



KONYA
TEKNİK ÜNİVERSİTESİ
1970

KONYA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
TASARIM-2
TINYML İLE KİŞİ ALGILAMA

HAZIRLAYANLAR:
HİLAL KELEŞ 181222020
TUĞBA GÖNCÜ 171222102

1.Giriş.....	3
1.1 Projenin Amacı.....	3
2.Tinyml Nedir?.....	3
3. Hangi donanımlara ihtiyaç duyulur?.....	3
3.1 Hangi yazılımlara ihtiyaç duyulur	5
4.Makine öğrenimi nedir?.....	6
5.Derin öğrenme nedir?.....	6
5.1 Derin Öğrenme Katmanları	8
5.1.1 Giriş (Input Katmanı).....	8
5.1.2 Konvolüsyon Katmanı.....	8
5.1.3 Aktivasyon Katmanı.....	8
5.1.4 Havuzlama Katmanı.....	8
5.1.5 Tam Bağlı Katman.....	8
5.1.6 DropOut Katmanı.....	8
5.1.7 Sınıflandırma Katmanı	9
5.1.8 Normalizasyon Katmanı.....	9
5.2 Derin Öğrenme Algoritmaları	9
5.3 Derin Öğrenme Uygulama Alanları.....	9
6.Model Seçimi	10
6.1 SSD Mobilenet.....	10
7.Data Nedir?.....	11
7.1 Data oluşturulurken neye dikkat edilir?.....	11
7.2DataSet Oluşturma	11
7.3 Nesne Etiketleme Araçları.....	15
8.LabelMap Oluşturma Ve Eğitim Ayaları.....	16
9.Hiperparametre Ayarlamaları Ve Seçimleri.....	17
9.1 Öğrenme Katsayısı.....	17
9.2 Mini Batch Boyutu.....	17
9.3 Eğitim Tur Sayısı.....	17
9.4 Loss Fonksiyonu.....	18

10.Proje Sonucu.....	18
11.Çıkarım.....	20
12. Projede Kullanılan Araçlar Ve Kütüphaneler.....	20
12.1 Anaconda.....	20
12.2 TensorFlow.....	20
13. Projede Kullanılan Cihazlar.....	21
13.1 Raspberry Pi 3 Model B+.....	21
13.1.1 Raspberry Pi Teknik Özellikleri.....	21
13.2 Raspberry Camera v1.3.....	21
13.2.1 Raspberry Camera Teknik Özellikleri.....	21

1.GİRİŞ

Bu rapor Konya Teknik Üniversitesi Mühendislik Ve Doğa Bilimleri Fakültesi Elektrik-Elektronik Mühendisliği Bölümü öğrencileri Hilal Keleş ve Tuğba Göncü tarafından Tasarım-2 dersi için hazırlanmıştır. Rapor ; tinymml'nin ne olduğunu ve tinymml ile kişi algılama projesinin detaylarını içermektedir.

1.1 Projenin Amacı

Yüksek güvenliğin önemli olduğu yerlerde video üzerinden siluetten insan tanıyarak daha güvenli bir ortam sağlamak ve görevli kişileri uyarmaktır.

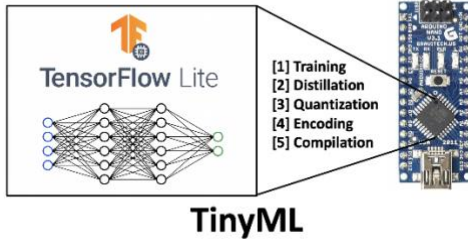


Şekil 1.1.1

2. Tinyml nedir?

Gelişen teknoloji ile birlikte uzun bir süredir kullandığımız cihazların çoğunda bir tür sinir ağı bulunmaktadır. Bu sinir ağı ortamda çalmakta olan müziği, fotoğrafı çekilmiş kişinin kim olduğunu tanıma gibi ve daha birçok uygulamaya olanak sağlar. Bunlar gömülü derin öğrenmeye dayanan örneklerden sadece birkaçıdır. Algoritmalar, tekrar kullanma olasılığımızın daha yüksek olduğu uygulamaları belirleyebilir ve gerekli olmayan uygulamaları kapatarak telefonun pil ömrü uzatılabilir. Bununla birlikte, gömülü AI için bazıları güç olan durumlar vardır. İşte TinyML'in devreye girdiği yer burasıdır. TinyML, son derece düşük güç tüketimine sahip gömülü sistemler ile birbirinden bağımsız olarak çalışan makine öğrenimi toplulukları arasındaki bir iş birliğidir. Tiny Machine Learning (veya TinyML), makine öğrenimi mimarileri, teknikleri, araçları ve yaklaşımları dahil olmak üzere "tam yığın" (donanım, sistem, yazılım ve uygulamalar) çözümleri gerektiren azaltılmış ve optimize edilmiş makine öğrenimi uygulamalarını entegre eden bir makine öğrenimi tekniğidir.

TinyML, otomatik görevleri gerçekleştirmek için sensörler veya mikro denetleyiciler gibi düşük enerjili sistemlere uygulanabilir. TinyML ile daha az olanak ile daha fazlası yapılabilir. Teknik olarak internet bağlantısı olmayan, daha az maliyetli ve daha az uğraş gerektiren makine öğrenimidir. Şekil 2.1



Şekil 2.1

3. Hangi donanımlara ihtiyaç duyulur?

Kullanacağımız gömülü cihazda derleme ve düzenlemeler yapacağımız için MacOS, Linux, Windows gibi işletim sistemlerine sahip, usb bağlantısı olan herhangi bir bilgisayar ihtiyaçlarımızı karşılayabilmektedir. Projede kullanacağımız veri kümelerini işleyebileceğimiz belli başlı platformların piyasaya sürülmesiye yüksek özellikte bilgisayara ihtiyaç duyulmamaktadır. Bu platformlara örnek olarak SaaS platformları (ör. Google Colaboratory) ve IaaS (ör. Amazon EC2 Bulut Sunucuları) verilebilir.

