

Proje Ana Alanı : Yazılım

Proje Tematik Alanı : Yapay Zekâ

Proje Adı (Başlığı) : Yapay Zekâ Uygulamaları ile Doğru Maske Kullanım Oranı ve Mutluluk İndeksi Hesaplama

Özet

Ülkemizde Türkiye İstatistik Kurumu (TÜİK) her yıl “Yaşam Memnuniyet Araştırması” yapmaktadır. Yayınladığı raporda TÜİK insanların mutluluk düzeyinin yaşa ve cinsiyete göre dağılımını ayrı ayrı hesaplamaktadır. 2019 raporuna göre mutlu olduğunu beyan eden erkeklerin oranı %47,6 iken, kadınlarda bu oran %57,0 oldu. Ayrıca Sinop %77,6 oranla en mutlu il olarak raporda ilk sırada yer aldı. Birleşmiş Milletler ve Avrupa İstatistik Ofisi de her yıl ülkeler düzeyinde anket yaparak Dünya Mutluluk İndeksi Raporu yayınlamaktadır.

Yapay Zekâ (YZ) uygulamaları ile mutluluk indeksi hesaplama Dünya’da henüz deneysel denilebilecek düzeyde kullanılsa da kişiden kişiye değişen öznel verilerin yapay zekâ algoritmalarıyla hesaplanabilmesi, önceden elde edilmesi çok zor veri analizlerinin yapılabilmesinin önünü açmıştır. Örneğin; IBM şirketi kendi çalışanlarının mutluluk oranlarını ölçmek için, Dubai’de kamu binalarında kamu hizmeti alan vatandaşların memnuniyet oranını hesaplamak için, mağazalar tüketicilerin memnuniyet oranını hesaplamak için kullanmaktadır.

Kendi geliştirdiğimiz model ile beraber toplam 4 adet evrişimli sinir ağını (CNN) birlikte kullanarak bir masaüstü uygulaması geliştirdik. Yazılım bir görüntü akışındaki insan yüzünü algılar, eğer bu insan yüzünde maske bulunuyorsa hangi oranda doğru maske kullanıldığını ölçer. Eğer maske bulunmuyorsa kişinin cinsiyetini, yaşını ve mutluluk oranını tahmin eder ve mutluluk düzeyinin cinsiyet ve yaş grubuna göre dağılımını hesaplar.

Doğru şekilde medikal yüz maskesi takmış, yanlış maske takmış ve maske takmamış şeklinde 3 sınıfa ayrılmış insan yüzü görselinden oluşan toplamda 24.000 adet CC0 (Creative Commons Zero) lisanslı veri kümesini kullanarak Python programlama dili ile geliştirdiğimiz CNN modelini eğiterek %97 test başarımla oranın ile görüntü sınıflandırması yapmayı başladık. Bu sayede bir kişinin hangi oranda doğru yüz maskesi taktığını ölçebiliyoruz.

Anahtar kelimeler: Makine Öğrenmesi, Mutluluk İndeksi, World Happiness Index, Görüntü Sınıflandırma, Medikal Yüz Maskesi Algılama

YAPAY ZEKA UYGULAMALARI İLE DOĞRU MASKE KULLANIM ORANI VE MUTLULUK İNDEKSİ HESAPLAMA (HAPPY INDEX)

Amaç

Dünyanın dört bir yanında, yeni tip korona virüse karşı alınan önlemlerin başında medikal yüz maskesi takmak geliyor. Yapay zekâ uygulamalarından derin öğrenme tekniğini ile, kameradan alınan video akışını kullanarak bir insanın maske takıp takmadığı, takıyor ise hangi oranda doğru maske kullanıldığını anlık hesaplayabiliyoruz. Örneğin yüz maskesi takılı iken burun açıksa doğru maske takma oranı %50 ile %70 arası çıkmaktadır. Bu veri istatistiksel olarak kullanılabilir veya geliştirdiğimiz yaklaşım çok farklı uygulamalara entegre edilerek de kullanılabilir. Örneğin bir kapı giriş-çıkış takip sistemine entegre edilerek, insanlar doğru yüz maskesi kullanmıyor ise uyarılabilir veya giriş izni verilmeyebilir.

Bir insanın yüz maskesi takip takmaması nesnel bir veridir kişiden kişiye değişmez, anlaması ve hesaplaması kolaydır. Her insan mutluluğu farklı bir şekilde tanımlasa da genel olarak insanlar mutlu olmanın hayatlarındaki en önemli hedef olduğunu söylerler. Mutluluk öznel bir kavramdır ve kişiden kişiye değişebilir. Öznel verileri baz alan anketler yapmak kendi içinde zorluklar içerir. Mutluluk indeksi araştırması yapan kişiler, şirketler, ülkeler ve kurumlar anlamlı veriye ulaşmada zorluklar yaşarlar. Projemizin amaçlarından biri de yapay zekâ algoritmalarını kullanarak bir ortamda bulunan insanların mutluluk oranının yaşa ve cinsiyete göre dağılımını kamera görüntüsü kullanarak hesaplamaktır.

Projemizi geliştirmek için ilk aşamada görüntü sınıflandırması yaparak insanların ne kadar doğrulukta yüz maskesi taktığını ölçen modeli tasarlamamız ve eğitmemiz gerekiyordu. Bu aşamada zorlandığımız konu uygun veri kümesini bulmaktı. Araştırmalarımız sonucu açık kaynaklı CC0 lisanslı doğru maske takmış, yanlış maske takmış ve maske takmamış olarak 3 sınıfa ayrılmış insan yüzü görsellerinden oluşan bir veri kümesini bulabildik.

İkinci aşamada masaüstü uygulamasını geliştirmemiz gerekiyordu. Uygulama bir video akışı içerisindeki insan yüzlerini algılayacak, daha sonra ne kadar doğrulukta yüz maskesi kullanıldığını hesaplayacak, eğer kişide yüz maskesi yok ise kişinin yaşını, cinsiyetini ve mutluluk oranını hesaplayacak. Bu aşamada uygulama içerisinde birbirleriyle entegre çalışan 4 farklı CNN (Convolutional Neural Network) ağını kullanmamız gerekiyordu. Bu durum çok fazla işlem gücü gerektirdiğinden ilk zamanlar saniyede sadece bir ölçüm alabiliyorduk. Etkili ve hızlı ölçüm yapabilmek için yazılım içindeki fonksiyonların birbirini bekletmemesi ayrıca her fonksiyonun 0,1 saniye gibi bir sürede çalışması gerekiyordu. Geliştirdiğimiz yazılımı sürekli iyileştirerek saniyede 10 ölçüm almayı başardık. Bunun için kendi ettiğimiz modeli daha az sayıda birimden oluşan yapay sinir ağı bulunacak şekilde yeniden tasarladık ve aynı sonucu almak için modelimizi daha uzun süre(dönem) eğittik. İlk zamanlar ölçüm alabilmek için kameranın önünde hareketsiz durmak gerekirken son aşamada kameranın önünden durmadan yürüyerek geçen insanlardan ölçüm almayı başardık.

