Bir Görselin Gerçekliğini Tespit Etme

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Rıfat Oktar

ORCID 0000-0000-0000-0000

Proje Danışmanı: Dr. Ögr. Üyesi Emre Şatır

Haziran 2024

Bir Görselin Gerçekliğini Tespit Etme

Öz

Çağımız da gelişen yapay zeka projeleri, ürettikleri görsellerin gerçek olabilecek kadar kusursuz olmasından ötürü, kullanıcıların gerçek veya sanal ortam da dolaşıma sokulan görsellerin gerçekliğini sorgulamasına yol açmaktadır. Böyle bir durumda kullanıcı, elinde bulundurduğu bir görselin bir yapay zeka tarafından mı yoksa bir insan tarafından mı üretildiğini tespit edebilmesi gerek telif hakları gerekse de gerçekliğinin tasdik edilebilmesiyle ilgili kullanıcı için büyük önem arz etmektedir.

Dolayısıyla bu proje bir kullanıcının elinde bulundurduğu bir görselin bir yapay zeka tarafından mı yoksa bir insan tarafından mı oluşturulduğunu tespit edebilmesi için geliştirilmiştir. Bu işlem de geliştiricilerin projelerin de sıklıkla kullandığı Python programlama dili ve kütüphaneleri, projeyle ilgili olarak üretilmiş görsellerden oluşan veri setleri, öğrenme modeli ve değerlendirme ölçütü kullanılarak gerçekleştirilmiştir.

Anahtar Sözcükler: Yapay zeka, öğrenme modelleri, değerlendirme ölçütleri, insan, Python programlama

Determining The Authenticity Of An Image

Abstract

In our era, the evolving artificial intelligence projects, due to their images being flawlessly realistic, raise questions about the authenticity of images circulated in real or virtual environments. In such a scenario, it is crucial for users to determine whether an image they possess was created by artificial intelligence or by a human, both for copyright purposes and to verify its authenticity.

Therefore, this project is developed for a user to be able to determine whether an image they possess was created by artificial intelligence or by a human. This process is carried out using the Python programming language and libraries commonly used by developers in projects, datasets consisting of images related to the project, a learning model, and evaluation criteria.

Keywords: Artificial intelligence, learning models, evaluation criterias, human, Python programming

Bu proje çalışması annem Rübeyda Oktar' a ve ablam Lülüfer Oktar' a adanmıştır.

İçindekiler

Şekiller Listesi

Kısaltmalar Listesi

NYP	Nesne Yönelimli Programlama
VSC	Visual Studio Code
IDE	Integrated Development Environment
YİDH	Yerel İkili Desen Histogramları
HDA	Hata Düzeyi Analizi
ORCID	Open Researcher and Contributor ID
AI	Artifical Intelligance
SDXL	Stable Diffusion XL

Giriş

İlk olarak Alan Turing (1950)' in "Makineler Düşünebilir Mi?" sorusuyla ortaya koyduğu yapay zeka kavramı, o zamanlar için gerçekliğe aykırı bir düşünce olmasının aksine son 10 yıl içerisin de gerçekleşen teknolojik gelişmeler sayesinde artık hayal olmaktan çıkmış ve insanların sıklıkla kullandığı araçlar halini almışlardır.

Yapay zeka, her geçen gün kendini yenileyen ve geliştiren bir alan olmayı sürdürmektedir. Günlük hayatta pek çok sorunu kolay ve hızlı bir şekilde çözebilen yapay zekalar, insanların sıklıkla etkileşime geçtiği ortamlar oluşturmuştur. Gelişen bu ortamlar yanında bazı sorunları da getirmiştir. Örneğin, sanal ortamlarda üretilen görsellerin ne kadarının gerçek ne kadarının ise sahte olduğunun bir kullanıcı tarafından tespitinin kolay ve hızlı bir şekilde yapılamıyor olmasıdır. Bu nedenle, yapay zekalar sıkı bir şekilde denetime ve takibe ihtiyaç duymaktadırlar.

Örnekte de belirtilen bu sorunun ışığın da üretilmesi gereken en gerçekçi çözüm yönteminin ne olması gerektiği konusun da araştırmalar yapılmış ve bazı gelişmeler gerçekleşmiştir. Bahsi geçen bu gelişmelerden biri de bir yapay zeka projesinin geliştirilip bir tahmin yürütülerek ilgili görselin kategorize edilmeye çalışılmasıdır.

Yürütülecek olan bu tahmin sayesinde kullanıcı, ilgili görselin bir yapay zeka tarafından mı yoksa bir insan tarafından mı oluşturulduğunu öğrenebilecektir. Böylelikle ilerleyen zamanlar da gerek ilgili görsel üzerindeki telif hakları korunmuş gerekse de sahtekarlığın önüne geçilmiş olunacaktır.

Ancak bu türden denetimlerin hem maliyetli oluşu hem de bazı özel bilgilerin geri dönülemez bir şekilde 3. taraf kurum veya kuruluşlara aktarılmasından ötürü, pek çok platform güvenilir bir şekilde çalışmamaktadır.

Bu projeyle birlikte, kullanıcının hem tespit maliyetinin önüne geçmek hem de ilgili görselin sanal ortama yüklenmesinin önüne geçerek ilerleyen zamanlarda oluşabilecek veri sızıntılarına ve siber saldırılara karşı önlem alınabilecektir.

Yapay Zeka Nedir?

Yapay zeka en temel anlamda, bir insanın yapabileceği bir işin daha hızlı ve daha stabil bir şekilde yapılabilmesine olanak tanır. Dolayısıyla, yapay zeka çağın getirmiş olduğu ihtiyaçlardan biri olan hıza daha kolay bir şekilde uyum sağlayabilmektedir. Çünkü verilecek olan bir komutun yerine getirilmesi birkaç saniyenin ötesine gitmemektedir. Ayrıca alınan cevabın tatminkarlığı da yüksektir.

Teknolojinin sunmuş olduğu bu hizmet her ne kadar pek çok olumsuz yoruma maruz kalsa da aslında durumun bu şekilde olmadığı gayet ortadadır.

> *"Robotlar insanların yerini almayacak, işlerini daha insancıl hâle getirecekler. Zor, aşağılayıcı, talepkâr, tehlikeli, sıkıcı - bunlar robotların alacağı işler.* (Sabine Hauert, Robot uzmanı)"

Sözünden de anlaşılacağı üzere yapay zeka insanlığın uğraştığı sıkıcı ve zaman alıcı işlerini devralacaktır. Ancak, günün sonunda bu teknolojiyi üreten insanoğlu olduğundan ve kendinden bir parçasının da bu tür yapay zekalara geçmesinden ötürü, en nihayetin de yapay zekalar insanoğlunun hem iyi yönlerini hem de kötü yönlerini yansıtmaktadırlar. John Hagel' in de dediği gibi;

> *"Eğer doğru yaparsak, benzersiz insan kabiliyetlerimize dokunan ve insanlığımızı eski haline getiren bir çalışma biçimi geliştirebiliriz. Nihai paradoks, bu teknolojinin insanlığımızı geri kazanmak için ihtiyaç duyduğumuz güçlü bir katalizör olabileceğidir.* (John Hagel, Yazar)"

Dolayısıyla, bir kullanıcının, bir yapay zeka dan ne almak istediği tamamen o kullanıcıya bağlıdır.

Ancak günün sonunda yapay zeka da bir teknolojik gelişme olmasından ötürü, bazı zamanlarda yanılma payının da olabileceğini hesaba katmak mühimdir. Dolayısıyla bir yapay zekadan alınacak olan bir cevabın her zaman için "doğru budur" yaklaşımına girilmesi doğru olmayacaktır. Bu nedenle, kendini sürekli geliştiren bir yapay zeka, her zaman için daha güvenilir hale gelebileceğinden ötürü bu tür yapay zekaları kullanmak daha akıllıca olacaktır. Aynı zaman da kendini sürekli geliştiren bir yapay zeka eğer ki yanlış ellerin eline geçerse gerçekte olmayan ancak gerçekmiş gibi algılanabilecek sonuçlar ortaya koyabilir. Bu durum da kullanıcının gerçekliği sorgulamasına ve hatta artık internet ortamında hiçbir şeye inanmaz hale gelmesine yol açabilecektir. Dolayısıyla, yapay zekanın her zaman için doğru eller tarafından sürekli denetlenmesi ve hakikatin sürekli bir şekilde vurgulanması önemlidir.

Bu nedenle yapay zeka kullanmak isteyen bir kullanıcının iyi bir şekilde teknoloji okur-yazarı olması son derece önemlidir. Çünkü elde edilecek olan bilgi eğer ki doğru bir bilgi değilse, bu durum hem yapay zekanın daha gelişemez hale gelmesine hem de kullanıcının bu teknolojiden giderek uzaklaşmasına yol açabilir. Dolayısıyla, burada hem yapay zekayı geliştiren geliştiriciye hem de bu yapay zekayı günlük hayatta kullanan kullanıcıya büyük sorumluluklar düşmektedir.

Ayrıca bahsedilen bu hız bazı çevreler için olumlu iken bazı çevreleri içinse olumsuzdur. Örneğin, karmaşık matematiksel bir işlemi 2 hafta da yapabilen bir matematikçi ile aynı işlemi sadece 30 saniyede yapabilen bir yapay zekanın olduğu bir ortamda tahmin edildiği üzere matematikçiye ihtiyaç kalmayacaktır. Bu durumda belli bir kesim daha da çok kazanırken belli bir kesimin de daha da çok kaybedeceği bir ortam oluşturur. Profesör Doktor Ng' nin de dediği gibi;

"Yapay zekânın yalnızlara sohbet ve rahatlık sağladığını gördük; yapay zekânın ırk ayrımcılığıyla uğraştığını da gördük. Ancak yapay zekânın

kısa vadede bireylere vereceği en büyük zarar işten çıkarmalar olacak çünkü yapay zekâ ile otomatikleştirebileceğimiz iş miktarı eskisinden çok daha büyük. Liderler olarak, her bireyin başarılı olma fırsatına sahip olduğu bir dünya inşa ettiğimizden emin olmak hepimizin görevidir. (Andrew Ng, Profesör)"

Temel anlamda yapay zekanın bazı hem iyi yönlerine hem de kötü yönlerine bakıldığına göre artık gerçekleştirilen bu projede neden yapay zeka kullanıldığı tartışılabilmektedir.

Projenin temelin de yapay zekanın kullanılmasının nedeni, üretilen bir yapay zekanın hem hızlı bir şekilde işlem yapabilmesi hem de bir kullanıcıya göre önceden oluşturulmuş bir görselin bir insan tarafından mı yoksa bir yapay zeka tarafından mı oluşturulduğunu daha kolay ayırt edebilmesindendir.

Böyle bir projenin oluşturulmasının nedeni ise, önceden elde edilmiş bir görselin birebir kopyasının bir yapay zeka ile oluşturulmasıyla sanatçının kendinde saklı tutabileceği telif haklarının artık kullanılamaz hale gelebileceğinden, aynı zamanda bu durum hem yapay zekaları daha kötü yaftalarla suçlayabilecek hem de bir sanatçının artık rekabet edemez hale gelebilecek olmasının önüne geçebilmek içindir.

Bu tür projelerle günün sonunda aslında yapay zekaların iyi bir şekilde eğitilip, sınırlandırılıp ve denetlendiği takdirde insanoğlunun düşmanı değil bilakis çok büyük iş ortağı ve dostu olabileceğini kanıtlayabilmektir.

Bir yapay zeka projesi geliştirilirken kodlama aşamasında yapılması gereken birkaç adım vardır. Bunlar;

2.1 Yapay Zekanın Geliştirilmesi İçin Gereken Veri Setinin Oluşturulması

Bir yapay zeka projesi geliştirilmek isteniyorsa, öncelikle bu projesinin eğitilmesi gerekir. Bu eğitim de söz konusu yapay zeka projesinin ne amaçla kullanılacaksa o

konuyla alakalı ister görsel materyaller ister metinsel veriler olsun bu dokümanlar eğitim için kullanılmak zorundadır.

Eğitim için kullanılan söz konusu veri setinin doğruluğu, basitliği veya karmaşıklığı, miktarı çok önemlidir. Çünkü, kullanılacak olan veri seti ne kadar "kaliteli" olursa eğitim süreci de o kadar kaliteli olacak ve daha kesin sonuçlar elde edilebilinecektir.

2.2 Oluşturulan Veri Setinin Proje İçerisinde Yapılacak İşlemlere Göre İşlenmesi

Bir yapay zeka projesi oluşturulurken kullanılabilecek makine öğrenmesi yöntemleri vardır. Bu yöntemler kullanılan veri setine ve en nihayetin de istenen sonuca göre değişiklik gösterebilir. Bazı öğrenme yöntemleri;

2.2.1 Denetimli Öğrenme (Supervised Learning)

Modelin eğitimin de bir dizi giriş verisi ve karşılık gelen bir eşleştirilmiş etiketin çıktı verisi kümesiyle eğitilmesi olarak tanımlanmaktadır. Yani, model kullanılan veri seti ile alınan sonuca bakılarak eğitilmesi durumudur.

Bu yöntem yapay zeka projelerinin büyük bir çoğunluğunda kullanılmaktadır. Denetimli öğrenme, temel de 2 sınıfa ayrılır. Bunlar; sınıflandırma ve regresyondur. Sınıflandırma ve regresyon da kendi içlerinde sınıflara ayrılırlar.