Ayrıca oluşturduğumuz programları test etmek için gömülü sistemlere ve projeye göre değişebilen harici parçalara ihtiyaç olacaktır. Bu doğrultuda piyasada proje ihtiyacını karşılayabilecek çeşitli alternatifler mevcuttur. Projenin özelliklerine ve ihtiyaçlarına göre doğru tercih yapılmalıdır. Örneğin kullanılacak dataset ve model büyük ise alanı büyük bir kart seçilmeli ya da proje kablosuz internete ihtiyaç duyuyorsa kablosuz internet özelliği olan bir kart seçilmelidir. İhtiyaçlar

belirlenip bu doğrultuda kart seçimi yapılmalıdır. Projenin ihtiyaçlarını karşılayabilecek piyasada bulunan birkaç kartı örnek verilecek olursa;

- Raspberry Pi (Şekil 3.1)



Şekil 3.1

Her yaştan insanın bilgisayarı keşfetmesini ve Scratch ve Python gibi dillerden nasıl programlanacağını öğrenmesini sağlayan kredi kartı boyutunda bir bilgisayardır. İnternet'te gezinmek ve yüksek tanımlı video oynatmaktan elektronik tablolar oluşturmaya kadar birçok işlemi yapabilir.

- Arduino (Şekil 3.2)



Şekil 3.2

Bileşenleri kodlamadaki ve kullanmada sağladığı basit yapısı, piyasada farklı ve her bütçeye uygun çeşitlerinin bulunabilmesi ve açık kaynak özellikleri Arduino Kartlarını prototiplemeye yeni başlayanlar için en uygun seçimdir

- Stm32 (Şekil 3.3)



Genellikle endüstriyel uygulamalarda kullanılan çok ünlü bir geliştirme kartı serisidir. Uyku modunda daha az akım çekmesi karın güç tüketimini düşürürken, sahip olduğu kod yürütme, veri aktarımları ve veri işleme hızlarıyla diğer ST Kartlarından daha çok tercih edilirler.

Şekil 3.3

- NodeMCU (Şekil 3.4)



Ağınızda bağlanabilen bir wifi çipinden ve çok sayıda dijital pinden oluşur. Analog pinlere sahip olmamasıyla Arduino kartından ayrılan NodeMCU, İnternete bağlanabilmesi ve sensörlerinizi çalıştırabilmeniz için basit komutlar sunmasıyla projelerin gözde kartlarıdır.

Şekil 3.4

3.1 Hangi yazılımlara ihtiyaç duyulur?

Projeye başlamadan önce hangi programlama dili ile çalışma yapılacağına verilmelidir. Bu dillere örnek olarak python, java, c++, c örnek verilebilir. Bu diller arasından tercih yaparken hakim olunan dile göre tercih yapmak kolaylık sağlayacaktır. Proje için gerekli kodun değiştirilmesi veya incelenmesi için MacOS, Linux, Windows gibi işletim sistemlerine uyumlu düzenleyiciye ihtiyaç olacak. Bu konuda birçok düzenleyici alternatifi mevcuttur.

4.Makine öğrenimi nedir?

Makine öğrenmesi esas olarak 1959 yılında bilgisayar biliminin yapay zekada sayısal öğrenme ve model tanıma çalışmalarından geliştirilmiş bir alt dalıdır. Makine öğrenmesi yapısal işlev olarak öğrenebilen ve veriler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran bir sistemdir. Bu tür algoritmalar statik program talimatlarını harfiyen takip etmek yerine örnek girişlerden veri tabanlı tahminleri ve kararları gerçekleştirebilmek amacıyla bir model inşa ederek çalışırlar. Makine öğrenmesine ait başlıca kavramların listesi açıklamalarıyla birlikte aşağıdaki gibidir:

► **Denetimli Öğrenme:** Veriler etkileşimli sistemlerden alınarak belirli bir düzende organize edilmesidir

► **Denetimsiz Öğrenme:** Sınıf bilgisi barındırmayan verilerin içerisindeki gruplar irdelenmesidir

► **Yarı Denetimli Öğrenme:** Bu kavram tam olarak yukarıdaki iki kavramın arasında yer alır ve etiketlenmemiş büyük miktarda bir veri ile etiketlenmiş küçük miktarda bir verinin beraber kullanılmasıdır.

► **Takviyeli Öğrenme:** Öğreticinin, sistemin ürettiği sonuç için doğru ya da yanlış olarak bir değerlendirmesidir.

► **Yoğun Öğrenme:** Hiyerarşik öğrenme olarak da bilinir. Bu öğrenme yöntemi derin grafiklerde birçok doğrusal ve doğrusal olmayan dönüşümlerden ve çoklu işlem katmanlarından oluşturulmuş verilerde, üst düzey soyutlamalar kullanılarak elde edilen model girişimlerine dayalı bir dizi algoritmalarla geliştirilmiş makine öğrenmesidir.

5. Derin öğrenme nedir?

Derin öğrenme, verilen bir veri seti ile sonuçları tahmin eden birden fazla katmandan oluşan bir makine öğrenme yöntemidir. Derin öğrenme, makine öğrenmesi ve yapay zekâ birbirinden farklı anlamları olan terimlerdir. Derin öğrenme, makine öğrenmesinin; makine öğrenmesi ise yapay zekânın alt dalı olarak özetlenebilir. Şekil 5.1.1



Şekil 5.1.1

Yapay zekâ, insan zekâsına benzer şekilde çeşitli görevleri yerine getiren ve sürekli kendini geliştiren sistemler veya makinalardır. 1950’lerde ortaya çıkan yapay zekâ hatalarından öğrenebilen sistemler olduğu için sistemi sürekli iyileştirmektedir. Makine öğrenmesi ise 1980’lerde ortaya çıkmış olup verilen bir veri kümesini işlemden geçirip tahminlerde bulunmak veya sınıflandırmaktır. Makine öğrenme algoritmalarında iki tür öğrenme biçimi vardır: gözetimli ve gözetimsiz öğrenme.

- Gözetimli öğrenme, etiketlenmiş verilerden öğrenmedir. Hem girdi hem de istenilen sonuç tanımlanır. Tahmin hakkındaki doğruluk geri bildirimi ise kişi tarafından yapılır.

- Gözetimsiz öğrenme, etiketsiz gözlemlerden öğrenme sürecidir. Sonuç verisiyle eğitime gerek yoktur. Algoritmanın kendisi veriden sonuca varır. Kendinin keşfetmesi beklenir.

Makine öğrenimi de yapay zekâ gibi uzun yıllar boyunca önemli bir gelişim gösterememiştir. 1990’lı yıllarda veri madenciliğiyle popülerliği artmış ve 2000’li yılların başında ise derin öğrenme gelişim göstermeye başlamıştır.

Derin öğrenme gözetimli, yarı gözetimli veya gözetimsiz olarak gerçekleştirilebilir. Derin öğrenmede çok sayıda veri girişiyle ayırt edici özellikleri kendisi öğrenir. Öğrenme işlemi için ne kadar çok veri girişi olursa o kadar çok başarılı olunur. Veriler birden çok katmandan geçer. Üst katmanlar daha çok ayrıntı çıkaran katmanlardır.