Giriş

Duygular insan zihninin psikolojik bir halini temsil etmektedir. Duygular uyarıldığında genellikle dış görünüşte, yüz ifadesinde, jestte ve duruşta değişiklikler meydana gelmektedir. Günümüzde insan-yapay zekâ etkileşimli akıllı sistemlerin yoğun bir şekilde geliştirilmesi ile bu değişiklikler, yapay zekâ tarafında belirli ölçülerde izlenebilir ve insanların duygu durumları dış görünüşlerinden saptanabilir hale gelmiştir (Doğan ve Özkol, 2018).

Günümüzde veri kurum ve kuruluşlar için son derece önemli bir hale gelmiştir. Özellikle öznel verilerin yapay zekâ algoritmalarıyla hesaplanabilmesi, önceden elde edilmesi çok zor olan veri analizlerinin yapılabilmesinin önünü açacağını öngörmekteyiz. Henüz deneysel denilebilecek uygulamalar bunu göstermektedir. IBM şirketi benzer yaklaşımı kendi çalışanlarının mutluluk oranını ölçmek için kullanıyor. Bu veri, iş gücü devir oranını (Turnover) azaltmak için kullanılabilir. Çin Halk Cumhuriyeti okullarda öğrencilerin mutluluk oranını hesaplamak için kullanıyor. Bu veri eğitim kalitesini artırmak için kullanılabilir. Dubai’de turistik destinasyonlarda MOBESE benzeri şehir içi kamera sistemine entegre bir sistemle turistlerin memnuniyet düzeyini ölçmek için kullanılıyor. Bu veri de turizmin ülke ekonomisine katkısını artırmak amacıyla kullanılabilir. Bazı ülkeler kamu kurumlarından hizmet alan vatandaşların memnuniyet oranını hesaplamak, bazı zincir mağaza şirketleri ise müşteri memnuniyet oranını ölçmek amacıyla kullanıyor.

Basit internet araştırmaları ile bu kullanım alanlarına ve alakalı makalelere ulaşılsa da çeşitli ülkelerin ve şirketlerin uyguladıkları uzun süreli deneysel çalışmaların ne kadar başarılı açık kaynak paylaşılarak olduğu verisine ulaşılamıyor. Ülkemizde de bu konu hakkında birçok akademik çalışma bulunmakta fakat uzun soluklu deneysel uygulamalara rastlamadık. Geliştirdiğimiz uygulamayla hem proje süreci sayesinde hem de sonrasında kodları bu konuya ilgi çekmek istiyoruz.

NOT: Geliştirdiğimiz yazılıma <https://sefakozan.github.io/happy-index/> adresinden ulaşabilirsiniz. Özel bir donanım ve sistem gerekesimi bulunmamakta fakat ekran kartı ortalamanın üzerinde olan güncel bir bilgisayarda denenmesini tavsiye ederiz. Yazılım ana ekranında saniyede işlenen görsel sayısı (FPS) değeri yazmaktadır yaklaşık 10 FPS alıyorsanız bilgisayar konfigürasyonlarınız yeterli demektir.

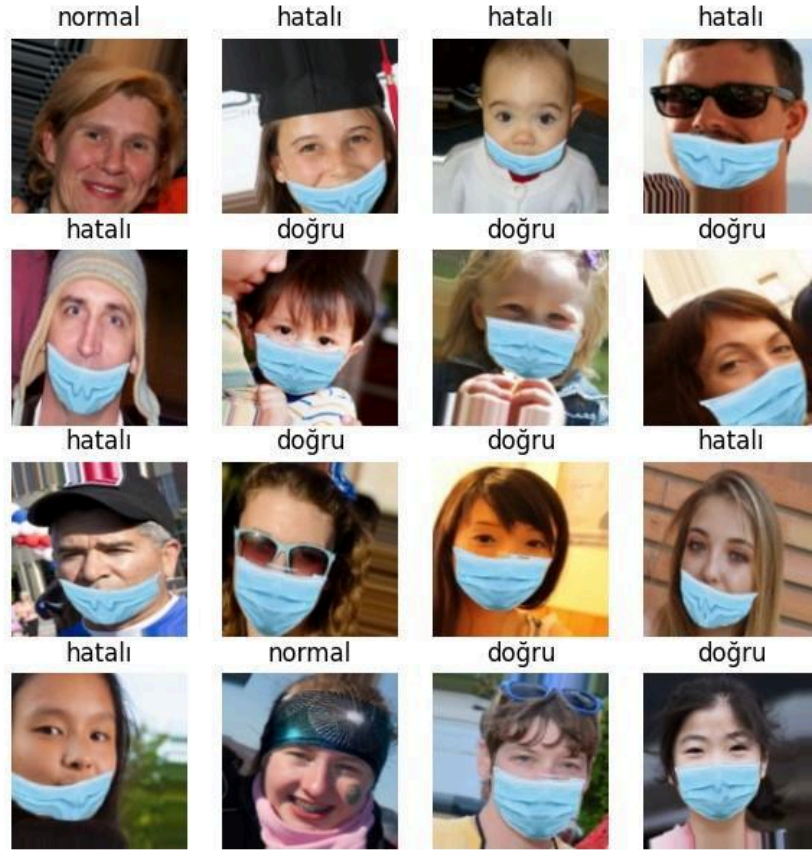
Yöntem

Geliştireceğimiz yazılım bir masaüstü uygulaması olacaktır. Yazılım çalışmaya başladığı andan itibaren video akışı içerisindeki insan yüzlerini ve konumunu tespit edecek. Bu konum bilgisi kullanılarak insan yüzü videodan ayrıştırılır. Yüz görselinde CNN ağı sayesinde ne kadar doğrulukta yüz maskesi kullanıldığı hesaplanacak. Eğer yüz maskesi yoksa yaş, cinsiyet ve mutluluk oranı hesaplanacak.

