

Kırıkkale Üniversitesi

2019-2020

Mühendislik Fakültesi

Bilgisayar Mühendisliği

Bitirme Projesi 1

Danışman : Yrd.Doç.Dr. Bülent Gürsel Emiroğlu

Öğrenci : Burak Şencan

150205008

İÇİNDEKİLER

Sayfa No:

**ÖZET………………………………………………………………………………………..…….…………………..3**

**Projenin Amacı…………………..……………………………………………………………………………….3**

**BÖLÜM 1 Giriş ve Tanım……………..……………..……………..……………..……………..…………4**

1 Yapay Zeka**…..……………..……………..……………..……………..……………..……………..….4**

1.1 Giriş**…..……………..……………..……………..……………..……………..……………..….….4**

1.2 Tanım**…..……………..……………..……………..……………..……………..……………..…..5**

1.3 Yapay Zekanın Amaçları**…..……………..……………..……………..……………..………6**

1.4 Yapay Zekanın Araştırma Alanları**…..……………..……………..……………..……….6**

1.4.1 Doğal Dil İşleme**…..……………..……………..……………..……………..………..6**

1.4.2 Görüntü İşleme**…..……………..……………..……………..……………..…………6**

1.4.3 Makine Öğrenmesi**…..……………..……………..……………..……………..……6**

1.5 Derin öğrenme**…..……………..……………..……………..……………..……………..……7**

2 Android **…..……………..……………..……………..……………..……………..……………..………8**

2.1 Giriş**…..……………..……………..……………..……………..……………..……………..………8**

2.2 Android İşletim Sistemi Mimarisi**…..……………..……………..……………..………….8**

2.3 Android Sürümleri **…..……………..……………..……………..……………..……………….9**

2.4 Android Uyglama Geliştirme**…..……………..……………..……………..……………..…9**

2.5 Android Proje Yapısı (Android Studio) **…..……………..……………..……………..…10**

3 IBM Watson**…..……………..……………..……………..……………..……………..………………..12**

3.1 Giriş**…..……………..……………..……………..……………..……………..……………..………12**

3.2 Yapay Zeka ve Makine Öğrenmesi Servis Hizmetleri**…..……………..…………….12**

3.3 Görüntü Tanıma Servisi (Visual Recognition Services) **…..……………..……..12**

**BÖLÜM 2 Android Uygulaması…..……………..……………..……………..……………..………13**

1.1 Giriş**…..……………..……………..……………..……………..……………..……………..…..…..13**

1.2 Layout (activity\_main.xml) **…..……………..……………..……………..……………..…...13**

1.3 Permissions (manifest.xml) **…..……………..……………..……………..………………..…14**

1.4 Activity (MainActivity.java) **…..……………..……………..……………..………………..…14**

**BÖLÜM 3 Sonuç ………..……………..……………………..……………..……………………………….17**

1.1 Ne Öğrendik**……..………………..………………..………………..………………..……………..17**

1.2 Kaynakça**……..………………..………………..………………..………………..………………….17**

**ÖZET**

Bu projede, IBM Watson görüntü tanımlama(Visual Recognition) hizmetinin kullanıldığı android tabanlı bir uygulama geliştirilmiştir. Uygulama fotografı çekilen besinleri bulut hizmeti (watson visual recognition) kullanarak tanımlamaktadır. Lokalde tutulan kalori degerleri ise açılır mesaj penceresinde (toast) görüntülenmektedir.

Projede Kullanılan Teknolojiler

1) IBM Watson Visual Recognition: Kameradan alınan görüntünün tanımlanması için kullanıldı. IBM Watson Visual Recognition hizmeti, olayların, nesnelerin, yüzlerin ve diğer içeriğin görüntülerini analiz etmek için derin öğrenme algoritmalarını kullanır.

2) Android (java): Uygulamanın geliştirildigi ortam.Dil olarak java kullanıldı. Android Linux tabanlı bir işletim sistemidir

**Projenin Amacı**

Günümüz toplumunda obeziteye neden olan fazla kalori alımı gibi sağlıksız beslenme alışkanlıkları giderek artmaktadır. Bu uygulama ile besinlerin kalori değerleri hakkında kişiye bilgi verilmesi, ve bu sebeple fazla kalori alımının önüne geçmesi amaçlanmaktadır.

**BÖLÜM 1 Giriş ve Tanım**

**1 Yapay Zeka**

1.1 Giriş

Bilgisayarların hayatımıza girmesi ile insanoğlunun milattan önceki çağlarda başlayan işleri mekanik araçlarla otomatikleştirme isteği yapay zeka kavramının ortaya atılmasına neden olmuştur. Sanayide bilgisayar teknolojisinin bant üretim süreçlerinde kullanılmasıyla da robot kavramı kullanılmaya başlanmıştır.

