



T.C.
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ

STAJ DOSYASI

BÖLÜMÜ : YAZILIM MÜHENDİSLİĞİ

NUMARASI : 170541023

ADI ve SOYADI : KARDELEN AKTOPRAK



FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
STAJ DOSYASI BELGESİ



BÖLÜMÜ: Yazılım Mühendisliği
Adı Soyadı: Kardelen AKTOPRAK

STAJ DÖNEMİ : Staj1
Okul No : 170541023

STAJ YAPILAN FIRMA / KURUM

ADI : EMELTEK BİYOMEDİKAL Yazılım, Bilişim, Elektronik, AR-GE Danışmanlık Tic. Ltd. Şti.
ADRESİ : Fırat Teknokent Ek Binası No:203 ELAZIĞ
TEL : 0 505 542 62 72 FAX :

STAJ İLE İLGİLİ FIRMA / KURUMSORUMLUSUNUN

ADI ve SOYADI : ENGİN AVCI
UNVANI : Şirket Müdürü

EMELTEK BİYOMEDİKAL SİSTEMLER
YAZ. BİL. HİZMET ELEKTRONİK SİS.
AR-GE DAN. SAN. VE TİC. LTD. ŞTİ.
Fırat Üniversitesi Fırat Teknokent Ek Binası No:203
Hazar Vii (Tarih, Kaşe ve İmza)
ONAY
(Tarih, Kaşe ve İmza)

STAJ BAŞLAMA TARİHİ : 24 / 08 / 2020
STAJ BİTİŞ TARİHİ : 18 / 10 / 2020

Bu kısım Bölüm İşyeri Eğitimi Komisyonu tarafından doldurulacaktır.
STAJ YAPILAN GÜN SAYISI : _____ gün

KABUL EDİLEN GÜN SAYISI : _____ gün

ONAY

Başarılı

Başarisız

(Tarih, Kaşe ve İmza)

24 / 08 / 2020 tarihinden 28 / 08 / 2020 tarihine Kadar bir haftalık çalışma programı

HAFTALAR	YAPILAN ÇALIŞMALAR	SAYFA NO	SAAT
Pazartesi	Staj işleyişi hakkında toplantı.	1	8
Salı	Görüntü işleme tekniği hakkında araştırma.	2	8
Çarşamba	Görüntü işleme alanları hakkında araştırma.	3	8
Perşembe	Kullanılacak programların araştırılması.	4	8
Cuma	Örneklerin incelenmesi.	5	8
ONAY	EMELTEK DİYOMEDİKAL SİSTEMLER YAZ. BİL. İMZA / EŞTEN ALCİ AR-GE DAN. SAN. VE TİC. LTD. ŞTİ. Fırat Üniversitesi Fırat Teknokent Ek Binası No: 203 Hazar V.D.: 333 072 5787 ELAZIG	Toplam Saat : 40	

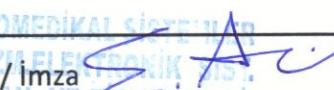
31 / 08 / 2020 tarihinden 04 / 09 / 2020 tarihine Kadar bir haftalık çalışma programı

HAFTALAR	YAPILAN ÇALIŞMALAR	SAYFA NO	SAAT
Pazartesi	Toplantı.	6	8
Salı	Python, PyCharm, Anaconda ve OpenCV kurulumu.	7	8
Çarşamba	Kurulan programların incelenmesi.	8	8
Perşembe	PyCharm kodların yazılması ve çalıştırılması.	9	8
Cuma	No module cv2 hatası.	10	8
ONAY	EMELTEK DİYOMEDİKAL SİSTEMLER YAZ. BİL. İMZA / EŞTEN ALCİ AR-GE DAN. SAN. VE TİC. LTD. ŞTİ. Fırat Üniversitesi Fırat Teknokent Ek Binası No: 203 Hazar V.D.: 333 072 5787 ELAZIG	Toplam Saat : 40	

07 / 09 / 2020 tarihinden 11 / 09 / 2020 tarihine Kadar bir haftalık çalışma programı