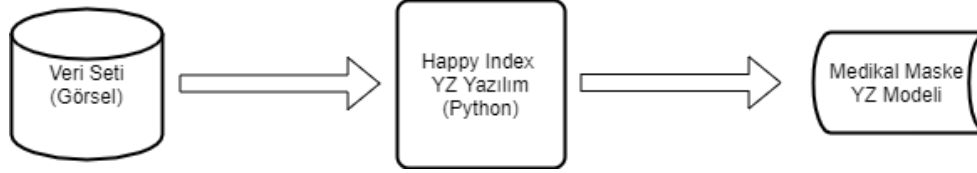
Projenin geliştirilmesi üç temel aşamada planlanmıştır. İlk aşama medikal yüz maskesinin ne kadar doğru kullanıldığını hesaplayacak YZ modelinin Python programlama dili ile geliştirilmesi ve TensorFlow kütüphanesi kullanılarak eğitilmesidir. İkinci aşama yazılımın arayüzünün geliştirilmesidir. Üçüncü aşama yazılımın kodlanmasıdır.

YZ Modelin Eğitilmesi

Bir insan yüzü görselinde medikal maske olup olmadığını algılamamız gerekiyordu. Bunun için ilk aşamada maskeli, maskesiz ve normal insan yüzlerinden oluşan veri kümesine ihtiyacımız vardı. MaskedFace-Net (bkz. 2) deposunda bulunan CC0 lisanslı görsellerden 8.000 maskeli, 8.000 maskesiz ve 8.000 normal görsel seçerek veri kümemizi oluşturduk.



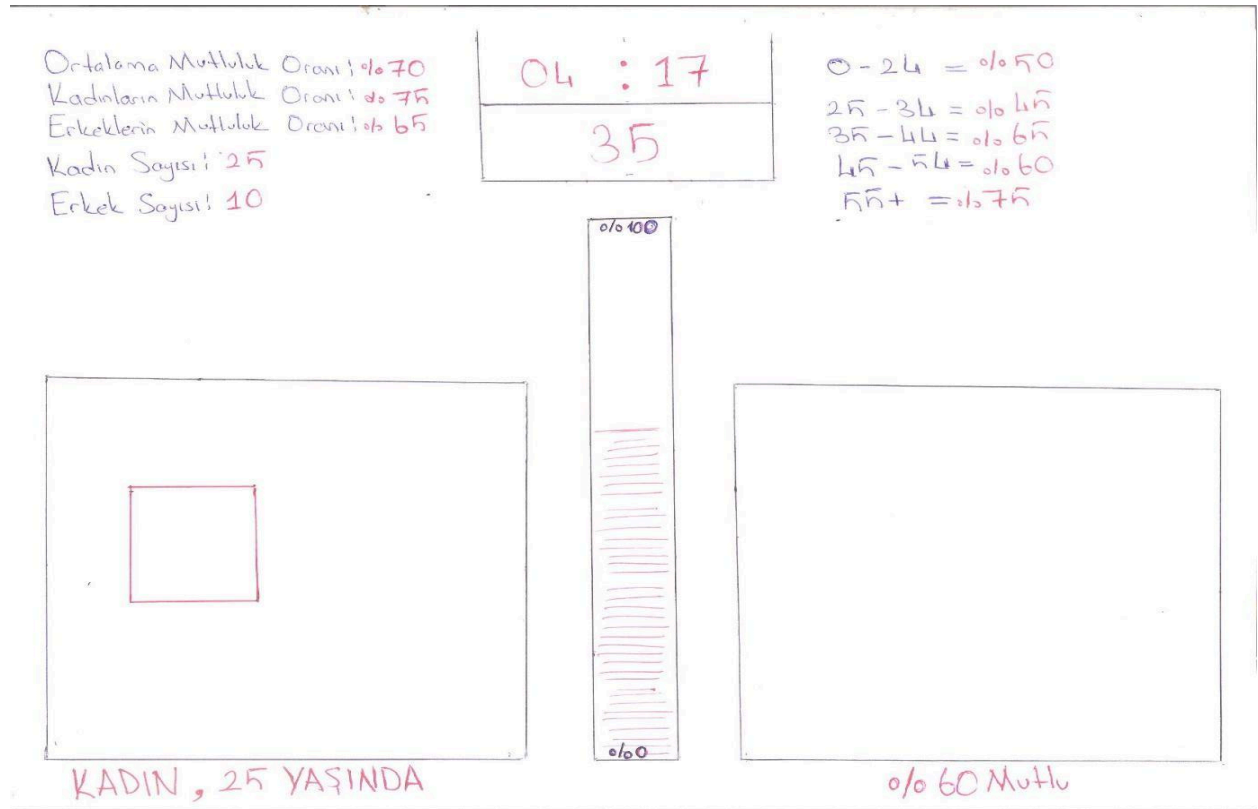
Bu veri kümesini kullanarak Python programlama dili ile geliştirdiğimiz evrimsel sinir ağı kullanan YZ modelimizi eğittik. Python tarafında TensorFlow kütüphanesi kullandık. Bu eğitim sonucunda %96 test başarımlı oranı elde ettik.



Eğittiğimiz model “Happy Index” yazılımında insan yüzlerinde maske olup olmadığını algılamak amacıyla kullanıldı. Ayrıca duygu durumunu, cinsiyet ve yaşı algılamak için TensorFlow.js kütüphanesinin hazır modellerini kullandık.

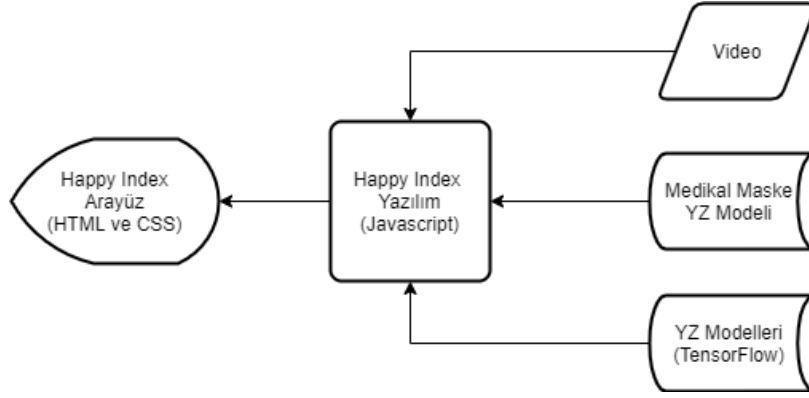
Arayüz Geliştirilmesi

İlk çizilen taslak arayüz şeklindeki gibidir. Değişkenleri kırmızı renk ile gösterdik. Bu değişkenleri ölçüm yaptıkça yazılım ile değiştireceğiz. Bu taslağı kullanarak yazılımı arayüzü HTML ve CSS kullanarak web teknolojileri ile geliştirdik.

