T.C

VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ

BİLGİSAYAR PROGRAMCILIĞI BÖLÜMÜ

Konu: Sanal Klavye Mouse Projesi

Ders: Nesne Tabanlı Programlama II

Koordinatör: Öğr. Gör. Osman GÜMÜŞ

Hazırlayanlar

Metin HAŞİMİ 22160019042

Yavuz Emre TUNCİL 21160019032

İÇİNDEKİLER

ÖZET**3**

1.KÜTÜPHANELER**3**

1.1.OPENCV**3**

1.2.OPENCV İLE NELER YAPILABİLİR**3**

1.3.MEDIAPIPE**4**

1.4.MEDIPIPE İLE NELER YAPILABİLİR**4**

2.PYTHON NEDİR NEDEN PYTHON**5**

3.PROJE**5**

3.1.PROJE KAPSAM VE AMACI**5**

3.2.PROJE İŞLEVİ**5**

3.3.PROJEYE NEDEN İHTİYAÇ DUYULDU**5**

4.YAPILABİLİRLİK**6**

5.SWOT ANALİZİ**6**

5.1.GÜÇLÜ YÖNLER**6**

5.2.ZAYIF YÖNLER**6**

5.3.TEHDİT YÖNLER**6**

5.4.FIRSAT YÖNLER**6**

6.VERİTABANI İLE İLGİLİ**7**

7.AKIŞ ŞEMASI**7**

8.BENZER ÇALIŞMALAR**7**

8.1.VİSUAL KEYBOARD**7**

8.2.VİSUAL MOUSE**7**

8.3.PROJE FARKI**7**

9.KOD YAPILARI**8**

9.1.HAND TRACKER**8**

9.2.MOUSE**8**

9.3.Sanal Klavye**9**

ÖZET

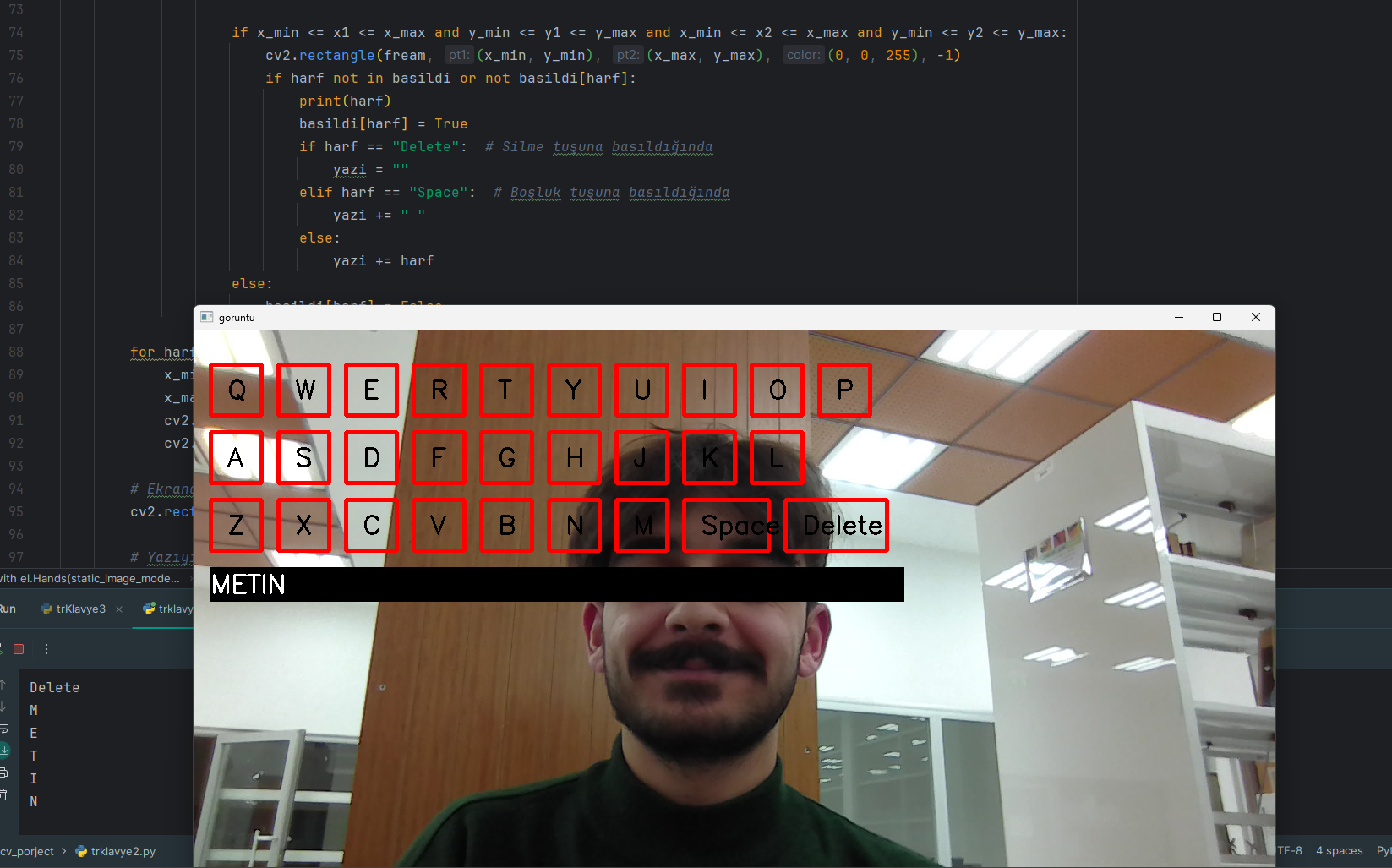
Projemiz basitçe Opencv ve Mediapipe gibi benzer python kütüphanelerini kullanarak Google’nin el iskelet ve görüntü modeli üzerine konularak oluşturulmuş klavye ve Mouse’u el hareketleri ile kullanabilme imkânı sağlamak.

