

YOLO ile Kiři yada Nesne Algılanması Çalışması

Bu çalışma örneğinde gerçek zamanlı nesne tespiti kütüphanesi olan YOLO ile kendi nesnelerimi tanıtip gerekli işlemleri yaparak modelimi eğitip çalışmamın sonuçlarını test ediyorum.

Bu çalışmada Barış Manço'nun profil resimlerini tespit etmeye çalıştım.

Çalışmanın işlem adımları;

1- Veriseti Görsellerini Oluşturdum

Google görsellerden indirdiğim Barış Manço resimlerini Darknet klasörü içerisindeki **data** klasörü altına **baris_dataset** adıyla yeni klasör oluşturarak kaydediyorum.

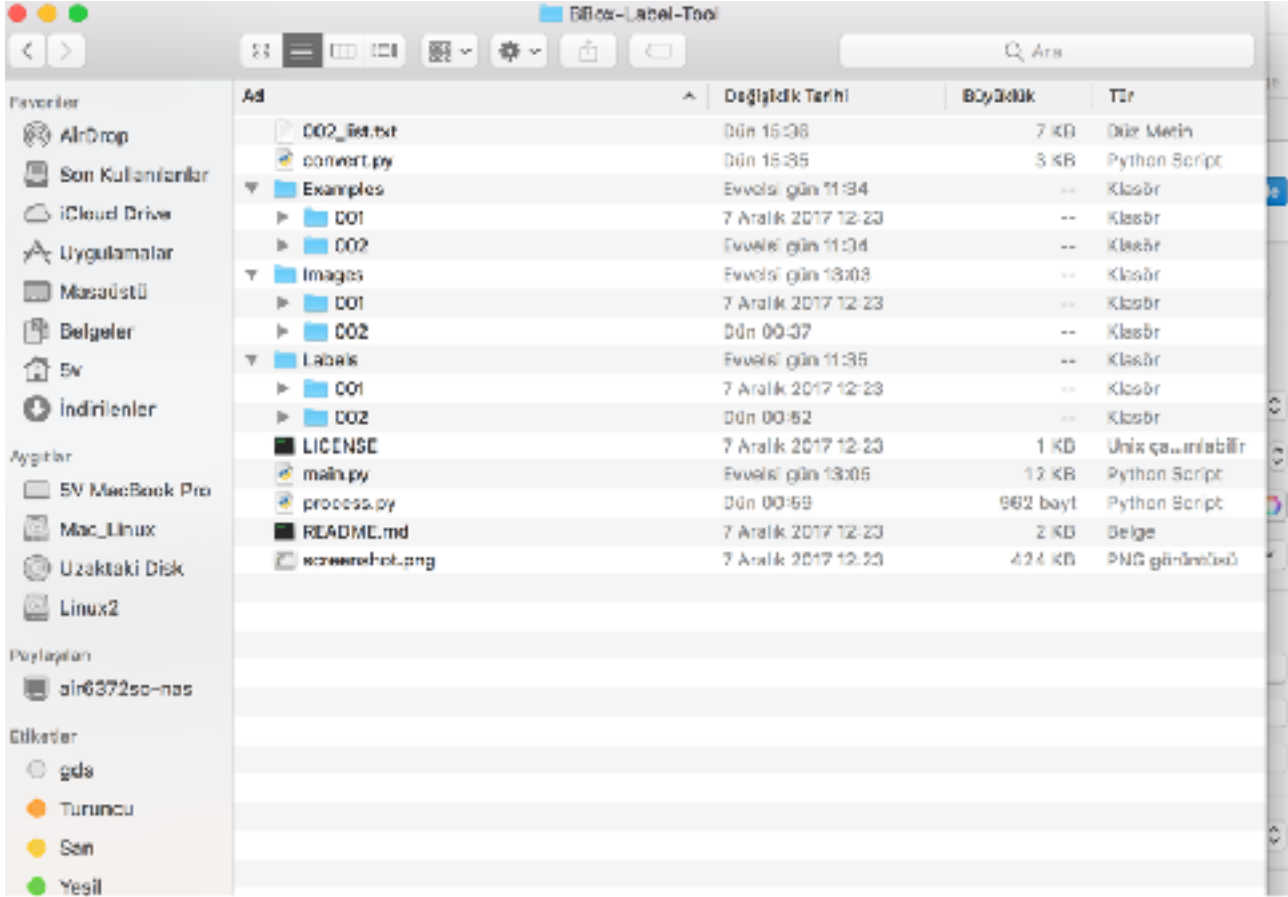
2- Tanınmasını İstediğim Kişiyi/Nesneyi Çerçeve İçerisine Alma

YOLO ile nesne tespiti için indirdiğimiz resimler içerisindeki tanımak istediğim kişi yada nesnenin çerçevesini belirlemem gerekiyor. Bunun için [BBBox Label Tool](#) aracını kullanıyorum. Öncelikler Github reposunu kendi bilgisayarıma klonlayarak indirdim.