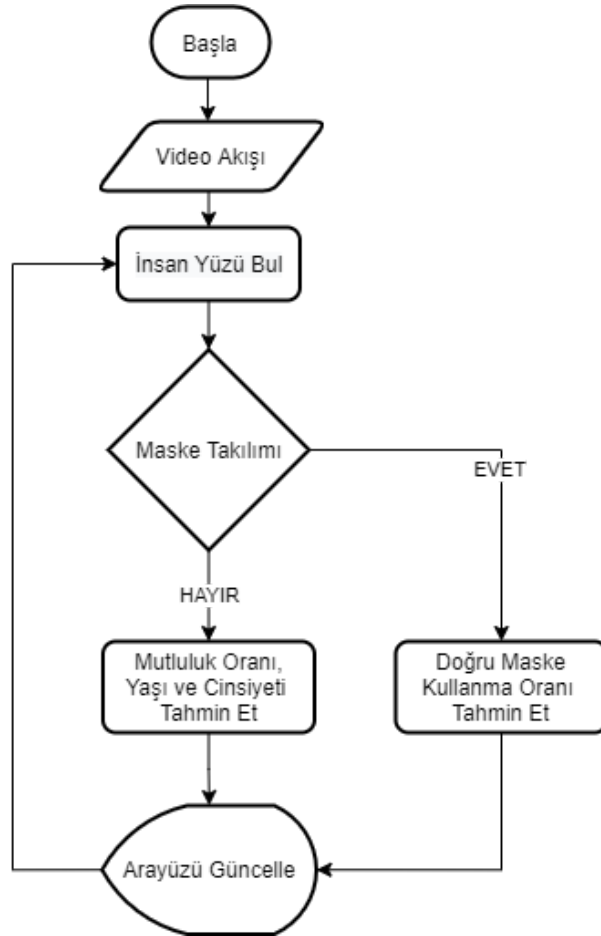


“Happy Index” Yazılımının Kodlanması

Yazılımı Javascript kullanarak geliştirdik. Javascript tarafında YZ modelleri ile tahminde bulunmak için TensorFlow.js kütüphanesini kullandık. Electron sayesinde “Happy Index” yazılımını web teknolojilerini kullanarak masaüstü GUI (Grafiksel Kullanıcı Arayüzü) uygulaması olarak geliştirdik.



Uygulama çalıştığı anda kameradan alınan video akışı içerisinde insan yüzü aramaya başlayacak eğer insan yüzü bulursa ne kadar doğrulukta maske takılı olup olmadığını, yaşı, cinsiyeti ve mutluluk oranını tahmin edecektir.



Proje İş-Zaman Çizelgesi

Geliştireceğimiz yazılımda kullanılan teknolojilerin kolay öğrenilebilir, yaygın ve geliştirme yapması kolay olması bizim için önemliydi. Bu yüzden özellikle literatür taraması yaparken bu konuyu da araştırdık. Evrişimli sinir ağı modeli tasarlamak ve eğitmek için Python programlama dilini ve TensorFlow kütüphanesini tercih ettik. Masaüstü uygulamasını Electron uygulaması olarak geliştirdik bu sayede HTML ve Javascript gibi yaygın ve kolay öğrenilebilir teknolojileri kullandık. Python tarafında, Python uyumlu TensorFlow kütüphanesi ile eğittiğimiz modeli, Javascript tarafında Javascript uyumlu Tensorflow.js kütüphanesi sayesinde masaüstü uygulamasında kullandık.

[illegible]

Bulgular ve Gerçekleme

Projeyi geliştirme sürecinde zamanı verimli kullanabilmek için hız odaklı, dağıtık çalışan bir sürüm kontrol ve kaynak kod yönetim sistemi olan Git kullandık. Halihazırda aşına olduğumuz HTML, CSS ve Javascript gibi web teknolojileri kullanarak yazılımı geliştirmeyi, YZ modelimizi ise Python kullanarak geliştirmeye karar verdik.

Yazılımda Kullanılan YZ Modelleri

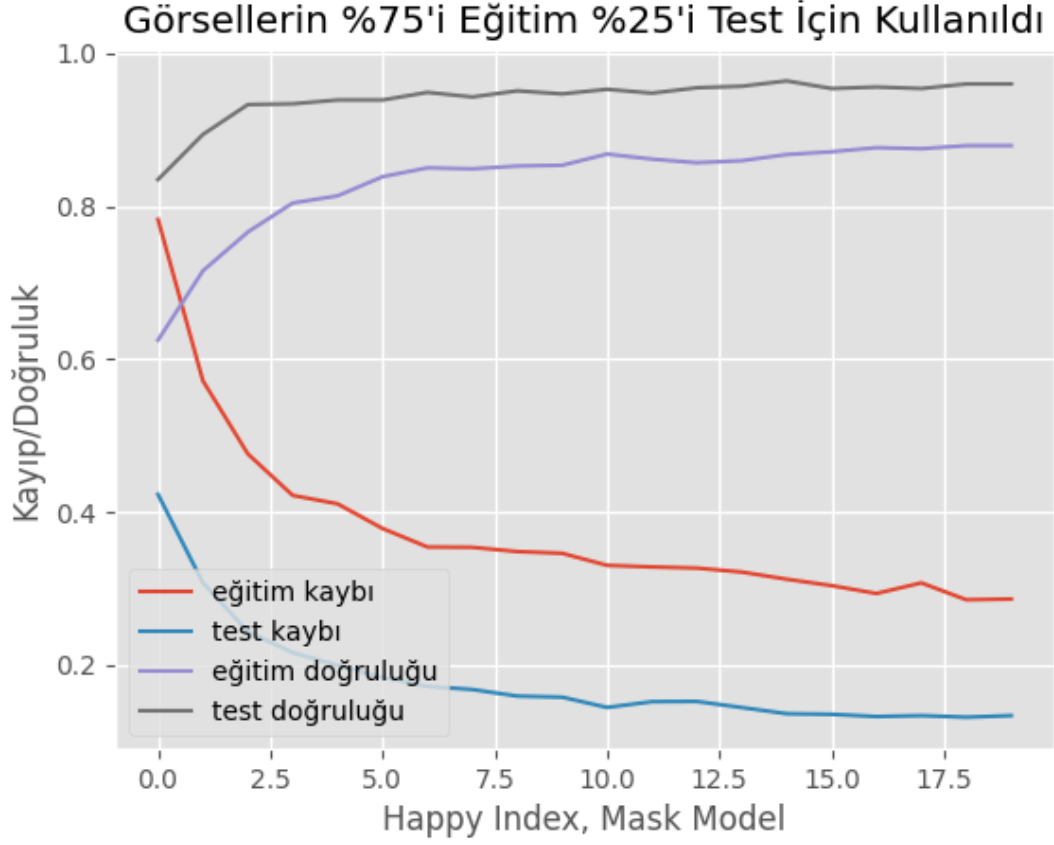
Toplam 4 adet CNN ağı kullandık. Bunlardan üçü için TensorFlow.js tabanlı FaceApi kütüphanesinin hazır modellerini kullandık (bkz. 5). Bu modelleri bir görseldeki insan yüzlerini tespit etmek; mutluluk oranını, yaşı ve cinsiyeti tahmin etmek için kullandık. Medikal yüz maskesinin ne kadar doğrulukta takıldığını tahmin eden modeli ise kendimiz geliştirdik.

