**Metal Damgalama Görüntüleri Üzerinden Karakter Tanımlaması**

Erdem Nayın 180202050 – Melih Yeşilyurt 180202060

Barış Arslan 180202112 – Ali Atilla Aydemir 180202063

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

**1.Problem**

Günümüzde birçok fabrikada ürün numaraları metal yüzey üzerine yazılmaktadır. Bu firmalara örnek olarak Aygaz ve birçok otomotiv firması tarafından kullanılmaktadır. Bu projede çözülmesi hedeflenen sorun ise yazı metni ile arka plan renginin aynı olmasından dolayı OCR modellerinin yazıyı tespit etmemesi.

**2. Karşılaşılan Sorunlar**

**a. Metin Rengi ve Arka Planın Aynı Renk Olması**

Bu projenin zor olmasının en büyük sebebi metin rengi ile arka planın aynı renk olmasıdır. Bu sebeple genel olarak kullanılan OCR kütüphaneleri ve çoğu model bu yazıları tespit etmede yetersiz kalmaktadır.

**b. Veri Seti Yetersizliği**

İnternette detaylı bir araştırma yapmamıza rağmen ilgili sitelerde veri seti bulamadık. Örnek olarak bir fotoğraf verildiğinden dolayı çalışmalarımızı o fotoğraf üzerinden gerçekleştirdik. Daha sonra dersin hocasıyla iletişime geçtik ama aldığımız veri seti kalitesizdi. Veri setinden 300 fotoğraf bulunmaktaydı. Veri madenciliği dersinden öğrendiğimiz bilgileri kullanarak veri setini temizledik. Elimizde 210 fotoğraf kaldı. Kalan fotoğrafların çok düşük çözünürlüğe sahip olmasından ve sadece sayılardan oluşmasından dolayı işimize yaramadı ve kullanamadık ama daha sonra yaptığımız araştırmalar sonucunda kullanabileceğimiz ve bize uygun olan 2 fotoğraf daha elde ettik. Çalışmalarımızı elimizdeki imkanlarla gerçekleştirmek zorunda kaldık.

**c. Tecrübe Yetersizliği**

Daha önceden aldığımız veri madenciliği dersinin devamı olarak aldığımız yapay zekâ dersi hakkında daha önceden tecrübemiz ve çalışmamız bulunmadığı için duyduğumuz her kavramı ilk kez duymuş bulunmaktayız. Bu sebeple projemizi geliştirmeden önce kendimizi yapay zeka alanında geliştirdik.

**d. Test Edilmiş filtrelerin Kodlarının Bulunmaması**

Geliştirdiğimiz programı daha verimli hale getirmek için çeşitli filtreler kullandık. Bu filtreleri hızlı bir şekilde deneyebilmek ve sonuçlarını görebilmek için bazı yararlı internet siteleri bulduk. Ama bu sitelerde denediğimiz her filtrenin kodunu bulamadık. Kodlarını bulabildiğimiz filtrelerin farklı farklı kombinasyonları denemek epey zamanımızı aldı.

**e. Benzer Projelerin Bulunmaması**

Bu projenin konusu zor olduğundan dolayı internette hazır proje örnekleri bulamadık. Bulduğumuz projeler genellikle normal yazı tespitiyle alakalıydı. Bizim problemimiz ise daha karmaşık olduğundan dolayı o projelerden yeteri kadar faydalanamadık.

**3. Denediğimiz Başarısız Çalışmalar**

Uygulama açıldığında uygulamanın üst kısmında üç sorguya ithafen yapılan üç fragmentin navigation barı görülür. Kullanıcı buradan sorgular arasında geçiş yapabilir.

Uygulama açıldığında kullanıcının karşısında ilk tip sorgunun yapıldığı birinci fragment kısmı gelmektedir. İlk tip sorgu olarak “En fazla yolcu taşınan 5 günü ve toplam yolcu sayısını listeleyiniz.” seçildi. İlgili butona tıklanıldığında sorgu CloudSQL’ e gönderilir, sorgu sonucu uygulamaya geri döndürülerek sorgu sonucunda elde edilen bilgiler ekranda gösterilir.

Kullanıcı eğer navigaton barı kullanarak ikinci tip sorgunun yapıldığı ikinci fragment kısmına girerse karşısına iki adet tarih girebileceği iki textbox gelir. İkinci tip sorgu olarak “İki tarih arasında seyahat edilen en az mesafeli 5 yolculuk hangisidir?” sorgusu seçildi. Kullanıcı istediği iki tarihi girdikten sonra ilgili butona tıkladığında sorgu CloudSQL’ e gönderilir, sorgu sonucu uygulamaya geri döndürülerek sorgu sonucunda elde edilen bilgiler ekranda gösterilir.