Yapay zekanın temellerinin; felsefe (milattan önce 428’den günümüze), matematik (8.yüzyıldan günümüze), psikoloji (1879’dan günümüze), bilgisayar mühendisliği (1940’dan günümüze) ve dilbilim (1957’den günümüze) disiplinlerinden oluştuğu görülmektedir.

Yapay zeka kavramının geçmişi görüldüğü gibi çok eskilere dayanmaktadır. Fikir babası,

"Makineler düşünebilir mi?" sorunsalını ortaya atarak makine zekâsını tartışmaya açan Alan Mathison Turing’dir. II. Dünya sava¸sı sırasında kripto analizi gereksinimleri ile üretilen elektromekanik cihazlar sayesinde bilgisayar bilimi ve yapay zekâ kavramları doğmuştur. Enigma makinesinin ¸sifre algoritmasını çözme amacı ile başlatılan çalışmalar, Turing’in prensiplerini oluşturduğu bilgisayar prototipleri olan Heath Robinson, Bombe Bilgisayarı ve Colossus Bilgisayarları, Boole cebirine dayanan veri işleme mantığı ile Makine Zekâsı kavramının oluşmasına sebep olmuştur.

Yapay zeka konusundaki çalışmalar 1960’lardan beri gündemde olmasına karşın yapay zeka uygulamalarının güçlü bilgisayarlara ihtiyaç duyması nedeniyle araştırmaların hızlı ilerlemesine engel olmuştur. Ancak günümüzde bilgisayar teknolojisinde yaşanan gelişmelerin sağladığı ucuz ve güçlü bilgisayarlar sayesinde yapay zeka alanında büyük ölçekli araştırma yapabilmek ekonomik açıdan mümkün hale gelebilmiştir. Bunun sonucu olarak, yapay zekanın bir alt alanı olan uzman sistemler (expert systems) konusunda daha şimdiden önemli gelişmeler sağlanmış olup, iş dünyasının karar verme sürecinde uzman sistemlerden önemli ölçüde yararlandığı gözlenmektedir

1.2 Tanım

Farklı uygulama alanları olan yapay zeka kavramının uygulama alanlarına bağlı olarak pek çok tanımı vardır. Bundan önce zekanın tanımı üstünde durmak faydalı olacaktır.

Zeka kavramının sözlük anlamına baktığımızda TDK (Türk Dil Kurumu) tarafından “˙Insanın

düşünme, akıl yürütme, nesnel gerçekleri algılama, kavrama, yargılama, sonuç çıkarma yeteneklerinin tümüdür.” (TDK) olarak tanımlandığı görülmektedir.

Lenat ve Feigenbaum (1991) zeka için “Karma¸sık bir problemi çözüm arama alanını daraltarak kısa yoldan çözmek için gerekli bilgileri toplayıp birleştirme kabiliyetidir” ifadesini kullanmıştır. Uğur ve Kınacı (2006) yapay zekayı Lenat ve Feigenbaum (1991)’ın tanımına bağlayarak; “bu özelliklere sahip organik olmayan sistemlerdeki zekadır.” ifadesiyle tanımlamıştır.

Yapay zekanın bazı tanımları ve karakteristikleri (Bakınız ¸ Sekil 1) de karar verme ve problem çözme üzerinde yoğunlaşmıştır.



Sekil 1.1: Yapay zekanın karakteristikleri

Sembolik dll işleme: Uzmanlar, yapay zekaya uygun tipte olan problemleri çözmek, problem içeriklerini tanımlamak için sembol kullanırlar ve bu içerikleri işlemek için değişik stratejiler ve kurallar uygularlar. Waterman’ın (1986) yapay zeka yaklaşımı bilgiyi, problem konseptlerini temsil eden semboller kümesi olarak tanımlar. Yapay zekanın teknik dilinde sembol gerçek dünyanın bazı içeriklerini temsil eden karakterler kümesidir.

Bulgusallık: Bulgusallık, kurallar yumağı olarak, tanımda yapay zekanın anahtar elemanı olarak kullanılmıştır. "Yapay zeka, bilgisayar biliminin, bilgiyi rakamlardan ziyade sembollerle temsil etme yöntemleri ve kurallar yumağını içeren bulgusal yapıda veya bilgiyi işleme metodları ile çalışan bilim dalıdır” (Copeland, 2017).