1.1.OpenCV

Opencv (*Open Source* Computer Vision Library, anlamı Açık Kaynak Bilgisayar Görüsü Kütüphanesi) gerçek-zamanlı bilgisayar görüsü uygulamalarında kullanılan açık kaynaklı bir kütüphanedir. İlk olarak Intel tarafından geliştirilmiş, daha sonra Willow Garage ve sonra Itseez (Intel tarafından satın alındı) tarafından sürdürüldü. Bu kütüphane çoklu platform ve BSD lisansı altında açık kaynaklı bir yazılımdır.

1.2.OpenCV İle Neler Yapılabilir

Opencv ve benzeri görüntü işleme kütüphaneleri, birçok farklı uygulama alanında kullanılır. İşte bunlardan bazıları:



**Görüntü işleme ve analiz:**

Nesne tanıma, yüz tanıma, hareket algılama, nesne takibi gibi birçok görüntü işleme ve analiz görevinde

kullanılır.

Robotik:

Robotik sistemlerde görüntü tabanlı algılama ve animasyon için kullanılabilir. Örneğin, bir robotun çevresini algılamak ve nesneleri tanımak için kullanılabilir.

Otomatik Sürüş:

Otomotiv endüstrisinde, sürücüsüz araçlar veya yardımcı sürüş sistemleri için görüntü işleme ve analiz kullanılabilir. Örneğin, araçların çevresini algılamak ve diğer araçları veya engelleri tanımak için kullanılabilir.

Güvenlik Ve Gözetim**:**

Güvenlik kameraları ve gözetim sistemleri için görüntü işleme, yüz tanıma, hareket algılama gibi işlevlerde kullanılabilir.

Tıp ve Sağlık:

Tıbbi görüntüleme sistemlerinde (örneğin, MRI, CT taramaları) veya hastalık teşhisi için görüntü analizi için kullanılabilir.

Sanal ve Artırılmış Gerçeklik**:**

Sanal gerçeklik ve artırılmış gerçeklik uygulamalarında görüntü işleme ve görüntü tabanlı takip için kullanılabilir.