Kullanıcı eğer navigation barı kullanarak üçüncü tip sorgunun yapıldığı üçüncü fragment kısmına girerse karşısına tarih girebileceği bir textbox gelir. Üçüncü tip sorgu olarak “Belirli bir günde en uzun seyehatın harita üstünde yolunu çiziniz. Başlangıç ve varış konumları lokasyonun merkezi kabul edip mesafeye göre bir yol bulunmalıdır” sorgusu seçildi. Kullanıcı istediği tarihi girdikten sonra ilgili butona tıkladığında sorgu CloudSQL’e gönderilir, sorgu sonucu uygulamaya geri döndürülür. Uygulama ara yüzünde harita açılır. Döndürülen veriler kullanarak harita üzerinde iki lokasyon arasında rota çizdirme işlemi yapılır ve harita yolculuk başlangıç lokasyonuna doğru zoomlanarak ilgili rota kullanıcıya gösterilir

**4. Bulduğumuz Çözüm**

1. MainActivity

Bu class uygulamanın çalışmasını sağlar. Ayrıca database ile olan bağlantıyı bu class gerçekleştirmektedir. Bütün sorgular bu class da bulunur. Diğer classlar tarafından çağırılırlar.

1. Fragment1

Bu class tab menüsünün birinci sekmesinin çalışmasını sağlar. İçerisindeki buton sayesinde birinci sorguyu çağırır. En fazla yolcu taşınan 5 günü ve toplam yolcu sayılarını ekrana bastırır.

1. Fragment2

Bu class tab menüsünün ikinci sekmesinin çalışmasını sağlar. İçerisindeki buton sayesinde kullanıcıdan tarihleri alır ve ikinci sorguyu çağırır. İki tarih arasında seyahat edilen en az mesafeli 5 yolculuğu ekrana bastırır.

1. Fragment3

Bu class tab menüsünün üçüncü sekmesinin çalışmasını sağlar. İçerisindeki buton sayesinde kullanıcıdan tarih alır ve yeni bir activity açarak, SubActivity classını çalıştırır.

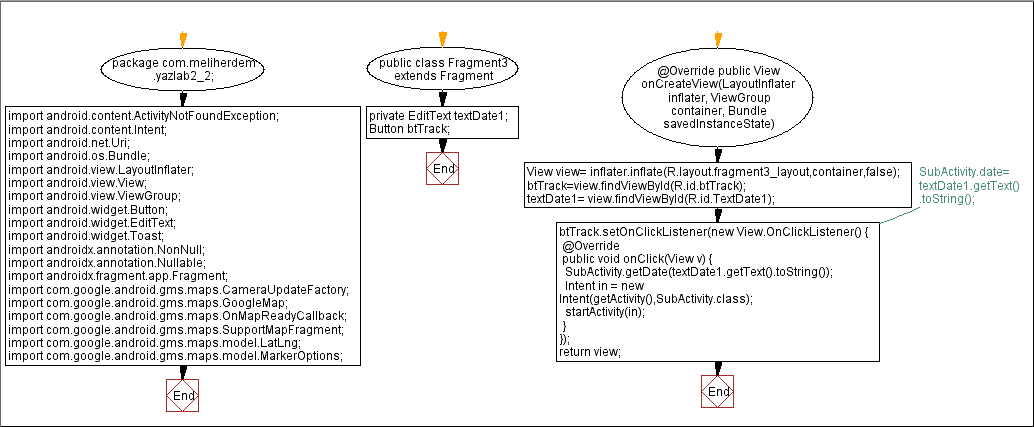
1. SubActivity

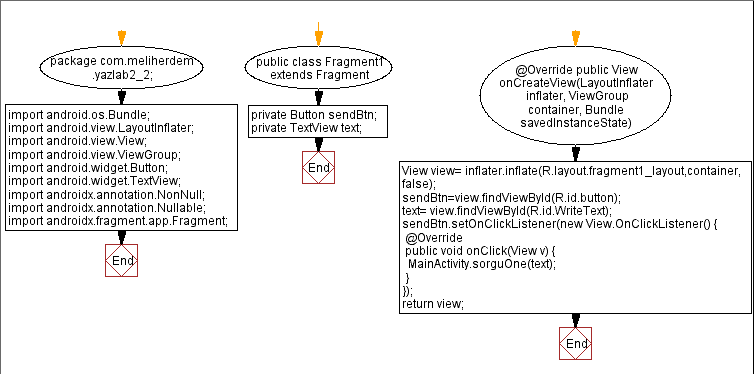
Bu class Fragment3 classı tarafından çalıştırılır. İçerisinde Google map apiye bağlanmış harita bulundurmaktadır. Fragment3 classı tarafından gönderilen tarihi kullanarak 3. ve 4. Sorguları çalıştırır. Elde ettiği noktaların koordinatlarını haritada bularak aralarındaki en kısa yol güzergahını çizdirir.

