Android'de Yapay Zeka ve Tensorflow Kütüphanesi

Caner Konuk
Matematik ve Bilgisayar Bölümü
Eskişehir Osmangazi Üniversitesi
Fen-Edebiyat Fakültesi
Meşelik Kampüsü, 26480, Odunpazarı, Eskişehir
canerkonuk@gmail.com

Özet

Bu proje ile android işletim sisteminde yapay zeka kütüphanelerinin incelenmesi amaçlanmış olup, android işletim sistemi üzerinde en çok kullanılan yapay zeka kütüphanelerinden biri olan tensorflow kütüphanesi üzerinde durulmuştur. Öncelikle android işletim sisteminde bulunan yapay zeka uygulamaları ile ilgili genel örnekler bulunmuştur. Bu örnekler üzerinden android işletim sistemi üzerinde tensorflow kütüphanesinin daha yaygın bir biçimde kullanıldığı görülmüştür. Buradan yola çıkarak belirlenen YOLO(You Look Only Once) isimli android tensorflow uygulamasının model dosyalarının normalden daha da geliştirilmiş bir hale getirilmesi sağlanmıştır.

Anahtar Kelimeler – Android, Tensorflow, Python, Yolo Tensorflow

I. GİRİŞ

Proje android işletim sisteminde yapay zeka alanının gelişimi ile ilgilir. Bu proje sayesinde tensorflow kütüphanesi ile android işletim sistemi üzerinde neler yapılabildiğinin görüntülenmesi sağlanmıştır. YOLO örneğinin tam olarak yaptığı işlev gerçek zamanlı nesne tanımlanmasıdır. Örneğin orjinalinde varsayılan model dosyası sadece 20 tanımlı nesne içermektedir. Fakat burada model dosyası yeni bir model dosyası yaratılıp bu dosya ile değiştirilmiştir. Böylelikle uygulama 20 nesneyi tanımlamak yerine 80 nesneyi tanımlar hale gelmiştir.

II. VERİ SETİNİN TOPLANMASI

Bu proje için Darknet Yolo veri seti kullanılmıştır. Önceden eğitilmiş olan modelin ağırlık dosyaları(.weights) ile önceden hazırlanmış ağ dosyaları(.cfg) ile tensorflow aracılığı ile birleştirilerek protobuff(.pb uzantılı model) modeli ortaya çıkartılmıştır. Bu modeli çıkartmamızın amacı android studio platformu üzerinde protobuff modellerinin daha kullanışlı olmasıdır. Protobuff modelini elde ederken yanında gelen .meta uzantılı dosya sayesinde de modellerin hangi nesneleri tanımladığını labellar aracılığı ile öğrenilmiştir.

III. VERİ SETİNİN KULLANIMI

Verilmiş olan tablo ile kullanılan ağırlık dosyası ve ağ dosyaları arasındaki kombinasyonlar ile hangi modellerin ortaya çıktığının gösterilmesi amaçlanmıştır.

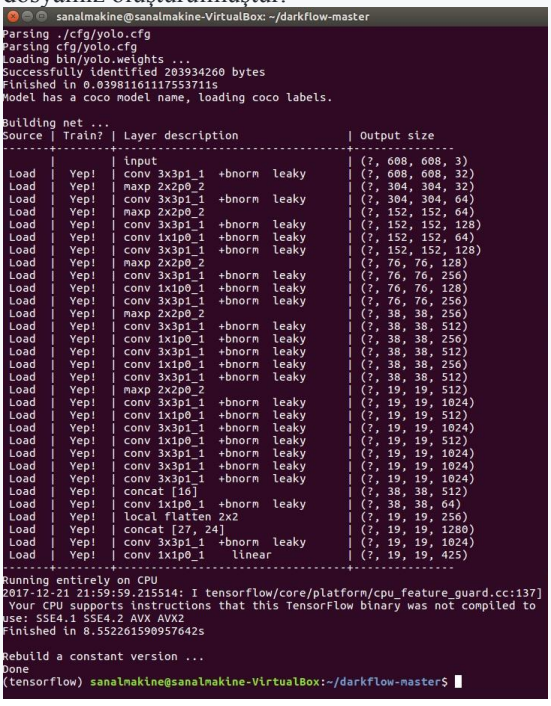
TABLO
MODEL DOSYALARININ OLUŞUMU VE VERİ SETLERİNİN KULLANIMI

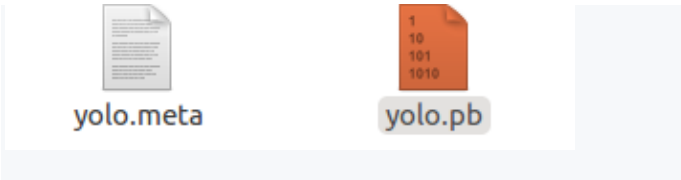
Kullanılan ağırlık dosyası	yolo.weights
Sırası ile kullanılan ağ dosyaları	yolo.cfg, yolo-voc.cfg, tiny-yolo.cfg
Sırası ile ortaya çıkan .pb uzantılı modeller	yolo.pb, yolo-voc.pb, tiny-yolo.pb
Sırası ile ortaya çıkan .meta uzantılı label dosyaları	yolo.meta, yolo-voc.meta, tiny-yolo.meta

.