Endüstriyel Otomasyon:

Endüstriyel sistemlerde ürün kalitesinin kontrolü, nesne tanıma ve takip için kullanılabilir.

Eğlence ve Oyunlar:

Oyun geliştirme ve eğlence sektöründe, karakter tanıma, hareket algılama gibi birçok görüntü işleme görevi için kullanılabilir. Bu sadece birkaç örnek. Görüntü işlemenin kullanım alanları oldukça geniştir ve sürekli olarak yeni uygulamalar geliştirilmektedir.

1.3.MediaPipe

MediaPipe, Google tarafından geliştirilen açık kaynaklı bir çerçeve ve kütüphane koleksiyonudur. Video analizi ve medya işleme uygulamaları geliştirmek için kullanılır. MediaPipe, kullanıcıların çeşitli işlemleri gerçekleştirebilecekleri modüler ve araçlar sunar ve bu işlemler genellikle video ve görüntü verileri üzerinde yapılır.

1.4.MediaPipe İle Neler Yapılabilir

MediaPipe Graphs (MediaPipe Grafikleri):

MediaPipe, veri akışını tanımlamak ve işlemek için bir grafik tabanlı yapı kullanır. Bu grafikler, farklı işlevleri gerçekleştiren modüllerin bir araya getirilmesini sağlar. Örneğin, yüz algılama, el izleme, vücut tespiti gibi görevler için hazır modüller sağlanır ve bu modüller birleştirilerek istenen işlemler gerçekleştirilir , bizde bu sayede istediğimiz projeyi uygulayabiliyoruz.

Önceden Eğitilmiş Modeller:

MediaPipe, bir dizi önceden eğitilmiş derin öğrenme modeli içerir. Bu modeller, nesne algılama, yüz tanıma, insan vücudu algılama gibi görevler için kullanılabilir.

MediaPipe Python API:

MediaPipe’ in Python API'si, geliştiricilere MediaPipe grafiklerini oluşturmak, çalıştırmak ve sonuçları işlemek için bir arayüz sağlar.

MediaPipe C++ API:

MediaPipe, C++ API aracılığıyla da kullanılabilir. Bu API, performansı artırmak ve platform bağımsız uygulamalar geliştirmek için kullanılabilir . MediaPipe, genellikle gerçek zamanlı video işleme ve analizi uygulamaları geliştirmek isteyen araştırmacılar, geliştiriciler ve mühendisler tarafından kullanılır. Örneğin, artırılmış gerçeklik uygulamaları, hareket tanıma sistemleri, insan etkileşimi ara yüzleri gibi çeşitli alanlarda kullanılabilir

2.PYTHON NEDİR , NEDEN PYTHON

Python, 1990 yılından başlanarak Hollandalı Guido Van rossum’un Hollanda’nın

Amsterdam kentinde geliştirmeye başladığı, mihrakı nesne olan, etkileşimli, birimsel ve yüksek kademeli yani kullanıcı derecesinde kodlanan bir programlama dilidir. Bir başka deyişle Python’un bir derleyici kanalı yardımıyla makine diline derlenmesi elzemdir. Kodlanış açısından Bu dil diğer yüksek kademeli dillerden daha kolaydır. Python ile birçok proje daha basit bir şekilde gerçekleştirilebilir. Python’un yazım şekli diğer dillere nazaran daha kolaydır. Ayrıca Python sürekli güncellenen dinamik bir dildir. Python için tüm dünyadan gönüllüler modüller yazmaktadır. Ayrıca görüntü işleme veri analizi ve madenciliği gerektirdiğinden görüntü işlemenin en rahat ve en kolay yapılabildiği dil Python’dur. Python’da veri analizi ve madenciliği ile ilgili birçok modül vardır. Bu çalışmada tanıtımı yapılan projede kullanılan OpenCv modülüde Python ortamına çok rahat bir şekilde adapte edilebilmektedir. Bu nedenden dolayı projemizde Python yazılım dilini kullanacağız.