**5. Karşılaşılan Problemler**

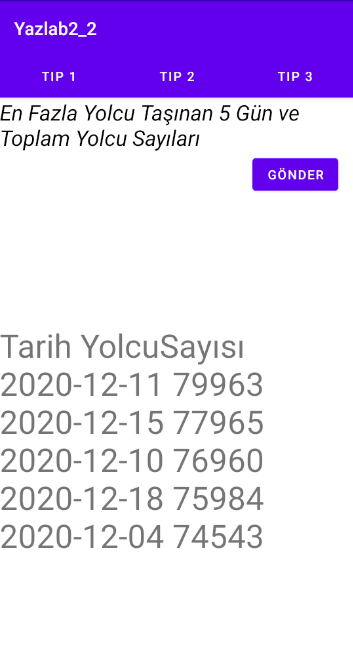
1. Android ile uygulama geliştirme konusunda deneyimli olmadığımız için arayüz kısmını tasarlama aşamasında biraz zorlandık. Yaptığımız araştırmalar sonucunda fragment yapısını ve fragmentler arasında geçiş yapmayı sağlayan navigaton bar ikilisini kullanmaya karar verdik. Kullanıcı arayüzünü bu sayede 3 ayrı bölmeye ayırıp sorguların ve sorgu sonuçlarının birbirlerinden ayrılmasını sağladık.
2. Daha önceden bize verilen .csv uzantılı taksi kaydı datalarının cloud ortamına yüklenmesi konusunda zorluk yaşadık. Projede ilk önce Firebase Firestore kullanmaya karar verdik. CSV uzantılı dosyamızı Firestore database’ine direkt olarak yükleyemeyeceğimizi öğrendik. Bu dosyayı yüklemek için ilk önce JSON formatına getirip daha sonra yükleyebileceğimizi öğrendik. CSV dosyamızı JSON formatına dönüştürürken zorluklar yaşadık. CSV dosyasında yaklaşık 1.5 milyon kayıt bulunduğundan online converterlar dosyayı çeviremiyordu. Dosyadan kayıt silip çevirmeye çalıştığımızda bazı dataların hatalı şekilde çevrildiğini ve data kaybı yaşadığımızı gördük. Daha sonra çeşitli programlar aracılığıyla bu dosyayı çevirmeye karar verdik. Kullandığımız hiçbir program bu dosyayı çeviremedi. Daha sonra python kullanarak bu dosyayı çevirebileceğimizi öğrendik ve bulduğumuz python kodunu ve kütüphaneyi kullanarak datamızı çevirmeye çalıştık. Fakat data çok büyük olduğundan ve datada hatalı değerler bulunduğundan yine çeviremedik. Bu dosyadan daha küçük olan taxi zones dosyasını ise kolaylıkla çevirebildik. Daha sonra bu data çok büyük olduğundan datayı kırparak python koduyla birlikte çevirmeyi başardık ve Firestore’a yükleyebildik.
3. Firestore’a yüklediğimiz datalar üzerinde sorgularımızı yapmaya çalıştık fakat Firestore genelde güncel android uygulaması yazılımı koşullarını karşılıyordu. Bizim istediğimiz şekilde dataları sorgulamayı başaramadık. Firestore relational database olmadığı için işimize tam anlamıyla yaramadığını, sadece basit data al ver işlemlerini yapabildiğini gördük. SQL sorgularını Firestore sorgularına çeviren bir kütüphane bulduk ve onu kullanarak sorgulama yapmaya çalıştık fakat bu kütüphane de işimizi görmedi. Bu yüzden yine Firebase’e ait olan RealtimeDatabase’e geçmeye karar verdik.
4. Firestore RealtimeDatabase’e csv dosyamızı kolaylıkla import edebildik çünkü RealtimeDatabase bu desteği kendi ara yüzünde sunmuştu. Yüklediğimiz datalar üzerinde Google Cloud hizmeti olan BigQuery hizmetini kullanarak bu datalar üzerinde sorgulama işlemlerimizin demosunu yapmış olduk ve sorgularımız hazırlanmıştı. Daha sonra RealtimeDatabase sorguları nasıl yapılır araştırmaya başladık. Sonuç olarak yine hüsrana uğradık ve RealtimeDatabase’in sorgularının Firestore sorgularından bile daha az kapsamlı olduğunu ve sql queryleri gibi yapılmadığını gördük. Bizim istediğimiz şekilde sorgular yapılamıyordu.
5. Daha sonra olarak yine Google’ın Cloud hizmeti olan ComputeEngine sanal bilgisayarı üzerinde tanımladığımız yine Google’ın Cloud hizmeti olan CloudSQL for MySQL kullanmaya karar verdik. Sorgularımızı MySQL sorgualarına çevirdik. Yaşadığımız bazı hatalardan dolayı CloudSQL for MySQL kullanmayı bırakıp CloudSQL for MSSQL kullanmaya başladık. Microsoft SQL Server Management Studio kullanarak bağlandık ve sorgularımızı tekrardan MSSQL sorgularına uygun haline getirdik. CloudSQL ile uygulamamzın bağlantısını yaparak uygulama içinden sorguları CloudSQL’e göndererek gerçekleştirdik.
6. Üçüncü tip sorgunun Google Maps API kullanılarak harita üzerinde gösterilmesinde sorunlar yaşadık. Araştırdığımız bazı çözümler, izlediğimiz bazı tutoriallar eski olduğundan depracated olmuş birçok fonksiyon ve kullanım yöntemleriyle karşılaktık. Bu sorunları Android Developers dökümantasyonlarını kullanarak düzeltmeye ve güncellemeye çalıştık fakat olmadı. Yaptığımız sayısız araştırma sonucu güzel bir biçimde nasıl çizdirebileceğimizi bulduk ve uygulamamızın harita üzerinde rota çizimi işlemini böylece gerçekleştirmiş olduk.
7. Üçüncü tip sorgunun sonucunda elde edilen lokasyon bilgilerinin fragment ve activity arasında iletilmesinde sorun yaşadık. Kullandığımız taxi\_zones.csv dosyasında latitude ve longitude veri tiplerini float olarak belirlemiştik ve bu şekilde importlamıştık fakat bunu uygulamaya çekerken data kaybı olduğunu gördük. Bu yüzden tabloyu silip baştan bütün değerleri nvarchar(max) veri tipinde yükledik ve uygulamanın içinde value casting işlemlerini gerçekleştirdik. Böylece rota çizdirme sorunlarımızın da üstesinden gelmiş olduk

