Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 36, S. 32-35, Mayıs 2022

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 36, pp. 32-35, May 2022

Research Article

Detecting Sign Language from Hand Gestures and Translating it into Text

Pinar Kirci[1*], Burcin Berk Durusan[2], Baha Ozsahin[3]

[1*] Bursa Uludag University, Faculty of Engineering, Departmant of Computer Engineering, Bursa, Turkey, pinarkirci@uludag.edu.tr
[2] Bursa Uludag University, Faculty of Engineering, Departmant of Computer Engineering, Bursa, Turkey
[3] Bursa Uludag University, Faculty of Engineering, Departmant of Computer Engineering, Bursa, Turkey

(1st International Conference on Engineering and Applied Natural Sciences ICEANS 2022, May 10-13, 2022)

(**DOI:** 10.31590/ejosat.1097389)

ATIF/REFERENCE: Kirci, P., Berk Durusan, B. & Ozsahin, B. (2022). Detecting Sign Language from Hand Gestures and Translating it into Text. *European Journal of Science and Technology*, (36), 32-35.

Abstract

The sign language recognition project to be designed is aimed to be realized in an optimized way using up-to-date technologies. The machine learning part of the project will be done over TensorFlow using Keras and Sklearn. TensorFlow was chosen considering the possibility of moving the project to a mobile environment in the future. The object recognition method to be used was chosen as MediaPipe Holistic.

Keywords: Real Time Sign Language Recognition, Machine Learning, Deep Learning, Object Recognition, Computer Vision

El Hareketlerinden İşaret Dilini Algılayıp Yazıya Dönüştürme

Öz

Tasarlanacak işaret dili tanıma projesinin güncel teknolojiler kullanılarak optimize biçimde gerçeklenmesi amaçlanmıştır. Projenin makine öğrenmesi bölümü Keras ve Sklearn kullanılarak TensorFlow üzerinden yapılacaktır. TensorFlow, ilerleyen aşamalarda projeyi mobil bir ortama taşıma ihtimali göz önünde bulundurularak seçilmiştir. Kullanılacak nesne tanıma yöntemi MediaPipe Holistic olarak seçilmiştir.

Anahtar Kelimeler: Gerçek Zamanlı İşaret Dili Tanıma, Makine Öğrenmesi, Derin Öğrenme, Nesne Tanıma, Bilgisayarla Görü

[*] Corresponding Author: pinarkirci@uludag.edu.tr

1. Giriş

Görsel kanalı kullanarak karşı taraf ile iletişim kurmak için kullanılan dillere işaret dili denir. Yüzdeki hareketlerin ve mimiklerin de yardımıyla birlikte el hareketleriyle ifade edilirler. İşaret dillerinin de günümüzde insanların kullandığı konuşma dilleri gibi kendilerine ait kelime dağarcıkları ve dil bilgisel yapıları vardır ve bu durum işaret dillerini doğal dil kategorisine sokmaktadır.

Duyma engelli bireylerin içinde bulunduğu toplumların hepsinde doğal olarak duyma engelliler için bir iletişim yolu olarak gelişmiştir. Genellikle sağır insanlar tarafından kullanılan işaret dili iletişim yöntemi aynı zamanda farklı sağlık sorunları sebebiyle sesli iletişim kuramayan insanlar ya da yakın çevresinde veya aile üyelerinden birisinde işitme engeli bulunan işiten bireyler tarafından da kullanılmaktadır.

Dünyada kaç tane işaret dilinin bulunduğu bilinmemekle birlikte hemen hemen her ülkenin kendisine ait bir işaret dili bulunmaktadır. Bunlardan bazıları resmi olarak kabul edilirken bazılarının herhangi bir resmiyet statüsü bulunmamaktadır.

Ko tarafından teklif edilmiş sistem, insanların vücutlarındaki çeşitli anahtar noktaları belirlemek ve bu el yüz ve poz anahtar noktalarını kullanarak işaret dili çevirisi yapmaya odaklanmıştır. İlgili el yüz ve poz anahtar noktalarını görüntüden çıkarmak için OpenPose kütüphanesi kullanılmıştır. OpenPose, gerçek zamanlı birden fazla kişiyi algılayabilen bir anahtar nokta detektörü görevi görmektedir. Yapılmış anahtar nokta sonrasında RNN (tekrarlayan sinir ağları), LSTM (uzun kısa süreli bellek) ve GRU (kapılı tekrarlayan birimler) kullanılarak çeviri işlemi gerçekleştirilmiştir [1].

