



Teknoloji Fakültesi

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Türk İşaret Dili Tanıma ve Metne Dönüştürme: İngilizce Çeviri Destekli Uygulama

BİTİRME PROJESİ 2.

ARA RAPORU

Bilgisayar Mühendisliği Bölümü

DANIŞMAN

Prof. Dr. Serhat Özekes

İSTANBUL, 2025

MARMARA ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencileri Ayça Nilay ÖZOĞUL, Merve HASASLARI, Serra K.TALASLI tarafından “Türk İşaret Dili Tanıma ve Metne Dönüştürme: İngilizce Çeviri Destekli Uygulama” başlıklı proje çalışması, xxx tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Dr. Öğr. Üyesi xxx xxx
Marmara Üniversitesi
Prof. Dr. Xxx xxx
Marmara Üniversitesi
Prof. Dr. Xxx xxx
Marmara Üniversitesi

(Danışman)

(Üye)

(Üye)

(İMZA).....

(İMZA).....

(İMZA).....

ÖNSÖZ

Proje çalışmamız süresince karşılaştığım bütün problemlerde, sabırla yardım ve bilgilerini esirgemeyen, tüm desteğini sonuna kadar yanımda hissettiğim değerli hocalarım, sayın Dr. Öğr. Üyesi Xxx xxx ve sayın Prof. Dr. Xxx xxx' a en içten teşekkürlerimi sunarım.

Bu proje çalışması fikrinin oluşması ve ortaya çıkmasındaki önerisi ve desteğinden dolayı değerli hocam Dr. Öğr. Üyesi Xxx xxx' a teşekkür ederim.

Proje çalışmam sırasında maddi ve manevi desteklerini esirgemeyen okul içerisinde ve okul dışında her zaman yanımda olan değerli çalışma arkadaşlarım ve hocalarım Doç. Dr. Xxx xxx ve Dr. Öğr. Üyesi ' xxx xxx a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

1. GİRİŞ
 - 1.1. Proje Çalışmasının Amacı ve Önemi
2. LİTERATÜR TARAMASI
 - 2.1. Türk İşaret Dili Tanıma Performansının İyileştirilmesi İçin Transfer Öğrenme Yaklaşımı
 - 2.2. Türk İşaret Dili Tanımada Video Transformer Network Kullanımı
 - 2.3. Convolutional Neural Network ve LSTM ile Türk İşaret Dili Tanıma
 - 2.4. İki Akışlı S3D ile Mobil Tabanlı İşaret Dili Tanıma
 - 2.5. BosphorusSign22k Veri Seti ile Türk İşaret Dili Tanıma
 - 2.6. Türk İşaret Dili Tanımada Çok Modlu Veri Seti Kullanımı
3. METOT VE YÖNTEMLERİ
 - 3.1. Veri Seti
 - 3.2. Veri Ön İşleme Adımları
 - 3.3. İskelet Çıkarımı Yöntemleri
 - 3.4. Mediapipe ile İskelet Çıkarımı
 - 3.5. Model Mimarisi
 - 3.5.1. Long Short-Term Memory (LSTM)
 - 3.5.2. Model Birleşimi
 - 3.5.3. Eğitim ve Test Aşamaları
4. BULGULAR VE TARTIŞMA
 - 4.1. Veri Seti Özellikleri
 - 4.2. Veri Ön İşleme
 - 4.3. İskelet Çıkarımı
 - 4.4. Model Mimarisi
5. Kullanılan Teknolojiler
6. Sistem Mimarisi ve Akış
 - 6.1. Kullanıcı Arayüzü (Frontend)

6.2. Sunucu ve Model Entegrasyonu (Backend)

6.3. Şifre Sıfırlama ve E-posta Entegrasyonu

7. Gerçekleştirilen Teknik Çalışmalar

8. Yapılacak Çalışmalar

KAYNAKLAR

ÖZET

Türk İşaret Dili Tanıma ve Metne Dönüştürme: İngilizce Çeviri Destekli Uygulama

Bu proje çalışmasında, Türk İşaret Dili (TİD) kullanılarak işaret dili tanıma ve metne çevirme işlemleri gerçekleştirilmiştir. Geliştirilen sistem, gerçek zamanlı olarak işaret dili hareketlerini algılayarak bu hareketleri metin formatına dönüştürme yeteneğine sahiptir. Amacı, işaret dili kullanıcıları ile Türkçe bilmeyenler arasında iletişimi kolaylaştırmaktır.

Metne dönüştürülen işaret dili ifadeleri, ardından İngilizce'ye çevrilecektir. Sonuç olarak, işaret dili tanıma ve çeviri süreçlerinin daha hızlı ve doğru bir şekilde gerçekleştirilmesi sağlanarak, işaret dili kullanıcılarının günlük yaşamlarında daha fazla erişilebilirlik sunulacaktır.

Haziran, 2025

Öğrenciler

ABSTRACT

Turkish Sign Language Recognition and Text Conversion: English Translation Supported Application

In this study, sign language, which is an essential communication tool for individuals with hearing and speech impairments, is examined in detail. Sign language is not just a visual communication method but a complex system that includes hand shapes, movements, orientations, speed, facial expressions, body posture, and gestures. These elements work together to convey meaning and context, forming a unique linguistic structure. However, the lack of widespread knowledge and proficiency in sign language creates significant challenges for hearing-impaired individuals, particularly in accessing education, healthcare, employment, and social interaction.

Linguistic studies have revealed that sign languages exhibit unique characteristics that differ from spoken languages and vary between countries and cultures. For example, Turkish Sign Language (TİD), which is commonly used by hearing-impaired individuals in Turkey, has its own grammatical structure distinct from both spoken Turkish and other sign languages worldwide. This diversity among sign languages often results in communication difficulties, emphasizing the need for systems capable of automatically translating between different sign languages.

Recent advancements in artificial intelligence (AI) and deep learning (DL) have introduced new opportunities for the automatic recognition and translation of sign languages. Modern sign language recognition systems utilize computer vision, image processing, and natural language processing (NLP) techniques to analyze hand movements and facial expressions, converting them into written or spoken text. These systems hold significant potential to eliminate communication barriers faced by hearing-impaired individuals, thus promoting social inclusion.