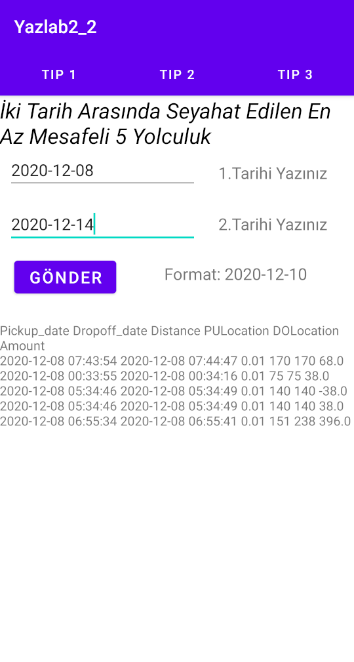
**6. Akış Diyagramı**

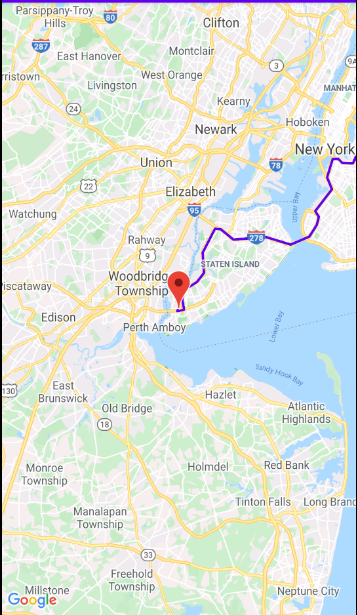




**7. Deneysel Sonuçlar**

****

****

****

**8. Kaynakça**

* <https://www.youtube.com/watch?v=TY-2Cx4IW9A&list=WL&index=40>
* <https://www.youtube.com/watch?v=fZIZ81fn7b4&list=WL&index=40>
* <https://www.youtube.com/watch?v=58AxNh2cWRU&list=WL&index=39>
* <https://www.youtube.com/watch?v=UnpbVqBSsf4&list=WL&index=37&t=451s>
* <https://introcs.cs.princeton.edu/java/44st/Location.java.html>
* <https://stackoverflow.com/questions/21646019/errors-in-sectionspageradapter-class-when-referring-to-specific-fragment>
* <https://stackoverflow.com/questions/10609268/what-is-the-difference-between-fragment-and-fragmentactivity>
* <https://medium.com/gokhanyavas/javada-dizi-kullanımı-5365ec363409>
* <https://www.youtube.com/watch?v=NwW2QTpArac>
* <https://www.youtube.com/watch?v=vmaVJazDcNc>
* <https://www.youtube.com/watch?v=TSFlTR7D9Wo>
* <https://www.youtube.com/watch?v=VR8RKM9LTyA>
* <https://www.udemy.com/course/tam-kapsaml-android-programlama-egitimi/learn/lecture/11786118?start=15#overview>
* <https://www.youtube.com/watch?v=WXKhmWRPEtY>