Her problemin çözümü için aynı yapay sinir ağı modeli kullanılamaz. Yapay sinir ağları kullanıldıkları yerlere ve amaca göre farklılık göstermektedir. Problemlerin yapısına göre tercih edilecek olan model değişkenlik göstermektedir. Aşağıda yer alan bölümde yapay sinir ağları modelleri görülmektedir. Sınıflandırma, tespit, tanı, tahmin, teşhis vb. sonuçların üretilmesi için farklı sinir ağı modelleri tercih edilir. Bu durum verinin yapısına göre de değişkenlik gösterebilmektedir. Bunlar;

- Konvolüsyonel Sinir Ağları (Convolutional Neural Network)
- Tekrarlayan Sinir Ağı (Recurrent Neural Network)
- Uzun-Kısa Süreli Hafıza (LSTM- Long Short-Term Memory)
- Kısıtlı Boltzmann Makinesi (RBM-Restricted Boltzmann Machine)
- Derin İnanç Ağı (DBN-Deep Belief Network)
- Derin Oto-kodlayıcılar (Auto Encoder)

5.1Derin öğrenme katmanları

Temel olarak 3 katmandan oluşmaktadır. Girdi katmanı, gizli katman ve çıktı katmanı.

5.1.1Giriş (Input) Katmanı

Veri giriş katmanı olarak bilinmektedir. Sinir ağlarında oluşturulacak olan veri seti ağın mimarisine göre belirlenmesi gerekmektedir. Sinir ağına girecek olan her bir örneklem giriş katmanına ağın eğitimi için girdi verisi olarak kullanılır. Her bir öğeye ait veri arka arkaya gelerek bir veriseti oluşturmaktadır. Bu verisetinin boyutu ağın hızını, test süresini ve bellek ihtiyacını artırmaktadır.

5.1.2Konvolüsyon(Convolution) katmanı

Konvolüsyonel sinir ağlarının temelini oluşturan katmandır. Bu katmanda girdi verisi üzerinde daha önceden belirlenmiş bir filtrenin girdi verisi üzerinde gezdirilerek girdinin belirgin özelliklerinin ortaya çıkmasını amaçlamaktadır. Filtreleme sonucunda giriş verisinden daha küçük bir matris elde edilmesini sağlar.

5.1.3Aktivasyon katmanı

Genellikle konvolüsyon katmanından sonra aktivasyon katmanı gelir. Aktivasyon fonksiyonu olarak simoid, hiperbolik tanjant, sinüs, step, eşik değer fonksiyonları kullanılmaktadır. Çok katmanlı yapay sinir ağları içerisinde doğrusal olmayan dönüşümler kullanılmaktadır.

5.1.4 Havuzlama (Pooling) Katmanı

Daha çok aktivasyon katmanı sonrasında yer alan havuzlama katmanı verilerin indirgenerek ağda sonraki katman için giriş boyutunu azaltır. Havuzlama işlemi ile veri üzerinde kayıplar oluşmaktadır. Ancak ağda dolaşacak olan veri miktarındaki azalma ağın daha hızlı olmasını sağlamaktadır.

5.1.5 Tam Bağlı (Full-Connected) Katman

Bu katmandaki tüm nöronlar bir dizi şeklinde görünür. Katmandaki nöronların tamamı bu katmana bağlı önceki katmandaki aktivasyonların hepsine tam bağlı durumdadır.

5.1.6 Dropout Katmanı

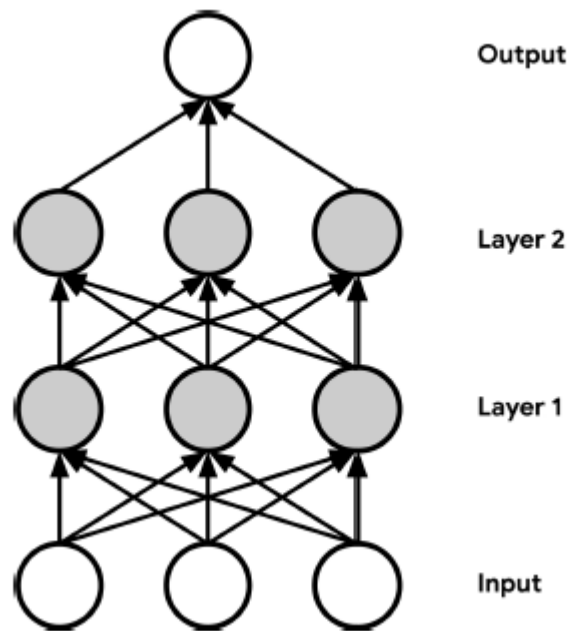
Dropout katmanı ağın ezberlenmesinin önüne geçmek için ağda ezber yapan bazı düğümlerin ortadan kaldırılmasıdır. Böylelikle ağın ezberlenmesi ortadan kaldırılmaya çalışılır.

5.1.7 Sınıflandırma (Classification) katmanı

Tam bağı katmandan sonra gelen sınıflandırma katmanı sınıflandırılması yapılacak öge sayısı kadar sonuç üretir.

5.1.8 Normalizasyon (Normalization) Katmanı

Eğitim süresinin azaltılmasının bir yolu nöronların aktivasyonlarını normalize etmektir. Normalizasyon katmanı geri beslemeli ağlarda gizli katmanlardaki durumları stabil hale getirmede oldukça etkilidir. Genellikle aktivasyon katmanı sonrasında normalleştirme gerçekleştirilir.



Şekil 5.1.2 İki katmanlı basit bir derin öğrenme ağı

5.2Derin Öğrenme Algoritmaları

- LeNet
- AlexNet
- ZF Net
- VggNet
- GoogleNet
- ResNet

5.3Derin öğrenmenin uygulama alanları:

- Yüz tanıma sistemlerinde
- Ses tanıma sistemlerinde
- Araçlarda otopilot özelliğinde veya sürücüsüz araçlarda kullanılmaktadır.
- Alarm sistemlerinde, kamera kayıtlarını sürekli kontrol etmek yerine, yalnızca olağandışı hareketlerde alarm sisteminin devreye girmesi gibi teknolojiler derin öğrenme sayesinde mümkün olmaktadır.
- Sağlık sektöründe kanser araştırmalarında, zaman kaybını ortadan kaldırmaktadır. Kanserli hücre örneklerinin tanıtıldığı derin öğrenme algoritmaları, yeni hücrelerin kanserli olup olmadığı tanısını koymakta hem daha hızlı hem de daha başarılı oluyor.g
- Görüntü iyileştirilmesinde
- Tavsiye sistemlerinde, beğenilebilecek müzik ve film önerileri sunmada
- Siber tehdit analizlerinde de derin öğrenme yöntemleri geliştirilebilir. Yukarıda verilen örneklerin dışında birçok örnek verilebilir.