Ayrıca BBox Label Tool **.JPEG** uzantılı resim dosyalarını ile işlem yaptığı için ve benim resim dosyalarımda **.jpg** olduğundan dolayı BBox Label Tool klasörü içerisindeki **main.py** dosyasının içeridindeki **.JPEG** değerlerinin hepsini **.jpg** olarak güncelledim.

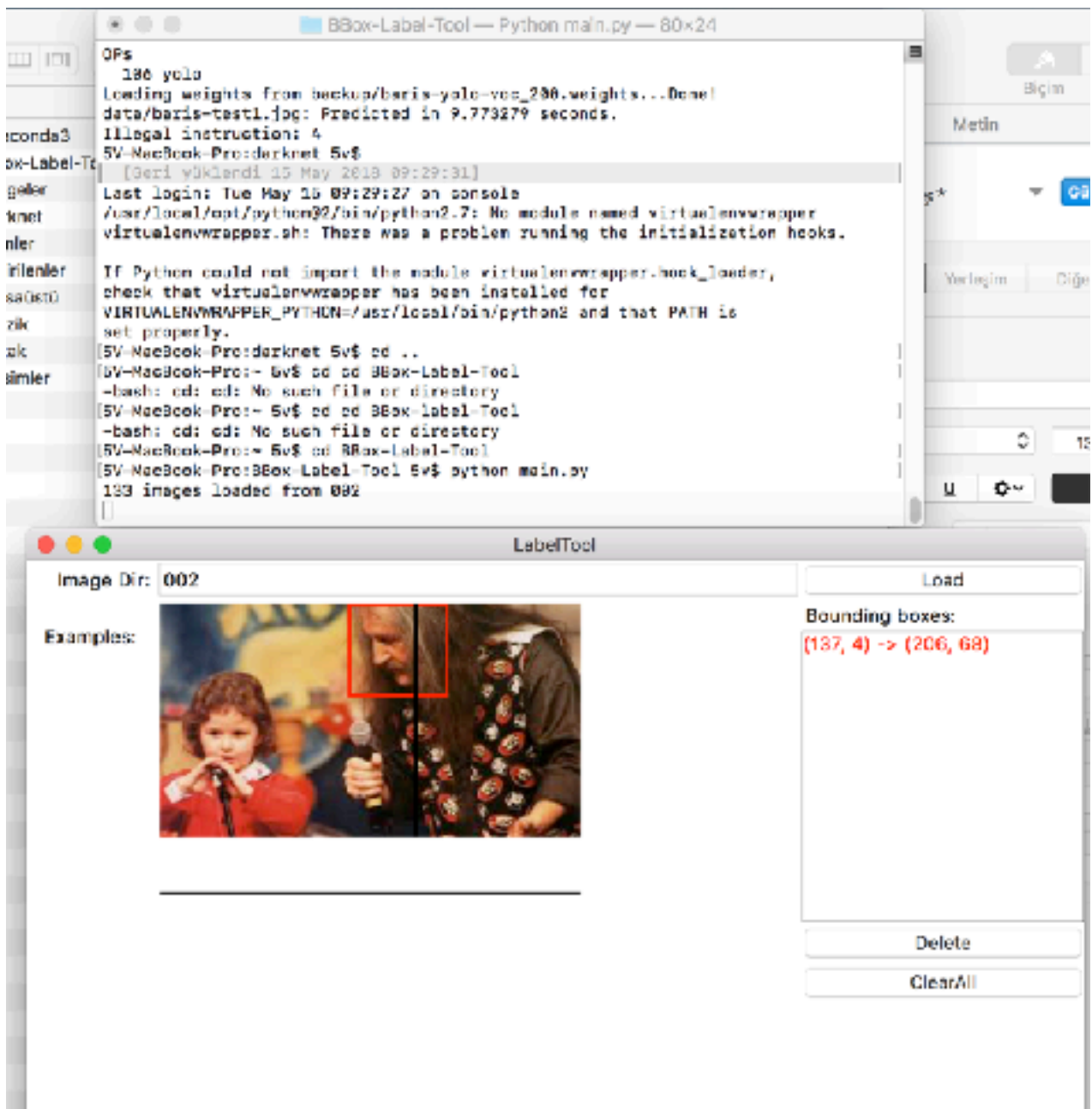
BBox-Label-Tool klasörü altında bulunan (Examples - Images - Labels) klasörlerine benim işlem yapacağım işlemlerin tanımlanması ve depolanması için bu 3 klasör altına, rakamla 002

isminde yeni dosyalar açtım. **BBox-Label-Too** klasörü içerisindeki Images klasörü altına yeni oluşturduğum 002 klasörü altına kendi oluşturduğum verisetimin resimlerini ekledim.



3- Resimlerin Etiketlenmesi işlemi

~/Users/5v/BBox-Label-Tool\$ python main.py komutunu çalıştırdığımda ekrandaki ekran geliyor karşıma ve bu ekran üzerinden tanınmasını istediğim kişi(Bariş Manço)'nın resimlerini etiketliyorum.