Various approaches have been developed to recognize sign languages automatically, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer models. Despite these efforts, challenges remain in capturing spatial

and temporal features simultaneously and accurately detecting subtle variations in hand shapes and facial expressions. Addressing these challenges requires robust and well-generalized models that can operate efficiently in real-world scenarios.

The primary goal of this research is to develop an innovative model for automatic recognition of Turkish Sign Language (TİD) using advanced deep learning techniques. The study aims to achieve high recognition accuracy while maintaining real-time performance, allowing effective interaction between hearing-impaired individuals and digital platforms. The model will integrate advanced methods such as 3D convolutional networks, attention mechanisms, and multimodal data fusion to accurately capture the spatial and temporal aspects of TİD gestures. Additionally, the use of transfer learning and data augmentation techniques will enhance the model's generalization and reduce overfitting.

Furthermore, the project will include an English translation feature, allowing recognized Turkish Sign Language to be converted into English text. This feature aims to bridge communication between Turkish sign language users and English-speaking individuals, promoting cross-linguistic accessibility.

The expected outcome of this research is to develop a robust and efficient application named *"Turkish Sign Language Recognition and Text Conversion: English Translation Supported Application"*. This system will be applicable in various real-world contexts, such as public services, healthcare, education, and social media platforms. By minimizing the need for human interpreters, the system aims to empower hearing-impaired individuals to communicate more freely and independently, promoting social inclusion and equality.

June, 2025

Students

KISALTMALAR

CUDA : Compute Unified Device Architecture

GPU : Graphical Processing Unit

CPU : Central Processing Unit

AUTSL : Large Scale Dataset for Turkish Sign Language

YOLOv8 : You Only Look Once versiyon 8

CNN : Konvolüsyonel Sinir Ağı (Convolutional Neural Network)

LSTM : Uzun Kısa Süreli Bellek (Long Short-Term Memory)

Mediapipe : Google tarafından geliştirilen bir makine öğrenimi çözüm paketi

NLP : Neuro Linguistic Programming

SL-GCN: Spatial-Temporal Graph Convolutional Network, işaret dili tanıma gibi zaman serisi verilerinde uzamsal ve zamansal ilişkileri öğrenmek için kullanılan bir derin öğrenme modeli.

DANN: Domain-Adversarial Neural Network, farklı veri kümeleri (kaynak ve hedef alanlar) arasında öğrenmeyi genelleştirmek için kullanılan bir adaptasyon tekniği.

DSBN: Domain-Specific Batch Normalization, farklı veri dağılımlarına sahip veri kümeleri için ayrı ayrı normalizasyon yapan bir yöntem.

JAN: Joint Adaptation Network, özellik dağılımlarını hizalayarak transfer öğrenmeyi iyileştiren bir yöntem.

MCC: Minimum Class Confusion, sınıf belirsizliğini en aza indirgeyerek modelin daha iyi genelleşmesini sağlayan bir yöntem.

RGB: Red, Green, Blue. Görüntü ve video verilerinde kullanılan temel renk uzayı.

BLSTM: Bidirectional Long Short-Term Memory, dizisel veriler için çift yönlü bilgi işleyebilen gelişmiş bir RNN modeli.

VTN (Video Transformer Network): Video tabanlı görevlerde uzun vadeli bağımlılıkları modellemek için Transformer mimarisini kullanan bir ağ türü.

S3D: Separable 3D Convolution, 3D evrişimleri ayrıştırarak hesaplama maliyetini düşüren bir video analiz yöntemi.

IDT (Improved Dense Trajectories): Hareket tabanlı özellik çıkarımı yapan bir video analizi yöntemi.

3D Residual Networks (MC3): 2D ve 3D evrişimleri birleştirerek hareket bilgisini yakalayan bir sinir ağı modeli.

GCN (Graph Convolutional Network): Grafik yapılarındaki verileri işlemek için kullanılan bir sinir ağı modeli.

I3D (Inflated 3D ConvNet): 2D konvolüsyonları 3D'ye genişleterek video tabanlı derin öğrenme modellerini geliştiren bir yöntem.

CSL (Chinese Sign Language Dataset): Çince İşaret Dili tanıma için kullanılan büyük ölçekli bir veri seti.

ŞEKİL LİSTESİ

Şekil 1.1 Yolo algoritmasının görselleştirilmesi	2
Şekil 1.2 MediaPipe algoritmasının görselleştirilmesi	8

TABLO LİSTESİ

Tablo 1.1 Veri setinin özellikleri

6

1. GİRİŞ

İşaret dili, duyma ve konuşma engelli insanların dış dünyayla iletişim kurması için en önemli araçlardan biridir. Çoğunlukla görsel bir iletişim tarzı olarak değerlendirilse de işaret dili yalnızca el hareketlerinden ibaret değildir; el yönleri ve hızları, yüz ifadeleri, postür ve mimikler gibi çeşitli komponentler, bu dili oluşturan temel unsurlar arasında bulunmaktadır [1]. İşaret dilini etkin şekilde kullanabilen birey sayısının kısıtlı olması, işitme engelli toplulukların eğitim, istihdam, sağlık hizmetleri ve sosyal etkileşim gibi temel sahalarda dezavantajlı konuma düşmelerine sebep olmaktadır (World Federation of the Deaf, 2023).

Dilbilimsel çalışmalar, her dilin kendine has bir işaret dili olduğunu ve bu dillerin de kendine özgü fonolojik, morfolojik ve sentaktik özelliklere sahip olduğunu göstermektedir [3]. Türk İşaret Dili (TİD), Türkiye'deki işitme engelli bireylerin anadilidir ve diğer işaret dillerinden farklı bir dilbilgisi yapısına sahiptir [4]. Bu durum, işitme ve konuşma engelli insanların farklı diller konuşulan ülkelere seyahat etmeleri veya farklı kültürlerden bireylerle iletişim kurmaları sırasında zorluklarla karşılaşmalarına yol açmaktadır. Küresel bir işaret dili ölçütünün olmaması, yurt dışından gelen işitme engelli bireyler arasındaki iletişimi oldukça zorlaştırmaktadır. Bu bağlamda, işaret dilleri arasında otomatik çeviri yapabilecek teknolojilere duyulan gereksinim zamanla artmaktadır [5].