3.1.Proje Kapsam Ve Amacı

Bu çalışma kapsamında anlık kamera görüntülerinden el hareketlerini tanıyarak, işlemeye ve bu el hareketlerini otomatik olarak sanal klavye ve fare uygulamasına yarayan projemizde, görüntü işleme algoritmaları ve sezgisel tanıma sistemleri incelenmiştir. Ayrıca görüntü tanıma sistemlerinde sıklıkla kullanılan ve alt yapıları oluşturan OpenCv ve MediaPipe kütüphanesi tanıtılmıştır. Sezgisel hareket tanıma sistemleri sade ve anlaşılır olmalıdır. Ayrıca gerçek zamanlı görüntü kaynağından tanıma ve tanımlama işlemleri gerçekleşeceği için birden fazla görüntüyü otomatik olarak işleme kapasitesine sahip olmalıdır ve bu sebeple performanslı çalışmalıdır. Bu çalışmanın amacı günlük hayatta Klavye ve Mouse kullanımına farklı bir boyut kazandırmak ve Eski demode Kompakt fiziksel temas gerektiren sistem yerine daha yeni ve modern kullanımlara sahip geliştirilebilir ve önü açık bir klavye Mouse sistemi geliştirmek ve Kullanıcı Performansını iyileştirme.

3.2.Proje İşlevi

Bu çalışmaya konu olan projemizin işlevi herhangi bir görüntü kaynağından yakalanan görüntüdeki el hareketlerini tespit ederek yazılıma önceden öğretilmiş kişinin el hareketleri sayesinde klavye ve fareye ihtiyaç duymaksızın klavye ve fare işlevine yarayacak bir işlemdir.

3.3.Projeye Neden İhtiyaç Duyuldu

Klavye ve fare bilgisayar donanım araçlarının önemli iki parçasıdır. Fare ve klavye olmadan bir bilgisayar hayal etmek çoğu insan için imkansız gibi bir şeydir. Bizde bu projemizde bu algıyı kırarak fare ve klavyeye ihtiyaç olmadan bir bilgisayar mümkündür dedik. Bu proje sayesinde kullanıcılar 2. ve 3. Araç olmadan bilgisayar kullanabileceklerdir. Bunu aynı zamanda hem nakit tasarrufu hem de kuvvet tasarrufu olarakta algılayabilirsiniz. Kısacası Günümüzde kullanılan Klavye ve Mouse deneyimini iyileştirme ve kullanıcıların daha sağlıklı bir kullanım deneyimi sağlamasına yardımcı olma ve Klavye Mouse algısına yeni bir boyut eklemek.

4.YAPILABİLİRLİK

Python'un geniş kütüphane desteği sayesinde klavye ve fare simülasyonu için gerekli olan işlevlerin kolayca yazılabilmesi. Mediapie ve OpenCV gibi kütüphanelerin belirlenen gereksinimleri karşılaması ve projenin teknik zorluklarını çözmede yardımcı olmasıYeteri kadar zaman ve sabırla yapılabilir.

5.SWOT ANALİZİ

Basitçe SWOT analizinin amacı Güçlü , Zayıf , Tehdit , Fırsat değerlendirmesi yapmaktır .

5.1.Güçlü Yönler

Python dilinin esnekliği ve geniş kütüphane desteği sayesinde projenin hızlı bir şekilde geliştirilebilmesi.Mediapie ve OpenCV gibi güçlü kütüphanelerin kullanılmasıyla, klavye ve fare simülasyonu için gerekli olan görüntü işleme ve etkileşim mekanizmalarının sağlanabilmesi. Klavye ve fare simülasyonuna olan talebin artmasıyla, projenin pazar payını genişletme ve yeni kullanıcı kitlesine ulaşma fırsatı sağlar

Ek olarak bu proje Günümüzden geleceğe bir Sanal Cihaz ön hazırlığı olur

Hiç olmazsa bunun için ilgide oluşturularak Eğitim adına güzel bir proje olarak geçer .