Anlam Çıkarma: Yapay zeka, makinenin sebep bulma yeteneği sergilemesini gerektirir. Sebep bulma, bulgusal yada diğer arama yaklaşımlarını kullanarak, olaylardan ve kurallardan anlam çıkarma sürecinden oluşur. Bu süreç üzerinde semboller üzerinde gerçeklenen en basit sekildeki örüntü uydurma ve tanıma (pattern matching, recognation) bu işin esasını oluşturmaktadır. Yapay zeka, bu yaklaşımını uygulayarak anlam çıkarmada eşsizdir

1.3 Yapay Zekanın Amaçları

Yapay zeka çalışmalarının amacı, insan zekasını örnek alarak, insan zekası gerektiren görevleri yapabilecek makinalar yapmaktır. Yani şu anda insanların bilgisayarlardan daha iyi yaptığı şeyleri bilgisayarların daha iyi yapmasını sağlama çalışmasıdır. Genel olarak yapay zekanın amacı üç ana başlık altında toplanabilir:

a. Makinaları daha akıllı hale getirmek: Gelece˘gin bilgi toplumunun kurulmasında önemli rol oynayacak ‘genel bilgi sistemleri geliştirmek,

b. Zekanın ne olduğunu anlamak: İnsan beyninin fonksiyonlarını bilgisayar modelleri yardımıyla anlamaya çalışmak

c. Makinaları daha faydalı hale getirmek: Yapay zeka iş yardımcıları ve ‘zeki robot timleri’ geliştirmek.

1.4 Yapay Zekanın Araştırma Alanları

Yapay zeka çalışmaları sadece bilgisayar bilimlerinde değil, oyun, doğal dil anlama ve çeviri işlemlerinde, görüntülerin işlenmesinde, genel bilgi sistemlerinde, makine öğrenmesinde, bilgi tabanlı sistemlerde, veri madenciliğinde, robotik gibi farklı alanlarda gerçeklekleştirilmektedir. Her geçen gün farklı alanlarda yapay zekanın kullanıldığını görmekteyiz.

1.4.1 Doğal Dil İşleme

Bu alandaki çalışmalar, otomatik tercüme, doğal dilde yazılmış metinlerin açıklanması ve üretilmesi ve konuşmaların otomatik işlenmesi gibi faaliyetleri kapsar. Son yıllarda CYC ve EDR gibi genel bilgi sistemleri kullanılarak tercüme sistemlerindeki doğruluk oranı arttırılmaktadır. Makine tarafından bir cümlenin anlaşılması, birçok bilgiyi devreye sokan bir süreçtir. Fakat kimi zaman gürültüler ve akustik degişkenlik benzesen sinyalin işlenmesini zorlaştırır. Günümüzde deneysel insan-makine diyalogu sistemleri geliştirilmiş durumdadır.

1.4.2 Görüntü İşleme

Görme bir makinenin çevresini fark etmeye yönelik başka bir özellğidir. Görme probleminin basitleştirilmesi, basit ¸sekillerin algoritmik ¸sekil tanıma metotları yardımıyla tanımlanmasından ibarettir. Bunlar bir metin içindeki harfler veya bir uydu resmi üzerindeki özel bölgeler olabilir. Bir sahnenin veya görüntünün gerçek olup olmadığının anlaşılması, tıbbi teşhis amacı ile radyolojik görüntülerin açıklanması, basılı ya da el yazısı bir metnin anlaşılması robotbilime aittir. Görüntülerin işlenmesi yapay zekanın endüstriyel alandaki ilk uygulamalarından biridir.

1.4.3 Makine Öğrenmesi

İlk çalışmalar “algılayıcılar” (perceptrons) adı verilen, aygıt düzeyinde basit sistemler üzerinde başlamıştır. Daha sonra sembol düzeyi öğrenme metotları geliştirilmeye başlanmıştır. 1970’lerin sonlarına doğru bilgi tabanlı sistemlerin ortaya çıkmasıyla bilgi düzeyi öğrenme metotları geliştirilmeye başlanmıştır.

1.5 Derin Öğrenme

Makine öğrenme, yapay zekânın bir koludur. Makine öğrenme, veri ile eğitilebilir. Daha sonra, elde ettiği bilgi ile tahminlerde bulunabilir.

Derin ögrenme ise makine öğrenmenin bir dalıdır. Büyük veri üzerinde çalışan, ön eğitimli (farklı işler için özelleşmiş) çok sayıda yapay zekâdan oluşan, çok sayıda düğüme sahip ağlar derin öğrenme olarak adlandırılır. Derin öğrenme uygulamaları büyük ve etiketsiz veri üzerinde çalışırken öğretmensiz ön eğitim (pre- unsupervised trained), ardından öğretmenli öğrenme süreçlerinin çalıstırıldığı sistemlerdir. Bu uygulamaların temel özelliklerin biri de ağdaki düğüm miktarını çok büyük olmasıdır. Örneğin, Google Brain’de 1 milyar düğüm mevcuttur. Böyle bir yapının en büyük eksiği bir objenin ö˘ğrenilmesinin uzun zaman almasıdır. Örneğin, bir obje, 10 milyon videodan, 16000 bilgisayarla 3 günde derin öğrenme ile öğrenilmiştir.