2.2.1.1 Sınıflandırma (Classification)

Denetimli öğrenmenin bir kolu olan sınıflandırma, daha çok metinsel bir verinin tasnif edilmesi için kullanılan bir öğrenme yöntemidir. Örneğin sınıflandırma, yoğun olarak e-posta sağlayıcılar da "Naive – Bayes" algoritması kullanılarak spam filtreleme yapılmaktadır. Burada, model eğitilirken kullanılan anahtar kelimeler önemlidir.

Örneğin, araba satışı yapan bir firmanın göndereceği her e-posta spam veya normal olmak zorunda değildir. Burada önemli olan mailin içerdiği kelimelerdir. Bazı

kelimeler her kullanıcı için geçerli olsa da bazı kelimeler sadece ilgili kullanıcı için olabilir veya çok önemli bir e-posta olabilir. Bu nedenle, e-posta sağlayıcısı burada belirleyeceği anahtar kelimeleri doğru seçmek zorundadır.

2.2.1.2 Regresyon (Regression)

Denetimli öğrenmenin bir diğer kolu olan regresyon, kesikli veya sürekli verilerin bulunduğu veri setlerinde kullanılmaktadır. Regresyon, modelin eğitimin de girdi olarak kesikli veya sürekli değerleri kullandığı için metinsel verilerde sınıflandırma kadar doğru sonuçlar ortaya koymayacaktır.

Örneğin, bir hastane de tanısı konan kanser hastalarının sayısal verilerine göre yeni gelen hastalara daha hızlı ve kolay bir şekilde tanı konması sağlanmaktadır. Böyle bir senaryoda en can alıcı nokta kanser veya sağlıklı hastaların sayısıdır.

2.2.1.3 Doğrusal Regresyon (Linear Regression)

Regresyon' un alt kollarından biri olan doğrusal regresyon, bir veya daha fazla kesikli veya sürekli bir verinin kesintisiz bir ölçekten bir değeri tahmin eden öğrenme modelidir.

Örneğin, bir dersin öğrencilerinin bir dersin sınavlarından alacakları notları tahmin edebilmek için önceden alınan notlara, öğrencilerin ilgili ders için ayırdıkları zamana bakılması gerekir. Böylelikle hem ders için çok zaman harcayıp hem de yüksek not alan öğrencilerin diğer sonuçları da daha kolay ve hızlı bir şekilde tahmin edilebilir.

2.2.1.4 Lojistik Regresyon (Logistic Regression)

Lojistik regresyon girdi olarak verilen kesikli veya sürekli verilerin kategorik olarak bir çıktıyı tahmin edebilme yeteneğine sahip bir denetimli öğrenme modelidir.

Örneğin, bir çiftçinin ekeceği buğday ekininin o sene için elde edeceği buğday hasadını önceden tahmin edebilmesidir. Buradaki değişkenler ekine verilen

yağmurun, güneş ışığının miktarlarıdır. Böyle bir senaryoda, her ikisinin de az olması, o sene için hasadın da az olacağı tahminin yürütülmesine yol açacaktır.

2.2.1.5 Destek Vektör Makineleri (Support Vector Machine)

Destek vektör makineleri, girdi olarak verilen bir kesikli veri setinin değerlerini bir düzlem üzerine yerleştirir ve sınıflandırılmak istenen noktadan bir hiperdüzlem çizer. Bu hiper düzleme en yakın olan değerlere 90^o lik doğrular çizer. Çizilen bu doğruların uzunluğuna göre referans olarak belirlenen değer sınıflandırmaya tabi tutulur. Uzunluğu kısa olan doğruya sahip sınıf, ilgili veriyi kendi sınıfına alır. Böylelikle eldeki değer sınıflandırılmış olur.

Örneğin, bir okuldaki etüt sınıflarından birine bir öğrencinin yerleştirilmek istendiği düşünülsün. Böyle bir senaryo da her etüt sınıfındaki öğrencilerin ders notları belirlenir. Daha sonrasın da ise ilgili öğrencinin ders notları belirlenir. Elde edilen değerler karşılaştırılır ve ilgili öğrencinin notlarına en yakın hangi notlar hangi sınıfta varsa o sınıfa ilgili öğrenci yerleştirilir.

2.2.1.6 Sinir Ağları (Neural Networks)

Sinir ağları, diğer denetimli öğrenme modellerine göre daha farklı olarak, eğitim için verilen belli veri girdisini belli bir çıktıların ağırlığının dengelenmesi adına karmaşık matematiksel işlemler gerçekleştiren bir denetimli öğrenme modelidir.

Örneğin, bir lise mezuniyet fotoğrafındaki yüzlerce öğrencinin yüzlerinin tanımlanması sonucu hızlı ve kolay bir şekilde isim etiketlerinin yazılması verilebilmektedir.

2.2.1.7 Karar Ağacı (Decision Tree)

Karar ağacı, modelin eğitimini kullanılan veri setini aldıktan sonra yazılım geliştirmenin temel özelliklerinden biri olan if – else koşulunu uygulayarak sonuç tahminini sağlar.

Örneğin, bir zincir marketin müşterilerini elinde tutmak için uyguladığı bir model düşünülsün. Burada müşteriler zincirin hangi şubesinden hangi ürünü daha çok alıyor veya ilgili ürünü hangi şubede bulamıyor ve ona göre devamlılığını sağlayabiliyorsa bu duruma göre şubelerin denetiminin artması veya ürün gamının arttırılması gibi sonuçlara varılır.

2.2.1.8 K-En Yakın Komşu (K-Near Neighbours)

K-en yakın komşu, hem kesikli ve sürekli veri setleriyle hem de metinsel veri setleriyle çalışabilen bir sınıflandırma modelidir. Bu model bir düzleme yerleştirilmiş olan veriler ile sınıflandırılmak istenen veriyi bir araya getirir. Sonrasın da ilgili veriye en yakın olan verilere 90^o lik doğrular çizer. Bu doğruların da en yakın olan değerin sınıfı artık ilgili verinin de sınıfı halini almış olur.

Örneğin, yeni mezun bir yazılım mühendisinin çalışmak için isteyeceği iş ilanlarının bir listesinin olduğu düşünülsün. Bu listede kendi özelliklerine en yakın olan iş ilanlarına başvuru yapması istenecektir. Böylelikle bir işe girebilmesi hem daha kolay hem de hızlı olacaktır.

2.2.1.9 Rastgele Orman (Random Forest)

Rastgele orman, bir veri sınıflandırma modelidir. Model önceden oluşturulmuş pek çok karar ağacı modellerini bir araya getirir. Sonrasında elde edilen bu ağaçları farklı özelliklere sahip olacak şekilde eğitir. Daha sonrasında ise en doğru değere yakın olan ağacın sonuçlarını sınıflandırma yapmak için kullanır.

Örneğin, iris çiçeğine ait olan 3 ayrı türdeki özelliklerin bir arada olduğu bir veri seti kullanılır. İlgili türün hangisi olduğunun belirlenmesi için rastgele orman modeline tabi tutulur ve sınıflandırılması gerçekleştirilir.

2.2.2 Yarı Denetimli Öğrenme (Semi-Supervised Learning)

Yarı denetimli öğrenme modeli de hem denetimli öğrenme hem de denetimsiz öğrenme modeline yakın bir öğrenme modelidir. Bu model, veri girdisini

gerçekleştiren kullanıcının kendisine belli oranda yardım etmesini ister. Bunu da verilerin bazılarını el yordamıyla kullanıcının etiketlemesini bekler.

Etiketlenen veriler aynı özellikteki verilerle aynı kümelemeye tabi tutulur. Modelin en önemli artısı, çok az miktarda el yordamıyla yapılan etiketleme ile çok büyük veri setlerinin eğitilmesi sağlanır.

2.2.3 Denetimsiz Öğrenme (Unsupervised Learning)

Denetimli öğrenmenin aksine, denetimsiz öğrenme metodun da çıktı verisini herhangi bir etiketleme veya kümeleme işlemine tabi tutmadan algoritma giriş verilerine atanan durumu ifade eder.

Denetimsiz öğrenmenin temelindeki sınıfı kümelemedir.

2.2.3.1 Kümeleme (Clustering)

Denetimsiz öğrenmenin bir alt kolu olan kümeleme, benzer özellikteki veri kümelerini aynı kümeye yerleştirirken farklı özellikteki kümeleri farklı kümelere yerleştirir.

Örneğin, potansiyel bir siber saldırıyı önceden tahmin etmek için sisteme bağlanan farklı ağlardan gelen verileri aynı özellikteki kümelere aktarma işlemi sonucu tahmin yürütülme işlemidir.

2.2.4 Pekiştirmeli Öğrenme (Reinforcement Learning)

Pekiştirmeli öğrenme modeli, temelin de deneme/yanılma ve ödül/ceza süreçlerini kullanarak öğrenme işlemini gerçekleştirir. Model, ilgili sistemin gerçekleştirmek istediği bir eylemi gerçekleştirmek için bazı deneme/yanılma yoluyla elde ettiği bilgilerin ne kadar iyi çalışıp çalışmadığına bakmak için ödül/ceza geri bildirimine başvurmaktadır. Bunun sonucunda elde edilen geri bildirime göre öğrenme süreci tamamlanmış olur.

Örneğin, bir robotun ev içi temizlik yapması isteniyor olsun. Bu robotun çevresinden elde edeceği deneyimlerle eğitimini sağlayıp işini yüksek doğruluk içerisinde gerçekleştirmesidir.

2.2.5 Derin Öğrenme (Deep Learning)

Derin öğrenme, önceden eğitimine başlanmış sinir ağlarının büyük miktarda veriden elde ettiği makine öğrenmesinin bir alt kümesidir. Yani insan beynin bir modelinin oluşturulup bilgisayarların da anlayabileceği bir hale getirilmesidir.

Örneğin, doğa da daha önce keşfedilmemiş bir hayvan türünün resminin foto kapan tarafından alınıp diğer hayvan türleriyle bir ilgisinin olup olmadığının sorgulanabilmesini sağlayabilir.

2.3 Verinin Belli Bir Süreçten Geçirilerek Bazı Hesaplamalara Tabi Tutulması

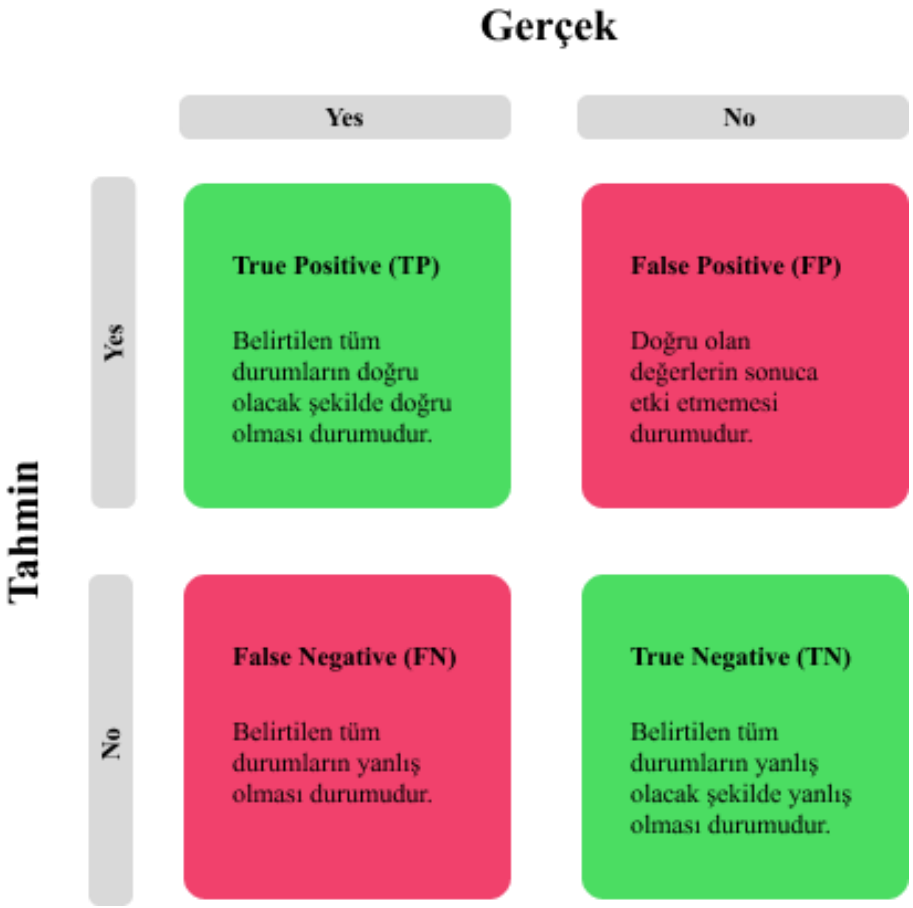
İlgili veri setini ve geliştirilecek yapay zekanın özelliklerine göre öğrenme yöntemi seçilmesinin ardından, elde edilen sonuçların yorumlanması hem geliştirilen yapay zekanın hangi ölçüde doğru çalıştığını hem de ilgili platformdaki performans iyileştirmelerini gözlemlemek için önemlidir.

Dolayısıyla, ilgili projeye uygulanan öğrenme yöntemlerindeki sınıflandırmaların ve veri setinin "Kesinlik, Doğruluk, F-1 Puanı, Karmaşıklık Matrisi ve Duyarlılık" gibi değerlendirme ölçütlerine tabi tutulması önemlidir. Bu projede doğruluk değerlendirme ölçütü kullanılmaktadır.