Son yıllarda yapay zekâ (YZ) ve derin öğrenme (DL) temelli teknolojiler, işaret dili tanıma sistemlerinin geliştirilmesinde önemli bir gelişme kaydetmiştir. Gelişmiş görüntü işleme ve doğal dil işleme yöntemleri sayesinde, kameralar aracılığıyla kaydedilen işaret dili hareketleri otomatik olarak analiz edilerek metne veya farklı dillere çevrilebilmektedir [6]. Bu tür teknolojiler, işitme engelli bireylerin iletişim duvarlarını aşmalarına, bilgiye ulaşmalarının kolaylaştırılmasına ve toplumsal yaşama daha aktif katılımlarına imkan sağlamaktadır [7].

Bu çalışmada, Türk İşaret Dili'nin (TİD) otomatik tanınmasını sağlamak amacıyla son teknoloji derin öğrenme yaklaşımlarının incelenmesi ile daha yüksek doğruluk oranına ve kullanıcı dostu yapıya sahip bir model geliştirilmesi hedeflenmektedir. Modelin, kamu ve özel sektör işlemlerinde işaret dili tercümanı ihtiyacını azaltarak işitme engelli bireylerin sosyal ve profesyonel hayata daha aktif iştirak etmesi amaçlanmaktadır. Bu

sayede, işaret dili ile iletişimin evrensel ölçekte daha erişilebilir duruma getirilmesi ve işitme engelli bireylerin uluslararası etkileşimlerinin kolaylaştırılması hedeflenmektedir.

2. LİTERATÜR TARAMASI

2. 1 Türk İşaret Dili Tanıma Performansının İyileştirilmesi İçin Transfer Öğrenme Yaklaşımı

Türk İşaret Dili tanımada kısıtlı veri setlerinin başarısını artırmak amacıyla transfer öğrenme yöntemleri uygulanmıştır. AUTSL ve BSign22k veri setleri birleştirilerek ortak bir çalışma alanı oluşturulmuştur. Çalışmada SL-GCN, DANN, DSBN, JAN ve MCC gibi modeller kullanılmıştır. Karma yöntemlerle %98,8 doğruluk oranına ulaşılmıştır [8].

2. 2 Türk İşaret Dili Tanımda Video Transformer Network Kullanımı

AUTSL veri seti üzerinde VTN modeli kullanılmıştır. İnsan vücut noktaları ve el hareketleri gibi ek bilgilerle model doğruluğu %92,92'ye ulaşmıştır. RGB verisi kullanıldığında ise doğruluk oranı %82 olarak hesaplanmıştır [9].

2. 3. Convolutional Neural Network ve LSTM ile Türk İşaret Dili Tanıma

CNN ve LSTM modelleri, geniş çaplı AUTSL veri setinde kullanılmıştır. BLSTM ve Attention mekanizmaları ile %95,95 doğruluk oranına ulaşılmıştır. Kullanıcıdan bağımsız testlerde doğruluk %62,02 olarak elde edilmiştir [10].

2. 4. İki Akışlı S3D ile Mobil Tabanlı İşaret Dili Tanıma

RGB ve iskelet verilerini kullanan İki Akışlı S3D modeli ile AUTSL veri setinde %93,75 doğruluk oranına ulaşılmıştır. LSA64 veri setinde ise %99,38 doğruluk elde edilmiştir [11].

2. 5 BosphorusSign22k Veri Seti ile Türk İşaret Dili Tanıma

BosphorusSign22k veri seti, 744 işaret içermekte olup Microsoft Kinect v2 ile kaydedilmiştir. IDT ve MC3 modelleriyle doğruluklar sırasıyla %88,53 ve %78,85 olarak rapor edilmiştir [12].

2.6 Türk İşaret Dili Tanımda Çok Modlu Veri Seti Kullanımı

AUTSL veri seti üzerinde GCN, I3D ve BLSTM gibi modellerle %98,42 (RGB

Track) ve %98,53 (RGB+D Track) doğruluk oranlarına ulaşılmıştır [13].

3. METOT VE YÖNTEMLERİ

3.1 Veri Seti

Bu çalışmada, Türk İşaret Dili (TİD) tanıma üzerine yapılan analizlerde AUTSL veri seti kullanılmaktadır. AUTSL, Türk İşaret Dili için hazırlanmış kapsamlı bir veri setidir. İçerisinde 226 farklı işaret bulunmakta olup toplamda 38.336 örnek içermektedir. Bu veri seti, 43 farklı kişinin gerçekleştirdiği işaretlerden oluşmakta olup, her işaret için ortalama 170 örnekle geniş bir çeşitlilik sunmaktadır. AUTSL, içerdiği örnek sayısı açısından, CSL veri setinden sonra en büyük ikinci veri seti olma özelliğine sahiptir [14].

Property	Description
Number of signs	226
Number of signers	43
Total samples	38,336
Number of different backgrounds	20
Mean sample per sign	169.6
RGB and depth resolution	512x512
FPS	30

Tablo 1.1 Veri setinin özellikleri

AUTSL'yi diğer büyük ölçekli işaret dili veri setlerinden ayıran en önemli özelliklerden biri, 20 farklı arka plan içermesi ve çeşitli zorlukları barındırmasıdır. Veri setindeki videolar, hem iç hem de dış mekânlarda, farklı ışıklandırma koşullarında çekilmiş olup, gerçek hayat senaryolarına uygun şekilde hazırlanmıştır. Özellikle dış mekân çekimlerinde, rüzgârla hareket eden nesneler, arka planda geçen insanlar ve değişken ışık koşulları gibi dinamik unsurlar yer almaktadır. Bu sayede, geliştirilen modellerin gerçek dünya koşullarına daha iyi uyum sağlaması ve genelleme yapabilmesi hedeflenmektedir [14].

Veri seti oluşturulurken, günlük hayatta en sık kullanılan işaretlerin seçilmesine özellikle dikkat edilmiştir. Ayrıca, el hareketlerinin çeşitliliğini dengeli bir şekilde dağıtmak ve birbirine benzeyen ancak anlamı farklı olan işaretleri de dahil etmek amacıyla titiz bir çalışma yürütülmüştür. Bu süreçte, Türk İşaret Dili eğitmenleriyle iş birliği yapılarak işaretlerin kapsamlı ve temsil edici olmasına özen gösterilmiştir. Seçilen işaretler, el şekilleri ve hareket dinamikleri açısından geniş bir yelpazeyi kapsamaktadır. Modelin farklı varyasyonları öğrenebilmesini sağlamak amacıyla çeşitlendirilmiştir.