Sekil 2: Bir geriye yayılım sinir ağının topolojisi

Derin öğrenme sistemlerinin hem öğrenme hem de karar verme aşamasında çıkarılan özelliklerden yakın olanların bütünleştirilmesi aşaması alt küme (Pooling) oluşturmadır. Milyonlarca farklı verinin (görüntü, ses ve bunun gibi) tüm özellikleri birebir aynı olmayacağı için bu safha önemlidir. Oluşturulan alt kümelerin stokastik olarak elenmesine silme (dropout) denir. Bu sayede genellikle zayıf ilişkiye ait olan bağlar koparılır. Silme katmanları sadece kıvrımlı ağ için kullanılmazlar. Diğer ağlar için de kullanılabilir. Yapılan işlem gürültü giderme işlemine benzemektedir. Verideki gürültü ise girdi sayısı çıktı sayısına eşit olan yapay sinir ağlarıyla (autoencoder) sağlanır. Derin öğrenme sistemlerinde öğrenme ve karar verme karar verme süreçleri göz önünde bulundurulduğunda aşağıdaki temel bileşenlerin her derin öğrenme ağında bulunduğu anlaşılmaktadır.

Parametreler

Katmanlar

Aktivasyon fonksiyonları

Zarar fonksiyonları

Optimizasyon yöntemleri

**2 Android**

**2.1 Giriş**

Android, kelimem anlamı olarak insan şeklinde robot anlamına gelen ingilizce bir kelimedir Terim olarak ele aldığımızda ise önceleri sadece mobil cihazlarda kullanılması için düşünülerek tasarlanan bir linux tabanlı mobil işletim sistemidir.

Android işletim sistemi açık kaynak ve özgür yazılım hareketinin en ünlü örneklerinden olan Linux Kernel üzerine inşa edilmiş açık kaynak kodlu mobil işletim sistemidir

Android işletim sisteminin temelleri ilk olarak 2003 yılında Kaliforniya’da Andy Robin, Rich miner , Nick Sears ve CrisWhite tarafından atıldı. Bu 4 mühendis linux tabanlı bir işletim sistemi geliştirmeye başladı. 2005 te Google tarafından satın alınan proje üzerindeki Alliance(OHA) adında bir birlik kuruldu

**2.2 Android İşletim Sistemi Mimarisi**

Bu işletim sisteminin parçaları çekirdek (kernel), sistem kütüphaneleri, uygulama geliştirme çatıları (frameworks) ve yerleşik temel uygulamalardan oluşmaktadır. Androidmimarisi aşağıdaki gibidir.

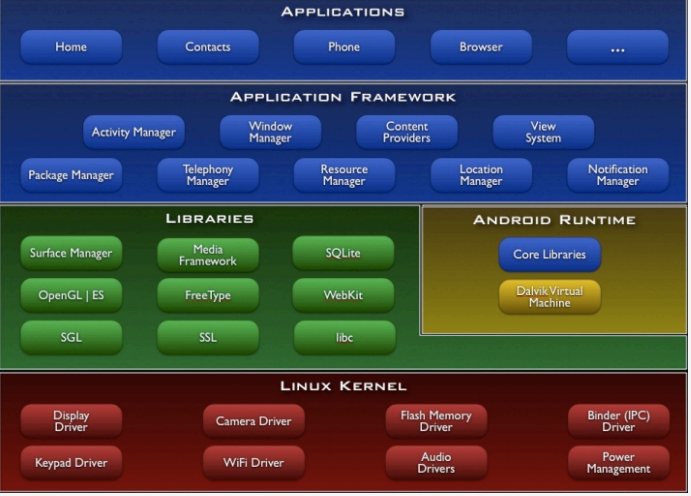
Çekirdek: Linux kernelidir. Güvenlik, hafıza yönetimi, süreç yönetimi, ağ yığınları ve sürücü modellerini içermektedir.

Android Runtime: Sanal makinedir. Dalvik Sanal Makinesini de içermektedir. 5.0 ile Dalvik kaldırılmış ve ART'ye geçilmiştir.

Kütüphaneler: Veritabanı kütüphaneleri, web tarayıcı kütüphaneleri, grafik ve arayüz kütüphanelerini içermektedir.

Uygulama Çatısı: Uygulama geliştiricilere geniş bir platform sunan kısımdır.

Uygulama Katmanı: Doğrudan Java (programlama dili) ile geliştirilmiş uygulamaları içermektedir.