2.3.1 Karmaşıklık Matrisi (Confusion Matrix)

Karmaşıklık matrisi, geliştirilmiş olan sınıflandırma işleminin ardından yapılan işlemler sonucu elde edilen değerlerin hangi oranda doğru olduğunu hangi oranda yanlış olduğunu tespit edebilmek için oluşturulmuş bir hata tablosudur.

Bu tablo, 2 satır ve 2 sütun olmak üzere 4 adet değere sahiptir. Bunlar “Doğru Pozitif (TP), Doğru Negatif (TN), Yanlış Pozitif (FP), Yanlış Negatif (FN)” şeklindedir. Elde edilen bu veriler ışığında daha sonra anlatılacak olan değerlendirme ölçütlerinin hesaplanması gerçekleştirilmektedir (bakınız Şekil 2.1).



Şekil 2.1: Karmaşıklık matrisi

2.3.2 Doğruluk (Accuracy)

Doğruluk, karmaşıklık matrisinden elde edilen değerler ışığın da hesaplaması yapılan bir değerlendirme ölçütüdür. Bu ölçüt hesaplanırken, doğru olan değerlerin tüm değerlerin oranı dikkate alınır. Sonuç %’ li bir ifade olarak ortaya çıkar.

Yapılan işlem sonucu elde edilen değer %100’ e ne kadar yakınsa o kadar kullanılan öğrenme, sınıflandırma modeli ve veri seti doğrudur yorumu yapılabilmektedir. (bakınız Şekil 2.2)

$$\text{Doğruluk} = \frac{TP + TN}{TP + FP + TN + FN}$$

Şekil 2.2: Doğruluk formülü

2.3.3 Kesinlik (Precision)

Kesinlik, karmaşıklık matrisinden elde edilen değerlerdeki pozitif tahminlerin yüzde kaçının doğru çıktığını hesaplamak için kullanılmaktadır.

Dolayısıyla, kesinliğin hesaplanması için doğru pozitiflere ve tüm pozitif değerlere ihtiyaç duyulmaktadır (bakınız Şekil 2.3).

$$\text{Kesinlik} = \frac{TP}{TP + FP}$$

Şekil 2.3: Kesinlik formülü

2.3.4 Duyarlılık (Recall)

Duyarlılık, karmaşıklık matrisinden elde edilen değerlerdeki yapılan negatif tahminlerin yüzde kaçının doğru olduğunu hesaplamak için kullanılmaktadır.

Dolayısıyla, duyarlılığın hesaplanması için doğru pozitiflere ve yanlış negatiflere ihtiyaç duyulmaktadır (bakınız Şekil 2.4).

$$\text{Duyarlılık} = \frac{TP}{TP + FP}$$

Şekil 2.4: Duyarlılık formülü

2.3.5 F-1 Puanı (F-1 Point)

Eğer ki yapılan sınıflandırma işlemlerin de tam kesinlik isteniyorsa o halde en uçlardaki yanlışların veya doğrularında hesaba katılması gerekmektedir. Bu durum da yapılacak olan şeyin normal bir şekil de ortalama hesaplamanın olduğu düşünülebilir. Ancak geliştirilen F-1 puanı hesaplaması sayesinde ortalamanın oluşturacağı sonuçtan daha keskin sonuçlar elde edilebilmektedir. Bunu da önceden hesaplanan kesinlik ve duyarlılık ölçütleriyle gerçekleştirmektedir.

Sonuç olarak, önceden elde edilen kesinlik ve duyarlılık ölçütlerinin kullanıldığı bir formül üretilmiştir. Ancak unutulmamalıdır ki F-1 puanı hesaplanırken 2 farklı ölçüt kullanılmasından ötürü, bilgisayarlarda performans düşüklüğüne, hesaplama zamanın da artışa neden olmaktadır (bakınız Şekil 2.5).

$$\text{F-1 Puanı} = 2 * \frac{\text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}}$$

Şekil 2.5: F-1 puanı formülü

2.4 Hesaplamaların Anlamlandırılması ve Neticesinde Bir Sonuca Varılması

Önceden geçekleştirilen sınıflandırma yöntemleri ve değerlendirme ölçütleri sonucu elde edilen değerler ışığında, kullanılan sınıflandırma ve öğrenme yönteminin ilgili proje için ne kadar sağlıklı olduğu ve hangi oranda hata barındırabileceği gözlemlenebilmektedir.

2.5 Talebe Göre Görsel veya Metinsel Olarak Bir Sonuç Elde Edilmesi

En nihayetinde bir yapay zeka projesi geliştirilmek isteniyorsa, bu projeden alınmak istenen bir amaç bulunmaktadır. Bu amaç ister anlık olarak bir cevabın elde edilmesi olsun isterse de elde edilen değerlerin başka platformlarca kullanılıp yeni bir ürüne dönüştürülmesidir. Sonuç olarak ilgili yapay zeka projesi, doğruluğunun veya yanlışlığının tartışılacağı bir sonuç ortaya koyacaktır.

Şeklinde sıralanabilir.

Bahsedildiği üzere gerek telif haklarının korunması gerekse de sahte görsellerin veya metinlerin oluşturulamaması için gereken denetim mekanizmasının sağlıklı bir şekilde işleyebilmesi adına bu projede tersine bir sıralamadan bahsedilebilir. Yani elde edilen görselin tahmin edilmesi işlemidir. Ancak yapılacak olan bu tahminin yüzdesel oranı yüksek olmak zorundadır. Aksi takdirde söz konusu denetim mekanizması sağlıklı bir şekilde çalışmayacaktır.

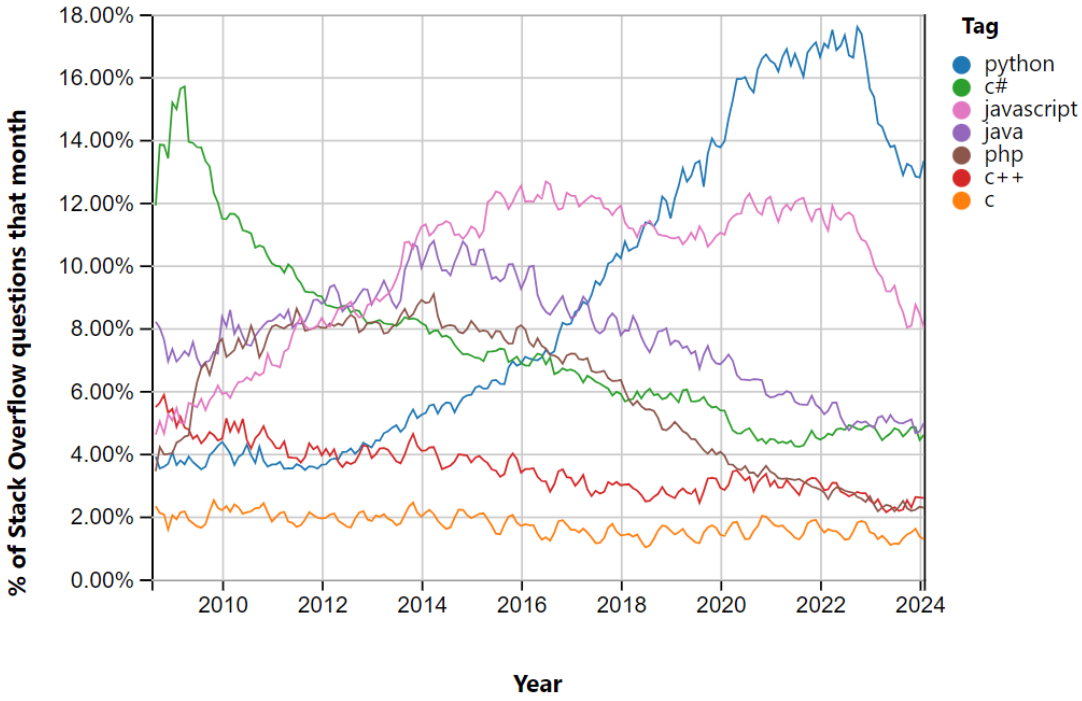
Dolayısıyla bu projede hem hassas sonuçların alınması hem de yüksek olasılıklı tahminlerin yürütülebilmesi adına “Python” programlama dili ve ilgili kütüphaneleriyle “Rastgele Ormanlar” öğrenme modeli kullanılmaktadır.

Python Programlama Dili

Python, Hollandalı bir yazılım geliştirici olan Guido van Rossum tarafından geliştirilmiş ve ilk kez 20 Şubat 1991 tarihinde duyurulmuştur. Programlama dili ismini de BBC de 1969 ile 1974 tarihleri arasında yayınlanan "Monty Python's Flying Circus" adlı diziden almıştır. Bu dizinin önceki zamanlarda yayınlanan kitaplarını okuyup etkilenen Van Rossum' a göre "Python" ismi hem bir miktar gizem içerirken hem de aklındaki diğer isimlere göre daha kısa ve dile getirilebilmesi kolaydır.

Python, öğrenilmesi ve pratikte uygulanması görece diğer programlama dillerine göre daha kolay olmasından ötürü, pek çok proje alanında yoğunlukla kullanılabilmektedir. Bu nedenle geleneksel programlama dillerine göre daha fazla kullanım ortamına sahip olmuştur.

Bu durumu, yazılım geliştiricilerinin sıklıkla kullandığı ve burada kendilerini pek çok yazılım dilinde geliştirmek isteyen geliştiricinin soru-cevap şeklinde topluluk kurduğu bir platform olan Stack Overflow' un yayınladığı ve Python programlama dilini diğer dillerle kıyaslandığı grafik kanıtlar niteliktedir (bakınız Şekil 3.1).



Şekil 3.1: Stack Overflow' un içerdiği Python' ın diğer dillere göre kullanım oranı grafiği

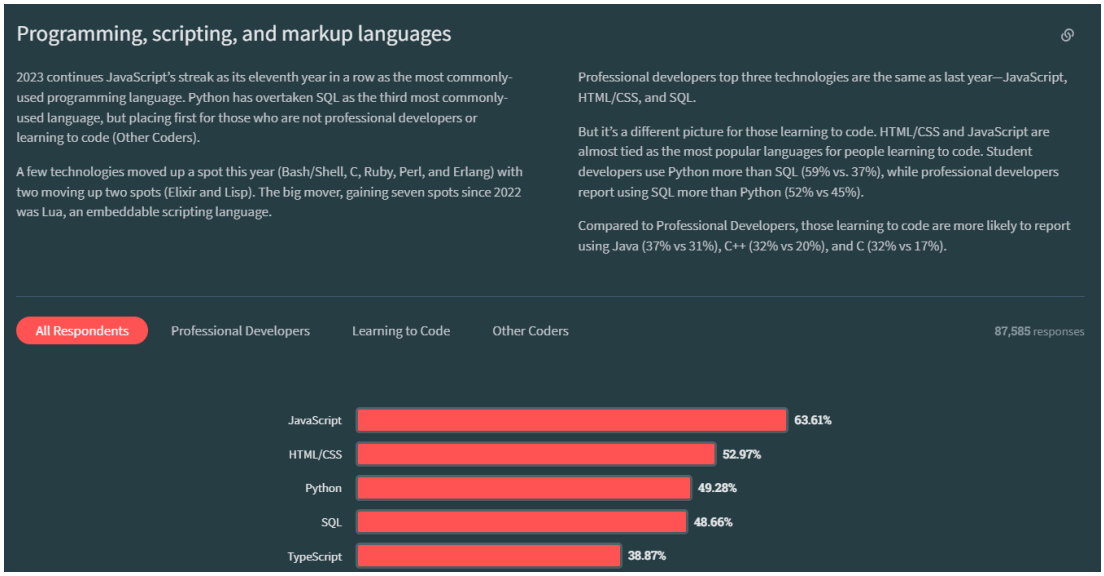
Dolayısıyla, genel-amaçlı programlama yapılmak istenen projeler için oldukça kolaylıklar sağlamaktadır. Örneğin, NYP, veri bilimi, yapay zeka projeleri, mobile ve web tasarımı, yazılım otomasyonları... gibi alanlar da yoğunlukla kullanılmaktadır. Böylelikle, Python belirtilen bu nedenler çerçevesinde büyümeye devam etmektedir.

Python, belirtilen bu kolaylıkları sağlayabilmesinin en önemli sebeplerinden birisi Python' ın büyük bir çoğunluğunu açık kaynak olarak geliştirilen sayısız kütüphaneler oluşturur. Bu kütüphaneler, bir yazılım geliştiricinin projesine kolay bir şekilde entegre edebilmesine ve hızlı bir şekilde geri dönüt alabilmesine olanak tanır. Böylelikle geliştiricinin kullanılan kütüphanedeki kodları sürekli baştan yazmasının da önüne geçilmektedir. Dolayısıyla hem zamandan hem de maliyetten kazanç sağlanmaktadır.

Python farklı farklı ortamlarda ve IDE' ler de çalıştırılabilmesinden ötürü, her bilgisayarın ve her geliştiricinin kendine has stiline ayak uydurması kolaydır. Örneğin yaygın bir şekilde kullanılan ve Microsoft tarafından geliştirilen VSC IDE' sine kolay ve hızlı bir şekilde entegre olabilirken, pek çok platformu içinde

barındıran Anaconda' nın geliştirdiği bir platform olan Jupyter Notebook IDE' sinde de kolay ve hızlı bir şekilde kullanılabilmektedir. Böylelikle, geliştiriciler için pek çok sayıda olasılığa sahip olmakta ve yoğun bir şekilde kullanılmaktadır.

