalanadigan modullarimizni o'rnatamiz. O'rnatish printsipi juda ham oddiy: **`pip install`**va **`modullning nomi`** (modullarni nomini quyidan topishingiz mumkin). Bu quyidagiga o'xshaydi:

```
pip install tensorflow # Tensorflow dasturiy ta'minotini o'rnatamiz
```

Qaysi modullarni o'rnatishimiz kerak:

- **Numpy;**
- **SciPy;**
- **OpenCV (opencv-python);**
- **Pillow;**
- **Matplotlib;**
- **H5py;**
- **Keras;**
- **ImageAI (link)**

Siz rasmiy ImageAI hujjatlari veb-saytida barcha modullar va ularni o'rnatish haqidagi to'liq ma'lumotlarni ko'rishingiz mumkin.

- **Python** 3.5.1 or higher, Download Python
- **pip3** , Download PyPi
- **Tensorflow** 1.4.0 or higher

```
pip3 install --upgrade tensorflow
```

- **Numpy** 1.13.1 or higher

```
pip3 install numpy
```

- **SciPy** .19.1 or higher

```
pip3 install scipy
```

- **OpenCV**

```
pip3 install opencv-python
```

- **Pillow**

```
pip3 install pillow
```

- **Matplotlib**

```
pip3 install matplotlib
```

- **h5py**

```
pip3 install h5py
```

- **Keras**

```
pip3 install keras
```

Retina Net

Endi yuklab olish kerak faylRetina Net modeli uchun . U tasvirlardagi obyektlarni aniqlash jarayonida ishtirok etadi. Bu faylda dasturimiz aniqlay oladigan narsalarning tahsifi berib o'tilgan. Aniqroq esa ImageAI hujjatlari da tanishib chiqishingiz mumkin.

Modullaro'rnatilgandan so'ng, rasmlardagi obyektlarni aniqlash uchun birinchi kod satrlarini yozish mumkin. `.py` kengaytmali **FirstDetection** faylini yaratishimiz kerak . Quyidagi kodni yaratilgan faylga qo'ying. Bundan tashqari, **Retina** modelining faylidan(ya'ni yuqorida yuklab olgan faylimizdan) nusxa olamiz va **Python** fayli bilan bir katalogga joylashtiramiz. So'ngra tekshiriladigan rasmni o'sha katalogga joylashtirmiz, ya'ni barchasi bir katalogda bo'lishi lozim.

Sinov

Fayl yarating va quyidagi kodni joylashtiring:

```
from imageai.Detection import ObjectDetection
import os

exec_path = os.getcwd()

detector = ObjectDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath(os.path.join(
       exec_path, "RETINA NET MODELI FAYLI")
)
detector.loadModel()

list = detector.detectObjectsFromImage(
       input_image=os.path.join(exec_path, "TEKSHIRILADIGAN FAYL.jpg"),
       output_image_path=os.path.join(exec_path, "TEKSHIRILGAN FAYL.jpg"),
       minimum_percentage_probability=90,
       display_percentage_probability=True,
       display_object_name=False
)
```

Kodni ishga tushirish va konsolda natijalar paydo bo'lishini kutish qoladi.
Keyinchalik, `FirstDetection.py` fayl turgan katalogga o'ting . Bu erda yangi yoki bir nechta rasm paydo bo'lishi kerak. Nima bo'lganini yaxshiroq tushunish uchun siz asl va yangi rasmni ochishingiz kerak.

Kod qanday ishlashini ko'rib chiqamiz:

```
from imageai.Detection import ObjectDetection
import os

exec_path = os.getcwd()
```

Qatorlarning tavsifi:

- **1 qator:** obyektlarni qidirish uchun `ImageAI`modulidan va `ObjectDetection` sinfni ulaymiz;
- **2-qator:** Pythonda o'rnatilgan `os` modulini ulaymiz;
- **4-qator:** Python, RetinaNet, model va rasmli fayllar joylashga katalogga yo'l ko'rsatiladigan o'zgaruvchini yaratamiz.

```
detector = ObjectDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath(os.path.join(
       exec_path, "resnet50_coco_best_v2.0.1.h5")
)
detector.loadModel()

list = detector.detectObjectsFromImage(
       input_image=os.path.join(exec_path, "objects.jpg"),
       output_image_path=os.path.join(exec_path, "new_objects.jpg"),
       minimum_percentage_probability=90,
       display_percentage_probability=True,
       display_object_name=False
)
```

Qatorlarning tavsifi:

- **1 qator:** Obyektlarni qidirish uchun yangi sinfni e'lon qilamiz;
- **2 qator:** RetinaNet modelining turini belgilaymiz;
- **3-qator:** RetinaNet modeliga yo'lni ko'rsatamizi;
- **6-qator:** modelni qidirish uchun sinf ichida yuklaymiz;
- **8 qator:** aniqlash funktsiyasini chaqirish (*ob'ektlarni aniqlash*) va boshlang'ich va oxirgi rasmlarning yo'llarini parsing qilishni boshlaymiz.