5.2.Zayıf Yönler

1. Teknolojik karmaşıklık nedeniyle geliştirme sürecinde karşılaşılabilecek potansiyel engeller. Ör(“Versiyon problemleri ”)
2. Yeterince Hızlı ve Duyarlı olamaması
3. Çeşitli Kütüphane Versiyonlarını desteklememesi
4. Tüm IDE’ler de Çalışmaması
5. Kameranın zorunlu olması İçi içe Birden fazla görünüm kullanamıyor oluşu (hem klavye hem Mouse aynı anda çalışamıyor)
6. Stabilize ve Optimizasyon Problemi
7. Kütüphaneler Tam Verimde Çalışmaması
8. Çok Farklı İç içe Kombinasyonlara açık olmaması
9. Türkçe Q , F Gibi Temel Klavye Tür Problemlerini Yaşaması

(Her ülkenin Klavye Türü Farklı )

1. Rahat Bir Kullanıma Sahip olmaması

5.3.Tehdit Yönleri

Log dosyası ve Veri Tabanı Kayıtlarını tutmadığmız için Her hangi bir içerden ve ya dışardan

Tehdit oluşturulamaz Anca bir Hacker gelecekte Virüs yerleştirecek te ona göre Keyloger dan bakacak bilgilere ve anca o şekilde bir tehdit oluşturur yada kod yapısını bozar .

5.4.Fırsat Yönleri

İlerde Bu projenin modeli kullanarak daha farklı bir evreye varabilir

Belki de engelliler için pozitif yönde bir uygulama olma potansiyeli taşıyor .

6.VERİTABANI İLE İLGİLİ

Çalışmamızda veri tabanına ihtiyaç duymadık çünkü bu proje ile ilgili veri sınırsız denecek kadar çok. Projede veriler kullanıcı tarafından girildiği için kullanıcıdan kullanıcıya göre veri türleri değişecektir veri bundan dolayı veri tabanını belirli bir şekilde kalıplandıramıyoruz.

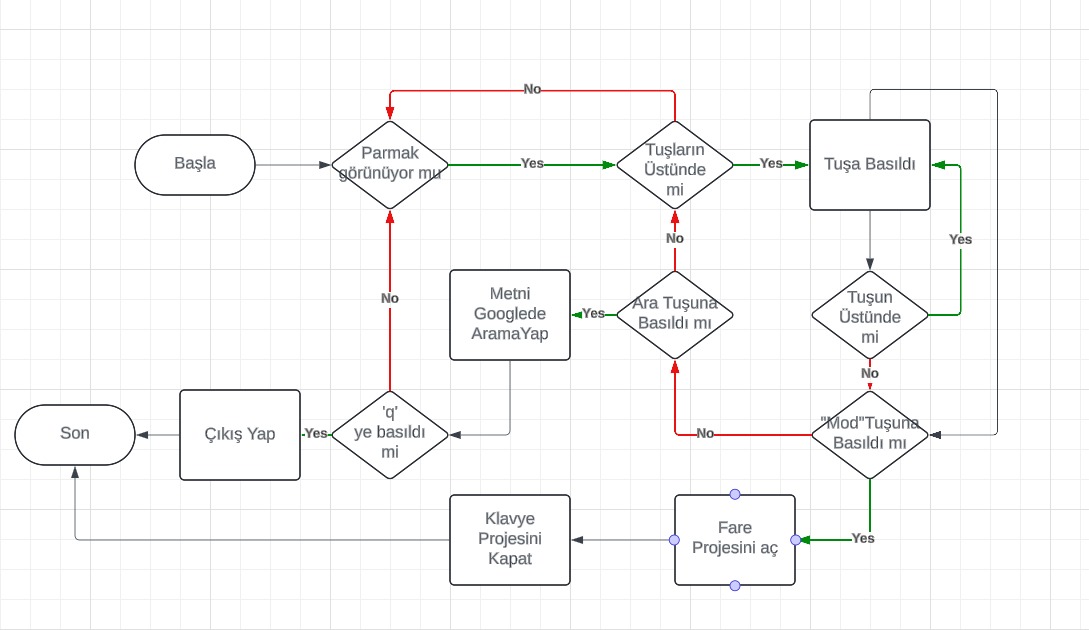
Aynı şeklide Kullanıcılar bir oturuma sahip olmadığı için Veri karmaşıklığı problemi beraberinde geliyor .