4- Nesne Çerçeve Bilgisini YOLO'ya Uygun Hale Dönüştürme

BBox Label Tool ile oluşturduğum .txt dosyaları içerisindeki koordinat bilgileri YOLO kütüphanesi tarafından desteklenmiyor, bu sebepten dolayı .txt dosyası içerisindeki koordinat bilgilerini YOLO'ya uygun yani 0-1 formatına dönüştürüyorum.

Bu .txt'ler içerisindeki değerleri YOLO'nun sevdiği aşağıdaki formata dönüştürmem ve piksel değeri yerine 0-1 arasında oran belirtmem gerekiyor.

Bu formata dönüştürmek için internetten bulduğum Guanganh Ning'nin yazdığı scriptinden yararlandım. İlgili **convert.py** script dosyasını kullandım.

Convert.py dosyası içerisinde bazı değişiklikler yaptıktan sonra Convert.py dosyasını çalıştırıyorum ve dosyada belirttiğim çıktı yolunda 0-1 değerleri olarak düzenlenmiş .txt dosyaları oluşuyor.

```
~/Users/5v/BBox-Label-Tool$ python convert.py
```

5- Eğitim ve Test Veri Kümesini Oluşturma

Bu aşamada oluşturduğum verisetinden ne kadar resimin eğitim ve test için kullanmasını istediğimi belirtiyorum.

Eğitmek için oluşturduğum resim ve YOLO'ya uygun formatta düzenlenmiş koordinat bilgilerini içeren .txt dosyalarının bulunduğu klasör ("**/Users/5v/darknet/data/baris_dataset/**") altına, eğitim ve test kümelerini ayırmama yarayacak olan script yardımı ile bu işlemi gerçekleştiriyorum.

process.py scriptinin dosyasının 7.satırındaki **path_data** değerini '**data/baris_dataset/**' olarak resim ve .txt dosyalarının bulunduğu veriseti klasörü değiştirdim. Ve scriptim çalıştırdığımda bana eğitim için %90 ve test için %10 veriseti yolunu içerecek train.txt ve test.txt dosyalarını oluşturuyor.

6- YOLO için Yapılandırma Dosyalarını Hazırlama

Kendi verisetim için darknet klasörü altındaki cfg klasörü altında bulunan **obj.data**, **obj.names**, **yolo-obj.cfg** dosyalarını kendi adıma oluşturuyorum.

Yeni bir **baris-obj.data** dosyası oluşturuyorum ve aşağıdaki şekilde düzenliyorum.

.data dosyası, kaç sınıf eğiteceğimi, test ve doğrulama kümelerinin dosya yollarını, eğiteceğim sınıfın etiket adının bulunduğu dosyanın yolunu ve eğitim sırasındaki yedek dosyalarının bulunacağı klasör bilgisini belirtiyorum.

Örneğin;

```
1 classes= 1
2 train  = /Users/5v/darknet/data/baris_dataset/train.txt
3 valid  = /Users/5v/darknet/data/baris_dataset/test.txt
4 names  = /Users/5v/darknet/cfg/baris-obj.names
5 backup = /Users/5v/darknet/backup/
```

Yeni bir **baris-obj.names** oluşturarak içerisine etiket adını yani “Barış Manço” yazdım.

Son olarak hazırda var olan **yolo-voc.cfg** dosyasının bir kopyasını oluşturup dosya adını **baris-yolo-voc.cfg** değiştirdim.

baris-yolo-voc.cfg ağ dosyasını aşağıdaki şekilde düzenledim.

- 3. satırdaki **batch** değerini 64 olarak güncelleyelim (siz istediğiniz şekilde ayarlayabilirsiniz)
- 4. satırdaki **subdivisions** değerini 8 olarak güncelleyelim(siz gpu ve cpu işlemci sayınıza göre ayarlayabilirsiniz)
- 237. satırdaki $\text{num}/3 * (\text{classes} + 5) * 5$ formülüne uygun olarak bizim sınıf sayımız 1 olduğundan dolayı $(1+5) * 5 = 30$ olduğundan **filters=30** olarak güncelleyelim
- 244.satırdaki **classes** değerini bizim sınıf sayımız 1 olduğundan 1 olarak güncelleyelim.

Artık eğitim işlemini başlatmam için son bir adım kaldı.

Sıfırdan eğitmek yerine Imaget veriseti ile önceden eğitilmiş ağırlıkları kullanarak modelimi eğittim.

Ağırlık dosyasını **ekteki** linkten indirerek bu eğitilmiş dosyanın ağırlıklarını kullandım.

Artık son aşama olarak eğitmiş olduğum modeli test ettim,

7- Eğitilmiş Modeli Test Ediyorum

Test ve train veriseti içerisinde bulunmayan başka bir resim indirip ve en son kaydedilen baris-yolo-voc_200.weights ağırlık dosyasını kullanarak örnek resmin içerisindeki Barış Manço'yu tanıyarak çerçeve içerisine alıp alamayacağını test ettim ancak kontrollerim sonunda Mac Os'ta bulunan bir bug(hata) sebebi ile ilgili test işlemi başarılı sonuç vermiyor.