HAFTALAR	YAPILAN ÇALIŞMALAR	SAYFA NO	SAAT
Pazartesi	Görsel açma, okuma ve yazma.	11	8
Salı	Görselden yüz ve nesne tespiti.	12	8
Çarşamba	Videodan insan tespiti.	13	8
Perşembe	Webcam'den görüntü alma.	14	8
Cuma	Görüntü boyutlandırma.	15	8
ONAY	EMELTEK DİYOMEDİKAL SİSTEMLER YAZ. BİL. İMZA / EŞTEN ALCİ AR-GE DAN. SAN. VE TİC. LTD. ŞTİ. Fırat Üniversitesi Fırat Teknokent Ek Binası No: 203 Hazar V.D.: 333 072 5787 ELAZIG	Toplam Saat : 40	

14 / 09 / 2020 tarihinden 18 / 09 / 2020 tarihine Kadar bir haftalık çalışma programı

HAFTALAR	YAPILAN ÇALIŞMALAR	SAYFA NO	SAAT
Pazartesi	Görüntü grileştirme.	16	8
Salı	Webcam video kaydetme.	17	8
Çarşamba	Alınan görüntünün bir kısmını ayrı pencerede açma.	18	8
Perşembe	Morfolojik filtreleme.	19	8
Cuma	Toplantı.	20	8
ONAY	EMELTEK BİYOMEDİKAL SİSTEMLER YAZ. BİL. HİZMETLERİ LTD. ŞTİ. AR-GE DANIŞMANLIK VE İLG. LTD. ŞTİ. Kaşe / İmza 	Toplam Saat : 40	

Fırat Üniversitesi Fırat Teknokent Ek Binası No: 203
Hazar V.D.: 333 072 5787 ELAZIG

STAJ İŞLEMLERİ TAKİP KİLAVUZU

1. Staj için gerekli işlemler, öğrencinin Fırat Üniversitesi Teknoloji Fakültesi ilgili Bölümünün Web Sayfasından elde edeceği dilekçeler ile başlatılır. Öğrenci, **Başvuru Dilekçesi (Form-1)** ile Staj yapmak istediği Firmaya/Kuruma başvurur.
2. Staj isteği kabul edilen öğrenci, **Firma/Kurum Bilgi Formunu (Form-2)** Firmaya/Kuruma onaylatarak **Bölüm Staj Komisyonuna** teslim eder.
3. Bölüm Staj Komisyonu toplanarak, **kurumun uygunluğuna** karar verir.
4. **Staj yeri uygun bulunmayan öğrenciler**, ikinci bir kurum bulmak için **1. maddeden itibaren işlemlerini tekrar başlatırlar.**
5. Öğrenci STAJ Komisyonuna **Kabul belgesi**, orijinal imzali ve mühürlü halini teslim eder.
6. 5510 Sayılı Sosyal Sigortalar ve Genel Sağlık Sigortası Kanunu'na göre öğrencilerin iş ve meslek hastalıklarına karşı sigortalanması Fırat Üniversitesi tarafından yapılacaktır.
7. Öğrenci sigorta işlemlerini fakültede bulunan ilgili memurdan takip eder ve satağı ile paralel olarak sigortasının başlamasını sağlar. Sigorta evraki bölüm sekreterliğine teslim edilir.
8. Öğrenci kendisinde muhafaza ettiği **Staj Dosyası** ile birlikte ilgili sömestre başlangıcında kurumuna giderek işlemlerini başlatır.
9. Tüm işlemlerin staja başlamadan 1 hafta önce tamamen bitirilmelidir.
10. Stajını Fırat Üniversitesi Teknoloji Fakültesi **Staj Yönergesi** kapsamında tamamlayan öğrenci, **hazırladığı ve onayları (imza, kaşe vs.) tamamlanmış Staj Dosyasını yarsa ekleri ile birlikte Bölüm Staj Komisyonuna** teslim eder.

*Staj hakkında detaylı bilgi için **Staj Yönergesine** başvurulabilir.