Bu durumu kanıtlar nitelikte olan, Stack Overflow' un her sene gerçekleştirmiş olduğu ve platformu kullanan yazılım geliştiricilerine yönelttiği "Developer Survey" isimli anketin sonuç raporuna göre 2023 senesin de Python' ın yapay zeka alanında nasıl bir ivme kazandığı gösterilmektedir (bakınız Şekil 3.2).



Şekil 3.2: Stack Overflow da Python için hem metinsel açıklama hem de grafiksel olarak diğer dillerle karşılaştırma grafiği

3.1 Python Bu Projede Nasıl Bir Rol Üstlenmektedir?

Python, yapay zeka projesi geliştirmek isteyen bir geliştirici için pek çok yönden kolaylıklar sağlamaktadır. Söz konusu bu kolaylıklar 6 alt başlık altında incelenebilir;

3.1.1 Basitlik ve Okunabilirlik

Python doğası gereği sözdiziminin kolay ve anlaşılabilir olmasından ötürü, ister geliştiricinin kolay bir şekilde kodlarını yazabilmesi ve hata ayıklayabilmesine isterse de daha sonraları ilgili kodun bakım ve onarımının yapıldığı zaman o zamanki geliştiricinin kolay bir şekilde anlayabilmesine olanak tanır. Dolayısıyla hem yapay zeka alanında bir proje geliştirmek isteyen hem de yapay zeka teknolojilerini öğrenmek isteyen bir geliştiricinin atması gereken en önemli adımlardan biri Python programlama dilini öğrenmektir.

3.1.2 Yaygın Kütüphaneler ve Çerçeveler

Python ortamında bir proje geliştirilmek istendiğinde açık kaynak olarak ilgili projeye yönelik kütüphanelere erişim kolaydır. Dolayısıyla bir yapay zeka projesi geliştirilmek istendiğinde, bu alan için özel hazırlanmış kütüphanelere de erişim o kadar kolaydır. Yapay zeka alanın da yoğun bir şekilde TensorFlow, Scikit-Learn gibi kütüphaneler yoğun bir şekilde kullanılmaktadır.

3.1.3 Esneklik ve Ölçeklenebilirlik

Python, sürekli kendini geliştiren bir platform olmasından ötürü, geliştirmeler aşamasında olabildiğince geliştiricilere kolaylık sağlayabilmektedir. Bunlardan biri de diğer platformlara kolay ve hızlı bir şekilde entegre olabilmesidir. Gerek bir Back-End sürecinde veri tabanını inşa etmede gerekse de bir yapay zeka projesi için gereken veri setlerini işleyebilmede önemli kolaylıklar sağlamaktadır.

3.1.4 Güçlü Topluluk Desteği

Python' ın her geçen gün gelişen geliştirici topluluğu hem diğer geliştiricilerin Python' ı yapay zeka projelerine kolay ve hızlı bir şekilde entegre edebilmesine hem de daha çok çeşitte proje için açık kaynağa erişimi konusun da iş birliği ortamı oluşturur. Böylelikle proje geliştirilirken oluşacak kafa karışıklıklarının veya sorunların daha kolay ve hızlı bir şekilde üstesinden gelinebilmektedir. Dolayısıyla,

bu topluluk yapay zeka projeleri için önemli olan noktaların daha çok geliştiriciye erişimini sağlar. Bu durum yapay zekanın daha stabil ve güvenilir bir şekilde geliştirilebilmesine olanak tanımaktadır.

3.1.5 Hızlı Prototipleme ve Geliştirme

Python, özel olarak yapay zeka için geliştirilen anlaşılır ve açık kaynaklı kütüphaneler sayesinde, yapay zeka konusunu hiç bilmeyen bir geliştiriciyi en düşük seviyeden alıp en üst seviyeye hızlı ve kolay bir şekilde taşıyabilmektedir. Böyle bir özelliğinin olmasından ötürü, her kesimden geliştiricinin yapay zeka teknolojilerine entegrasyonu kolay ve hızlı olmakta ve ilgisini arttırmaktadır.

3.1.6 Veri Bilimi Araçlarıyla Entegrasyon

Python için özel geliştirilmelerin yapıldığı veri bilimi kütüphaneleri olan Pandas, NumPy, SeaBorn, SciKit... gibi kütüphaneler, ilgili veri setlerini kolay ve hızlı bir şekilde işleyebilmesinden ötürü, yapay zekanın bel kemiğini oluşturan veri biliminin yani veri işleme süreci için Python çok önemli bir araç olmaktadır. Ayrıca Python pek çok bu tarz da kütüphanelerin bir arada kolay ve hızlı bir şekilde kullandırabilmesiyle birlikte daha az kod satırı ile daha çok işlemler gerçekleştirilebilmektedir. Bu durum hem geliştirici için hem de daha sonraları bakım ve onarım için ilgili kod satırlarını inceleyen geliştirici için kolaylıklar sağlamaktadır. Bunun yanında, Python pek çok bilgisayara entegre olabilmesi için bilgisayar performansında gözlemlenebilir artışlar sağlamasıyla büyük veri setlerini kolay ve hızlı bir şekilde işleyebilmektedir.

Belirtilen bu başlıklar, bir yapay zeka projesi geliştirmek isteyen her seviyedeki geliştiriciler için çok önemli noktalara değinmektedir. Dolayısıyla Python, bir yapay zeka projesi geliştirmek için uygulanması gereken adımlar adına kullanılabilecek en stabil, güvenilir, açık kaynaklı, kendini sürekli yenileyen, güçlü desteklere sahip bir platformdur. Sonuç olarak bu projede kullanılan yapay zeka aracının geliştirilmesin de Python programlama dili ve kütüphaneleri kullanılmaktadır.

Kullanılan Python Kütüphaneleri

Bu projede, yapay zeka aracını daha kolay bir şekilde geliştirmek için toplam da 6 adet Python kütüphanesi kullanılmaktadır.

4.1 Numpy

Numpy, temel anlamda Python programlama dili içerisin de geliştirilmiş olan sayısal dizi türleridir. Bu diziler aynı türde olmak zorundadır. Yani sayısal bir diziden bahsediliyorsa o zaman sadece sayısal değerlerden oluşan bir dizi de Numpy kullanabilir.

Numpy, içerisin de barındırdığı sayısal diziler vasıtasıyla Python da geliştirilen pek çok alan da kolaylıkla kullanılabilmektedir. Ancak günün sonun da Numpy bir açık kaynak kod olarak geliştirilen bir kütüphane olduğundan ötürü, her proje başlangıcın da "import" edilmesi gerekir. Proje içerisin de daha kolay bir tanımlama yapılabilmesi adına import işlemi sırasında "np" ifadesi sıklıkla geliştiriciler tarafından kullanılmaktadır.

4.2 Skimage

Skimage, görsel işleme yeteneği kazandırılmak istenen bir yapay zeka projesi geliştirmek için kullanılabilecek olan açık kaynak kodlu bir Python kütüphanesidir. Bu kütüphane içerisin de araştırma, eğitim ve endüstriyel projelerin de geliştirilebileceği algoritmaları barındırmaktadır.

Skimage, alt yapısın da farklı işlevler için geliştirilen alt kütüphanelere sahiptir. Bunlar Io, Color, Data, Filter, Img_as_ubyte... şeklindedir. Bu projede de io, color ve img_as_ubyte alt kütüphaneleri kullanılmaktadır.

4.2.1 Io

Skimage kütüphanesinin bir alt kütüphanesi olan io, proje içerisin de bir görselin okunmasın da ve yazılmasında kullanılmaktadır.

4.2.2 Color

Skimage kütüphanesinin bir diğer alt kütüphanesi olan color, görsellerdeki renk düzenlemeleri için kullanılmaktadır.

Görüntü işleme sırasın da color kütüphanesine olan ihtiyaç, ilgili görseldeki nesnelerin veya şekillerin anlamlandırılabilmesine olan gereksinime bağlıdır. Bu nedenle color alt kütüphanesi pek çok görüntü işleme projelerin de sıklıkla kullanılmaktadır.

Color, kendi içerisin de pek çok alt kütüphaneye ayrılır. Bunlar; combine_stains, convert_colorspace, rgb2gray... şeklindedir.

4.2.3 Img_as_ubyte

Skimage kütüphanesinin bir diğer alt kütüphanesi olan img_as_ubyte, bir görüntünün [0, 255] değerleri arasında etiketsiz bayt biçimine dönüştürerek tanımlanabilmesine olanak tanımaktadır.

Bir görüntünün görüntü işleme sırasın da böyle bir işleme maruz kalmasının nedeni, görüntünün sayısal bir değere dönüştürülerek, görüntünün barındırdığı şekillerin daha kolay bir şekilde anlamlandırılabilmesine olanak tanınmaktadır.

Dolayısıyla, img_as_ubyte alt kütüphanesi pek çok görüntü işleme projelerin de sıklıkla kullanılmaktadır.

4.3 Scipy.stats

Scipy.stats kütüphanesi, Python için geliştirilmiş ve içerisin de istatistiksel işlemlerin olduğu (olasılık dağılımı, özet ve frekans istatistikleri, korelasyon fonksiyonları ve istatistiksel testler, maskelenmiş istatistikler, çekirdek yoğunluğu tahmini, Monte Carlo benzeri işlevsellik... gibi) bir kütüphanedir.

Kütüphane istatistiksel ifadeler içerdiği için pek çok alanda kullanılabilmektedir. Örneğin, bir görsel işleme projesin de önceden işlenen görüntünün barındırdığı renkler ve bytesal karşılığı sonucu yürütülebilecek tahminlerin daha kolay bir şekilde anlamlandırılabilmektedir.

Scipy.stats kütüphanesi kendi içerisin de alt kütüphanelere de sahiptir. Bunlar Skew, Kurtosis, Mode, Expectile... şeklindedir.

4.3.1 Skew

Scipy.stats kütüphanesinin bir alt kütüphanesi olan skew, önceden belirlenmiş olan bir veri setindeki örnek çarpıklığını hesaplamak için kullanılmaktadır. Çünkü normal bir şekilde oluşturulmuş bir veri setin de çarpıklık sıfıra yakın olmalıdır. Eğer ki örnek çarpıklığı sıfırdan uzaklaşıyorsa, üretilmiş olan veri seti yanlış olabilir kanısına varılmaktadır.

Bu nedenle, bir veri seti kullanılmadan önce ilgili veri setinin örnek çarpıklığının hesaplanması gerekmektedir.

4.3.2 Kurtosis

Scipy.stats kütüphanesinin bir alt kütüphanesi olan kurtosis, bir veri setinin basıklığını hesaplamak için kullanılır.

Basıklık kısaca dördüncü merkezi momentin varyansın karesine bölümü şeklinde tanımlanabilir.

Basıklığın hesaplanması, elde bulundurulan veri setinin ne kadar objektif bir şekilde geliştirildiğini tespit etmek açısından önemlidir. Yani, eğer ki veri seti bir noktaya daha çok değiniyor diğer noktalara neredeyse hiç değinmiyorsa bu durumda, öğrenme aşamasın da yanlış kararların verilmesi sorunu ortaya çıkmaktadır. Bu nedenle bir veri setin de basıklığın sıfıra yakın olması önemlidir.

4.4 Sklearn.ensemble

Sklearn.ensemble kütüphanesi, ensemble öğrenme yöntemini içerisin de barındıran bir kütüphanedir. Ensemble öğrenme ise, bir proje içerisin de kullanılan öğrenme yöntemlerine tek tek sorular sorarak her birinden aldığı yanıtlara göre bir kanıya varılmasına olanak tanıyan bir öğrenme yöntemidir. Bu yöntem, kesinliğe daha yakın sonuçlar verebileceği düşüncesiyle sıklıkla yapay zeka projelerin de kullanılır.

Sklearn.ensemble kütüphanesi de bu nedenle alt kütüphaneleri olarak RandomForestClassifier, BaggingClassifier, ... gibi öğrenme yöntemlerini barındırmaktadır.

4.4.1 RandomForestClassifier

RandomForestClassifier, sklearn.ensemble kütüphanesinin bir alt kütüphanesidir. Bu kütüphane bilindik ve sıkılıkla kullanılan rastgele orman denetimli öğrenmenin sınıflandırma yöntemini bir Python ortamında koda dökmek için kullanılmaktadır.

4.5 Sklearn.model_selection

Sklearn.model_selection kütüphanesi, makine öğrenmesi konusun da geliştirilen bir proje olan Scikit-learn kütüphanesinin bir alt kütüphanesidir. Scikit-learn kütüphanesi içerisin de Python da geliştirilen NYP özelliğini barındırmaktadır. Böylelikle pek çok yapay zeka projelerin de sıklıkla kullanılmaktadır.

Sklearn.model_selection kütüphanesi bir alt kütüphane olarak train_test_split kütüphanesini de barındırmaktadır.

4.5.1 Train_test_split

Sklearn.model_selection kütüphanesinin bir alt kütüphanesi olan train_test_split, bir makine öğrenmesi işlemin de kullanılacak olan bir veri setindeki verileri belli oranda gruplayarak, belli bir miktara sahip grubu modelin eğitmek belli miktara sahip grubu ise modeli test etmek için kullanmaktadır.

Bu oranlar projeden projeye, geliştiriciden geliştiriciye değişiklik göstermektedir. Örneğin "%80 train, %20 test" şeklinde ayrılmış olan bir projede ilgili veri setindeki veriler rastgele bir şekilde %80' si eğitim için %20' ise test için ayrılmış olur. Bu aradaki oranlama ne kadar doğru olursa, modelin öğrenmesi de o kadar kolay ve kesin sonuçlar verebilir hale gelmektedir.