Bazı işaretlerde eller birbirini veya yüzü kapatabilmektedir, örneğin:

Ellerin birbirini kapattığı işaretler: *Ayakkabı, Bal...*

Ellerin yüzü kapattığı işaretler: *Beklemek, Üzgün...*

Bazı işaretlerde eller derinlik yönünde hareket etmektedir, örneğin:

İtmek, Terzi...

Bazı işaretlerde sağ ve sol el çapraz pozisyonda bulunmaktadır, örneğin:

Yardım, Tehlike...

Ayrıca, veri setinde bileşik işaretler de yer almaktadır. Örneğin, *Hastane* işareti, *Doktor* ve *Bina* işaretlerinin sıralı ve ardışık şekilde yapılmasıyla oluşmaktadır. Benzer şekilde, *Yemek* ve *Ocak* işaretleri ile oluşturulan *Yemek pişirmek* bileşik işareti de veri setine dahil edilmiştir.

AUTSL veri setinde yer alan 43 işaretleyici, farklı deneyim düzeyine ve profillere sahiptir:

- 6 kişi Türk İşaret Dili eğitmeni
- 3 kişi Türk İşaret Dili tercümanı
- 1 kişi işitme engelli birey
- 1 kişi işitme engelli ebeveynlerin çocuğu (Coda - Children of Deaf Adults)
- 25 kişi Türk İşaret Dili kursiyeri
- 7 kişi veri setindeki işaretleri öğrenmiş eğitilmiş işaretleyiciler

İşaretleyicilerin 10'u erkek, 33'ü kadın olup 2'si solaktır. Veri seti, işaretlerin dağılımı açısından dengeli bir yapıya sahiptir. Ancak, bazı işaretleyicilerin örnek sayısı diğerlerine kıyasla daha fazla olabilmektedir. Bunun temel nedeni, aynı işaretleyicilerin farklı kıyafetlerle veya değişen arka planlarda birden fazla kez kayıt altına alınmış olmasıdır. Bu çeşitlilik, modelin farklı ortam ve koşullara daha iyi uyum sağlayabilmesi için bilinçli olarak eklenmiştir [14].

Sonuç olarak, AUTSL veri seti, geniş kapsamı, farklı arka plan çeşitliliği, dinamik çevresel koşulları ve dengeli işaret dağılımı ile Türk İşaret Dili tanıma alanında önemli bir kaynak niteliği taşımaktadır. Bu çalışmada, modelin eğitimi ve değerlendirilmesi süreçlerinde bu veri seti etkin bir şekilde kullanılacaktır.

3. 2 Veri Ön İşleme Adımları

Bu çalışmada, Türk İşaret Dili tanıma amacıyla kullanılan AUTSL veri setine ait video verileri belirli ön işleme adımlarından geçirilerek analiz için uygun hale getirilmiştir. Veri setindeki videoların işlenmesi sürecinde aşağıda sıralanan adımlar takip edilmiştir:

- 1. Videoların Başında ve Sonunda Boş veya Durağan Karelerin Tespiti ve Kaldırılması:** Veri setindeki videoların başlangıç ve bitiş kısımlarında yer alan boş veya durağan kareler, ardışık kareler arasındaki farklar analiz edilerek tespit edilmiştir. Bu işlemde, OpenCV kütüphanesinin **cv2.absdiff()** fonksiyonu kullanılmış ve belirlenen eşik değerinin altında kalan kareler otomatik olarak kaldırılmıştır. Böylece, videoların yalnızca anlamlı hareketler içeren bölümleri işleme alınmıştır. Böylece, yalnızca işaret dili hareketlerini içeren bölümler veri setinde tutulmuştur.
- 2. Tüm Karelerin Aynı Çözünürlüğe Ölçeklenmesi:** Veri setindeki videoların farklı çözünürlüklere sahip olması, modelin eğitimi sırasında tutarsızlıklara yol açabileceğinden tüm kareler belirli bir standart çözünürlüğe uyumlu hale getirilmiştir. Bu çalışmada, tüm videolar **224x224 piksel** boyutlarına yeniden ölçeklendirilerek veri kümesinin homojen bir giriş formatına sahip olması sağlanmıştır. Bu işlem, OpenCV kütüphanesinin **cv2.resize()** fonksiyonu kullanılarak gerçekleştirilmiştir.
- 3. Bozuk veya Eksik Videoların Tespit Edilmesi:** Veri setinde bazı videolarda eksik kare veya bozuk içerik bulunabileceği göz önüne alınarak tüm videolar **cv2.VideoCapture** yöntemiyle işleme tabi tutulmuş ve düzgün şekilde okunamayan dosyalar tespit edilerek veri kümesinden çıkarılmıştır. Bu süreçte, her video dosyasının sorunsuz bir şekilde açılıp açılmadığı ve ilk karelerinin başarıyla okunup okunmadığı dikkatlice kontrol edilmiştir.
- 4. Sıfır Süreli veya Çok Kısa Videoların Filtrelenmesi:** Veri setinde yer alan çok kısa süreli videolar, modelin yeterli bilgiyle eğitilmesini zorlaştırabileceğinden filtrelenmiştir. Her bir videonun süresi, **cv2.CAP_PROP_FPS** ve **cv2.CAP_PROP_FRAME_COUNT** kullanılarak hesaplanmış ve belirlenen eşik sürenin (örneğin 1 saniye) altında kalan videolar veri setinden çıkarılmıştır. Böylece, yalnızca

anlamli ve yeterli bilgi ieren videolar iřlenerek modelin daha verimli bir řekilde ğrenmesi saėlanmıřtır.

5. Düşük Çözünürlüklü veya Bozuk Görüntülü Videoların

Ayıklanması: Düşük çözünürlüklü veya görüntü kalitesi yetersiz olan videolar, iřaret dili hareketlerinin doėru řekilde analiz edilebilmesi iin veri setinden ıkarılmıřtır. Bu amala, videoların ilk kareleri incelenmiř ve belirlenen minimum çözünürlük (**224x224 piksel**) altında kalanlar ayıklanmıřtır. Ayrıca, bozuk kareler tespit edilerek bu videoların iřlenmesi önlenmiř ve yalnızca net ve kaliteli görüntü ieren videoların modele dahil edilmesi saėlanmıřtır.