IV. MODELİN OLUŞTURULMASI

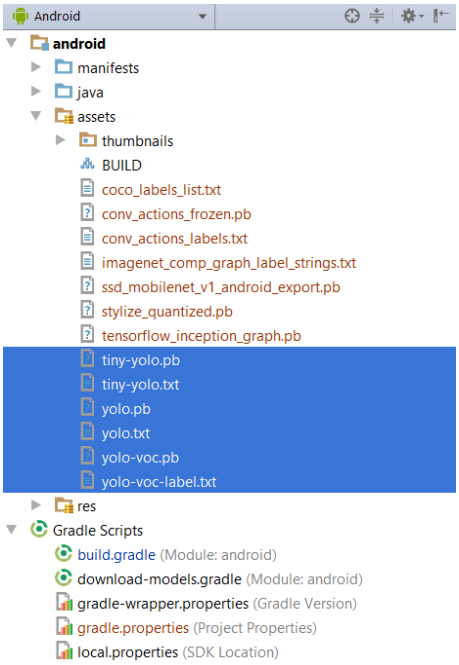
Modeli oluşturabilmemiz için python3 üzerine tensorflow r.1.4 sürümlü kütüphanesi dışında gerekli olan diğer kütüphanelerin(opencv 3.4.0 sürümü ve cython) kurulumu yapılmıştır. Ubuntu işletim sisteminde bulunan komut penceresine " python3 flow --model cfg/yolo.cfg --load bin/yolo.weights –savepb " komutu yazılarak(Bu komut yolo.cfg ve yolo.weights içindir.) aşağıdaki resimde bulunan çıktı elde edilmiştir. Bu çıktının sonunda resimde de görüldüğü üzere .pb uzantılı modelimiz ve .meta uzantılı label dosyamız oluşturulmuştur.





V. MODELİN ANDROİD STUDİO ÜZERİNDE KULLANIMI

Öncelikle android studio üzerinde yapay zeka modellerinin "assets" isimli klasör içerisine konulacağı belirlenmiştir. Protobuff (.pb) uzantılı model dosyalarını assets klasörü içine atıp .meta uzantılı label dosyasından da .txt uzantılı label belgesi elde edilmiştir. Labelların .txt uzantılı belge haline getirilmesinin nedeni android studio üzerinde hangi nesnelerin model içerisinde tanımlandığının daha kolay bir şekilde görüntülenmesini sağlamaktır.

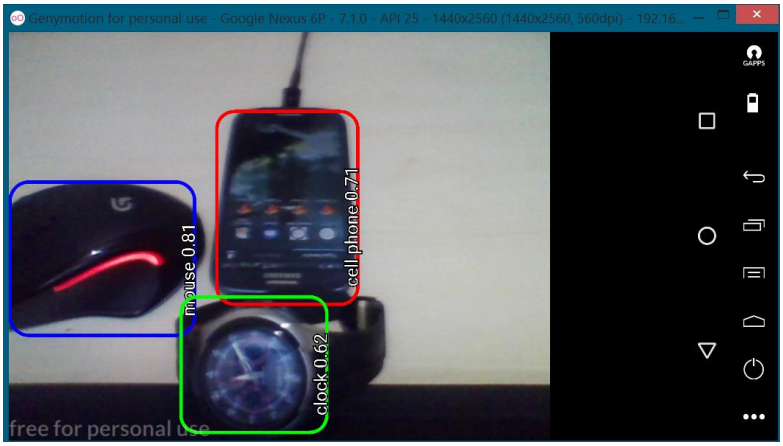


Bu işlemlerin ardından model dosyalarının uygulama üzerinde çalışması için android tensorflow örneğimizin DetectorActivity.java classı üzerinde değişiklikler yapılmıştır. Bu yapılan değişiklik, örneğin 20 nesne tanımlı model dosyasınının, kendi yarattığımız 80 nesne tanımlı .pb uzantılı model dosyası ile değişimini sağlamıştır. ("yolo.pb" isimli modelin kullanılabilmesi için yapılan değişiklik aşağıdaki kod satırında belirtilmiştir. Kod satırı diğer modellerin isimlerine göre değişiklik göstermektedir.)

```
private static final String
YOLO_MODEL_FILE =
"file:///android_asset/yolo.pb";
```

VI. SONUÇ

Sonuç olarak uygulama başarılı bir şekilde 20 nesne yerine 80 nesneyi gerçek zamanlı bir şekilde tanımlayacak hale getirilmiştir.



KAYNAKLAR

[1] https://www.tensorflow.org/mobile/android_build
[2] https://www.tensorflow.org/tutorials/image_recognition
[3] https://www.tensorflow.org/mobile/tflite/
[4] https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android
[5] https://pjreddie.com/darknet/yolo/
[6] https://github.com/pjreddie/darknet
[7] https://github.com/thtrieu/darkflow
[8] https://www.anaconda.com/download/#linux
[9] https://opencv.org/releases.html