Medikal Yüz Maskesi Algılama Modeli

Medikal maske algılamak için kullanacağımız modeli, Python programlama dili ile TensorFlow kütüphanesini kullanarak geliştirdik. MobileNet evrişimli modelinde öğrenme aktarımı (transfer learning) tekniği kullanarak geliştirdiğimiz model toplam 10 katmandan oluşuyor.

```
93 # Medikal Yüz Maskesi Algılama Modeli, "Transfer Öğrenme" yöntemi kullanılmıştır.
94 # CNN, Evrişimli Sinir Ağı katmanı girdi olarak görsel alır.
95 # Görüntü boyu ve genişliği 128px. SHAPE = (128,128,3)
96 MaskeModel = keras.Sequential(name="happy_index")
97 inputLayer = layers.Input(shape=SHAPE)
98 baseLayer = MobileNetV2(weights="imagenet", include_top=False, input_shape=SHAPE, input_tensor=inputLayer)
99 for layer in baseLayer.layers: layer.trainable = False
100 MaskeModel.add(baseLayer)
101
102 # Pooling (havuzlama) katmanı boyutsallığı azaltma amacıdadır.
103 # Gereksiz özellikler yok sayılarak daha önemli özelliklere odaklanılır.
104 # Bu sayede hem gereken işlem gücü azalır.
105 MaskeModel.add(layers.MaxPooling2D(pool_size=(4, 4), name="pooling_layer"))
106 MaskeModel.add(layers.BatchNormalization(axis=-1, momentum=0.8))
107
108 # Klasik Sinir Ağı için verileri hazırlar.
109 MaskeModel.add(layers.Flatten( name="flatten_layer"))
110 MaskeModel.add(layers.Dropout(0.4))
111
112 # 128 birimden oluşan klasik katman, relu aktivasyon fonksiyonu kullanıldı.
113 MaskeModel.add(layers.Dense(160, use_bias=False, name="dense_layer"))
114 MaskeModel.add(layers.BatchNormalization(axis=-1, momentum=0.8))
115 MaskeModel.add(layers.ReLU())
116
117 # Eğitim işlemi sırasında katmandan rasgele bir dizi veri üretir.
118 # Aşırı öğrenmeyi(overfitting) engellemek için kullanır.
119 # 0-1 arasında değer alır, %50(0.5) oranında rasgele veri üret anlamına gelir.
120 MaskeModel.add(layers.Dropout(0.4, name="dropout_layer"))
121
122 # Çıktı katmanı 3 adet çıktımız olacak, maske_dogru, maske_yanlis, maske_yok.
123 MaskeModel.add(layers.Dense(NUM_CLASSES, activation="softmax", name="output_layer"))
124 MaskeModel.compile(optimizer=OPTIMIZER, loss="categorical_crossentropy", metrics=['categorical_accuracy'])
```

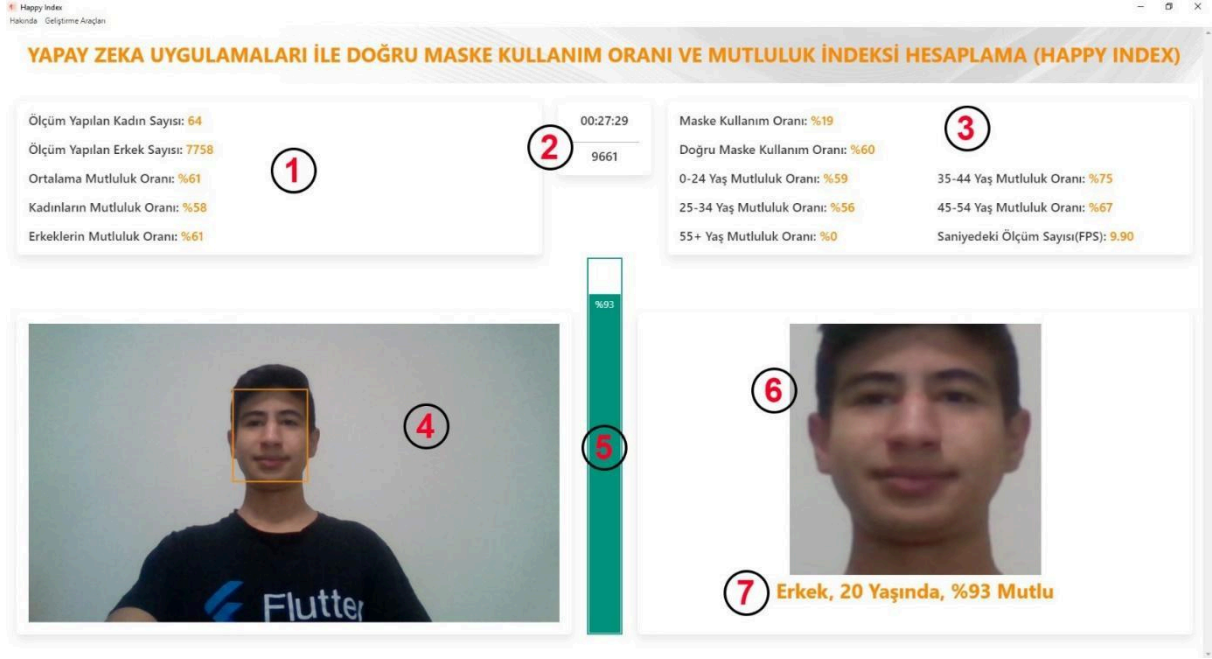

İnsan yüzlerinde medikal maske takılı olup olmadığını tespit etmek için kendi eğittiğimiz modeli kullandık. Maskeli ve maskesiz olarak CC0 (Creative Commons Zero) lisanslı 24000 görsel kullanarak eğittik. Eğitim sonucunda %96 doğrulama başarımlarını elde ettik.