Yukarıda belirtilen veri ön iřleme adımları sayesinde, AUTSL veri setindeki verilerin kalite ve tutarlılıėı önemli ölçüde artırılmıřtır. Bu iřlemler sonucunda elde edilen temiz ve standart hale getirilmiř veri kümesi, iřaret dili tanıma modelinin daha güvenilir ve verimli bir řekilde eėitilmesine olanak tanımaktadır.

3. 3 İskelet Çıkarımı Yöntemleri

İskelet ıkarma iřlemi iin öncelikle videolar karelere (frame) ayrılmaktadır. Çünkü iskelet ıkarma iřlemi, videolar yerine tekil kareler üzerinde alıřıldığı zaman daha doėru sonuçlar vermektedir. Bunun nedenleri řunlardır:

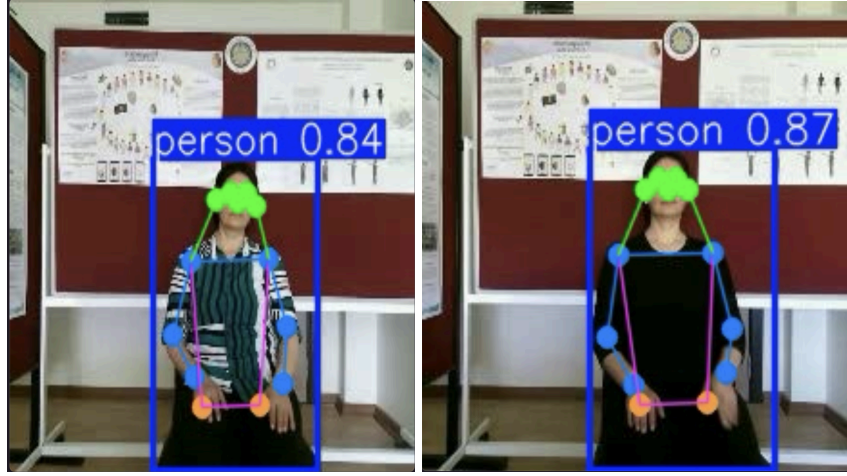
- Anlık Tespit: Video yerine her kareyi ayrı ayrı iřleyerek, hareket bulanıklılıėını ve modelin sürekli deėiřen pozisyonlara uyma zorunluluėunu ortadan kaldırır.
- Baėımsız İřleme: Tekil kareler üzerinde yapılan analizler, ardışık kareler arasındaki olası hata durumunu minimize eder.
- Paralel İřlemeye Uygunluk: Kare bazlı iřlemede, GPU veya oklu iř paracıėı kullanımıyla hızlandırılabilir.

Bu durumda iskelet ıkarma iřlemi iin farklı yöntemler denenmiř ve ařaėıdaki sonuçlara ulařılmıřtır:

YOLO ile İskelet Çıkarımı

YOLO (You Only Look Once), nesne tespiti iin kullanılan bir derin ğrenme modelidir. İřaret dili tanıma sürecinde, el hareketlerini algılamak iin YOLO'dan faydalanabileceėi düşünölmüřtür. Ancak:

- YOLO bütün vücuda odaklanırken, el ve parmak detayları yeterince iyi tespit edilmemiřtir.
- İřaret dili iin parmak hareketleri kritik öneme sahip iken, bu yöntemin kullanımı uygun görülmemektedir.



Şekil 1.1 Yolo algoritmasının görüntülenmesi

OpenPose ile İskelet Çıkarımı

OpenPose; yüz, vücut ve ellerin iskelet noktalarını çıkarabilen gelişmiş bir kütüphanedir. İşaret dili tanıma sürecinde kullanımı açısından avantajlı durumları şunlardır:

- Tüm vücut hareketlerini ve parmak detaylarını tespiti.
- İskelet bazlı hareket analizlerinde yüksek doğruluk sunmak.

Fakat, OpenPose'un kullanımı sırasında şu zorluklarla meydana gelmektedir:

- Kurulum Süreci Karmaşıktır: CUDA, cuDNN, CMake, Visual Studio gibi ek yazılımlar gerektirmektedir.
- Sürüm Uyumsuzlukları: Çeşitli donanım ve yazılım kombinasyonlarında uyumluluk sorunları yaşanmaktadır.
- Zaman Kaybı: Uzun kurulum ve yapılandırma süreçleri nedeni ile pratik kullanım açısından verimli bulunmamaktadır.

Bu nedenlerden OpenPose'un kullanımı tercih edilmemektedir.

MediaPipe Hands ve MediaPipe Body ile İskelet Çıkarımı

MediaPipe, Google tarafından geliştirilen, hafif ve GPU destekli bir kütüphanedir. İşaret dili tanıma sürecinde vücut ve el hareketleri birlikte önemli olduğundan, MediaPipe'in Hands (Eller) ve Body (Vücut) modülleri birlikte kullanılmaktadır.

Neden MediaPipe Tercih Edildi?

- Kurulumu Kolaydır: Python kütüphanesi olarak hızlıca entegre edilebilmektedir.
- GPU Desteği Sunar: Gerçek zamanlı analiz için hız avantajı sağlamaktadır.
- El ve Parmakları Detaylı Tespiti: Her el için 21 anahtar nokta belirler ve bu noktaların x, y, z koordinatlarını JSON formatında kaydetmek mümkündür.



Şekil 1.1 MediaPipe algoritmasının görselleştirilmesi

Veri Kaydı ve Kullanımı

- Frame Bazlı İşleme: Videolar belirli adımlarla karelere ayrıldıktan sonra, her kare için iskelet verileri çıkarılmıştır.
- Paralel İşleme: Çoklu iş parçacığı kullanılarak hızlandırılmıştır.
- Analiz İçin JSON Formatı: Elde edilen koordinatlar, işaret dili tanıma modeline giriş olarak verilmek üzere JSON formatı ile saklanmıştır.

Denenen bu yöntemler arasında MediaPipe Hands ve MediaPipe Body, kurulum kolaylığı, gerçek zamanlı çalışabilmesi ve detaylı el takibi sunması nedeniyle işaret dili tanıma sürecinde en uygun yöntem olarak seçilmiştir

3. 4 Mediapipe ile İskelet Çıkarımı

1. Koordinatların Elde Edilmesi:

Videolardan elde edilen her bir karede, MediPipe yardımıyla elin 21 anahtar noktası (landmark) saptanacaktır. Bu noktalar X, Y ve Z koordinat bilgilerini içermektedir.