**T.C.
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
STAJ YÖNERGESİ**

BİRİNCİ BÖLÜM
Amaç, Kapsam, Dayanak ve Tanımlar

Amaç

MADDE 1- (1) Stajın amacı;

- a) Öğrencilere lisans programlarıyla ilgili işyerlerini yakından tanıtmak,
 - b) Öğrencilerin öğrenim süreleri içinde kazandıkları bilgi ve deneyimlerini pekiştirmek için uygulama yaparak lisans programlarına ilişkin bilgi ve görgülerini artırma imkânı sağlamak,
 - c) Öğrencilere almış oldukları teorik bilgileri kullanabilme ve uygulamaya aktarma becerisini kazandırmak,
 - ç) Öğrencilere Staj yaptıkları kurum veya kuruluşun görevli personeli ve müşterileri ile uyumlu çalışma ve iyi iletişim kurabilme alışkanlığını kazandırmak,
 - d) Öğrencilerin sektörde yaşanan teknolojik gelişmeleri tanımlarını sağlamak,
- olarak tanımlanmıştır.

Kapsam

MADDE 2- (1) Bu yönerge, Fırat Üniversitesi Teknoloji Fakültesi öğrencilerinin; yurtiçi/yurtdışı, kamuya/özel sektörde veya sivil toplum örgütlerine ait kurum veya kuruluşlarda yapacakları işyeri eğitimleriyle ilgili uygulama esaslarını kapsar.

Dayanak

MADDE 3- (1) Bu yönerge, 3308 Sayılı Mesleki Eğitim Kanunu, 5510 Sayılı Sosyal Sigortalar ve Genel Sağlık Sigortası Kanunu ve Fırat Üniversitesi Önlisans ve Lisans Eğitim-Öğretim Yönetmeliği hükümlerine dayanılarak hazırlanmıştır.

Tanımlar

MADDE 4- (1) Bu yönergede adı geçen;

- a) Bölüm İşyeri Eğitimi Komisyonu: İlgili lisans diploma programının işyeri eğitimi işlemleriini yürütmek, yönetmek ve aynı zamanda işyeri eğitimini değerlendirmek üzere, bölüm kurulu tarafından oluşturulan, ilgili lisans programındaki 3 (üç) öğretim üyesinden oluşan komisyonu,
- b) Denetçi Öğretim Elemanı: staj izlenmesi ve değerlendirilmesi için işyerlerine giderek staj alan öğrenciyi denetlemekle görevli kişiyi,
- c) Fakülte Staj Kurulu: Dekan veya dekan yardımcısı başkanlığında, bölüm başkanları ve Fakülte-Sanayi Koordinatöründen oluşan kurulu,
- ç) Fakülte-Sanayi Koordinatörü: Dekan tarafından görevlendirilen ve öğrencilerin bu yönerge doğrultusunda işyeri eğitimi yapmalarını koordine eden öğretim elemanını,
- d) İşyeri: Öğrencinin eğitim gördüğü lisans programının gereklere ve öğrenme çıktılarına uygun olarak uygulama yapabileceği, işyeri eğitiminin gerektirdiği fiziksel ortam, personel ve diğer nitelikleri taşıyan; öğrencilerin eğitimleri süresince kazandıkları bilgi ve deneyimlerini, işyeri eğitimi yoluyla pekiştirdikleri, mal veya hizmet üreten yurtiçi/yurtdışı kamuya, özel sektörde veya sivil toplum örgütlerine ait kurum veya kuruluşları,
- e) Staj: Öğrencilere, bir yarıyilda (2 sınıf veya 3. sınıf öğrenim gördükleri lisans programı ile ilgili işyerlerindeki faaliyetlere katılmalarını sağlayarak, öğrenim süresince aldıkları bilgilere dayalı uygulama becerisi kazandırma çalışmalarını,
- f) Staj Yetkilisi: İşyeri yönetici tarafından görevlendirilmiş ve öğrencinin eğitim alacağı işyerindeki lisans derecesine sahip kişiyi,
- g) Staj Yöneticisi: Staj eğitiminin uygulanacağı yurtiçi/yurtdışı kamu, özel sektör veya sivil toplum örgütlerine ait kurum veya kuruluşları adına imzalamaya yetkili kişiyi, ifade eder.

İKİNCİ BÖLÜM **Görev, Yetki ve Sorumluluklar**

MADDE 5- (1) Dekan, Staj organizasyonun en üst yetkilisi ve yöneticisidir.

(2) Fakülte Dekanının;

a) Fakülte staj Kurulunu oluşturmak ve Fakülte-Sanayi Koordinatörünü görevlendirmek,

b) Fakülte Staj Kurulunun yaptığı değerlendirme sonuçlarını Öğrenci İşleri Daire Başkanlığına bildirmek, yetkileri arasındadır.