**2.3 Android Sürümleri**

|  |  |  |
| --- | --- | --- |
| **Versiyon** | **Kod adı** | **Yayın tarihi** |
| 2.3 | [Gingerbread](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9HaW5nZXJicmVhZA) | 9 Şubat 2011 |
| 4.0 | [Ice Cream Sandwich](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9JY2VfQ3JlYW1fU2FuZHdpY2g) | 19 Ekim 2011 |
| 4.1 | [Jelly Bean](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9KZWxseV9CZWFu) | 9 Temmuz 2012 |
| 4.2 | 13 Kasım 2012 |
| 4.3 | 24 Temmuz 2013 |
| 4.4 | [KitKat](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9LaXRLYXQ) | 31 Ekim 2013 |
| 5.0 | [Lollipop](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9Mb2xsaXBvcA) | 3 Kasım 2014 |
| 5.1 | 9 Mart 2015 |
| 6.0 | [Marshmallow](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9NYXJzaG1hbGxvdw) | 5 Ekim 2015 |
| 7.0 | [Nougat](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9Ob3VnYXQ) | 22 Ağustos 2016 |
| 7.1 | 4 Ekim 2016 |
| 8.0 | [Oreo](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9PcmVv) | 21 Ağustos 2017 |
| 8.1 | 5 Aralık 2017 |
| 9 | [Pie](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF9Q) | 6 Ağustos 2018 |
| 10 | [10](https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF8xMA) | 3 Eylül 2019 |

**2.4 Android Uygulama Geliştirme**

Android uygulamarı Java veya kotlin dilinde, Android studio da yazılır. Uygulama geliştirilirken, geniş ve zengin Views kullanılır. Bu yapı, listeleri, textleri, butonları hatta gömülü web tarayıcısını içerir.

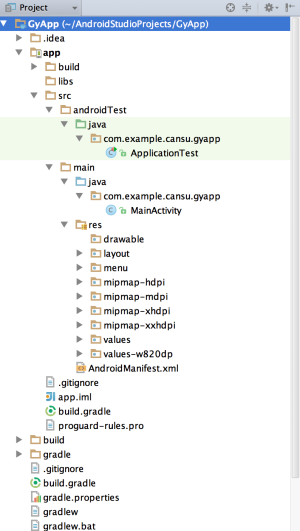
Content Providers, uygulamaların birbirlerinin verilerine erişmelerini ve verilerinin paylaşmalarını izin verir.

Resource Manager, kodlama gerekmeyen verilere erişim sağlar.

Notification Manager,status barda tüm uygulamaların uyarılarının gözükmesini sağlar.

Activity Manager,uygulamaların yaşam döngüsünü yönetir ve navigasyon sağlar.

**2.5 Android Proje Yapısı (Android Studio)**

****

build klasörü: Belirli derleme değişkenlerine göre derleme dosyalarını içerir. Ana uygulama modülü içinde bulunur.

libs klasörü: Eğer uygulamamızda Android SDK içinde yer almayan bir Java kütüphanesi (örneğin AdMob) kullanıyorsak, ilgili JAR dosyalarını bu klasöre atmamız gerekir. JAR dosyası ekledikten sonra dosyaya sağ tuşla tıklayıp Add To Build Path seçeneğini seçmeyi unutmamalıyız. Aksi takdirde derleyici jar dosyasını göremez.

src klasörü: Kaynak dosyaları burada yer alır. Bu klasörün içinde Java dosyalarını tutacağız.

src/androidTest klasörü: Araçlı test belgelerini içerir.

src/main/java klasörü: Uygulama aktiviteleri için Java kaynak kodlarını içerir.

src/main/jni/ klasörü: Java Native Interface (JNI) kullanan yerli kodları içerir.

src/main/gen/ klasörü: R.Java ya da AIDL dosyalarından üretilmiş arayüz belgeleri gibi Android Studio tarafından yaratılan belgeleri içerir.

src/main/assets/ klasörü: Genelde boştur. raw değerli dosyalarınızı barındırmak için kullanabilirsiniz. Buraya koyduğunuz dosyalar, .apk dosyasının içine derlenir.

src/main/res klasörü: Uygulamayla ilgili her türlü kaynak bu klasörde tutulur. Burada tutulan dosya tipleri resim dosyaları, ekran tasarımları ve metin dosyaları olabilir. Şimdi bu klasörü biraz daha detaylı inceleyelim:

drawable klasörleri: Bu klasörler uygulamada kullanılan resim dosyalarını içerir. Bu dosyalar PNG ya da JPEG formatında olabilir. Klasörün yanındaki hdpi (high dpi), ldpi (low dpi), mdpi (medium dpi) vexhpi (extra hight dpi) cihaza özel ekran çözünürlüklerine göre dosya çağırmamızı sağlar. anim klasörü: Animasyon objeleri için derlenen XML dosyalarını içerir.