Model, bu ayrıştırma sonucun da %80' lik kısımdan öğrendiğini %20' lik kısım üzerinde test ederek doğruya en yakın sonucu üretmeye çalışmaktadır.

4.6 Sklearn.metrics

Sklearn.metrics kütüphanesi, Scikit-learn kütüphanesinin bir alt kütüphanesidir. Bu kütüphane, makine öğrenmesi temelli projelerde yararlanılan değerlendirme ölçütlerini içerisinde barındırır. Bunlar, accuracy_score, precision_score... şeklindedir.

4.6.1 Accuracy_score

Accuracy_score, Sklearn.metrics kütüphanesinin bir alt kütüphanesidir. Bu kütüphane, değerlendirme ölçütlerinden bir olan "doğruluk" ölçütünü hesaplamak için Python kütüphanesi olarak geliştirilmiştir.

Elde edilen doğruluk değeri ne kadar %100' e veya 1' e yakınsa o kadar model doğru bir şekilde çalışıyor anlamına gelmektedir.

Veri Setinin Oluşturulması

Bir yapay zeka projesi geliştirilmek isteniyorsa, kullanılacak olan öğrenme modeli ve yöntemi her proje türü için farklı olsa da kullanılacak olan test ve eğitim için doğru veri setini belirlemek çok önemlidir. Dolayısıyla, yapılacak olan projenin temelin de metinsel bir durum söz konusuysa kullanılması gereken veri setini anahtar kelimelerin, cümle yapılarının doğru olarak tasnif edildiği bir veri setini seçmek önemlidir. Yapılacak olan projenin temelin de bir görsellik varsa bu durum da kullanılması gereken veri setini ilgili konuyla alakalı görsellerin oluşturması gerekmektedir.

Test ve eğitim amacıyla kullanılacak olan görsellerin de iyi bir şekilde tasnif edilmiş yani gürültüden uzak olması yapılacak olan yapay zeka projesini daha doğru sonuçlar elde edebilir hale getirebilmektedir. Ayrıca, eğer ki ilgili veri setinin kendini sürekli olarak yenileyen bir yapısı varsa bu durum daha da iyi sonuçlar elde edilebilmesini sağlamaktadır. Yani, ilgili yapay zeka projesi sürekli olarak güncel bir veri seti kullanıyorsa ileriki aşamalar da daha iyi bir şekilde eğitilebileceğinden, daha da yüksek sonuçlar elde edilebilecektir.

Bu projede yapay zeka aracının kendini iyi bir şekilde eğitebilmesi adına hem önceki yapay zeka projelerinin ürettiği sahte görsellerin hem de fotoğraf sanatçılarının oluşturduğu gerçek görsellerin olduğu bir veri seti kullanılmaktadır. Böylelikle bu projedeki yapay zeka aracının hem gerçek hem de sahte görsellerle elde ettiği eğitimi sayesinde yüksek sonuçlara ulaşabilmektedir.

Dolaysıyla, bu projede kendini sürekli olarak yenileyen, topluluk desteğinin yüksek seviyede olduğu, pek çok veri bilimiyle ve makine öğrenimiyle ilgilenen geliştiricilerin yapay zeka projelerini ve kaynak kodlarını içerisinde barındıran, telif haklarının ister ücretli isterse de ücretsiz olduğu bir veri seti platformu olan “Kaggle”

tarafından sağlanan “AI_Generated_Images” adlı veri setiyle yine aynı şekilde kendini sürekli yenileyen, geliştiren, gerçek görsellerin olduğu, telif haklarının ister ücretli olduğu isterse de ücretsiz olduğu bir görsellik platformu olan “Freepik” tarafından sağlanan “Kids Jumping Images” adlı veri setinden elde edilmektedir.

Bu iki platformdan 5’ er tane olmak üzere toplamda test ve eğitim verisi için 10 adet görsel kullanılmaktadır. Böyle bir tercihin sebebi ise, ileriki aşamalarda gerektiğin de ücretli telif haklarına sahip bir veri seti kullanılmak zorunda kalındığın da test ve eğitim maliyetlerini en düşükte tutup en yüksek seviyede verim alınabilmesinin istenmektedir. Bu nedenle bu proje için hem telif haklarının ücretsiz olduğu hem de kendini yenileyen ve yüksek çözünürlükte olan görseller kullanılmaktadır.

Kullanılan Python Kodları

Şekil 6.1 de belirtilen kodlar, Python kütüphanelerinin içerisinde bulunan fonksiyonların proje içerisin de kullanılabilmesi için proje dosyasına entegre edilmesi işlemidir. Bu işlem yapılmadığı takdirde, fonksiyonlar proje içerisin de kullanılamaz hale gelmektedir (bakınız Şekil 6.1).

```
import numpy as np
from skimage import io, color, img_as_ubyte
from scipy.stats import skew, kurtosis
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

Şekil 6.1: Projedeki kodların birinci bölümü

Şekil 6.2 de ki kod bloğu Python programlama dilinde fonksiyonların yazılmasında kullanılan yapıya sahiptir. Bu kod bloğunun tanımlanmasının amacı, veri setinde belirtilen görsellerin her birinin proje içerisin de okunmasını, gri tonlamalarının dönüştürülmesini, görsellerin 2 boyuttan 1 boyuta indirgenmesini ve görsellerin istatistiksel olarak hesaplanmasını sağlamaktadır.

Böyle bir işlem sayesin de hangi tür görselin bir insan tarafından hangi tür görselin ise bir yapay zeka projesi tarafından elde edildiğini anlayarak, sınıflandırma ve öğrenme işlemini daha doğru yapacak hale gelinebilmektedir (bakınız Şekil 6.2).

```
def get_image_features(image_path):

    img = io.imread(image_path)


    gray_img = color.rgb2gray(img)


    flat_img = gray_img.flatten()

    mean_value = np.mean(flat_img)
    std_dev = np.std(flat_img)
    skewness = skew(flat_img)
    kurt = kurtosis(flat_img)

    return [mean_value, std_dev, skewness, kurt]
```

Şekil 6.2: Projedeki kodların ikinci bölümü

Şekil 6.3 de ki kod bloğu, önceden belirlenmiş olan veri setlerindeki görsellerin proje içerisin de tanımlanması işlemini yürütmektedir.

“human_images” listesi, insanlar tarafından oluşturulmuş olan görsellerin listelendiği bölümdür. “ai_images” listesi ise, yapay zeka projeleri tarafından oluşturulmuş olan görsellerin listelendiği bölümdür (bakınız Şekil 6.3).

```
human_images = ['real1.jpg', 'real2.jpg', 'real3.jpg', 'real4.jpg', 'real5.jpg']
ai_images = ['ai1.png', 'ai2.png', 'ai3.png', 'ai4.png', 'ai5.png']
```

Şekil 6.3: Projedeki kodların üçüncü bölümü

Şekil 6.4 de ki kod bloğu, bir önceki kod bloğun da tanımlanan veri seti görsellerinin sınıflandırıldığı bölümdür. Bu bölüm işlemlerini “X” ve “y” listelerine “0” ve “1” etiketlerini eklemesiyle gerçekleştirmektedir.

Eğer ki alınan görsel insanlar tarafından üretilmiş bir görsel ise o zaman X listesine görsel eklenmekte ve y listesine ise 0 etiketi eklenmektedir.

Eğer ki alınan görsel yapay zeka projeleri tarafından üretilmiş bir görsel ise o zaman X listesine görsel eklenmekte ve y listesine 1 etiketi atanmaktadır (bakınız Şekil 6.4).

```
X = []
y = []

for path in human_images:
    X.append(get_image_features(path))
    y.append(0)

for path in ai_images:
    X.append(get_image_features(path))
    y.append(1)
```

Şekil 6.4: Projedeki kodların dördüncü bölümü

Şekil 6.5 de ki kod bloğu, önceden elde edilmiş olan X ve y listelerindeki sonuçları kullanarak, elde edilen bütünleşik veri setini rastgele bir şekilde %20 oranında test için %80 oranın da ise eğitim için bölüştürmektedir (bakınız Şekil 6.5).

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Şekil 6.5: Projedeki kodların beşinci bölümü

Şekil 6.6 da ki kod bloğu, bir önceki kod bloğundan elde edilen eğitim verisini ve Rastgele Orman sınıflandırma yöntemini kullanarak eğitme işlemini gerçekleştirmektedir (bakınız Şekil 6.6).

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

Şekil 6.6: Projedeki kodların altıncı bölümü

Şekil 6.7 de ki kod bloğu, bir önceki kod bloğundan elde edilen eğitim işlemi sonucu test verisiyle tahmin edebilme işlemini gerçekleştirmektedir (bakınız Şekil 6.7).

```
y_pred = model.predict(X_test)
```

Şekil 6.7: Projedeki kodların yedinci bölümü

Şekil 6.8 de ki kod bloğu, önceki kod bloklarından elde edilen test ve eğitim verilerini alarak doğruluk değerlendirme ölçütüyle hesaplama işlemi gerçekleştirmektedir. Sonrasında elde edilen doğruluk değerini ekrana yazdırmaktadır (bakınız Şekil 6.8).

```
accuracy = accuracy_score(y_test, y_pred)
print(f"Doğruluk: {accuracy}")
```

Şekil 6.8: Projedeki kodların sekizinci bölümü

Şekil 6.9 da ki kod bloğu, eğitim, test ve doğruluk değerinin hesaplanması sonucu kullanıcının yükleyeceği resmin projeye eklenmesini gerçekleştirmektedir. Burada

kullanılan “new_image2.png” görsel dosyası kullanıcının projeye dahil etmek ve tahmini gerçekleştirmek istediği görselin ismidir.

Sonrasında elde edilen kullanıcı görseli test ve eğitim verileriyle tahmin işlemine sokulmaktadır (bakınız Şekil 6.9).

```
new_image_path = 'new_image2.png'
new_image_features = get_image_features(new_image_path)
prediction = model.predict([new_image_features])
```

Şekil 6.9: Projedeki kodların dokuzuncu bölümü

Şekil 6.10 da ki kod bloğu bir önceki kod bloğundan elde edilen tahmin değerine göre koşullu bir sonuç elde etmektedir (bakınız Şekil 6.10).

Eğer ki tahmin değeri 0 ise kullanıcının yüklediği görselin bir insan tarafından üretildiği tahminine varılmaktadır.

Eğer ki tahmin değeri 0’ dan farklı bir değere sahip ise o zaman kullanıcının yüklediği görselin bir yapay zeka tarafından üretildiği tahminine varılmaktadır.