Diyelim ki Veri Tabanı yaptık ve Klavyede girilen bilgileri Veri Tabanında Tutuk.

Peki Bu veriler ne işe yarayacak?

Ve şunu söylemek gerekirse Keşke bir Veri Tabanını da Kapsayacak bir Çalışma olsa da Üzerine Düşünüp geliştirile bilmesi gibi süreçler yapabilsek.

7.AKIŞ ŞEMASI



8.BENZER ÇALIŞMALAR

8.1.Visual Keyboard:

<https://www.youtube.com/watch?v=jzXZVFqEE2I&t=1749s&ab_channel=Murtaza%27sWorkshop-RoboticsandAI>

Burada Sanal Klavyeyi Oluşturulmayı anlatılmış ve uygulamalı bir şekilde anlatılmış.

8.2.Visual Mouse:

<https://www.youtube.com/watch?v=8gPONnGIPgw&t=19s&ab_channel=Murtaza%27sWorkshop-RoboticsandAI>

Burada ise Sanal Mouse’yi Nasıl yapılacağını gösterilmiş

8.3.Proje Farkı

Hem klavye hem de Mouse’yi aynı projede kullanıp modlar arası geçişi sağlayarak bir kullanım elde etmek ve yapılan diğer projedeki eksiklikleri de halletmek. (mesela: TR Q klavye)

9.KOD YAPILARI

9.1.Hand Tracker:

Class handDetector():

init: Elleri algılamak için MediaPipe kütüphanesini ve ilgili özellikleri başlatır.

Bu özellikler aşağıda görüldüğü gibidir

* mode=false : Kullanılacak modun kapalı olduğunu döndürür
* maxhands=2: Algılanacak Maximum el saysını belirtir
* detectioncon=false : Algılama Kontrolu kapalı olduğnu söyler
* econ=0.5 : Yoğunluk hassasiyetinin ½ olduğunu belirtir .

findHands: Bir görüntüde elleri algılar ve isteğe bağlı olarak bu ellerin çizimini yapar.

findPosition: Ellerin pozisyonunu bulur ve bir sınırlayıcı kutu (bbox) çizer.

fingersUp: Parmakların yukarıda olup olmadığını kontrol eder.

findDistance: İki parmak arasındaki uzaklığı bulur.

Def Main():

Kameradan gelen görüntüyü alır ve el izleme işlevlerini uygular.

Parmak pozisyonlarını ve FPS (kare hızı) değerini görüntü üzerine ekler.

Bu kod, bir kameradan görüntü alır, elleri algılar ve her bir parmağın pozisyonunu bulur. Ayrıca, eldeki

parmakların yukarıda olup olmadığını kontrol eder ve iki parmak arasındaki uzaklığı hesaplar.

9.2. Mouse:

Class Smart\_Mouse()

* İnit: Webcam'i başlatır, el algılamak için HandTracking modülünü kullanır. Fare konumunu belirlemek için ekran genişliği ve yüksekliğini alır.
* Run: Elleri algılar ve fareyi kontrol etmek için belirli el hareketlerini kullanır.

Def Run()

Her bir karede el algılamayı çalıştırır ve belirli el hareketlerine karşılık gelen fare hareketlerini yapar.

Parmaklar kaldırıldığında ve parmaklar belirli bir pozisyonda olduğunda fareyi hareket ettirir veya tıklar. Bu method kodun çalışmasını sağlar

Ellerin yönüne bağlı olarak fareyi hareket ettirir veya tıklar.

Görüntüye bir FPS (kare hızı) metni ekler ve gösterir.