- X ve Y koordinatları elin konumunu belirtir.

- Z koordinatı, derinlik bilgisini içerir ve elin kameraya olan uzaklığını belirtir.

2. Koordinatların Normalizasyonu:

Elde edilen koordinatlar, her kullanıcının el yapısı ve pozisyonuna göre değişiklik gösterebileceğinden, normalizasyon(standartlaştırma) işlemi uygulanacaktır.

- Min-Max Normalizasyonu: Koordinatlar $[0,1]$ aralığına ölçeklendirilecektir.
- Merkezleme: Elin merkezi (bilek noktası) referans alınarak koordinatlar yeniden belirlenecektir.

3. Zaman Serisi Yapısının Elde Edilmesi:

Videolardan çıkarılan her frame(kare), bir zaman serisi oluşturacak şekilde sıraya alınacaktır. Ardışık karelerden elde edilen koordinat verileri, modelin zamansal tutarlılığı öğrenebilmesi için birleştirilecektir.

3. 5 Model Mimarisi

Bu model, el hareketlerinin zamansal analizine odaklanan iki katmanlı çift yönlü LSTM (Bidirectional LSTM) yapısı üzerine kurulmuştur. Görsel özellik çıkarımı doğrudan modelin giriş verisinde hazır olarak sağlandığı varsayılmakta, bu nedenle CNN yerine, önceden çıkarılmış görsel özellikler LSTM'ye doğrudan beslenmektedir.

1. Girdi Özellikleri

- Modelin girişi, her biri görsel bir kareyi temsil eden ve belirli sayıda özelliğe sahip sıralı veri (zaman serisi) olarak tanımlanmıştır.
- Girdi şekli: (zaman_adımı, kare_özellik_sayısı)

2. Bidirectional LSTM Katmanları

- **İlk Katman:** 128 birimli çift yönlü LSTM katmanı, hareketlerin zamansal örüntülerini her iki yönde öğrenmek üzere yapılandırılmıştır. `return_sequences=True` sayesinde çıktılar bir sonraki LSTM katmanına tam sırayla aktarılır.
- **İkinci Katman:** 64 birimli ikinci çift yönlü LSTM katmanı, zaman içindeki daha derin ilişkileri öğrenir. Her iki katmanda da L2 regularizasyonu uygulanarak aşırı öğrenme (overfitting) önlenmeye çalışılmıştır.

- Her LSTM katmanından sonra sırasıyla:
 - **Batch Normalization**: Öğrenmeyi stabilize eder.
 - **Dropout (0.4)**: Rastgele nöronları devre dışı bırakarak aşırı öğrenmeyi önler.

3. Havuzlama ve Tam Bağlantılı Katman

- **GlobalAveragePooling1D**: Zaman serisi boyunca ortalama alınarak boyut düşürülür.
- **Dense Katmanı**: 128 nöronlu, ReLU aktivasyon fonksiyonlu tam bağlantılı katman, öğrenilen zamansal özellikleri işleyerek sınıflandırma için uygun forma getirir.
- Ardından yine **Batch Normalization** ve **Dropout (0.4)** uygulanır.

4. Çıkış Katmanı

- Son katman, sınıf sayısı kadar nörona sahip olup **Softmax** aktivasyon fonksiyonu ile sınıflandırma gerçekleştirir.
- Bu yapı sayesinde model, her hareket dizisini belirli bir sınıfa ait olarak etiketleyebilir.

5. Hiperparametre Optimizasyonu (Grid Search)

Modelin başarımını artırmak için **Grid Search** yöntemiyle farklı hiperparametre kombinasyonları denenmiştir:

- **Dropout oranı**,
- **Öğrenme oranı (learning rate)**,
- **Batch size** gibi parametreler sistematik olarak test edilmiştir.

Bu aramalar sonucunda en yüksek doğruluk sağlayan parametre kombinasyonu belirlenmiş ve eğitimde bu değerler kullanılmıştır.

3. 5. 3 Eğitim ve Test Aşamaları

1. Veri Bölme:

Veri seti, % 78 eğitim, %12 validation ve %10 test olarak bölünecektir. Eğitim sırasında çapraz doğrulama (k-fold cross-validation) uygulanacaktır.

3. 5. 4 Model Eğitimi

1. Eğitim Parametreleri:

- Optimizasyon Algoritması: Adam (Adaptive Moment Estimation) kullanılacaktır.
- Kayıp (Loss) Fonksiyonu: Kategorik çapraz entropi kaybı (Categorical Cross Entropy) tercih edilecektir.
- Öğrenme Oranı (Learning Rate): 0.001 ve 0.0001 olarak denenecektir.
- Batch Boyutu: 16, 32, 64 olarak denenecektir.
- Epoch Sayısı: 150 epoch boyunca eğitim yapılacaktır.

2. Model Değerlendirme:

- Doğruluk (Accuracy): Modelin sınıflandırma başarısını gösterecektir.
- F1 Skoru: Kesinlik(Precision) ve Duyarlılık(Recall) değerlerinin harmonik ortalamasını hesaplanacaktır.
- Karışıklık Matrisi (Confusion Matrix): Hatalı ve doğru sınıflandırmaları gösterecektir.

3. 5. 5 DeneySEL Sonuçlar

Modelin eğitimi ve değerlendirme sonucunda elde edilen performans değerleri analiz edilecektir.

- Başarı Oranı Karşılaştırması: Literatürdeki halihazırdaki metotlar ile bu çalışma kapsamında oluşturulan modelin performansı karşılaştırılacaktır.
- Hata Analizi: Yanlış sınıflandırma yapılan işaretlerin özellikleri analiz edilecek ve modelin iyileştirilmesi için öneriler sunulacaktır.
- Gerçek Zamanlı Tanıma: Modelin canlı video akışında performansı test edilecektir.