```
if prediction == 0:
    print("Yüklediğiniz resim büyük bir ihtimalle bir insan tarafından üretilmiştir.")
else:
    print("Yüklediğiniz resim büyük bir ihtimalle bir yapay zeka tarafından üretilmiştir.")
```

Şekil 6.10: Projedeki kodların onuncu ve son bölümü

Literatür Taraması

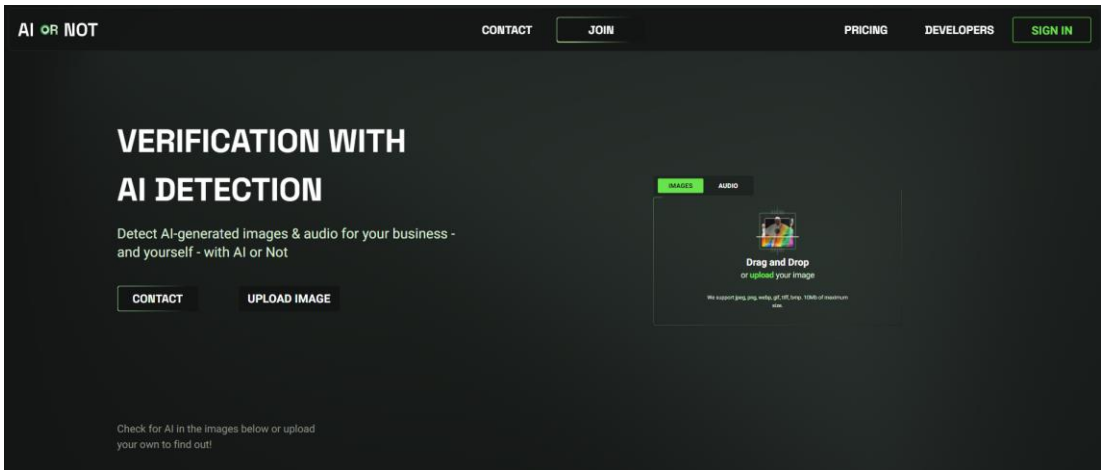
Yapay zekanın her geçen gün gelişmesiyle birlikte, kötü amaçlarla kullanılması da yaygınlaşmaktadır. Bu nedenle, pek çok kullanıcının “elimde bulunan bu görselin gerçek olup olmadığını nasıl anlayabilirim veya nasıl tasdik edebilirim?” gibi sorularını sorması gayet doğaldır. Dolayısıyla, her ne kadar yapay zeka projeleri işimizi büyük ölçüde kolaylaştırsa da kötü amaçlar için kullanılmasını, eğer ki kullanılacaksa bile bunu nasıl denetleyip tespit edebiliriz sorusunun sorulması çok önemli hale gelmektedir.

Yapay zeka projeleri geliştiren bazı geliştirici de bahsedilen bu soruyu sormuşlardır ve bir takım denetleme mekanizmaları geliştirmeye çalışmışlardır. Bunların en önemlilerinden birisi de bu projenin de ana konu başlığı olan “Bir Görselin Gerçekliğini Tespit Etme” konusunda pek çok proje geliştirmişlerdir. Bunlar; “AI or Not, Is It AI, Illuminarty, Haggingface, Content at Scale, Sightengine, Fake Image Detector, SynthID” isimli projelerdir.

7.1 AI or Not

AI or Not, web tabanlı ve abonelik ücretine sahip olan bir yapay zekayla üretilmiş olan görsel ve ses tespit etme projesidir.

Bu proje kullanıcının elinde bulunan ses veya görüntüyü sisteme yüklemesi ve belli oranda tahmin elde etmesi üzerine bir çalışma prensibi bulunmaktadır. Ancak kullanıcı projeye internet vasıtasıyla erişim sağlayabilmesinden ötürü, elindeki sesin veya görselin çok önemli veya internet ortamında yayılmasının riskli olacağı bir durum söz konusu olduğunda, bu projeyi maalesef kullanamamaktadır. Dolayısıyla, şu an da geliştirilen bu projede herhangi bir internet bağlantısı veya abonelik ücreti olmamasından ötürü hem veri güvenliği açısından hem de maliyet açısından daha avantajlı bir duruma geçilmektedir (bakınız Şekil 7.1).



Şekil 7.1: AI or Not projesinin web sitesinin ana sayfa görüntüsü

7.2 Is It AI

Yapay zeka aracı olarak üretilen bu proje, görsellerin bir insan tarafından mı yoksa bir yapay zeka projesi tarafından mı oluşturulduğunu belirlemek için analiz ederek bir sonuca varmaktadır.

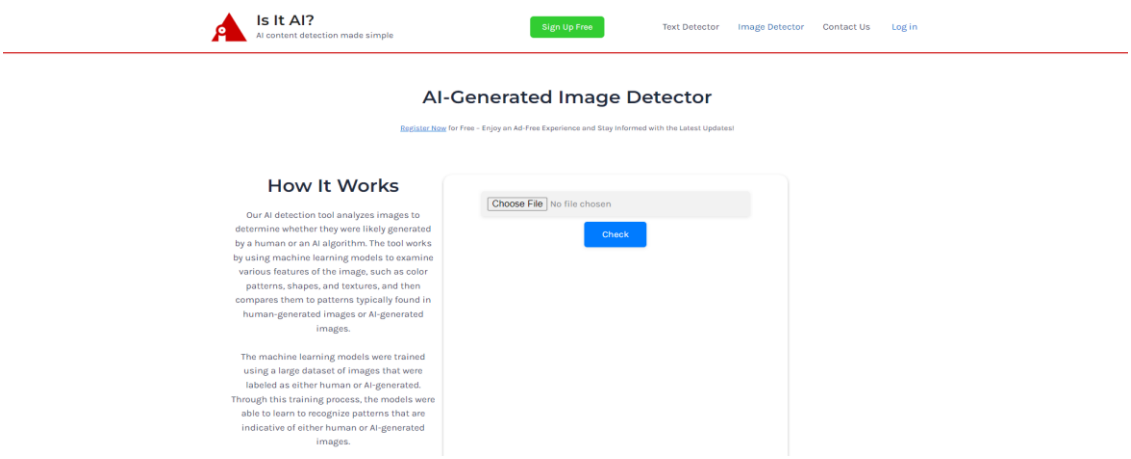
Proje, görüntünün renk desenleri, şekilleri ve dokuları gibi çeşitli özelliklerini incelemek için makine öğrenimi modellerini kullanarak çalışmaktadır. Ardından bunları genellikle insan tarafından oluşturulan görsellerde veya yapay zeka tarafından oluşturulan görsellerde bulunan desenlerle karşılaştırmaktadır.

Makine öğrenimi modelleri, “insan veya yapay zeka tarafından oluşturulmuştur” olarak etiketlenen geniş bir görsel veri kümesi kullanılarak eğitilmektedir. Bu eğitim süreci sayesinde modeller, insan veya yapay zeka tarafından oluşturulan görsellerin göstergesi olan kalıpları tanımayı öğrenebilmektedir.

Anlatıldığı üzere, Is It AI projesi, geliştirilen bu projeye çalışma prensibi açısın da oldukça benzemektedir. Ancak, internete bağlanılarak ilgili görselin işleme sokulmasından ve geliştiricinin hangi amaçla bu projeyi geliştirdiğinin anlaşılamamasından ötürü, kullanıcı hem veri güvenliğini riske atmış olur hem de

bilgisayarına herhangi bir siber saldırının gerçekleşip gerçekleşmeyeceğini de tespit edememektedir. Dolayısıyla bu projenin kullanımı pek de sağlıklı değildir.

Geliştirilen bu proje, bilgisayara kurularak kullanıldığı için internete herhangi bir veri sızıntısı veya bir siber saldırıya maruz bırakmaktadır (bakınız Şekil 7.2).



Şekil 7.2: Is It AI projesinin web sitesinin ana sayfa görüntüsü

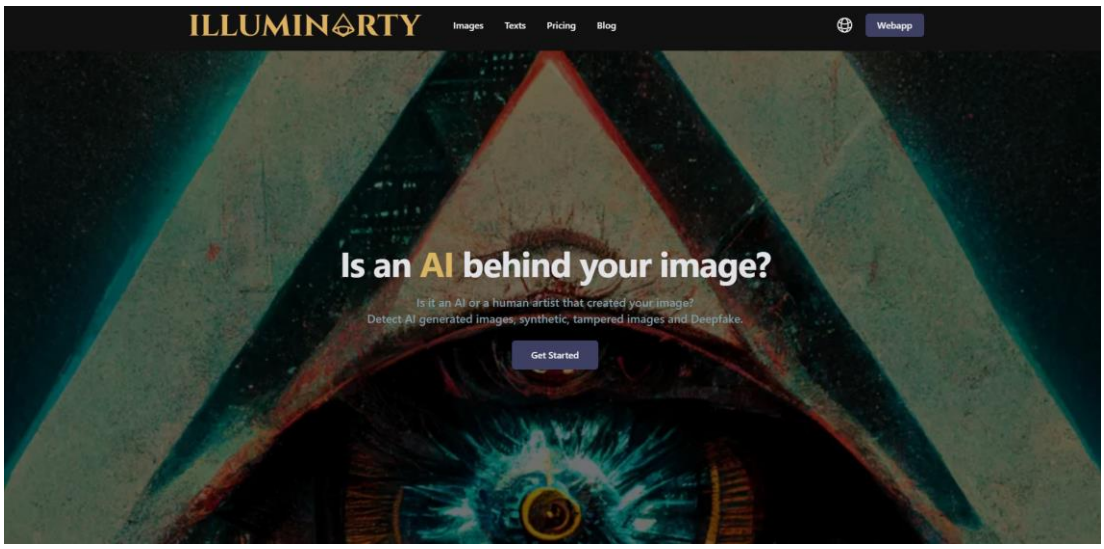
7.3 Illuminarty

Bir görselin bir yapay zeka tarafından mı yoksa bir insan tarafından mı oluşturulduğunu tespit etmek için kullanılan Illuminarty projesi, belirli bir görüntü için yapay zeka tarafından oluşma olasılığını ortaya koymaktadır.

Illuminarty, görüntünün halka açık bir yapay zeka modeli tarafından üretilme olasılığını sağlamak için çeşitli bilgisayarlı görme algoritmalarını birleştirmektedir. Bu işlemi de ilgili görselin sisteme yüklenmesi ve bunun karşılığında belli bir abonelik ücretin ödenmesiyle gerçekleştirmektedir.

Illuminarty projesi, görselin kendi sistemine yüklenmesini istemesinden ötürü hem veri sızıntılarına hem de tespit maliyetlerinin artmasına neden olmaktadır. Dolayısıyla, geliştirilen bu projede görsel hiçbir şekilde internet ortamına maruz

bırakılmamakta ve bu işlem için herhangi bir ücret talep edilmemektedir (bakınız Şekil 7.3).



Şekil 7.3: Illuminarty projesinin web sitesinin ana sayfa görüntüsü

7.4 Haggingface

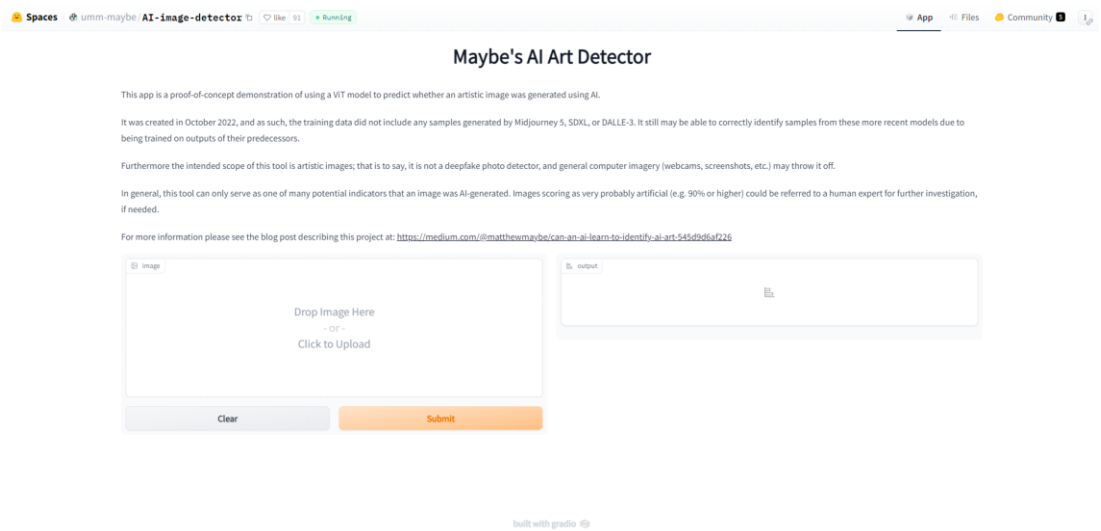
Haggingface, yapay zeka kullanılarak sanatsal bir görselin oluşturulup oluşturulmadığını tahmin etmek için Vision Transformers öğrenme modelinin kullanıldığı bir görsel tespit projesidir.

Proje, Ekim 2022'de oluşturulduğundan eğitim verileri Midjourney 5, SDXL veya DALLE-3 tarafından oluşturulan hiçbir örneği içermemektedir. Önceki modellerin çıktıları üzerine eğitilmiş olması nedeniyle, bu proje daha yeni modellerden örnekleri hâlâ doğru bir şekilde tanımlayamamaktadır.

Ayrıca bu aracın amaçlanan kapsamı sanatsal görsellerdir; diğer bir deyişle, bu bir deepfake görsel dedektörü değildir ve genel bilgisayar görüntüleri (web kameraları, ekran görüntüleri vb.) projeyi yanıltabilmektedir. Bu nedenle Haggingface' in kullanım alanı oldukça dardır.

Genel olarak bu proje, bir görüntünün yapay zeka tarafından oluşturulduğunu gösteren birçok potansiyel göstergeden yalnızca biri olarak hizmet edebilmektedir. Büyük ihtimalle yapay (örneğin %90 veya daha yüksek oranlıkla) olarak değerlendirilen görselleri tespit edebilmektedir. Bu nedenle daha düşük oranlı görsel tahminleri için bu proje kullanılamamaktadır.

Haggingface, belirtilen bu özellikleri sayesinde kullanılacak bir proje gibi durmamaktadır. Burada veri sızıntısından veya siber saldırı riskinden bahsedilmesine bile gerek kalmadan belli orandaki görsellerin ve eski görsellerin kullanılması gerektiğini savunmaktır. Bu nedenle bu projenin kendini sürekli geliştiren bu proje ile mukayese bile edilmemesi gerekmektedir (bakınız Şekil 7.4).



Şekil 7.4: Haggingface projesinin web sitesinin ana sayfa görüntüsü

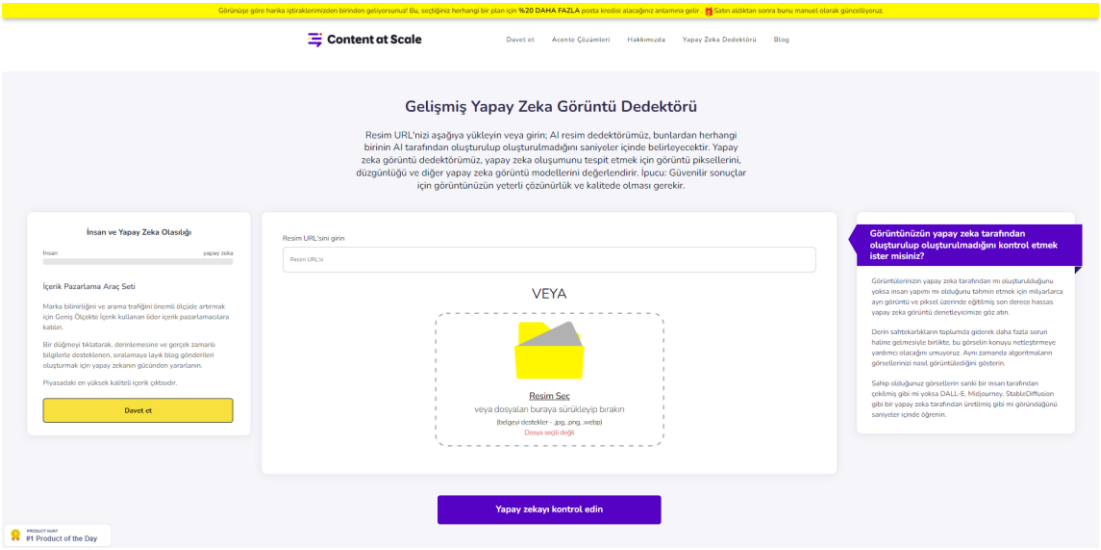
7.5 Content at Scale

Content at Scale, bir kullanıcının elinde bulunan bir görselin bir yapay zeka tarafından mı yoksa bir insan tarafından mı oluşturulduğunu tespit etmek için üretilmiş olan bir projedir.

Ancak, ilgili görüntünün bir yapay zeka tarafından mı oluşturulduğunu yoksa bir insan yapımı mı olduğunu tahmin etmek için büyük bir veri setine ihtiyaç

duyulmaktadır. Bu durum da projenin bir bilgisayar ortamına kurulmasına imkan vermemekle birlikte işlemin de uzun süreler almasına neden olmaktadır. Bu nedenle kullanıcı elindeki görseli mecburen hem veri sızıntılarını hem de herhangi bir siber saldırı riskini göz ardı ederek sisteme yüklemek zorundadır.

Dolayısıyla, geliştirilen bu projeyle birlikte Content at Scale' a göre hem internet ortamına verilecek herhangi bir veri sızıntısının hem de herhangi bir siber saldırı için kullanılacak bir virüsün cihazlara geçmesinin önüne geçilmektedir (bakınız Şekil 7.5).



Şekil 7.5: Content at Scale projesinin web sitesinin ana sayfa görüntüsü

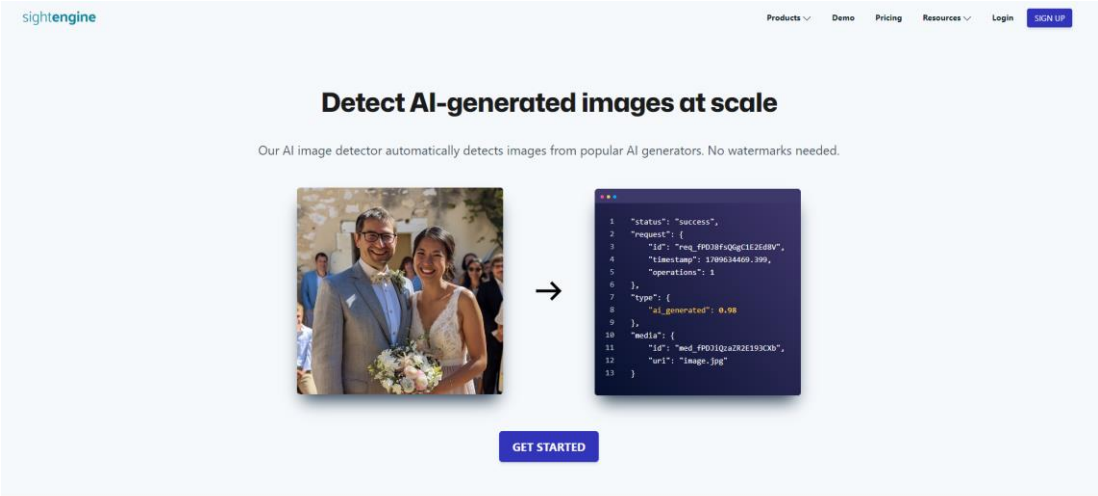
7.6 Sightengine