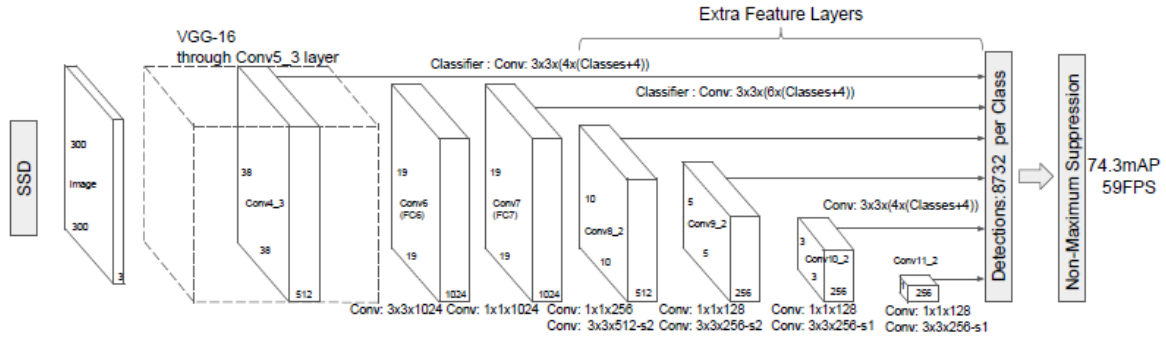
6.Model Seçimi

Geniş bir sorunu çözmek için tasarlanmış birçok derin öğrenme modeli mimarisi vardır. Bir model eğitilirken mimariyi tasarlamayı ya da araştırmacılar tarafından geliştirilen mevcut bir mimariyi kullanmak tercih edilebilir. Birçok yaygın sorun için, önceden eğitilmiş modeller ücretsiz olarak bulunabilir. Örneğin nesne tanıma projesi gerçekleştiriliyorsa Tenserflow Object Detection API aracılığı ile önceden eğitilmiş modellerden probleme uygun olanı bilgisayara indirilip,gerekli kurumlar sonrası kullanılabilir. Raporun devamında yapmış olduğumuz kişi algılama projesinde kullanmış olduğumuz model,data,donanım ve diğer gerekli araçlar anlatılacaktır.

6.1 SSD Mobilenet

Projemizde kullanmak üzere seçmiş olduğumuz SSD (Single Shot MultiBox Detector) ile tek seferde nesne tanımlama yapılabilmektedir. R-CNN ile bölge önerisinde ve bölgedeki sınıflandırmayı iki farklı aşmada yapılmaktadır. İlk önce resimde nesne olma ihtimali olan bölgeler belirleniyor daha sonra FC Layer ile bu bölgeler sınıflandırılmaktadır. SSD bu iki işlemi tek seferde gerçekleştirmektedir. Öncelikle giriş olarak alınan resim Konvolüsyonel sinir ağından geçirilmektedir. Bu şekilde karşımıza farklı boyutlarda özellik haritaları oluşmaktadır. Tüm özellik haritalarında 3x3 Konvolüsyonel filtre yardımıyla az miktarda sınırlayıcı dikdörtgenler elde edilmektedir. Her dikdörtgen için aynı zamanda hem sınırlar hem de sınıflandırmalar belirlenmektedir. Bu dikdörtgenler her aktivasyon haritasında olduğu için hem küçük nesneleri

hem de büyük nesneleri tanıyabilmektedir. SSD sadece tek ileri gidiş yaptığı için düşük donanımlı cihazlarda bile nesne sınıflandırma yapabilmektedir. Şekil 6.1.1 de SSD modelinin mimarisi gösterilmektedir.



Şekil 6.1.1 SSD model mimari yapı

7.Data nedir?

Data Türkçe karşılık olarak veri anlamına gelmektedir. Veri; yorumlanmayan ve analiz edilmeyen herhangi bir işlenmemiş gerçek, değer, metin, ses veya resim olabilir. Veri, tüm veri analitiği, makine öğrenimi, yapay zeka'nın en önemli parçasıdır. Veriler olmadan herhangi bir modeli eğitemeyiz. Yapılacak olan projeye göre hazır bulunan veri kümesi ya da projeyi oluşturacak kişinin hazırlayacağı veri kümesi ile proje gerçekleştirilebilir.

Yapılacak olan projeye göre alakalı data oluşturulur. Raporda bahsedilecek olan proje 'tinyml ile kişi algılama' olduğu için oluşturulmuş olan data insan fotoğraflarını içermektedir.

7.1 Data oluşturulurken neye dikkat edilir?

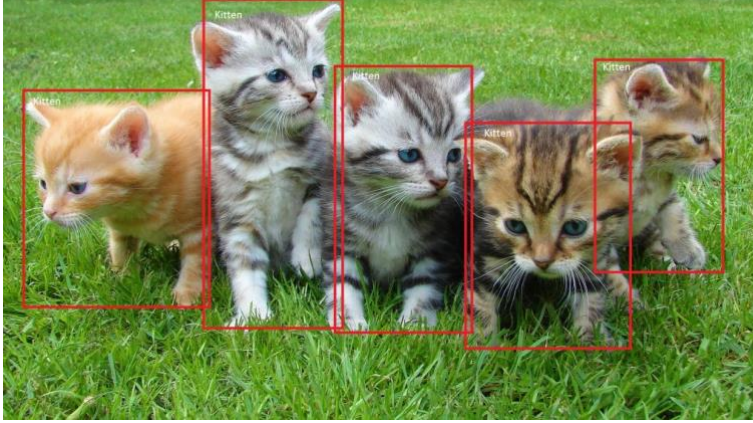
Oluşturulacak olan data projeye alakalı olmalıdır. Örneğin nesne tanıma projesi yapılıyor ise data nesne içeren fotoğraflardan, yüz tanıma yapılıyor ise insan yüzü içeren fotoğraflardan, ses tanıma projesi yapılıyor ise ses içeren datalardan oluşturulmalıdır. Projeden iyi sonuç alınabilmesi için data geniş tutulmalıdır. Örneğin bir kişi algılama projesi için en az 300-400 fotoğraf kullanılmalıdır. Bu fotoğraflar kişilerin farklı açılardan çekilmiş fotoğrafları ve karanlık, aydınlık ya da farklı özellikteki ortamlarda çekilmiş fotoğraflar olmalıdır.