Eğittiğimiz modeli, test etmek için ne eğitim ne de doğrulama sürecinde kullanılmamış 3.000 adet görselle test ettik. Test sonucunda ortalama %97 test başarımlarını elde ettik.

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| maske dogru | 0.95 | 0.97 | 0.96 | 1000 |
| maske yanlış | 0.97 | 0.95 | 0.96 | 1000 |
| maske yok | 1.00 | 1.00 | 1.00 | 1000 |
| accuracy | | | 0.97 | 3000 |
| macro avg | 0.97 | 0.97 | 0.97 | 3000 |
| weighted avg | 0.97 | 0.97 | 0.97 | 3000 |

Uygulamanın Arayüzü



Uygulamada toplam Şekil 9’da gösterilen 7 alan bulunmaktadır, bunlar;

1. Mutluluk oranının cinsiyete göre dağılımı.
 - i) “Ölçüm Yapılan Kadın Sayısı”
 - ii) “Ölçüm Yapılan Erkek Sayısı”
 - iii) “Ortalama Mutluluk Oranı”
 - iv) “Kadınların Mutluluk Oranı”
 - v) “Erkeklerin Mutluluk Oranı”
2. Uygulamanın çalıştığı toplam süre ve ölçüm yapılan toplam insan yüzü sayısı.
3. Maske kullanım oranları ve mutluluk oranının yaşa göre dağılımı.
 - i) Maske kullanım oranı
Örneğin %20 gibi bir oran, kameradan algılanan yüzlerin %20’sinde maske bulunduğu %80’inde ise maske bulunmadığı anlamına gelir. Hiç maske takmamış bir kişi algılanırsa. Maske kullanım oranı düşüyor fakat doğru maske kullanım oranı değişmiyor.
 - ii) Doğru maske kullanım oranı
%60 gibi bir oran, maske takan insanların yüzde kaç oranla doğru maske kullandığını ifade etmektedir. Maske takarken burnunu açıkta bırakmış bir kişi algılanırsa maske kullanım oranı artıyor fakat doğru maske kullanım oranı azalıyor.
 - iii) “0-24 Yaş” arası mutluluk oranı
 - iv) “25-34 Yaş” arası mutluluk oranı
 - v) “35-44 Yaş” arası mutluluk oranı
 - vi) “45-54 Yaş” arası mutluluk oranı
 - vii) “55+ Yaş” arası mutluluk oranı
 - viii) Saniyedeki ölçüm sayısını (FPS)
4. Kameradan alınan canlı video akışı.
5. Anlık hesaplanan mutluluk oranı ve doğru maske kullanım oranı.

%100 ile %0 arasında değişen değerlerini dikey bir barda göstermekteyiz. Mutluluk oranı değiştikçe barın dolum oranı yüzdesel olarak değişmektedir. %100 çok mutlu, %50 normal, %0 çok üzgün (mutsuz) anlamına gelmektedir. 100-50 aralığındaki diğer değerlerin anlamı çok mutlu ve normal arasında oransal olarak değişmektedir. Örneğin, %75 değeri orta mutlu anlamına gelir. 50-0 aralığındaki değerler ise normal mutlu ve üzgün (mutsuz) arasında oransal olarak değişmektedir. Örneğin, %25 yarı üzgün anlamına gelmektedir.

Yüz maskesi varsa bu alan mavi yoksa yeşil renkte gösterilmektedir. Yüz maskesi tespit edildiğinde bu alan anlık doğru maske kullanım oranını göstermek için de kullanılmıştır. %100 maskenin doğru kullanıldığı anlamına gelir. Burun biraz açıksa bu oran %70-80 aralığına, çok açıksa %40-50 aralığına, ağıza kadar acıksa %0-%10 aralığına düşmektedir.

6. Yakalanan insan yüzü akışı.

Saniyede 10 ölçüm yapıyoruz ve en fazla 10 insan yüzü yakalayabiliyoruz. Bu yakalanan yüzleri bu ekranda 10 FPS (Saniyedeki Kare Sayısı) akış olarak gösteriyoruz.

7. En son hesaplanan cinsiyet, yaş, mutluluk ve doğru maske kullanım oranı.

Burada insan yüzünde ölçüm yapılan verilerin 7. Bölgenin altına anlık olarak verileri “Cinsiyet, Yaş, Mutluluk Oranı” veya “Doğru Maske Kullanım Oranı” verisini yazdırıyoruz.

Yazılımın Geliştirilmesi

Bu kısımda sadece arka planda çalışan Javascript kodlarını göstereceğiz arayüzü geliştirilmesi için standart HTML ve CSS işaretleme dilleri kullanılmıştır.

YZ Modellerinin Yüklenmesi

Bu kod kesiminde kullanılan 4 CNN ağını yüklüyoruz. “maskeTahminiYap” fonksiyonu girdi olarak algılanan insan yüzünü alır çıktı olarak ise ne kadar doğrulukta maske kullanıldığını verir.

```
var MaskeModel = null;
async function LoadMaskModel() {
    await tf.setBackend('webgl');
    MaskeModel = await tf.loadLayersModel('ai_mask_model/model.json');
}

function maskeTahminiYap(canvas_el) {
    tf.engine().startScope();
    let inputTensor = tf.browser.fromPixels(canvas_el).toFloat();
    const offset = tf.scalar(127.5);
    inputTensor = inputTensor.sub(offset).div(offset);
    inputTensor = inputTensor.reshape([1, FACE_CANVAS_SIZE, FACE_CANVAS_SIZE, 3]);
    const prediction = MaskeModel.predict(inputTensor).dataSync();
    const data = Array.from(prediction)
    tf.engine().endScope();
    return data;
}

Promise.all
(
    [
        faceapi.nets.ssdMobilenetv1.loadFromUri('ai_tensorflow_models'),
        faceapi.nets.faceExpressionNet.loadFromUri('ai_tensorflow_models'),
        faceapi.nets.ageGenderNet.loadFromUri('ai_tensorflow_models'),
        LoadMaskModel()
    ]
).then(startVideo);
```

YZ Modellerinin Çalıştırılması

Bu kod kesiminde ilk satır video akışından insan yüzü tespit etmeye çalışıyor. Sonrasında tespit edilen yüz yoksa hiçbir şey yapmadan kod kesiminden çıkıyoruz. Eğer yüz tespit edilmiş ise yüz resmini alarak doğru maske takma oranını buluyoruz. Bu oran sıfır değilse maske takılmıştır bu yüzden ilgili verileri kaydederek arayüzü güncelliyoruz. Sonrasında kod akışını durduruyoruz.