color klasörü: Renkleri tanımlayan XML dosyalarını içerir.

mipmap klasörü: Uygulamanızın açılış logolarını içerir.

layout klasörü: Burada ekranlara dair tasarım dosyaları bulunur. xml formatındaki bu dosyalar her ekrana ait tasarımları barındırır. Bir ekran Activity ile oluşturulduğunda onCreate metodu içinde ilgili layout çağırılır ve ekranda yer alacak öğeler oluşturulur.

raw klasörü: İsteğe bağlı raw değerli dosyaları içerir. MP3 ve Ogg gibi medya dosyalarını kaydetmek için uygun bir klasördür.

menü klasörü: Eğer bir ekranda cihazın Menü tuşuna basıldığında bir menü çıkmasını istiyorsak, menü elemanlarını bir xml dosyasında tanımlayarak bu klasör içine saklarız.

values klasörü: Uygulamada kullanılan sabit değişkenler burada saklanabilir. strings.xml dosyası uygulamada kullanılan ve ekranlarda kullanıcıya gösterilen her türlü metni saklar. Anahtar – veri mantığıyla saklanan bu değerler kod içinde ya da layout dosyalarında çağırılır. Aynı zamanda bu değerler R.java dosyasında işaretlenir. Bunun dışında colors.xml dosyası layout dosyalarında kullanılacak önceden tanımlanmış renkleri içerir. styles.xml dosyası ise ekranlarda kullanılan ve yinelayout dosyalarından çağrılan stilleri içerir. values klasörünün bir başka özelliği de cihazın ayarlanmış ana diline göre yerelleşebilmesidir. Bunu ilerleyen bölümlerde anlatacağız.

xml klasörü: aPreferenceScreen, AppWidgetProviderInfo, or Searchability Metadata gibi değerleri tanımlayan çok yönlü XML dosyalarını saklar.

.gitignore dosyası: git tarafından reddedilen kayıt dışı dosyaları tanımlar.

app.iml dosyası: IntelliJ IDEA modülü.

build.gradle dosyası: Derleme sistemi için değiştirilebilir özellikleri içerir.

AndroidManifest.xml dosyası: Bu dosya uygulamanın kalbidir. Uygulamayla ilgili her türlü özellik ve uygulamanın işletim sisteminden talep edeceği bütün izinler burada tanımlanır. Aynı zamanda uygulama içinde kullanılan her ekran burada kaydedilip tanımlanmak zorundadır.

**3 İBM WATSON**

**3.1 Giriş**

Watson, IBM tarafından geliştirilen ve doğal dilde sorulan sorulara cevap vermek için tasarlanan bir yapay zekâ programıdır. Adı, IBM'e önayak olan Thomas J. Watson'dan gelmektedir; Watson, IBM'in DeepQA araştırma projesinin bir parçası olarak geliştirilmektedir. Program, POWER7 işlemcisi tabanlı sistemler üzerinde çalışmaktadır.

Watson, Java ile C++ ile yazıldı ve IBM'in DeepQA yazılımı ile SUSE Linux Enterprise Server 11 kullanmaktadır.

Watson, açık alan soru cevaplamasının alanına doğal dil işleme, bilgi alma, bilgi gösterimi ile akıl yürütme ve makine teknolojileri öğrenmede ileri bir uygulamadır. Çekirdeğinde, Watson, hipotez üretimi, büyük kanıt toplama, analiz ve puanlama için IBM'in DeepQA teknolojisinde inşa edildi.

**3.2 Yapay Zeka ve Makine Öğrenmesi Servis Hizmetleri**

IBM Watson, eğitim verilerini toplamak ve hazırlamak ve makine öğrenimi modelleri tasarlamak, eğitmek ve dağıtmak için kullanabileceğiniz yapay zeka araçlarıyla birlikte çalışan bir ortamdır. Sundugu hizmetler 3 kategori altında toplanabilir.

i) Visual recognition (görüntü tanımlama)

ii) Natural language classification (doğal dil sınıflandırması)

iii) Watson Machine Learning (makine öğrenimi)

**3.3 Görüntü Tanıma Servisi (Visual Recognition Services)**

Özellikle görsel tanımaya odaklanan araçları barındıran hizmetidir.

Built-in models: Herhangi bir eğitim almadan sahneler, nesneler ve diğer birçok kategori için görüntüleri analiz etmek için kullanılabilen hizmetidir.Projemizde Önceden besinler üzerinde Eğitimi Yapılmış “Food” modelini kullandık.

A model builder: tanımladığınız sınıflara göre görüntüleri sınıflandırmak için bir modeli eğitmeyi hızlı ve kolay hale getirir.