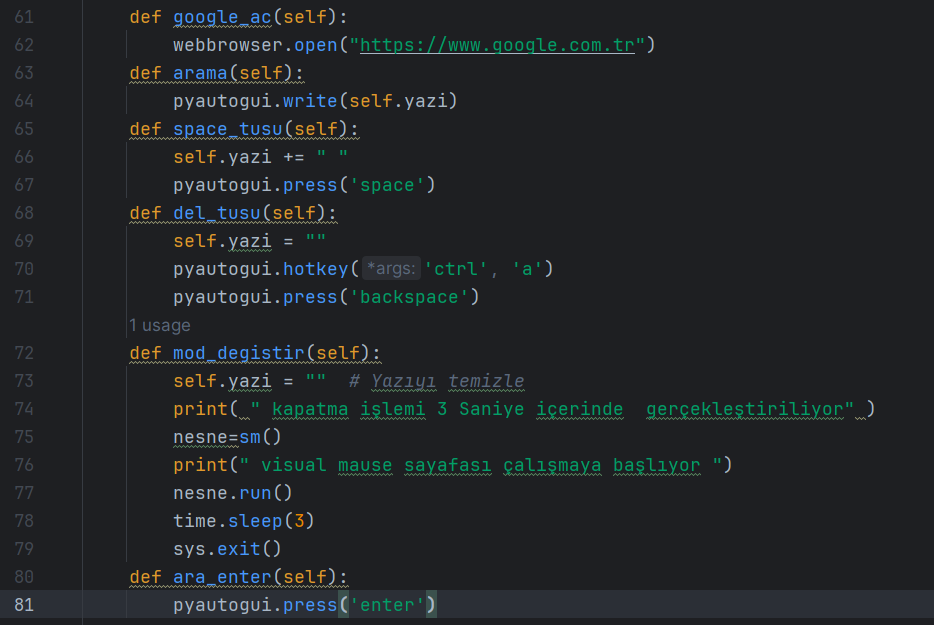
'q' tuşuna basıldığında programdan çıkar.

Bu kod, ellerinizi kullanarak bilgisayar fare kontrolü sağlar. İki parmağınızı kaldırarak fareyi hareket ettirebilir ve iki parmağınızı bir araya getirerek fareyi tıklayabilirsiniz.

9.3.Sanal Klavye:

Class EllerleKlavye()

Çeşitli Fonkisyonlar



* İnit: Webcam'i başlatır, el algılamak için Mediapipe kütüphanesini kullanır ve klavye harflerinin konumlarını belirler.
* google\_ac: Google'ı açmak için tarayıcıyı kullanır.
* arama: Belirli bir metni yazmak için pyautogui kullanır.
* space\_tusu: Boşluk tuşuna basmak için.
* del\_tusu: Yazıyı silmek için.
* mod\_degistir: Fare kontrol modunu başlatmak için **Mouse** modülünden **smart\_mouse** sınıfını kullanır.
* ara\_enter: Arama yapmak için enter tuşuna basar.
* run: Elleri algılar ve belirli klavye girişlerine karşılık gelen işlevleri çalıştırır.

Def Run()

cv2.VideoCapture ile webcam başlatılır ve kare boyutları ayarlanır.

with self.el.Hands(static\_image\_mode=False, max\_num\_hands=1, min\_detection\_confidence=0.5) as eller ile el algılama başlatılır.

While döngüsü içinde her bir karede elleri algılar ve belirli hareketlere karşılık gelen klavye girişlerini simüle eder.

q tuşuna basıldığında döngüden çıkılır ve webcam kapatılır.

Klavye girişlerine göre, belirli hareketler yapabilirsiniz. Örneğin, parmak pozisyonlarına göre harf yazma, boşluk tuşuna basma, silme, mod değiştirme (fare kontrol modu), Google'a gitme, arama yapma, ve enter tuşuna basma gibi.

Formun Üstü

KAYNAKÇA

* https://en.wikipedia.org/wiki/OpenCV/
* <https://developers.google.com/mediapipe/solutions>
* https://github.com/GuldenizBektas/VirtualKeyboard/
* <https://chat.openai.com/>
* <https://burakbagatarhan.medium.com/mediapipe-kullanarak-i%CC%87leri-seviye-computer-vision-e36dbb4c60f2>