Doğru maske takma oranı sıfır ise yani maske takılmamış ise cinsiyet, yaş ve mutluluk oranı tahminlerini yapıyoruz. Bu kod kesiminde sonu “_hort” ile biten “maske_hort”, “cinsiyet_hort”, “yas_hort” ve “mutluluk_hort” fonksiyonları bulunmaktadır. Bu fonksiyonlar ilgili son üç değerin hareketli ortalamasını almaktadır.

Yapay zekâ modelleri doğası gereği tahminde bulunmaktadır. Ve bu tahminler yanlış olabilmektedir. Örneğin güldüğümüzde mutluluk oranı artmakta fakat yüzdeki kırısklık da artığından yaşı büyük tahmin etmekte. Saniyede 10 ölçüm alıyoruz, son tahmin edilen 3 değerin hareketli ortalamasını alarak bazı durumlarda anlık gerçekleşen yanlış ölçümleri eleterek gerçeğe daha yakın değerler üretebildik.

```
var detection = await faceapi.detectSingleFace(video_el, DETECTOR_OPTIONS);
// eğer hiç bir insan yüzü algılanmamışsa hiç bişey yapma
if (!detection) return;

const resized_detection = faceapi.resizeResults(detection, DisplaySize);
const resized_detection_box = resized_detection.detection.box;

const tempCanvas = await yuzResminiAl(video_el, resized_detection_box);
FaceCanvasContext.beginPath();
FaceCanvasContext.drawImage(tempCanvas, 0, 0, FACE_CANVAS_SIZE, FACE_CANVAS_SIZE);

const maskeTahmini = maskeTahminiYap(face_canvas_el);
let dogruMaskeTakmaOrani = dogruMaskeTakmaOraniHesapla(maskeTahmini);

//maske takılı hesaplama yap
if (dogruMaskeTakmaOrani != 0) {
  dogruMaskeTakmaOrani = maske_hort(dogruMaskeTakmaOrani);
  ToplamMaskeliYuzSayisi++;
  DogruMaskeKullanımToplami += dogruMaskeTakmaOrani;
  updateMaskGUI(dogruMaskeTakmaOrani);
  return;
}

ToplamMaskesizYuzSayisi++;
detection = await faceapi.detectSingleFace(video_el, DETECTOR_OPTIONS).withFaceExpressions().withAgeAndGender();

//gender, tahmin edilen cinsiyeti alıyoruz
const cinsiyet = cinsiyet_hort(detection.gender);

//age, tahmin edilen yaşı alıyoruz
const yas = yas_hort(detection.age);

// happiness rate, mutluluk oranini hesaplıyoruz
let mutlulukOrani = mutlulukOraniHesapla(detection.expressions);
mutlulukOrani = mutluluk_hort(mutlulukOrani);
GenelMutlulukToplami += mutlulukOrani;
```

Ortalama Değerlerin Hesaplanması

Mutluluk oranı ve doğru maske takma oranı hesaplarında YZ modellerinde elde edilen değerler 0,0 ile 1,0 arasında ondalık sayılardı. Bu değerlerde dönüşümler yaparak yüzdelik oran hesapladık. Örneğin kendi geliştirdiğimiz model görüntüyü 3 farklı kategoriye göre maske doğru, maske yanlış ve maske yok diye sınıflandırıyor. Bu sınıflandırma sonucunda 0,0 ile 1,0 aralığında 3 farklı oransal değer üretiyor. Bu değerleri 100 ile çarparak yüzdelik oran buluyoruz daha sonra. “maske_yok” değeri diğer iki değerden büyükse maske olmadığı anlamına gelen 0 değerini döndürüyoruz. Büyük değilse 0 maske yok, 50 ortalama doğru maske takmış, 100 tam doğru maske takmış anlamlarına gelen oransal tek bir değer döndürüyoruz. Örneğin burun biraz açıksa bu değer %70 gibi, çok açıksa %50 gibi, maske takılı fakat burunla beraber ağızda açıksa bu değer %10 gibi oluyor. Bu oransal değere doğru maske kullanma oranı diyoruz.

Aynı şekilde duygu durumunu analiz etmek için kullandığımız model üzgün, normal ve mutlu anlamlarına gelen 3 farklı değer üretiyor. Bu değerleri yine oranlayarak tek bir değer üretiyoruz. Bu değere de mutluluk oranı diyoruz. Örneğin %10 ise üzgün %50 ise normal, %80 gibi bir değerse mutlu anlamına gelmekte. Test yaparak bu değerlerin kalibrasyonunu kendimiz yaptık.

```
function dogruMaskeTakmaOraniHesapla(data) {
    let sonuc = 50;

    // doğru maske takılı olama ihtimali
    const maske_dogru = data[0]*100;
    // yanlış maske takılı olama ihtimali
    const maske_yanlis = data[1] * 100;
    // maske takılı olamama ihtimali
    const maske_yok = data[2]*100;

    if (maske_yok > maske_dogru && maske_yok > maske_yanlis) {
        // maske yok
        return 0;
    }
    else {
        if (maske_dogru > maske_yanlis) {
            sonuc += maske_dogru/2;
        }
        else {
            sonuc -= maske_yanlis/2;
        }

        // 0 hiç maske olmadığı anlamına geliyor
        if (sonuc < 1) sonuc = 1;
    }
    return sonuc;
}

function mutlulukOraniHesapla(data) {
    let sonuc = 50;
    const uzgun = data.sad * 100;
    const dogal = data.neutral * 100;
    const mutlu = data.happy * 100;

    if (dogal>uzgun && dogal>mutlu) {
        sonuc = 40;
        sonuc = sonuc + dogal/10;
    }

    if(mutlu>uzgun){
        sonuc += mutlu/2;
    }
    else if(uzgun>mutlu){
        sonuc -= uzgun/2;
    }
    }
    else {
        if(mutlu>uzgun){
            sonuc += mutlu/2;
        }
        else if(uzgun>mutlu){
            sonuc -= uzgun/2;
        }
    }
    return sonuc;
}
```

Sonuç ve Tartışma

Python programlama dili ile geliştirdiğimiz evrişimli sinir ağını eğiterek %96 test başarımları elde ettik. Yapay zekâ modellerinin tasarlanmasında ve eğitiminde yaygın olarak Python dili kullanıldığından bu dili tercih ettik. Geliştirdiğimiz modelin daha kısa sürede daha az işlem gücüyle çalışabilmesi için geliştirmeler yaptık.