Fakülte İşyeri Eğitimi Kurulunun Görevleri

MADDE 6: (1) Fakülte Staj Kurulunun;

a) Staj verilecek işletmeleri belirlemek,

b) Staj planlanmasını, koordinasyonunu ve uygulanmasını sağlamak,

c) Öğrencilere Staj yeri temini hususunda ilgili birimlerle işbirliği yapmak ve gerekli bilgi sistemini oluşturmak,

d) Staj yapacak öğrencilerin sigorta işlemlerini planlamak ve takip etmek,

e) Öğrencileri, bu yönerge hükümleri doğrultusunda işyeri eğitimi ile ilgili bilgilendirmek,

f) Staj kullanılacak evrakları düzenlemek,

g) Öğrencilerin Staj yapacakları işyerlerine dağılımlarını yapmak,

g) Öğrencilerin Staj süresince hazırlayacakları çalışma faaliyetlerinin kapsamını ve Staj Dosyasının içeriğini belirlemek ve öğrencilere duyurmak,

h) Staj yönetmelik, yönerge usul ve hükümlerine uygun olarak düzenli sürdürülmesi için gerekli önlemleri almak,

i) Staj ilgili olarak doğabilecek aksaklık ve problemleri çözmek,

i) Bölüm Staj komisyonlarında hazırlanan Staj Sicil değerlendirme sonuçlarını karara bağlamak, görevleri arasındadır

Fakülte-Sanayi Koordinatörünün Görevleri

MADDE 7: (1) Fakülte sanayi koordinatörünün;

a) Staj konusunda fakülte ile işyeri arasındaki iletişimini ve koordinasyonu sağlamak,

b) Stajla ilgili olutablecek sorunları çözmek, çözülemeyen hususları fakülte Staj kuruluna iletmek, görevleri arasındadır

Bölüm İşyeri Eğitimi Komisyonunun Görevleri

MADDE 8: (1) Bölüm Staj Komisyonunun;

a) Satış verilecek işletmeleri belirlemek,

b) Öğrencilerin işyeri eğitimi süresince hazırlayacakları çalışma faaliyetlerini ve Staj dosyasının içeriğini belirlemek,

c) Her yarıyıl başında o yarıyıl için gerekli hazırlıkları yapmak, yarıyıl sonlarında teslim edilmiş Staj dosyalarını ve öğrencileri değerlendirmek,

ç) Fakülte işyeri eğitimi kurulunun belirleyeceği görevleri yerine getirmek,

d) İlgili mevzuat hükümleri kapsamında işyeri eğitiminin yürütülmesini sağlamak,

e) İşyeri eğitimiyle ilgili olutablecek sorunları çözmek, çözülemeyen hususları fakülte staj kuruluna iletmek, görevleri arasındadır.

Denetçi Öğretim Elemanının Görevleri

MADDE 9- (1) Denetçi Öğretim Elemanının;

a) Öğrencilerin stajdaki eğitimlerini denetlemek,

b) Denetim sonucunda Staj Denetim Formunu Bölüm staj Komisyonuna teslim etmek, görevleri arasındadır.

İşyeri Yöneticisinin Sorumlulukları

MADDE 10- (1) İşyeri Yöneticisinin;

a) Staj için öğrenci kabul edecek işyerinin kendi personeline sağladığı konaklama, beslenme ve sosyal imkânlardan öğrencilerin de yararlanması için gerekli çabayı göstermek,

b) Öğrenci için gerekli oryantasyon çalışmalarını gerçekleştirmek,

c) Öğrencinin staj bu yönerge esaslarına ve işyeri kurallarına göre yapabilmesi için gerekli şartları taşıyan bir İşyeri Eğitimi Yetkilisini görevlendirmek, sorumlulukları arasındadır.