Camgöz, RWTH-PHOENIX veri setinin bir uzantısı olan PHOENIX14T'yi sürekli işaret dili çevirisi için kullanılmak üzere oluşturdu [2]. Çalışmada RNN kullanımının yeterli olmadığı durumlar için Evrişimsel Sinirsel Ağları (CNN) ve dikkat tabanlı sinirsel makine çeviri metotlarını (NMT- Neural Machine Translation) birlikte kullanan bir yapıyı önerdi. Önerdiği yapıda makine çevirisine kaynak ve hedef dizilerin birbirlerine tokenizasyonu ve bu tokenlerin bir uzaya yansıtılması ile başlanır. Bu metodu kullanmaktaki temel amaç her kelimenin birbirine eşit uzaklıkta bulunduğu seyrek haldeki one-hot vektörleri, daha yoğun bir hale getirmektir. İlgili tokenlerin elde edilmesi için Camgöz 2D CNNleri kullanmıştır [3].

Konstantinidis, işaret dili çevirisini Ko gibi eklem anahtar noktalarını kullanarak gerçekleştirmeyi planlamıştır. Geliştirilen model mevcut olarak önceden eğitilmiş ImageNet VGG-19 ağının yanında CNN ve LSTM kullanmaktadır önceden eğitilmiş bir ImageNet VGG-19 ağı üzerinde evrişim katmanları çalıştırarak elde etmiştir. Sinir ağının üzerinde çalıştığı LSA64 Arjantin işaret dili özelinde en sık kullanılan 64 el işaretinin 10 farklı kişi tarafından yapılmasıyla oluşturulmuştur.

Eğitim aşamasında, veri seti rastgele biçimde %80 eğitim, %20 test seti olarak bölünmüştür. Bu işlem 5 kere tekrarlanıp, bütün döngüler arasındaki en iyi sonuç veren model seçilmiştir [4].

Hosain, çalışmasında Amerikan işaret dili üzerinde çalışırken el işaretlerini yakalamak için elle işaretlenmiş el görselleri ve yüksek özgüven değerli tahminleri CNN eğitmek için kullanmıştır. Sürekli hareketleri algılamak için eğitilen CNN tarafından yapılan yerleştirmeler üzerinde eğitim gören RNN tarafından yapılmıştır.

Yapılan çalışmada derin CNN eğitimi için tekrarlı bir eğitim modeli önerilmiştir. Bunun yanında ileride yapılacak çalışmalara yardımcı olmak üzere GMU-ASL51 veri setinin her kesiti için el işareti işaretlemesi yapılmıştır [5].

Zhang, sürekli işaret dili çevirisi problemi özelinde pekiştirmeli öğrenme (reinforcement learning) ve transformer kullanmayı önermiştir. Gözetimli öğrenmenin (supervised learning) yaratabileceği çeşitli hatalardan kaçınılması için transformerı direkt olarak kelime hata oranı (word error rate (WER)) gibi metrikler üzerinden pekiştirmeli öğrenmeye tabi tutulması uygun bulunmuştur. Önerilen model, işaret dili görüntülerinden elde edilecek özellikleri 3D-ResNet kullanarak çıkarttıktan sonra çıktıların transformera beslenmesi ve transformer üzerinde pekiştirmeli öğrenme gerçekleşmesi üzerine kurulmuştur [6].

2. Materyal ve Metot

2.1. Sunulan Metot

Proje ile ilgili diğer konu olan bilgisayarlı görü, görüntüyü yeniden işlemeyi, görüntüyü kesmeyi, 3D bir sahneyi bu sahnenin 2D görüntüsünden anlamayı açıklamaya çalışan disiplinler arası bilimsel bir alandır. Bu alanda bilgisayar yazılımı ve donanımı kullanılarak insan görüşü modellenmeye ve taklit edilmeye çalışılır.

Bilgisayarlı görü; görüntü işleme, örüntü tanıma, fotogrametri alanlarıyla tam olarak örtüşmektedir. Fakat bilgisayarlı görü ve görüntü işlemeyi karşılaştırdığımızda, görüntü işlemede resimden resme bir çeviri olduğu görülmektedir. Yani giriş datasının da çıkış datasının da bir resim olduğu görülmektedir. Diğer taraftan bilgisayarlı görü ise objelerin, onların görüntülerinden, anlamlı bir açıklamasını çıkarmayı hedefler. Yani bilgisayarlı görünün çıkış bilgisi görüntünün açıklamasıdır.