7.2 Dataset oluşturma

Derin öğrenmede dataset çok önemli bir yer tutar. Dataseti yeterli büyüklükte ve doğru bir şekilde etiketlenmiş olması eğitim ve başarı için kritik bir rol oynar.

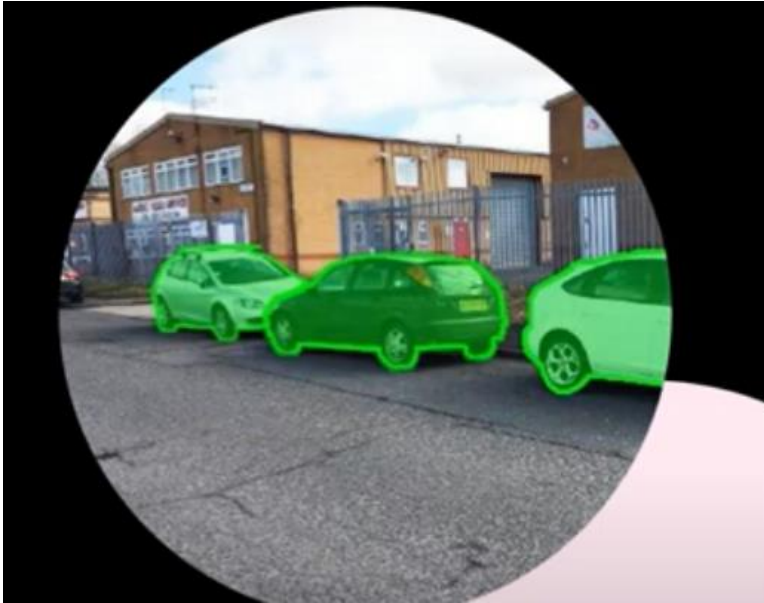
Resim etiketlemek için birkaç seçenek bulunmaktadır.

- **Sınırlayıcı Kutular:** Hedef nesnenin konumunu tanımlamak için kullanılan dikdörtgen şeklinde kutulardır. Nesne tespiti için en çok tercih edilen yöntemdir. Dikdörtgenin sol üst köşesindeki x,y eksen koordinatları ve sağ alt köşesindeki x,y eksen koordinatları belirlenebilir.Şekil 7.2.1 de gösterilmektedir.



Şekil 7.2.1

- **Çokgen Segmentasyonu:** Nesneler her zaman dikdörtgen şeklinde değildir. Bu fikirle çokgen bölümler, nesnenin şeklini ve konumunu çok kesin bir şekilde tanımlamak için dikdörtgenler yerine karmaşık çokgenlerin kullanıldığı başka bir veri etiketleme türüdür.Şekil 7.2.2 de gösterilmektedir.



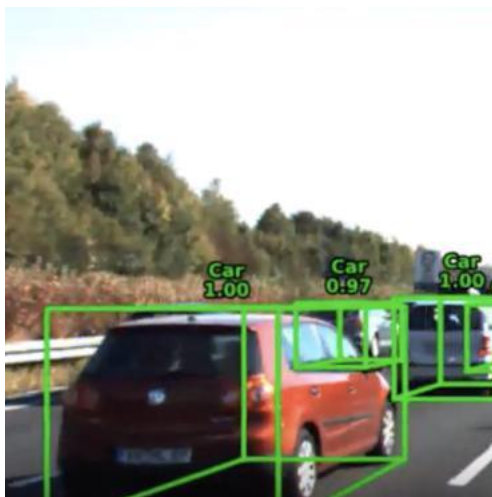
Şekil 7.2.2

- **Anlamsal Bölümleme** : Anlamsal bölümleme;Anlamsal bölümleme,görüntüdeki her pikselin bir sınıfa piksel bazlı bir açıklamadır.Bu sınıflar yaya,araba otobüs,yol,kaldırım vb. olabilir ve her piksel anlamsal bir anlam taşır.Anlamsal bölümleme , görüntüdeki her pikselin bir sınıfa piksel bazlı bir açıklamadır.Bu sınıflar yaya,araba otobüs,yol,kaldırım vb. olabilir ve her piksel anlamsal bir anlam taşır.Anlamsal bölümleme,öncelikle çevresel bağlamın çok önemli olduğu durumlarda kullanılır.Örneğin,kendi kendine giden arabalarda ve robotikte kullanılır çünkü modellerin içinde çalıştıkları ortamı anlamaları için bu tür daha iyidir.Şekil 7.2.3 de gösterilmektedir

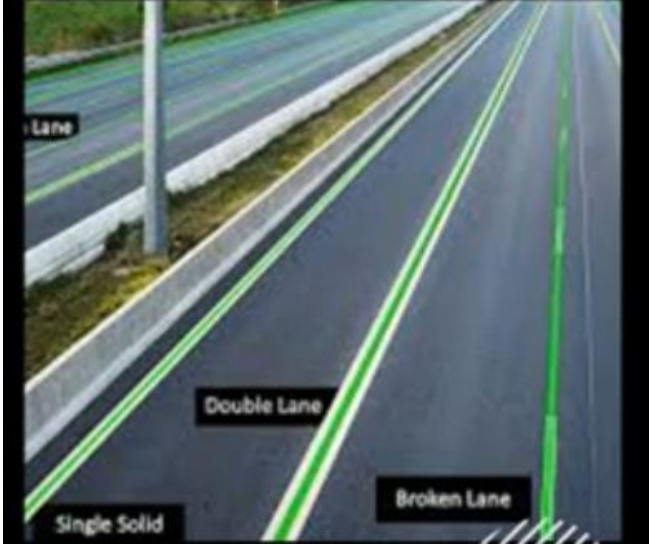


Şekil 7.2.3

- **3D Küpler:** 3D küpler,nesne hakkında ek derinlik bilgileri içerren sınırlayıcı kutulara benzer. Böylelikle 3D küplerle nesnenin 3D temsilini elde edebilir, sistemlerin 3D bir alanda hacim ve konum gibi özellikleri ayırt etmesine olanak tanır. 3D küplerin bir kullanım , nesnelerin arabadan mesafesini ölçmek için derinlik bilgisini kullanabileceği kendi kendine giden arabalardadır.Şekil 7.2.4 de gösterilmektedir.