Sigthengine, bir kullanıcının elinde bulundurduğu bir görselin bir yapay zeka tarafından mı yoksa bir insan tarafından mı oluşturulduğunu tespit etmek için geliştirilmiş olan bir projedir.

Proje, içerisin de barındırdığı pek çok görsel sınıflandırma, öğrenme modeli ve büyük bir veri setiyle birlikte yüksek oranda doğru tahminde bulunduğunu belirtmektedir. Ancak, kullanılan modeller ve veri setleri, işlemlerin çok uzun

sürmesine yol açmaktadır. Ayrıca projenin abonelik ücretine sahip olması ve görselin bu sisteme internet vasıtasıyla yüklenmesinden ötürü hem tespit maliyetlerini de yukarıya çekmekte hem de veri sızıntılarına imkan sağlamaktadır.

Bu nedenle, geliştirilen bu projede proje bilgisayara kurulmasından ve herhangi bir abonelik ücretlendirmesinin talep edilmemesinden ötürü hem tespit maliyetleri ortadan kaldırmakta hem de veri sızıntısı ve olası bir siber saldırıya karşı güven sağlanmaktadır (bakınız Şekil 7.6).



Şekil 7.6: Sightengine projesinin web sitesinin ana sayfa görüntüsü

7.7 Fake Image Detector

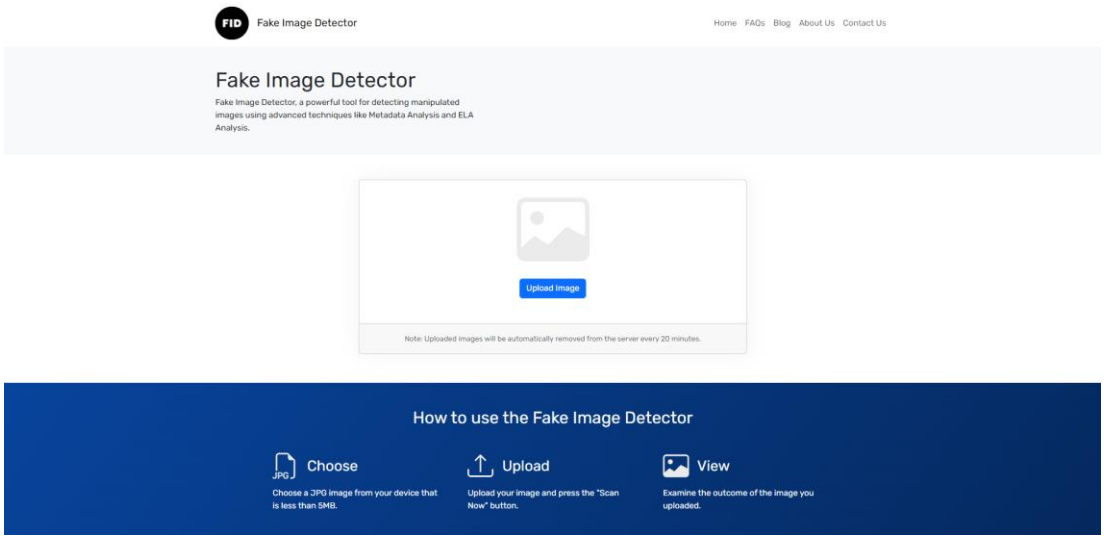
Fake Image Detector, bir görselin bir yapay zeka tarafından mı yoksa bir insan tarafından mı oluşturulduğunu tespit etmek için kullanılan bir projedir.

Fake Image Detector tarafından kullanılan önemli teknik, YİDH tanıyıcısını kullanan HDA' dır. Tipik olarak, bu bilgisayar talimatları seti yüz tanıma için ayrılmıştır. Bununla birlikte, sahte görüntülerin maskesinin kaldırılması bağlamında, YİDH tanıyıcı histogramlar oluşturulur ve bunları bir görselin gerçekliğinin belirlenmesi için karşılaştırılmaktadır.

YİDH tanıyıcı, görüntüye HDA uygulayarak görüntüyü ustaca analiz eder ve gerçekliğine ilişkin eğitimli bir karara varır. Bu yöntem gayet iyi sonuçlar verebilmektedir. Çünkü sahte görselleri ortaya çıkarmak için denenmiş ve onaylanmış bir yüz tanıma sisteminden yararlanılmaktadır.

Ancak anlatılanlar ve projenin çalışma prensibi ışığında Fake Image Detector, bir kullanıcıdan elinde bulundurduğu ilgili görseli o kullanıcıdan internet vasıtasıyla işleme sokabilmektedir. Ayrıca, sitenin kendi içerisin de ne türden virüsler barındırdığı da ilk bakışta bir kullanıcının anlayabilmesinin mümkün olamayacağından ötürü, olası bir siber saldırıya da ve veri sızıntılarına karşı korumasız kalınmaktadır.

Geliştirilen bu projeyle birlikte, kullanıcı ilgili görselini kendi cihazında ve herhangi bir internet bağlantısına ihtiyaç duymadan tespit işlemini gerçekleştirebilmektedir (bakınız Şekil 7.7).



Şekil 7.7: Fake Image Detector projesinin web sitesinin ana sayfa görüntüsü

7.8 SynthID

Google tarafından geliştirilen SynthID, bir görselin bir yapay zeka tarafından mı yoksa bir insan tarafından mı oluşturulduğunu tespit etmek için geliştirilmekte olan bir projedir.

SynthID, biri filigran eklemek için, diğeri tanımlamak için olmak üzere iki derin öğrenme modeli kullanmaktadır;

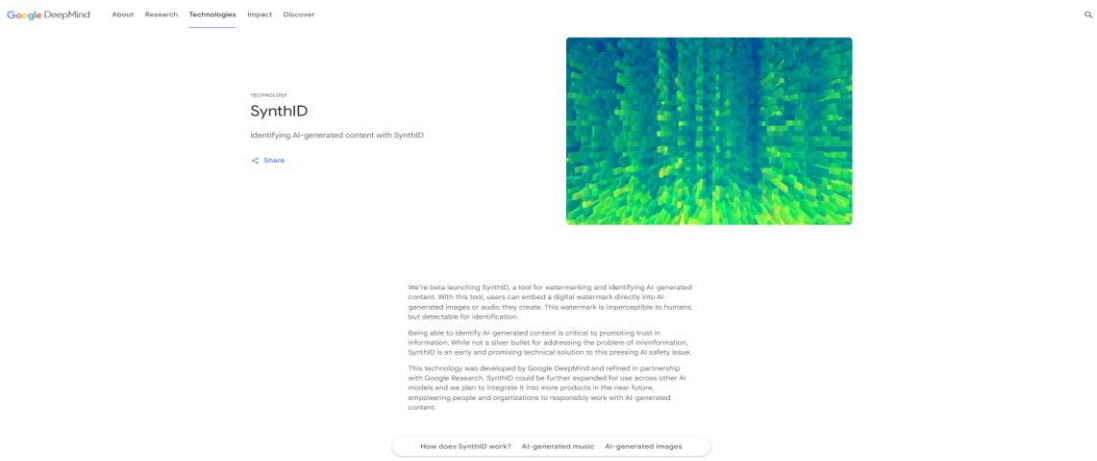
7.8.1 Filigranlama

SynthID, yapay zeka tarafından oluşturulan görsele doğrudan dijital filigran ekleyen yerleşik bir filigran teknolojisi kullanmaktadır. Birleştirilmiş model, filigranı gerçek içerikle hizalayarak algılanamazlığı iyileştirecek şekilde optimize edilmektedir.

7.8.2 Tanımlama

SynthID, dijital filigranı için görseli veya sesi tarayabilir ve kullanıcıların ilgili görselin veya bir kısmının yapay zeka modelleri kullanılarak oluşturulup oluşturulmadığını değerlendirmesine yardımcı olabilmektedir.

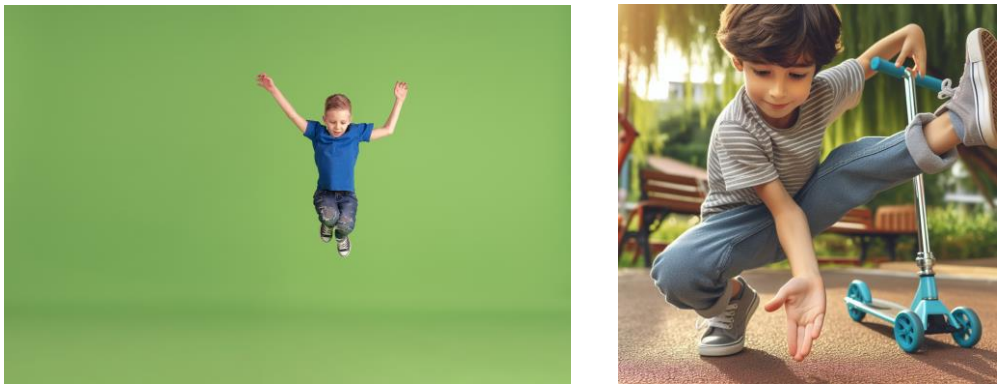
Ancak, ilgili görselin Google' ın sunucularına yüklenmesiyle işlevsellik kazanmasından ötürü, ileriki zamanlarda oluşabilecek olan veri sızıntılarına ve siber saldırılara karşı herhangi bir koruma sağlayamamaktadır. Bu nedenle, SynthID' nin aksine kullanılacak olan projeyle birlikte hem kullanıcının cihazına yüklenebilir olması hem de ücretsiz ve internet bağlantısı olmadan çalışabilmesi sayesinde daha güvenli, kolay ve hızlı bir şekilde tahminleme işlemi gerçekleştirilmektedir (bakınız Şekil 7.8).



Şekil 7.8: SynthID projesinin web sitesinin ana sayfa görüntüsü

Bulgular

Bu proje kapsamında kullanılan veri setleri ve öğrenme modeli sayesin de elde edilen veri incelendiğin de yüksek doğruluğa sahip tahmin sonuçları elde edilmektedir. Projenin geliştirilme aşamasında 5 adet gerçek ve 5 adet sahte olmak üzere toplam 10 adet görselden oluşan bir veri seti kullanılmaktadır (ilgili görseller "ekler" bölümünde belirtilmiştir). Ayrıca testlerin gerçekleştirilmesi için ise 1 adet gerçek ve 1 adet sahte görsel belirlenip sonuçlar alınmıştır (bakınız 8.1).



(a) (b)

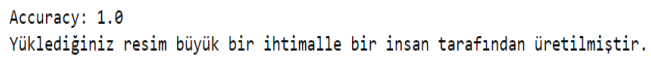
Şekil 8.1: Projenin testinde kullanılan görseller, (a) Gerçek görsel, (b) Sahte görsel

Testler sonucun da şekil 8.1' deki görsellerin önceden hangi kategoriye ait olduğu bilinmesine karşın, elde edilen sonuçlar ışığında projenin çıktı olarak verdiği cevapların her ikisi de doğru çıkmış ve doğruluk ölçütü yüksek elde edilmiştir.

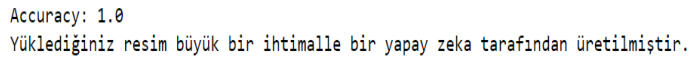
Dolayısıyla, veri setini oluşturan görsellerin sayısının az olmasına rağmen sonucun yüksek çıkması, bu projenin hem düşük performansa sahip bilgisayarlarda kolay ve hızlı bir şekilde çalıştırabileceğini hem de herhangi bir şekilde tespit maliyetine ihtiyaç duyulmayarak gerek kullanıcı bu projeye ücretsiz bir şekilde erişebilecek

gerekse de projenin çalışması için sürekli olarak veri setinin yenilenmesi ve saklanması için sunucu maliyetlerinin önüne geçilmiş olunacaktır.

Projenin geliştirilme aşamasın da elde edilmiş olan ve Şekil 8.1 de belirtilen test görselleri kullanılarak Şekil 8.2 de belirtilen sonuçlar ışığında “1,0” doğruluk ölçütü sonucu sayesin de projenin başarılı bir şekilde çalıştığı kanıtlanmıştır (bakınız Şekil 8.2).



(a)



(b)

Şekil 8.2: Projeden elde edilen sonuçlar, (a) Sonuç insan yapımı, (b) Sonuç AI yapımı

Tartışma

Kaggle ve Freepik platformları vasıtasıyla elde edilen veri setiyle projeden alınan doğruluk değerinin “1,0” olmasının nedeni, ilgili platformlarda yayınlanan veri setlerinin içerdiği görsellerin gayet açık ve yüksek çözünürlüklü olmasından ötürüdür. Bu nedenle, Kaggle üzerinden geliştirmeler yapmaya devam eden geliştiriciler, genellikle bu veri seti sayesin de yüz tanıma ve çocukların postürlerinin tespit edilmesi işlemlerini gerçekleştirmektedirler.

Ancak, kullanıcının elindeki görselin kalitesinin her zaman için yüksek olmasının mümkün olamayacağından ötürü, bu projede kullanılan veri setinin içerdiği görseller yeterli olamayacaktır. Bu nedenle aynı görsellerin daha bulanık veya çözünürlüğünün daha düşük olduğu ayrı bir veri seti oluşturularak, projenin yeniden eğitim ve teste tabi tutulması önemlidir.

Dolayısıyla, elde edilen 2 veri setinin birleştirilmesi sonucu kullanılacak olan yeni veri seti, yapılacak yeni bir eğitim ve test ayrıştırmasına tabi tutularak daha iyi sonuçlar gözlemlenebilecektir.

Sonuç

Geliştirilen bu projeyle birlikte, “Kaggle” ve “Freepik” dahil pek çok platformda olmayan ve rakiplerine göre hem tespit maliyetlerinin olmayıp hem de herhangi bir veri sızıntısı ve siber saldırı riskine maruz kalınmadan bir görselin gerçekliğinin tespit edilebilineceği “1,0” doğruluk ölçütü değeriyle birlikte ortaya konmuştur. Bu sayede bu proje diğer rakiplerine kıyasla daha güvenli ve az maliyetli bir seçenek olarak kullanılabilmektedir.

Bu başarılı sonucun kaynağı, oluşturulan veri setinin, “Rastgele Orman” öğrenme modelinin ve veri setinin gayet doğru bir şekilde “%20” test ve “%80” eğitim verileri olacak şekilde ayrıştırılmasıdır. Dolayısıyla, bu proje ile kullanıcılar artık elinde bulundurdukları değerli görselleri 3. taraf sanal ortamlarda paylaşmasının ve hatta abonelik gibi ücretlendirmelerinin önüne geçilmektedir.