İşaret dilini tanımak için kullanılan Python kaynak kodunu çalıştırmak için kullanılacak görüntü işleme ve makine öğrenmesi kütüphaneleri ve yöntemleri incelenmiştir. Projenin çalıştırılması için Google Colab tercih edilmiştir. Google Colab tercih edilmesinin sebebi, takım üyelerinin paralel çalışmasını desteklemesi, Google'ın sağladığı bulut üzerinden hesaplama imkanları sayılabilir. Projenin el tanıması için kullanılan yöntem olarak Google tarafından yaratılmış MediaPipe Holistic frameworkü kullanılmıştır. Sonraki aşamalarda el tanıma işlemini gerçekleştirecek kaynak kod oluşturulup mevcut MediaPipe Holistic kullanan program ile arasında karşılaştırma yapılacaktır. Bunların yanında projenin makine öğrenmesi kısmı için kullanılacak makine öğrenmesi platformları araştırılmıştır. Bunlardan sık kullanılan iki platform karşılaştırılıp projenin gereklerine uygun olan seçilmiştir. Bu platformlar;

- TensorFlow,
- PyTorch olarak belirlenmiştir.

Bu projede TensorFlow ile çalışılması öngörülmektedir.

PyTorch nesne yönelimli programlama stili açısından oldukça ünlü durumdadır. Örnek olarak, özel bir model veya özel bir veri seti oluşturulmak istendiğinde sıklıkla varsayılan PyTorch kütüphanelerini miras alan bir sınıf oluşturulur ve bundan sonra uygulanmak istenen asıl metoda adapte edilir. Sonuç olarak

PyTorch geliştiriciye bir yapı sunar fakat projenin kod uzunluğu açısından çok fazla satır sayısına sahip olmasına sebep olur.

Bir diğer tarafta TensorFlow kullanılırken sıklıkla Keras da kullanılır. Görüntünün bölümlere ayrılması, obje tanıma veya gözetimli görüntü sınıflandırılması gibi işler yapılırken Keras projeleri PyTorch projelerine göre satır sayısı açısından çok daha küçük boyutlu olur. Bu durum başlangıç ve orta seviye geliştiriciler için kodu okuma, anlama ve değişiklik yapabilme kapasitesi açısından oldukça iyidir. [7]

Projenin nesne tanıma ve takibi konusunda örnek olması açısından MediaPipe Holistic kullanılarak Google Colab üzerinde çalışan bir program yaratılmıştır. Bu kod, şekil 1'de gösterildiği gibi kameradan aldığı görüntü doğrultusunda görüntü üzerinde gerçek zamanlı olarak el, yüz ve poz bilgilerini bir dizi nokta olarak gösterebilmektedir.

Şekil 1. MediaPipe Holistic kullanılarakOpenPose kullanılarak anahtar nokta çıkarımı [2]

3. Araştırma Sonuçları ve Tartışma

Sorunun çözümüne yönelik yapılan araştırmalar sonucunda problemin çeşitli hareketleri tanımasını gerektirecek durumlar olduğu tespit edilmiştir. Bu sebeple çözümde kullanılacak sinir ağı modelinin, tekrarlayan bir sinir ağı modeli olmasına karar verilmiştir. Verilen bu karar doğrultusunda çeşitli sinir ağı modelleri incelendikten sonra LSTM (Long short-term memory) kullanma kararı alınmıştır.

Verinin toplanması, OpenCV ve MediaPipe Holistic kullanan bir script yazılarak tamamlanmıştır. Toplanan veriler, verisi toplanacak her harf özelinde 30 frameden oluşmaktadır. Veri gizliliği, dosya boyutu ve verinin daha kolay işlenebilmesi için toplanan veri fotoğraf şeklinde değil numpy arrayleri halinde tutulmuştur. İlgili numpy arrayleri MediaPipe Holistic tarafından işaretlenen anahtar noktalardan oluşmuştur. Verilerin toplanmasının ardından daha önceden yaratılmış script yardımıyla oluşturulmuş olan dosyalama sistemine veriler kaydedilmiştir.

Verilerin kayıt işlemi bittikten sonra ilgili veriler, numpy tarafından işlenmek üzere sisteme yüklenir. Elde edilen veriye yönelik veriler ve etiketleri ayrıştırıldıktan sonra sonrasında test için kullanılacak veri rastgele olarak Sklearn'ün sağladığı train_test_split fonksiyonu kullanılarak ayrılır.