4. BULGULAR VE TARTIŞMA

4. 1 Veri Seti Özellikleri

Bu çalışmada, Türk İşaret Dili (TİD) örneği için AUTSL veri seti kullanılıp, çeşitli ön işleme adımları gerçekleştirilmektedir. Veri seti, 226 farklı işaret ve toplam olarak 38.336 örnek içermektedir. 43 farklı işaretleyici tarafından sağlanan veri seti, arka plan çeşitliliği ve dinamik çevresel koşullar ile modelin gerçek dünya koşulları içinde daha iyi genelleme yapabilmesini düzenlenmektedir. Türk İşaret Dili öğretmenlerinin işbirliği ile, günlük hayatta sıkça kullanılan işaretler seçilmiş ve el hareketlerinin çeşitliliği dengeli bir biçimde dağıtılmaktadır.

4.2 Veri Ön İşleme

Veri setinde bulunan videoların işleme sürecinde, boş veya durağan karelerin tespiti, videoların aynı çözünürlükte ölçeklenmesi ve bozuk olan videoların ayıklanması gibi adımlar uygulanmaktadır. Yapılan bu işlemler sonucunda elde edilen temiz ve standart hale getirilmiş veri kümesi, modelin daha düzgün bir şekilde eğitilmesine olanak tanımaktadır.

4.3 İskelet Çıkarımı

MediaPipe kütüphanesi ile el ve vücut hareketleri detaylı bir şekilde tespit edilmektedir. Elde edilen koordinatlar ise video kareleri üzerinden çıkarılıp, JSON formatında saklanmaktadır. Bu yöntem ile kurulum kolaylığı ve gerçek zamanlı analiz yapabilme yeteneği nedeniyle tercih edilmektedir.

4.4 Model Mimarisi

Görsel ve zamansal bilgiyi içeren bir model oluşturulmuştur. İki katmanlı Bidirectional LSTM yapısı kullanılmıştır. Eğitim sürecinde Adam optimizasyon algoritması ve çapraz entropi (categorical cross-entropy) kaybı kullanılmıştır. En iyi performans için dropout, öğrenme oranı (lr) ve batch size değerleri Grid Search ile optimize edilmiştir.

5. Kullanılan Teknolojiler

Bu projede, hem kullanıcı deneyimini iyileştirmek hem de gerçek zamanlı işaret dili tanıma sistemini etkili biçimde uygulamak amacıyla çeşitli teknolojilerden yararlanılmıştır. Kullanılan başlıca teknolojiler aşağıda başlıklar halinde sunulmuştur:

- Frontend Teknolojileri:
Kullanıcı arayüzü, HTML5, CSS3 ve JavaScript kullanılarak geliştirilmiştir. Bu sayede kullanıcıların sisteme kolayca erişebilmesi, kamera ile görüntü alınabilmesi ve etkileşimli arayüz kullanımı sağlanmıştır. Tasarımda responsive (duyarlı) yapılar tercih edilerek tüm cihaz türlerine uyumlu hale getirilmiştir.
- Backend Teknolojileri:
Sunucu tarafı işlemlerinde Python programlama dili tercih edilmiş; Flask mikro web çatısı ve WebSocket protokolü ile gerçek zamanlı veri iletişimi sağlanmıştır. WebSocket, video karelerinin gecikmesiz biçimde işlenmesini mümkün kılmıştır.
- Makine Öğrenimi ve Görüntü İşleme:

Görüntü işleme ve işaret tanıma işlemleri için TensorFlow derin öğrenme kütüphanesi ile birlikte MediaPipe ve OpenCV kütüphanelerinden faydalanılmıştır. MediaPipe, özellikle el ve vücut pozisyonlarını belirlemek için kullanılmış, elde edilen landmark verileri TensorFlow modeline giriş olarak verilmiştir.

- E-posta Servisi:
Şifre sıfırlama işlemleri için üçüncü taraf e-posta servisi olan EmailJS kullanılmıştır. Bu servis sayesinde kullanıcılara otomatik olarak doğrulama kodu gönderimi yapılmaktadır.
- Veri Saklama:
Kullanıcı oturumları, kayıt bilgileri ve geçici veriler tarayıcı tabanlı localStorage sistemi ile yönetilmektedir. Bu yöntem, sunucuda kullanıcı verisi saklamaya gerek kalmadan hızlı ve kullanıcı bazlı veri erişimi sağlamaktadır.
- Yayınlama ve Dağıtım:
Uygulamanın internet üzerinde erişilebilir olması için Vercel veya Netlify gibi modern hosting platformları önerilmektedir. Bu platformlar, statik ve dinamik içerikleri kolaylıkla yayınlayarak projeyi genel kullanıma açma imkânı sunmaktadır.

6. Sistem Mimarisi ve Akış

6.1. Kullanıcı Arayüzü (Frontend)

Sistem, kullanıcı dostu ve etkileşimli bir arayüz ile donatılmıştır. Kullanıcılar şu işlemleri gerçekleştirilmektedir:

- Kayıt ve Giriş: Kullanıcılar yeni hesap oluşturabilir veya mevcut hesapları ile giriş yapabilir. Bu süreçte kullanıcı adı, e-posta ve şifre gibi bilgiler localStorage üzerinde saklanmaktadır.
- Şifre Sıfırlama: Şifresini unutan kullanıcılar, sistemde kayıtlı e-posta adreslerini girerek doğrulama kodu alabilirler. EmailJS aracılığıyla 6 haneli rastgele bir kod gönderilmekte; bu kodun arayüze girilmesinin ardından kullanıcı yeni şifresini belirleyebilmektedir.

- Kamera Tabanlı Tanıma: Kamera üzerinden alınan görüntüler anlık olarak işlenerek, işaret dili hareketleri tanımlanmakta ve tahmin edilen kelimeler kullanıcıya gösterilmektedir. Kullanıcı isterse bu kelimeleri anlamlı bir cümleye dönüştürebilmektedir.
- Responsive Tasarım: Tüm bu işlemler, farklı ekran boyutlarına uyum sağlayabilecek biçimde tasarlanmıştır. Media query'ler ile desteklenen CSS yapısı sayesinde masaüstü, tablet ve mobil cihazlarda sorunsuz bir kullanıcı deneyimi sağlanmaktadır.