Ayrıca, “Pyhton” programlama dili kullanılarak elde edilen performans iyileştirmeleri sayesinde projenin pek çok bilgisayara entegre edilmesi kolaylaştırılmaktadır. Bu sayede, performans sorunu yaşanabilecek olan bilgisayarlarda da bu proje rahat bir şekilde çalıştırılabilmekte ve sağlıklı bir şekilde doğru sonuçlar alınabilinmektedir.

Kaynaklar

Habibimoghaddam, F. (2024). *AI_Generated_Images*. Kaggle. https://www.kaggle.com/datasets/fhabibimoghaddam/ai-generated-images?resource=download

Candoğan, O. (2020, April 12). *Dünyaca Ünlü Bilim İnsanları Tarafından Yapay Zeka Üzerine Yapılmış 10 Düşündürücü Yorum.* Webtekno. https://www.webtekno.com/yapay-zeka-hakkinda-yapilmis-yorumlar h90025.html

Sönmez, C. (2023). *Yapay Zekaya Giriş*. İstanbul Teknik Üniversitesi. https://web.itu.edu.tr/~sonmez/lisans/ai/yapay_zeka_icerik1_1.6.pdf

Staff, C. (2024, April 3). *What Is Python Used For? A Beginner's Guide.* Coursera. https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

Yazar, İ. (2024, January 7). *Guido van Rossum.* Wikipedia. https://tr.wikipedia.org/wiki/Guido_van_Rossum

Trends, S. (2024). *Stack Overflow Trends.* Stack Overflow. https://insights.stackoverflow.com/trends?tags=python%2Cc%2Cc%23%2Cc%2B%2B%2Cjava%2Cphp%2Cjavascript

Robinson, D. (2017, September 6). *The Incredible Growth of Python.* Stack Overflow Blog. https://stackoverflow.blog/2017/09/06/incredible-growth-python/

Survey, S. (2023, May 1). *Developer Survey*. Stack Overflow Survey. https://survey.stackoverflow.co/2023/#most-popular-technologies-language

Zaidi, N. (2023, October 11). *Stack Overflow Survey 2023 Revealed*. Walturn. https://www.walturn.com/insights/stack-overflow-survey-2023-revealed

Buz, S. (2023, June 8). *Role of Python in Artificial Intelligence.* LinkedIn Blog. https://www.linkedin.com/pulse/role-python-artificial-intelligence-skillbuztrainings/

Blogger, B. (2023, December 14). *Denetimli ve Gözetimli Öğrenme Nedir? Supervised Learning Genel Bakış.* Bulutistan. https://bulutistan.com/blog/denetimli-ve-gozetimli-ogrenme-nedir-supervised-learning-genel-bakis/

Blogger, A. (2023). *Denetimli ve Denetimsiz Öğrenme Arasındaki Fark Nedir?* Amazon AWS. https://aws.amazon.com/tr/compare/the-difference-between-machine-learning-supervised-and-unsupervised/

Öztürk, M. (2022, April 13). *Python İle Sınıflandırma Analizleri Rastgele Orman Algoritması.* Miraç Öztürk Blog. https://miracozturk.com/python-ile-siniflandirma-analizleri-rastgele-orman-random-forest-algoritmasi/

Blogger, A. (2023). *Derin Öğrenme Nedir?* Amazon AWS Blog. https://aws.amazon.com/tr/what-is/deep-learning/

Blogger, O. (2023). *Derin Öğrenme Nedir?* Oracle Türkiye Blog. https://www.oracle.com/tr/artificial-intelligence/machine-learning/what-is-deep-learning/

Öğündür, G. (2019, November 9). *Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall) yada F1 Score?* Medium Blog. https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1l%C4%B1k-recall-ya-da-f1-score-300c925feb38

Şirin, E. (2017, July 2). *Hata Matrisi (Confusion Matrix) Yorumlama.* Veri Bilimi Okulu Blog. https://www.veribilimiokulu.com/hata-matrisini-confusion-matrix-yorumlama/

Ereken, Ö. F. (2022, June 3). *Denetimli Öğrenme Değerlendirme Ölçütleri.* Medium Blog. https://medium.com/@omereken/denetimli-%C3%B6%C4%9Frenme-de%C4%9Ferlendirme-%C3%B6l%C3%A7%C3%BCtleri-e87b2946d186

Maybe, M. (2022, October 15). *Can an AI Learn To Identify "AI Art"* Medium Blog. https://medium.com/@matthewmaybe/can-an-ai-learn-to-identify-ai-art-545d9d6af226

Zhong, N., Xu Y., Li S., Qian, Z. & Zhang, X. (2024, March 7). *PatchCraft: Exploring Texture Patch for Efficient AI-Generated Image Detection.* Cornell University. https://doi.org/10.48550/arXiv.2402.01123

Klingler, N. (2024). *How to Detect AI-Generated Content.* Viso Blog. https://viso.ai/deep-learning/ai-generated-content-detection/

Hsu, T. & Thompson, S. A. (2023, June 28). *How Easy Is It to Fool AI-Detection Tools?* The New York Times. https://www.nytimes.com/interactive/2023/06/28/technology/ai-detection-midjourney-stable-diffusion-dalle.html

Blogger, S. (2024, March 21). *How to Detect AI-Generated Images or Deepfake Videos.* Skimai. https://skimai.com/how-to-detect-ai-generated-images-or-deepfake-videos/

Baraheem, S. S. & Nguyen, T. V. (2023, August 10). *AI vs. AI: Can AI Detect AI-Generated Images?* Journal of Imaging. https://doi.org/10.3390/jimaging9100199

Epstein, D. C., Jain, I., Oliver, W. & Zhang, R. (2023, March 1). *Online Detection of AI-Generated Images.* Adobe Inc. https://openaccess.thecvf.com/content/ICCV2023W/DFAD/papers/Epstein_Online_Detection_of_AI-Generated_Images__ICCVW_2023_paper.pdf

Gillham, J. (2024, February 2). *Do AI Image Detectors Work? Accuracy Study.* Originality AI. https://originality.ai/blog/do-ai-image-detectors-work-accuracy-study

Chen, J., Yao, J. & Niu, L. (2024, April 20). *A Single Simple Patch is All You Need for AI-Generated Image Detection.* Cornell University. https://doi.org/10.48550/arXiv.2402.01123

Lu, L. & D. (2023). *Sentry-Image: Detect Any AI-Generated Images.* GitHub Repos. https://github.com/Inf-imagine/Sentry

Rodriguez F. M., Mojon, R. G. & Barciela, M. F. (2023, November 23). *Detection of AI-Created Images Using Pixel-Wise Feature Extraction and Convolutional Neural Networks.* National Library of Medicine. https://doi.org/10.3390/s23229037

Diaz, M. (2023, September 1). *Google's New Tool Can Detect AI-Generated Images, But It's Not That Simple.* ZDNet. https://www.zdnet.com/article/googles-new-tool-can-detect-ai-generated-images-but-its-not-that-simple/

Agarwal, S. (2023, June 14). *How To Detect AI-Generated Text and Photos.* Zapier Blog. https://zapier.com/blog/ai-content-detection/

Aktuğ, B. (2023). *Kombinasyon Yöntemleri.* Ankara Üniversitesi. https://acikders.ankara.edu.tr/pluginfile.php/2707/mod_resource/content/1/JFM212_11_Numpy.pdf

Walt, S., Schönberger, J. L., Iglesias, J. N., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E. & Yu, T. (2014, July 23) *Scikit-Image: Image Processing in Python.* Stellenbosch University, University of North Carolina, Victorian Life Sciences Computation Initiative, Princeton University, Mayo Clinic, AICBT Ltd, Joint Unit CNRS, Enthought Inc. https://arxiv.org/pdf/1407.6245

Şirin, E, (2017, December 19). *Ensemble Yöntemler (Topluluk Öğrenmesi): Basit Teorik Anlatım ve Python Uygulama.* Veri Bilimi Okulu Blog. https://www.veribilimiokulu.com/ensemble-yontemler-topluluk-ogrenmesi-basit-teorik-anlatim-ve-python-uygulama/

Koçulu, B. (2021, January 11). *Scikit-Learn Yapısı Kolay Açıklamalı Anlatım (Örneklerle).* Medium Blog. https://burcukoculu.medium.com/scikit-learn-yap%C4%B1s%C4%B1-kolay-a%C3%A7%C4%B1klamal%C4%B1-anlat%C4%B1m-%C3%B6rneklerle-afbbbb4593e5

Sağlamtunç, K. S. (2020, December 17). *AI-Artificial Intelligence / Yapay Zeka.* DM Danışmanlık Mühendislik Ltd. Şti. https://www.mmo.org.tr/sites/default/files/users/zeynep/AI%20-%20ARTIFICIAL%20INTELLIGENCE%20-%20YAPAY%20ZEKA.pdf

Blogger, İ. (2023, August 1). *Karmaşıklık Matrisi (Confusion Matrix).* Medium Blog. https://medium.com/@irem42/s%C4%B1n%C4%B1fland%C4%B1rma-problemlerinde-ba%C5%9Far%C4%B1-de%C4%9Ferlendirme-44fa01689e4e

Bilen, B. (2021, February 4). *Karışıklık Matrisi (Confusion Matrix).* Medium Blog. https://burhanbilen.medium.com/kar%C4%B1%C5%9F%C4%B1kl%C4%B1k-matrisi-confusion-matrix-990dfc718653

Ekler

Projenin Veri Setinde Kullanılan Görseller