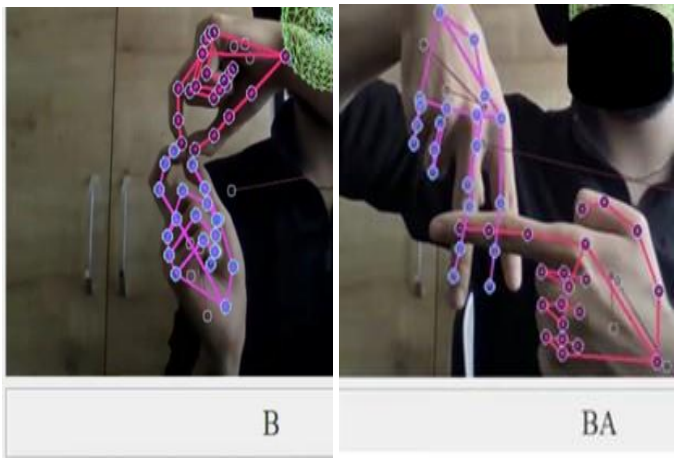
Modele beslenecek veri ön işlemeye tabi tutulduktan sonra model ardışık olarak tanımlanır. Keras kütüphanesinin sağladığı katmanlardan yararlanarak 64 birimlik LSTM, 128 birimlik iki LSTM, 64 birimlik son LSTM katmanından sonra bir 64 birimlik bir de 32 birimlik Yoğun katman oluşturulmuştur bu katmanlardan sonra çıktı katmanının birim sayısını tanınması gereken harf veya kelime sayısı tarafından belirlenmiştir.

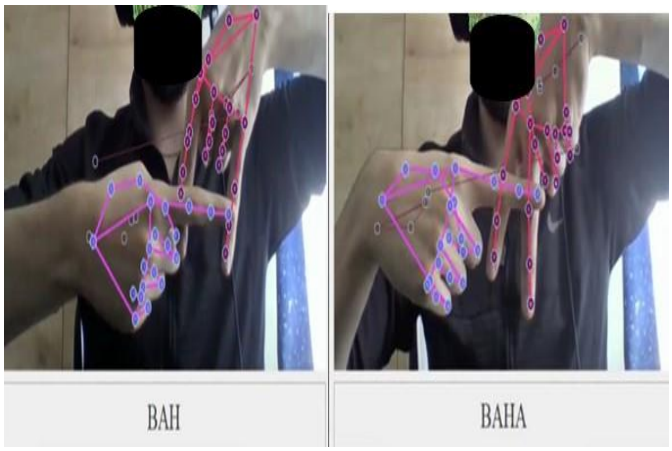
İlk 3 katman LSTM geri besleme yapmaktadır. Son LSTM katmanı ise sonrasında yoğun katmanlara besleme yapacağı için geri besleme yapmaz. Çıktı katmanından önce bulunan katmanların hepsi ReLU aktivasyon fonksiyonu kullanmıştır. Çıktı katmanında ise çıktıların normalizasyonunun sağlanması adına softmax aktivasyon fonksiyonu kullanılmıştır.

Modelin eğitimi sırasında eğitimin verimini arttırmak için Keras tarafından sağlanan callback fonksiyonlarının çözüm özelinde düzenlenmiş halleri kullanılmıştır. Model, kategorik isabetlilik değişkenine bağlı olarak değişimin arttığı sürece eğitimine devam etmiştir. Kategorik isabetlilik değişkeni düşmeye başladığı adımdan itibaren 8 adım boyunca yükselme göstermediği takdirde eğitim durdurulmuş ve oluşturulmuş script yardımıyla eğitim sırasında en iyi sonuca ulaşılmış ağırlıklar kayıt edilmiştir.