Yapay zekâ modelimizi eğittikten sonra ikinci aşamada bu modeli kullanarak ölçüm yapacak masaüstü uygulamasını geliştirmemiz gerekiyordu. Yazılımı web teknolojilerini kullanarak yani HTML ve Javascript kullanarak geliştirdik ve Elektronik uygulamasıyla masaüstü uygulamasına çevirdik. İlk zamanlar saniyede bir insan yüzünden ölçüm alabiliyorduk. Bu durumu hızlandırmak etkili bir ölçüm yapabilmek için yazılım içindeki fonksiyonların birbirini bekletmemesi ayrıca her fonksiyonun 100 milisaniye gibi bir sürede çalışması gerekiyordu. Geliştirdiğimiz yazılımı sürekli iyileştirerek saniyede 10 ölçüm yapmayı başardık.

Bunun için kendi ettiğimiz modeli daha az sayıda birimden oluşan yapay sinir ağı bulunacak şekilde yeniden tasarladık ve aynı sonucu almak için daha uzun süre eğittik. Birçok denememizde başarısız olduk bazı durumlarda model düzgün çalışıyor fakat bu modeli kullanarak ölçüm almamız 1 saniye kadar sürebiliyordu. Bazı durumlarda ise model hızlı çalışıyor fakat yanlış ölçüm alıyorduk. Modeli yeniden tasarlayıp tekrar tekrar eğittik. İnsan yüzünü algıladıktan sonra diğer ölçümleri paralel bir şekilde yapabiliirdik. Uzun bir iyileştirme sürecinden sonra 100 milisaniyede bir ölçüm yapmayı başardık. İlk zamanlar ölçüm alabilmek için kameranın önünde hareketsiz durmak gerekirken son aşamada kameranın önünden durmadan yürüyerek geçen insanlardan ölçüm almayı başardık.

Çeşitli yaş gruplarında yazılımı test ettik oldukça başarılı yaş, cinsiyet, mutluluk oranı ve doğru maske takma oranı tahminleri elde ettik. Siyah, beyaz ve mavi olmak üzere 3 farklı maske çeşidi ile testler gerçekleştirdik, bu testlerde de başarılı sonuçlar elde ettik. En çok zorlandığımız testler düşük mutluluk oranı hesaplama testleriydi. Test yapılan insanlar gülmekte hiç zorluk çekmiyor fakat üzülmesini istediğimizde bu durumu çok kolay gerçekleştiremiyorlardı. %0 mutluluk oranı ölçümünü test etmek için test yapılan insanın yüz ifadesinin ağlamaya yakın, kafasının biraz eğik ve gözlerinin kısık olması gerekiyor.

Öneriler

İnsan yüzleri kullanarak eğitilen yapay zekâ modellerini kullanan yazılımların başarısında, eğitimde kullanılan görsellerin yazılımın kullanıldığı toplumun kültürüyle uyumlu olması oldukça etkili olabilir. Doğru yüz maskesi takma oranını hesaplayan modelimizi eğitmek için kullandığımız veri setinde yok denecek kadar az başörtülü kadın görseli vardı. Test yaparken kişide baş örtüsü varsa ölçüm hassasiyetinin düştüğünü gözlemledik.

Bir yazılımda farklı işler yapan birden çok yapay sinir ağı kullanacaksanız oldukça fazla performans sorunları yaşanacaktır. Yazılım verimli çalışabilmesi için kendi yapay zekâ modelinizin oldukça az birimden oluşacak şekilde tasarlamanız ve etkili bir sonuç alabilmek için tekrar tekrar uzun süre eğitmeniz gerekecektir. Bu aşamada işlem gücü yüksek bir bilgisayarınızın bulunması çalışma sürenizi oldukça kısaltabilir.

Halihazırda aşına olmamızdan ve kodlamanın kolay olmasından dolayı masaüstü uygulamasını web teknolojilerini (HTML ve Javascript) kullanarak geliştirmeyi tercih ettik. Bundan dolayı birçok performans sorunu yaşadık. İyileştirmek için çok uğraştık. Öğrenmesi kolay olması sebebiyle bu teknolojileri tercih ettik fakat C++ gibi daha verimli yazılım dilleri tercih edilirse geliştirilen yazılımda daha verimli çalışabilir.

Kaynaklar

1. **NTV Haber, (2018).** Çalışanların mutluluğunu yapay zekâ belirleyecek.
<https://www.ntv.com.tr/video/teknoloji/calisanlarin-mutlulugunu-yapay-zeka-belirleyecek,IcsCgy7d60ukCVn3XhAT9A>
Erişim Tarihi: 05.2020
2. **MaskedFace-Net (tarih yok).** Eğitiminde kullanılan görseller.
<https://github.com/cabani/MaskedFace-Net>
Erişim Tarihi: 10.2020
3. **TensorFlow (tarih yok).** Yazılımda kullanılan makine öğrenmesi kütüphanesi.
<https://www.tensorflow.org/about>
Erişim Tarihi: 09.2020
4. **Doğan, K. ve Özkol İ. (2018).** Bilgi Merkezlerinde Görüntü Tanıma ve Duygu Analizi.
https://www.researchgate.net/publication/326352503_Bilgi_Merkezlerinde_Goruntu_Tanima_ve_Duygu_Analizi_-_Image_Recognition_and_Emotion_Analysis_in_Information_Centers
Erişim Tarihi: 11.2020
5. **FaceApi.js (tarih yok).** Yazılımda kullanılan modellerin bulunduğu kütüphane.
<https://github.com/justadudewhohacks/face-api.js/>
Erişim Tarihi: 11.2020
6. **Electron (tarih yok).** Yazılımda kullanılan kütüphane.
<https://www.electronjs.org/>
Erişim Tarihi: 12.2020
7. **TÜİK, Yaşam Memnuniyeti Araştırması, (2020).**
<https://data.tuik.gov.tr/Bulten/Index?p=Yasam-Memnuniyeti-Arastirmasi-2020-37209>
Erişim Tarihi: 02.2021
8. **McLaren S. (2019).** Here's How IBM Predicts 95% of Its Turnover Using Data.
<https://business.linkedin.com/talent-solutions/blog/artificial-intelligence/2019/IBM-predicts-95-percent-of-turnover-using-AI-and-data>
Erişim Tarihi: 09.2020
9. **Aydın, H. İ. (2020).** Mutluluk ve ekonomi: Yaşam memnuniyeti araştırması üzerinden bazı dikkatler.
<https://dergipark.org.tr/en/download/article-file/1127420>
Erişim Tarihi: 02.2020
10. **CC0 (tarih yok).**
<https://creativecommons.org/share-your-work/public-domain/cc0/>
Erişim Tarihi: 10.2020