6.2. Sunucu ve Model Entegrasyonu (Backend)

Sistemin arka planında, Python ile geliştirilmiş bir WebSocket sunucusu yer almaktadır. Bu yapı sayesinde, frontend tarafından gönderilen canlı kamera görüntüleri aşağıdaki adımlarla işlenmektedir:

1. Görüntü Alımı: Kamera görüntüleri, WebSocket protokolü üzerinden backend'e aktarılmaktadır.
2. Landmark Çıkartımı: MediaPipe kütüphanesi kullanılarak el ve vücut landmark'ları (anahtar noktalar) belirlenmektedir.
3. Model Tahmini: Elde edilen landmark verileri, TensorFlow tabanlı eğitilmiş bir derin öğrenme modeline aktarılmakta ve model bu veriler üzerinden ilgili işaret kelimesini tahmin etmektedir.
4. Sonuçların Gönderimi: Tahmin edilen kelime, frontend arayüzüne iletilmekte ve kullanıcıya sunulmaktadır.
5. Cümle Oluşturma: Kullanıcı tercihiyle ilgili olarak, art arda gelen tahminler Gemini API'si yardımıyla birleştirilerek anlamlı bir cümle haline getirilebilmektedir.

6.3. Şifre Sıfırlama ve E-posta Entegrasyonu

Şifre sıfırlama işlemi, EmailJS servisi aracılığıyla gerçekleştirilmiştir. Bu sürecin teknik detayları aşağıdaki gibidir:

- Kullanıcı, e-posta adresini girerek şifre sıfırlama isteğinde bulunur.
- Sistem, EmailJS API'si aracılığıyla bu adrese 6 haneli rastgele bir doğrulama

kodu gönderir.

- Kullanıcı bu kodu ilgili giriş alanına girdikten sonra yeni şifresini belirleyebilir.
- EmailJS şablonları dinamik olarak kullanıcı bilgilerine göre özelleştirilmekte, böylece kişiye özel ve güvenli bir sıfırlama süreci sağlanmaktadır.

7. Gerçekleştirilen Teknik Çalışmalar

Projenin geliştirme sürecinde aşağıdaki teknik çalışmalar başarıyla tamamlanmıştır:

- **Kullanıcı Yönetimi:** Kayıt olma, giriş yapma, oturum yönetimi ve profil düzenleme işlemleri localStorage kullanılarak güvenli bir şekilde gerçekleştirilmiştir.
- **Şifre Sıfırlama Sistemi:** EmailJS ile bütünleştirilen şifre sıfırlama akışı, doğrulama kodu gönderimi ve yeni şifre belirleme süreçlerini kapsamaktadır. Bu sayede kullanıcı güvenliği ve deneyimi artırılmıştır.
- **Gerçek Zamanlı Görüntü İşleme:** WebSocket aracılığıyla frontend'den backend'e gönderilen görüntüler MediaPipe ile analiz edilmiş, ardından TensorFlow modeli ile işaret dili tahminleri gerçekleştirilmiştir.
- **Anlamlı Cümle Oluşturma:** Tahmin edilen işaretlerin mantıksal bağlam içinde birleştirilmesini sağlayan özel bir cümle oluşturma fonksiyonu GPT ve Gemini LLM'leri kullanılarak geliştirilmiştir. Gemini modeli seçilerek projede kullanılmıştır.
- **Responsive Arayüz Tasarımı:** Tüm cihazlarda uyumlu çalışacak şekilde duyarlı tasarım ilkeleri benimsenmiş ve arayüz CSS medya sorguları ile desteklenmiştir.
- **Kullanıcı Deneyimi Geliştirme:** Modern bir tasarım dili ile kullanıcı arayüzü zenginleştirilmiş, hata mesajları ve bilgilendirme ekranlarıyla kullanıcı yönlendirmesi sağlanmıştır.

8. Yapılacak Çalışmalar

Kullanılan modelin hız ve doğruluk açısından performansı iyileştirilmeye çalışılacaktır.

Web sitesi arayüzünün kullanıcı performans testleri gerçekleştirilerek geliştirilmesi sağlanacaktır.

KAYNAKLAR

- [1] Stokoe, W. C. (1960). Sign language structure: *An outline of the visual communication systems of the American deaf*. Studies in Linguistics: Occasional Papers.
- [2] WFD (World Federation of the Deaf). (2023). *Guidelines for achieving sign language rights*. World Federation of the Deaf
- [3] Brentari, D. (1998). *A prosodic model of sign language phonology*. MIT Press.
- [4] Kurumu, T. D. (2023). iletişim. Nisan, 3, 2023.
- [5] Supalla, T., & Webb, R. (1995). CHAPTER FIFTEEN. *Language, Gesture, and Space*, 333.
- [6] Elakkiya, R., & Sumathi, V. P. (2020). Deep learning techniques for sign language recognition: A review. *Artificial Intelligence Review*, 53(8), 5685-5730.
<https://doi.org/10.1007/s10462-020-09804-3>
- [7] Starner, T., Weaver, J., & Pentland, A. (1998). Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12), 1371-1375.
<https://doi.org/10.1109/34.736314>
- [8] De Coster, G., & Sincan, S. (2023). *Isolated Sign Recognition From RGB Video Using Pose Flow*. arXiv. <https://arxiv.org/pdf/2403.14534>
- [9] De Coster, G. (2021). *Isolated Sign Recognition From RGB Video Using Pose Flow and CVPRW 2021*. ChaLearn.
https://openaccess.thecvf.com/content/CVPR2021W/ChaLearn/html/De_Coster_Isolated_Sign_Recognition_From_RGB_Video_Using_Pose_Flow_and_CVPRW_2021_paper.html
- [10] Sincan, S., & Koç, A. (2020). *Deep Learning Approaches to Sign Language Recognition*. IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/9210578>
- [11] Kocabey, E., & Şahin, F. (2023). *Sign Language Recognition Using Machine*

Learning. ACM Digital Library. <https://dl.acm.org/doi/abs/10.1145/3654522.3654559>

[12] Sincan, S., & Koç, A. (2020). *Large-Scale Sign Language Recognition Using Pose Flow*. arXiv. <https://arxiv.org/abs/2004.01283>

[13] Sincan, S., & De Coster, G. (2021). *Large Scale Signer Independent Isolated Sign Language Recognition*. ChaLearn.

https://openaccess.thecvf.com/content/CVPR2021W/ChaLearn/html/Sincan_ChaLearn_LAP_Large_Scale_Signer_Independent_Isolated_Sign_Language_Recognition_CVPRW_2021_paper.html

[14] O. M. Sincan and H. Y. Keles, "AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods," in *IEEE Access*, vol. 8, pp. 181340-181355, 2020, doi: 10.1109/ACCESS.2020.3028072.