ImageAIda obyektlarni topish uchun ko'plab turli xil sozlamalarmavjud. Masalan, rasmga ishlov berish jarayonida barcha topilgan obyektlarning ajratilishini va alohida massivga joylashirishni sozlashingiz mumkin. Qidiruv **sinfi** `image` deb nomlangan alohida katalog yaratishi, so'ngra barcha obyektlarga bo'lgan yo'l bilan ketma-ketlikni chiqarib, saqlashi va qaytarishi mumkin:

```
list, extracted_images = detector.detectObjectsFromImage
       (input_image=os.path.join(execution_path , "objects.jpg"),
       output_image_path=os.path.join(execution_path , "new_objects.jpg"),
       extract_detected_objects=True)
```

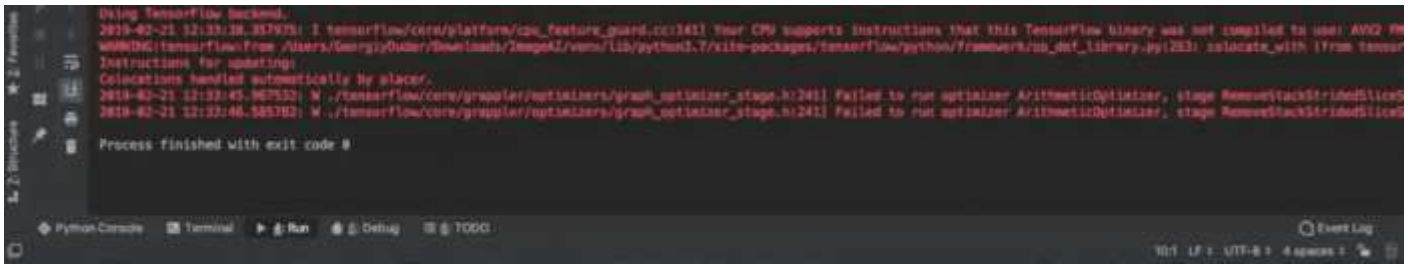
Natija

Albatta shuncha gapdan so'ng praktikasiz yakunlasak to'g'ri bo'lmaydi! ☺

Dasturning asl imkoniyatlarini tekshirish uchun ancha qiyinroq rasm tanlashga harakat qildim. Lekin modullning imkoniyatlari bundan ham ko'prog'iga qodir.

Sinov uchun rasm

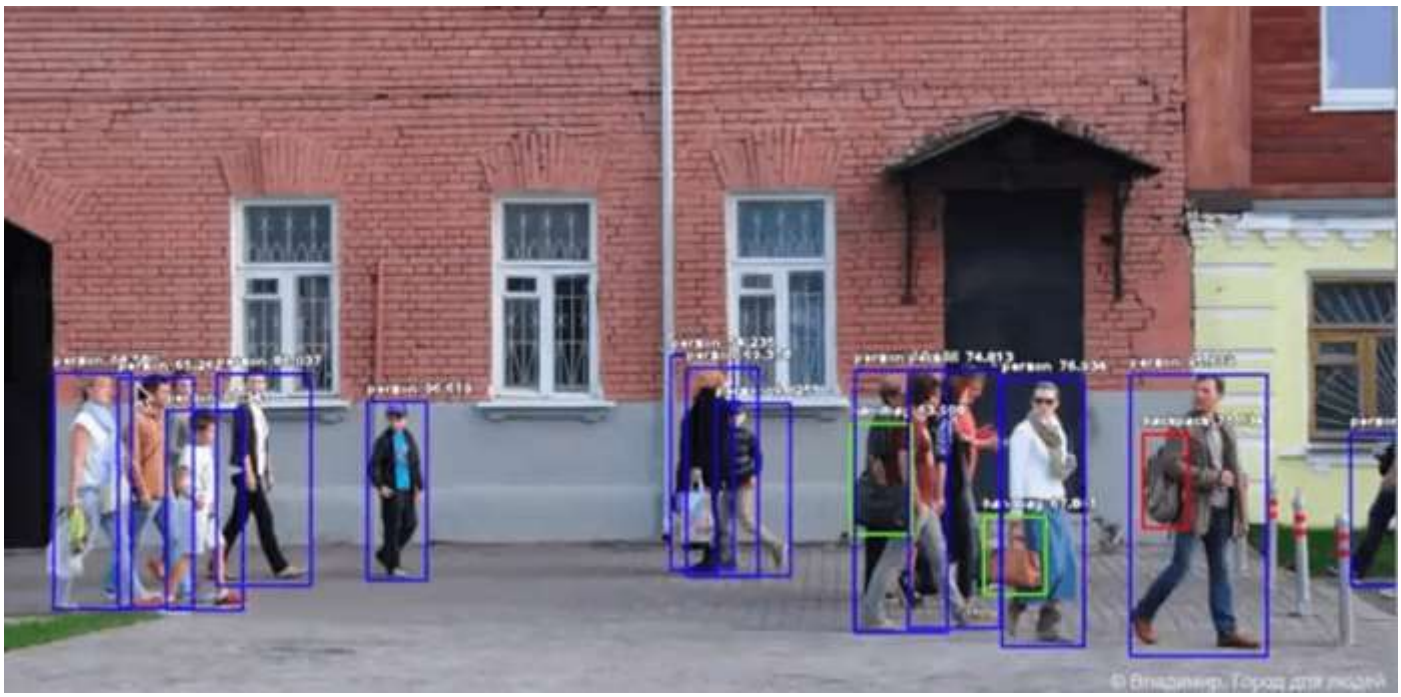
Rasmni **python fayl** joylashgan katalogga joylashtiramiz. So'ngra dasturni ishga tushiramiz:



Dasturni ishga tushuramiz

Ishga tushurish jarayonida ogohlantirishlar mumkin. Lekin e'tior bermasa ham bo'ladi.

Dastur yakullangach biz tahminan quyidagidek natija olamiz:



Shu bilan bugun sizga ko'rsatmoqchi va o'rgatmoqchi bo'lgan narsalarim shulardan iborat edi.

Agarda siz bu maqolamni oxiragacha o'qigan bo'lsangiz sizga katta rahmat aytmoqchiman.

Xayr, omon bo'ling!

Maqola muallifi: Azizmurod Murodov