Şekil 7.2.5

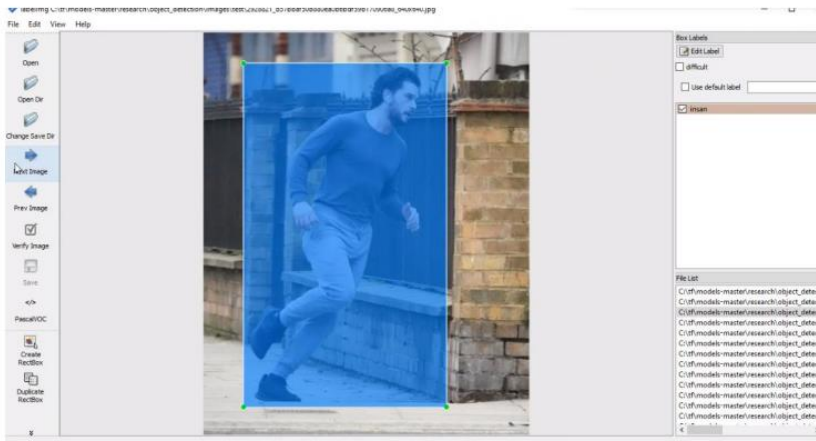


Şekil 7.2.7

7.3 Nesne Etiketleme Araçları

- MakeSense.AI
- Labelimg
- VGGImage annotator
- LabelMe
- Scalable
- RectLabel

Kişi algılama projesinde etiketleme aşamasında Labelimg kullanıldı. Örnek şekil 7.3.1 de gösterilmiştir.



Şekil 7.3.1 Fotoğraf Etiketleme Örneği

Fotoğrafların etiketleme işleminden sonra her fotoğraf için ayrı xml dosyası oluşur. Şekil 7.3.2


```

1 <?xml version="1.0"?>
2 <annotation>
3   <folder>train</folder>
4   <filename>5fb807fe5542800c58484c11.jpg</filename>
5   <path>C:\tensorflow1\models\tensorflow-train-object-detection-classifier-tutorial-master\research\object_detection\images\train\5fb807fe5542800c58484c11.jpg</path>
6   <source>
7     <database>Unknown</database>
8   </source>
9   <size>
10    <width>730</width>
11    <height>410</height>
12    <depth>3</depth>
13  </size>
14  <segmented>0</segmented>
15  <object>
16    <name>insan</name>
17    <pose>Unspecified</pose>
18    <truncated>1</truncated>
19    <difficult>0</difficult>
20    <bndbox>
21      <xmin>130</xmin>
22      <ymin>1</ymin>
23      <xmax>633</xmax>
24      <ymax>405</ymax>
25    </bndbox>
26  </object>
27 </annotation>

```

Şekil 7.3.2 Xml dosyası örneği

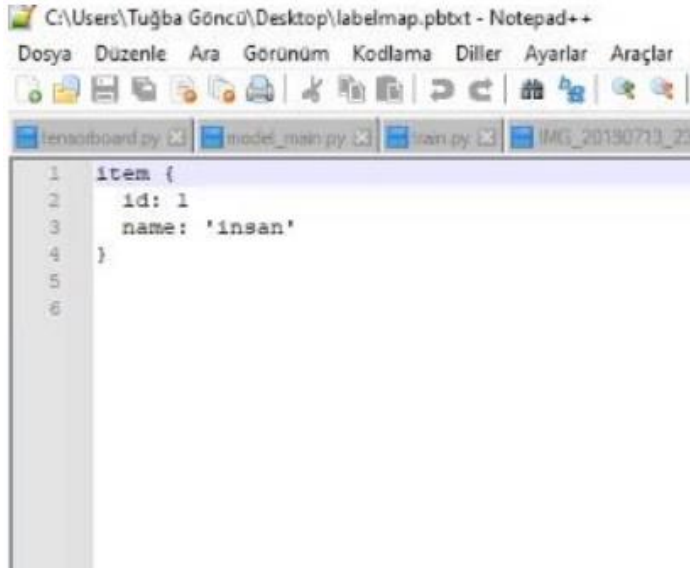
Xml dosyaları oluşturulduktan sonra eğitim için gerekli olan tfrecord dosyası oluşturulmuştur. Ancak tfrecord dosyası csv formatında veri girişi aldığı için daha önceden oluşturulan xml dosyaları başka bir program aracılığıyla csv formatına dönüştürülmüştür. Toplam xml dosyalarından train veri seti için 1 adet, test veri seti için 1 adet csv uzantılı dosya oluşturulmuştur. Şekilde (7.3.4) oluşturulan csv uzantılı dosya gösterilmektedir.

	A	B	C	D	E	F	G	H	I	J	K	L
1	filename,width,height,class,xmin,ymin,xmax,ymax											
2	0x0-1605538496637.jpg,1000,668,insan,250,7,879,668											
3	0x0-kirmizi-oda-son-bolum-izle-nihal-zafere-ayrilmak-istedigini-soyluyor-1620426224347.jpg,600,329,insan,40,1,440,328											
4	0_15_106.jpg,500,333,insan,42,59,160,300											
5	0_15_106.jpg,500,333,insan,260,68,352,272											
6	0_15_106.jpg,500,333,insan,358,51,465,304											
7	149496.jpg,652,400,insan,176,2,463,400											
8	163796.jpg,1001,600,insan,341,124,699,599											
9	163796.jpg,1001,600,insan,731,285,990,587											
10	163796.jpg,1001,600,insan,21,256,332,588											
11	1_01_02.jpg,992,661,insan,348,73,726,661											
12	24c3b510c32bd257c075ea0da4006fa3.jpg,2048,1367,insan,669,222,1208,1366											
13	2f7d87f2c377b43ea92eb62092e68543.jpg,1029,1029,insan,136,36,946,1025											
14	407051-nesrin-cavadzade-nin-cevabi-herkesi-guldu-60225372d48ca.jpg,680,374,insan,204,2,511,373											
15	410476-sherlock-the-abominable-bride-crop.jpg,640,360,insan,126,60,354,358											
16	410476-sherlock-the-abominable-bride-crop.jpg,640,360,insan,361,7,600,356											
17	5002708519_025_01.jpg,325,445,insan,68,11,273,445											
18	575623Image1.jpg,1799,1200,insan,551,29,1727,1200											
19	5cf8339011e2052819476217.jpg,750,563,insan,199,43,749,491											
20	5e8ebfd933ccb348c9d312f.jpg,1200,630,insan,270,2,614,630											
21	5e8ebfd933ccb348c9d312f.jpg,1200,630,insan,575,2,1041,624											

Şekil 7.3.4 Csv uzantılı dosya

8. Label Map Oluşturma ve Eğitim Ayarları

Eğitim modeline giriş olarak verilecek olan tfrecord dosyaları oluşturulduktan sonra eğitim aşamasına geçmeden önce label map oluşturulmuştur. Label map tfrecord dosyası oluştururken vermiş olduğumuz id ye göre nesnenin sınıfını belirlemiştir. Şekilde veri setimiz için oluşturulan label map gösterilmektedir. Eğitim için gerekli olan konfigürasyon dosyası yüklediğimiz Tensorflow API dosyasından seçilmiştir. Bu çalışmada kullanılan modellerin ilgili konfigürasyon dosyaları eğitim aşamasında seçilerek gerekli parametre ayarları ve veriler girilmiştir.