Staj Yetkilisinin Görevleri

MADDE 11- (1) Staj Yetkilisinin;

- a) Öğrencilerin fakültede almış olduğu teorik ve uygulama esashı bilgi ve becerilerin iş ortamındaki uygulama çalışmaları ile pekişmesini sağlamak,
- b) Öğrenciye haftalık çalışma planı ve sorumluluğu yüklemek,
- c) Öğrencinin işyerindeki sorumluluğunu üstlenmek,
- c) Öğrencilere meslek yeterliliğini ve disiplinini kazandırmak,
- d) Öğrencinin haftalık hazırladığı işyeri eğitimi dosyasını ve yaptığı faaliyetleri değerlendirmek,
- e) İşyeri eğitimi bitiminde, işyeri eğitimi değerlendirme formunu hazırlayıp Bölüm staj Komisyonuna kapalı zarf içinde gizli olarak göndermek, görevleri arasındadır.

Öğrencilerin Görev ve Sorumlulukları

MADDE 12- (1) Öğrenciler, staj Kabul formunun imzalanan işyerlerinde yapmak zorundadırlar.

(2) Öğrenciler, staj süresince işyeri eğitimi haftalık çalışma planını uygulamakla yükümlüdürler.

(3) Öğrenciler, staj yapacakları işyerinin kurallarına, mevzuatına ve Yükseköğretim Kurumları Öğrenci Disiplin Yönetmeliğine uymak zorundadırlar.

(4) Öğrenciler, günlük çalışma kayıtlarını içeren haftalık çalışma raporunu, sonraki haftanın ilk mesai günü bitimine kadar staj yetkilisine vermek zorundadırlar.

(5) Öğrenciler, işyerinden izinsiz ayrılamaz. staja devam zorunluluğu vardır. Hastalık, birinci derece yakınlarının vefati veya benzeri acil durumlar dışında izin kullanılamaz. İzinli olarak ayrılması gereken durumlarda işyeri eğitimi yetkilisi tarafından onaylı izin formu düzenlenir ve staj dosyasında sunulur.

(6) Öğrenciler, işyerindeki sendikal etkinliklere katılmazlar.

ÜÇÜNCÜ BÖLÜM

Stajla İlgili Düzenlemeler

Stajın Özellikleri

MADDE 13- (1) Öğrenciler, staj son sınıfın güz (2. sınıf) yaz döneminde (3. sınıf) yaz döneminde için devam zorunluluğu gerektiren dersi bulunmaması halinde gerçekleştirir. Öğrencinin 2. veya 2.sınıfında yaz döneminde $20+20=40$ işgünü staj yaparlar

Staj Süresi

İsyeri Eğitimi Başvurusu

MADDE 14- (1) Fakülte-Sanayi Koordinatörü, İşyeri Eğitimi Yetkilisi ve işyeri eğitimi yapacak olan öğrenci tarafından imzalanan İşyeri Eğitimi Sözleşmesi 3 (üç) nüsha olarak çoğaltılar ve taraflara verilir.

(2) Öğrenciler bütün derslerinden başarılı olsalar bile, işyeri eğitimi yapacakları süreler için kayıt yaptırıkmak ve katkı paylarını ödemek zorundadırlar.

İsyeri Eğitimi Dosyası Oluşturma

MADDE 15- (1) Her öğrenci, "staj dosyası" hazırlamak zorundadır.

(2) Staja başlayan öğrenci, staj Dosyasını ilgili bölüm sekreterliğinden temin eder.

(3) Staj Dosyası; öğrenci bilgilerinin bulunduğu kapak, staj dosyası, öğrenci tarafından hazırlanmış imzalı ve onaylı haftalık raporlar, işyeri eğitimi denetim formları ve staj yetkilisi tarafından kapalı zarf içerisinde yer alan dönem sonu staj sivil fişi oluşur.

(4) Staj Dosyası işyeri eğitimi süresince Staj Yetkilisinde bulunur, Staj sonunda değerlendirme amacıyla Bölüm staj Komisyonuna teslim edilir.

İsyeri Eğitimi Dosyasının Teslimi

MADDE 16- (1) Öğrenci, staj dosyasını işyeri eğitiminin bitiş tarihinden itibaren en geç bir hafta içerisinde Bölüm staj Komisyonuna elden teslim eder veya iadelî taahhütlü posta yoluyla gönderir.

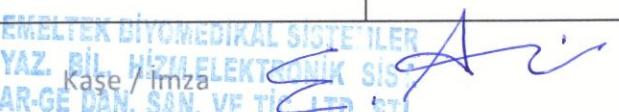
(2) Staj dosyasını, süresi içerisinde teslim etmeyen öğrencinin stajdan geçersiz sayılır.