Core ML support: görsel tanıma özel modellerinizi iOS uygulamalarında kullanmak için kullanılır.

**BÖLÜM 2**

**Android Uygulaması**

**1.1 Giriş**

Uygulamanın iç dinamikleri ve kod yapısını 3 alt başlık altında inceleyeceğiz

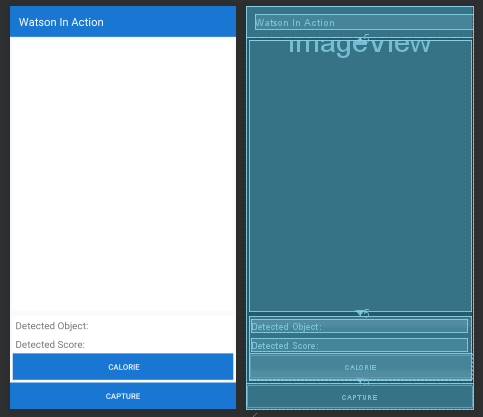
Layout (activity\_main.xml) = Arayüzün tanımlandığı dosya

Permissions (manifest.xml) = Gerekli izinlerin tanımlandıgı dosya(kamera,internet)

Activity (MainActivity.java) = Sınıfların ve Fonksiyonların yazıldığı temel uygulama dosyası

**1.2 Layout (activity\_main.xml)**

Arayüz tasarımında ekranın üst orda kısmında bir resim görüntüleyici, alt kısmında metin görüntüleyici, en altta ise 2 adet buton bulunmaktadır CAPTURE butonu Activity altında captureImage() isimli fonksiyonu çalıstırmaktadır. CALORİE butonu ise kalori değerini ekrana bastırmaktadır.



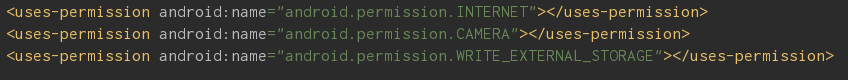
**1.3 Permissions (AndroidManifest.xml)**

AndroidManifest.xml dosyası uygulamanın kullanacagı özelliklere izin tanımlamak için kullanılır.Geliştirilen uygulama için:

İnternet : watson bulut servis hizmetine tanımlanması istenen resimi yüklemek ve cevabını almak için gerekli

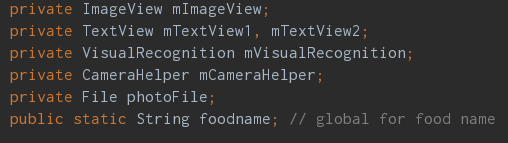
Kamera: Kalori değerinin öğrenilecegi besinin fotografı için gerekli

Hafıza : Çekilen resimlerin cihazda barınması için gerekli

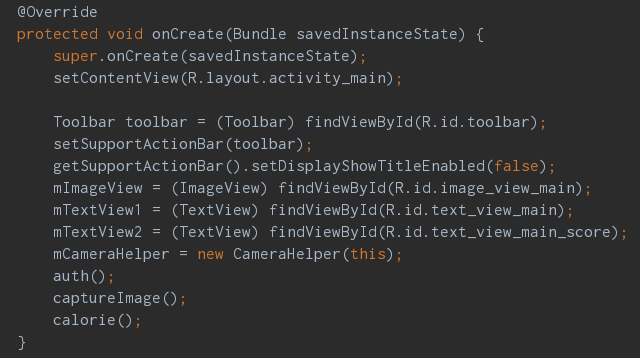


**1.4 Activity (MainActivity.java)**

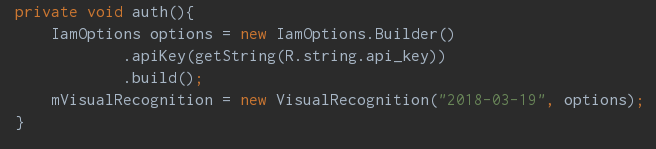
MainActivity.java dosyası uygulamanın asıl fonksiyonlarının tanımlandıgı dosyadır. VisualRecognition, CameraHelper isimli 2 adet global nesne tanımlandı. Bu nesneler ile watson hizmetine erişim ve kamera fonksiyonlarını yönetildi.



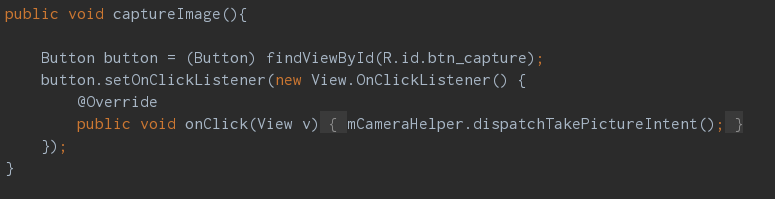
onCreate() fonksiyonu altında uygulamanın kullanıllanacagı fonksiyonlar ve activityler belirtildi.