Şekil 8.1.1 LabelMap

9.Hiperparametre Ayarlamaları Ve Seçimleri

9.1Öğrenme Katsayısı

Derin öğrenme algoritmalarında kullanılan en önemli parametrelerden biri öğrenme katsayısıdır ve bu değer her zaman iyi ayarlanmış olduğundan emin olmak gerekmektedir . Öğrenme katsayısına hata düzeltme katsayısı da denilmektedir. Bu katsayı çok büyük olduğunda hedefe ulaşmakta zorlanabilir, çok küçük olduğunda ise yakınsama uzun sürecektir . Birden fazla katmanlı sinir ağları için öngörülen değer 1 ile 10-6 arasında olmalıdır . Öğrenme katsayısını belirlemede modelin öğrenme süreci izlenerek optimum değeri bulmak mümkündür

9.2Mini Batch Boyutu

Derin öğrenme algoritmalarında modelin eğitim aşamasında kullanılan eğitim hızını, adım sayısını, hata hesaplamalarındaki gürültüleri etkileyen bir hiper parametredir. Mini-batch, bir adımda tüm verilerin eğitilmesi yerine veri setini parçalara bölerek, bu parçalar üzerinden eğitim yapılan bir tekniktir

9.3Eğitim Tur Sayısı

Eğitim tur sayısı tüm veri setinin bir ileri bir de geri yönde tüm ağıdan geçmesidir. Tur sayısı arttıkça modelin başarımı büyük oranda artmaktadır ancak belli bir tur sayısından sonra değişiklik çok küçük oranda olacağı için eğitim sonlandırılmalıdır

9.4 Loss Fonksiyonu

Loss fonksiyonu, tasarlanan modelin hata oranını ve bununla birlikte modelin başarımını ölçen fonksiyondur. Derin ağların son katmanı loss fonksiyonun tanımlandığı katmandır. Loss fonksiyonun ana görevi modelin yaptığı tahminin, gerçek değerden (ground truth) ne kadar farklı olduğunu hesaplamaktır. Dolayısıyla iyi tahminde bulunan bir model oluşturmamışsak, gerçek değer (ground truth) ile tahmin edilen değer arasındaki fark yüksek olacak yani loss değeri yüksek olacak, iyi modele sahipsek loss değeri az olacaktır. İyi bir modelden beklentimiz 0'a yakın loss değerine sahip olmasıdır.

9.5 Bias Değeri

Lineer fonksiyonlar temel olarak, $y = W.x + b$ şeklinde tanımlanan fonksiyonlardır. Bu fonksiyonlarda y değeri x'in değerine bağlıdır, dolayısıyla y değeri bağımlı değişken, x değeri ise herhangi bir girdi olabileceği için bağımsız değişken olarak tanımlanmaktadır. W ve b değerleri ise fonksiyonun parametreleridir.

10. Proje Sonucu

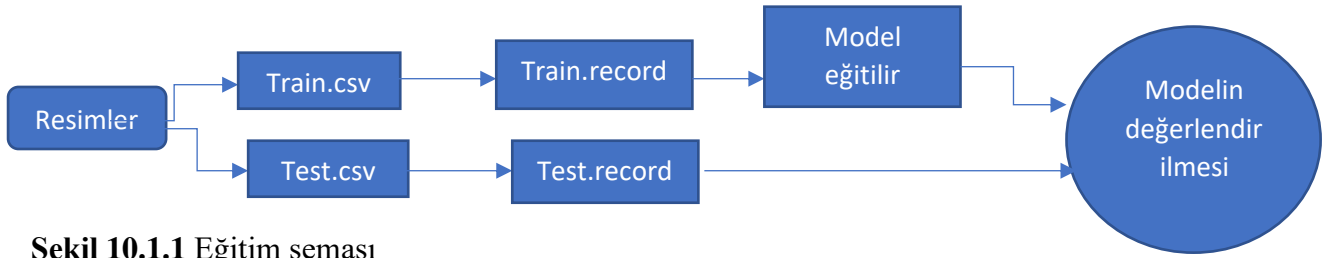
Bu proje çalışmasında derin öğrenme algoritmaları kullanılarak görüntü üzerinde yer alan kişi tanınması amaçlanmıştır. Bu amaç doğrultusunda öncelikle data seti çevreden toplanan yaklaşık 350 görüntü ile oluşturulmuştur. Toplanan bu görüntüler teker teker LabelImg programıyla etiketlenerek görüntü içerisindeki ilgili alanın tespiti yapılmıştır. Eğitim şeması şekilde gösterilmiştir. Açık kaynak kodlu derin öğrenme kütüphanesi olan TensorFlow ile Google tarafından eğitilmiş olan SSD Mobilenet mimari yapısı kullanılmıştır. İlgili model mimarileri eğitilmiş olan ağırlıkların transfer öğrenme tekniği kullanarak kendi data setimiz eğitilmiştir. Projemizde kullanılmış olan modelin hızı ve doğruluk oranı tabloda gösterilmiştir.