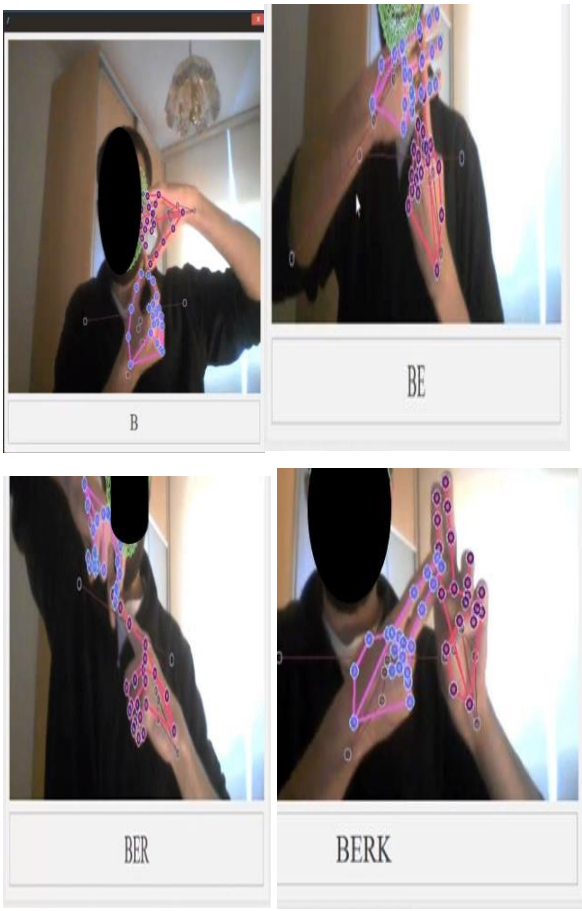
Eğitimin sonuçlarını kontrol etmek için eğitimin ardından test verisi üzerinde isabetlilik kontrolü yapılır.

Testin gerçek zamanlı yapılabilmesi ve çıktıların görüntülenebilmesi için şekil 2 ve şekil 3'de gösterildiği gibi Tkinter kullanılarak bir kullanıcı arayüzü oluşturulmuştur. Gerçek zamanlı test sırasında softmax fonksiyonundan gelen çıktıları 0.8'den büyük olan veriler doğru kabul edilip ekranın altındaki bölmeye yazılmıştır.





Şekil 2. Gerçek zamanlı test sırasında görülebilecek çıktıların örnekleri



Şekil 3. Gerçek zamanlı test sırasında görülebilecek çıktıların örnekleri

4. Sonuç

Projenin Google Colab ortamında geliştirilmiştir. Google Colab ortamında yaratılan kodun çalıştırılması için Jupyter Notebook kullanılarak yerel ortam bağlantısı kurulmuştur. MediaPipe Holistic frameworkü kullanılarak oluşturulan el, yüz ve poz tanıma kodu ile oluşturulmuş anahtar veriler başarılı biçimde çözüme aktarılabilmiştir. Aktarılmış veriler, oluşturulmuş olan makine öğrenmesi modeli tarafından verimli biçimde kullanılmıştır ve modelin eğitimi ardından başarılı biçimde test gerçekleştirilmiştir. Çalışmanın ilerleyen aşamalarında oluşturulmuş model için sesli okuma özelliği getirilecektir. Bunun yanında MediaPipe Holistic kullanmayan bir el tanıma modeli yaratılacaktır. Yaratılmış farklı el tanıma modellerine sahip çözümlerin işaret dili tanıma konusundaki verimleri karşılaştırılacaktır. Bu işlemlerin sonrasında ortaya koyulacak model karşılaştırması sonucunda projenin tamamlanması beklenmektedir.

Kaynakça

[1] Ko, S.-K., Kim, C. J., Jung, H., & Cho, C. (2019). Neural Sign Language Translation Based on Human Keypoint Estimation. Applied Sciences, 9(13), 2683. doi:10.3390/app9132683

[2] RWTH-PHOENIX-2014-T veri seti, https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX-2014-T/

[3] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural Sign Language Translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[4] Konstantinidis, D., Dimitropoulos, K., & Daras, P. (2018). Sıgn Language Recognıtıon Based On Hand And Body Skeletal Data. 2018- 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON). doi:10.1109/3dtv.2018.8478467

[5] Hosain, A. A., Santhalingam, P. S., Pathak, P., Rangwala, H., & Kosecka, J. (2020). FineHand: Learning Hand Shapes for American Sign Language Recognition. 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020). doi:10.1109/fg47880.2020.00062

[6] Zhang, Z., Pu, J., Zhuang, L., Zhou, W., & Li, H. (2019). Continuous Sign Language Recognition via Reinforcement Learning. 2019 IEEE International Conference on Image Processing (ICIP). doi:10.1109/icip.2019.8802972

[7] Pytorch vs Tensorflow 2021, https://towardsdatascience.com/pytorch-vs-tensorflow-2021-d403504d7bc3