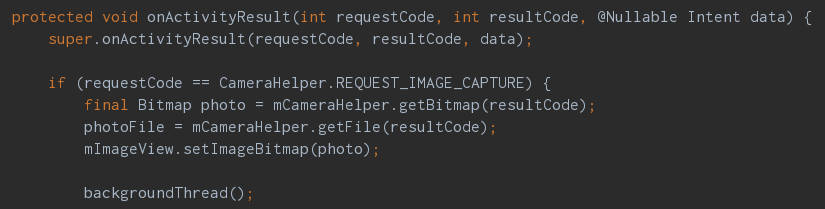


Watson görüntü tanıma servisine erişmemiz için gerekli keyi String.xml dosyasına belirttik. Bu keyden auth() fonksiyonu altında mVisualRecognitation nesnesi oluşturduk.



Arayüzdeki CAPTURE butonu ile ilişkilendirilen görevi resim android’te hazır kamera uygulamasını çalıstırmaktır.



OnActivityResult() yöntemi altında,ilk olarak çekilen resmi bitmap formatına çeviren getBitmap fonksiyonu kullanıldı. Daha sonra ağ çağrısı yapmak, sonucu ayrıştırmak ve sonucu kullanıcı arayüzünde görüntülemek için bir backgroundTread(); fonksiyonu tanımlandı. 

backgroundThread() fonksiyonu altınıda , classifyOptions ile tanımlama ayarlarımızı tanımlıyoruz(doğruluk en az kaç olcak, hangi eğitilmiş modeli kullanacak gibi) Daha sonra

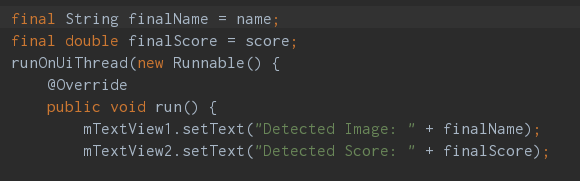
ClassifiedImages result = mVisualRecognition.classify(classifyOptions).execute();

Komutu ile mVisualRecognition nesnesi üzerinden ayarlarını belirttiğimiz resmimizi ibm watson’a gönderiyoruz gelen sonucu ise result değişkeninde tutuyoruz.

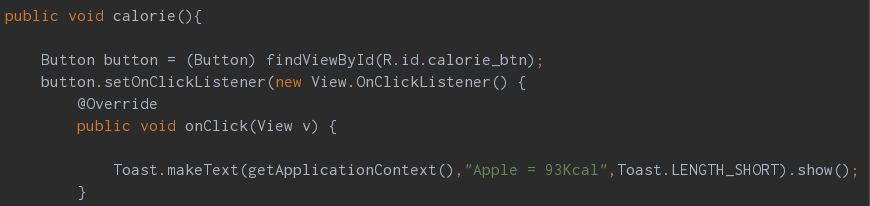
Java objelerini json stringine çeviren Gson kütüphanesini tanımlıyoruz.



İbm watson’ın json stringi içerisinde watson’ın gönderdigi tanımlamalar bulunmaktadır. Class ve Score ( isim, doğruluk) değerlerini name ve score değişkenlerine ekranda text viewde bastırmak üzere atıyoruz.



Son olarakta resmini gönderip tanımlamasını aldıgımız besinin kalori değerini bastıroyuruz



**BÖLÜM 3 Sonuç**

**1.1 Ne Öğrendik?**

Bu proje kapsamında yapay zekanın makine öğrenmesi alanı alında görüntü tanımlama teknolojileri hakkında bilgi edindik. Google firebase, IBM watson, Microsoft Azure gibi servisler hakkında bilgi edindik.

Android alanına giriş yapıp temel olarak Arayüz tanımlamayı, bileşen eklemeyi, eklenen bileşenlerin fonksiyonlarını tanımlamayı öğrendik. Uygulama Yaşam döngüsü ve idle hakkında bilgi edindik.

**1.2 Kaynakça**

<https://www.ibm.com/tr-tr/cloud/watson-visual-recognition>

<https://developer.android.com/guide>

<https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-201/bir-android-projesinin-yapisi>

<https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-201/android-mimarisi-ve-sistem-ozellikleri>

<https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQW5kcm9pZF8oacWfbGV0aW1fc2lzdGVtaSkjR2VsacWfdGlybWU>

<https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvV2F0c29uXyhiaWxnaXNheWFyKQ>

<https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/visual-recognition-overview.html>