Tablo 10.1

Model Adı	Hız (ms)	COCO mAP
SSD Mobilenet	29	22
Faster R-CNN	58	28

TensorFlow ile eğitilmiş modeli kişi tanımada kullanmak için inference graph (çıkarım grafiği) gerekmektedir. Bunun için etiketlenmiş görüntülerden oluşturduğumuz csv uzantılı dosyadan tfrecord dosyası oluşturulmuştur. Bu dosya eğitim aşamasında modele girdi olarak verilmiştir. Belirli bir adım eğitim gerçekleştirdikten sonra belirttiğimiz klasörde eğitimin sonuçları. ckpt uzantılı olarak oluşturulmuştur. Son olarak oluşturulan. ckpt uzantılı dosyalardan çıkarım grafiği

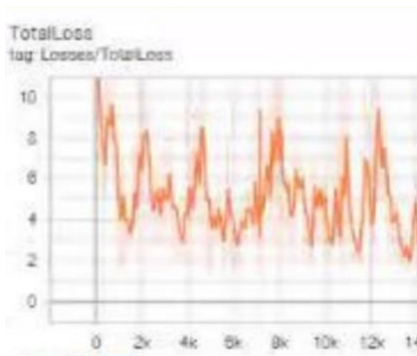
oluşturarak kişi tanıma algoritmasına girdi olarak verilmiştir. Şekil 10.1.1 de proje çalışmasında izlenen eğitim aşaması gösterilmektedir.



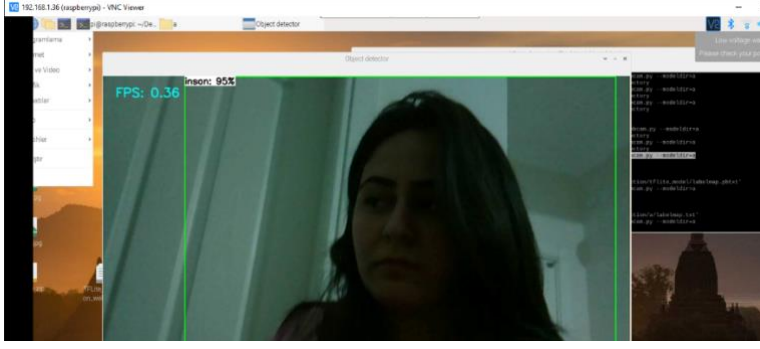
Şekil 10.1.1 Eğitim şeması

11.Çıkarım

Eğitim aşamasında kontrol edilmesi gereken en önemli parametrelerden biride loss grafiklerinin incelenmesidir. Bu aşamada tensorflow tarafından desteklenen tensorboard aracılığıyla grafikler oluşturulmaktadır. Bu grafiklere tensorboard sayfasından ulaşmak için öncelikle tensorboard – logdir= training kodunu çalıştırdıktan sonra internet tarayıcısına localhost:6006 yazarak ilgili grafiklere ulaşılabilir. Eğitimi yapılan modelin loss grafiği aşağıdaki şekilde sunulmuştur. Loss grafiğinde değer sıfıra ne kadar yakın olursa eğitimin o kadar başarılı gerçekleştiği anlamına gelmektedir. Bu başarıda eğitim adım sayısı ve öğrenme oranı önemli rol oynamaktadır.



Şekil 11.1.1



Şekil 11.1.2 SSD MobileNet ile yapılan proje fotoğrafı

12.Projede Kullanılan Araçlar Ve Kütüphaneler

12.1 Anaconda

Anaconda ücretsiz ve açık kaynaklı, Python ve R programlama dillerinin bilimsel hesaplama kullanımında paket yönetimini kolaylaştırmayı amaçlayan bir özgür ve açık kaynaklı dağıtımdır. Bu kurulum ayrıca jupyter notebook ve spyder gibi araçları da barındırır

12.2 Tensorflow

Yapmış olduğumuz projede Google tarafından geliştirilen derin öğrenme kütüphanesi olan tensorflow kullanılmıştır. TensorFlow tarafından açık kaynak kodlu olarak sunulan API sayesinde derin öğrenme uygulamaları yapılabilir. Ayrıca Google tarafından eğitilmiş derin öğrenme modelleri bulunmaktadır. Tensorflow Object Detection API ile hazır olarak tensorflow tarafından eğitilmiş modelleri kullanarak kendi görüntümüzde nesne sınıflandırma yapılmaktadır. Tensorflow bizlere birçok farklı SSD ve R-CNN modelleri sunmaktadır. Bu modellerin hızları ve isabet oranları değişiklik göstermektedir. Bu modeller tensorflow tarafından COCO data seti üzerinde eğitilmiştir. COCO data seti günlük hayatta yaygın olarak kullanılan 90 farklı sınıftan oluşmaktadır. Bu tez çalışmasında kullanılan sınıf COCO data setinde olmadığı için kendi oluşturduğumuz data seti üzerinde yeni bir model eğitmemiz gerekmektedir.

Sıfırdan model oluşturmak ve bu modeli eğitmek zor bir aşamadır. Zaman ve güçlü bilgisayar gerekmektedir. Bunun yerine kendi data setimizi oluşturulup ,Tensorflowun hazır modelleri kendi data setimiz üzerinde eğitilerek başarılı sonuçlar elde edilebilir.

13.Projede Kullanılan Cihazlar

13.1 Raspberry Pi 3 Model b+

Raspberry Pi, kredi kartı büyüklüğünde bir bilgisayardır.Raspberry Pi Model B'nin geliştirilmiş ve kart tasarımı değiştirilmiş sürümüdür. Model B'den farklı olarak 4 adet USB portu, normal SD kart yerine mikro SD kart slotu ve 26-pin yerine 40-pinli GPIO

konektörü bulunur. HDMI ve Ethernet bağlantıları yine bu kartta yer almaktadır. Kompozit video çıkışı bu kartta ayrı bir konektör olarak yer almamaktadır, 3.5 mm ses çıkış portundan 3'lü RCA tipi kabloyla bağlanır.Şekil 13.1.1 de gösterilmiştir.



Şekil 13.1.1.1

13.1.1 Raspberry Pi Teknik Özellikleri

- Raspberry Pi 3 Model B+ Teknik Özellikleri:
- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC 1.4GHz.
- 1GB LPDDR2 SDRAM.
- 2.4GHz ve 5GHz destekli IEEE 802.11.b/g/n/ac kablosuz ağ bağlantısı, Bluetooth 4.2, BLE destekli.
- USB 2.0 üzerinden gigabit Ethernet (maks. hız 300 Mbps)
- Genişletilmiş 40-pin GPIO header bağlantısı

13.2 Rasperi Camera 1.3

13.2.1 Raspberry Camera Teknik Özellikleri

- Sensör Tipi: OmniVision OV5647 (5 megapiksel)
- Sensör Boyutu: 3.67 x 2.74 mmI
- Piksel Sayısı: 2592 x 1944
- Piksel Boyutu: 1.4 x 1.4 um



Şekil 13.2.1.1



Şekil Projede kullanılan donanım görüntüsü