Hastalık ve Kaza Halleri

MADDE 17- (1) 5510 Sayılı Sosyal Sigortalar ve Genel Sağlık Sigortası Kanunu'na göre öğrencilerin iş ve meslek hastalıklarına karşı sigortalanması Fırat Üniversitesi tarafından yapılacaktır. Ancak, iş yerinin kusurundan dolayı meydana gelebilecek iş kazaları ve meslek hastalıklarından, 3308 sayılı Kanunun 25. Maddesine göre, işveren sorumludur.

(2) staj sırasında hastalanan ve resmi kurumlarca belgelenmek üzere hastalığı 7 (yedi) günden fazla süren veya herhangi bir kazaya uğrayan öğrencinin adı, soyadı, hastalıkın ve kazanın mahiyeti, işyeri tarafından Fakülte-Sanayi Koordinatörüne bildirilir. Bu durumlarda Sosyal Güvenlik Kurumu hüküm ve uygulamaları geçerli olup, telafi konusunu Fakülte İşyeri Eğitimi Kurulu belirler.

İÇİNDEKİLER

	Sayfa N o
Staj işleyisi hakkında toplantı.	1
Görüntü işleme tekniği hakkında araştırma.	2
Görüntü işleme alanları hakkında araştırma.	3
Kullanılacak programların araştırılması.	4
Örneklerin incelenmesi.	5
Toplantı.	6
Python, PyCharm, Anaconda ve OpenCV kurulumu.	7
Kurulan programların incelenmesi.	8
PyCharm kodların yazılması ve çalıştırılması.	9
No module cv2 hatası.	10
Görsel açma, okuma ve yazma.	11
Görselden yüz ve nesne tespiti.	12
Videodan insan tespiti.	13
Webcam'den görüntü alma.	14
Görüntü boyutlandırma.	15
Görüntü grileştirme.	16
Webcam video kaydetme.	17
Alınan görüntünün bir kısmını ayrı pencerede açma.	18
Morfolojik filtreleme.	19
Toplantı.	20
ONAY	 EMELTEK BİYOMEDİKAL SİSTEMLER YAZ. BİL. HİZ. ELEKTRONİK SİSTEMLER AR-GE DAN. SAN. VE TİC. LTD. ŞTİ. <small>Fırat Üniversitesi Fırat Teknokent Ek Binası No: 203 Hazar V.D.: 333 072 5787 ELAZIG</small>

Yapılan Çalışmanın Konusu: Staj işleyişi hakkında toplantı.

İlk olarak staj işleyişi, staj kuralları ve staj süresince yapılacak işler hakkında konuşulmak üzere bir toplantı yapıldı. Staj yapmış olduğum şirketin çalışma alanları ve projeleri hakkında kısa bir bilgi verildi ve benim hangi alanda daha başarılı olabileceğim konusunda konuştuğum. Staj süresince görüntü işleme alanında çalışma yapılmasına karar verildi ve bu konu hakkında istenilen araştırmalar ve çalışmalar için bir plan oluşturuldu. Stajın ilk 5 günü içerisinde araştırma yapılacak ve araştırmalar sonucu tekrar toplantı yapılması kararlaştırıldı.

Daha sonra yapılacak toplantıda bir proje konusu belirleneceği ve sonraki günlerde belirlenen konu üzerine çalışma yapılması ve yapılan çalışma sırasında oluşabilecek hatalar ve bu hataların nasıl düzeltilmesi gerekiği hususunda bilgilendirme yapılacağı belirtildi.

Toplantı bitiminde araştırmalarıma başladım. Görüntü işlemenin ne olduğu ve hangi alanlarda kullanıldığı konusunda araştırmalar yaptım. Görüntü işleme ile yapılan projeleri inceledim. Araştırmalarım sonucu edindiğim bilgileri not ettim.

Yapılan Çalışmanın Konusu: Görüntü işleme hakkında araştırma.

Bu gün ki çalışmam da görüntü işlemenin ne olduğu, hangi amaç için kullanıldığı, kullanım alanlarının ne olduğu ve hangi dillerin, ortamların ve kütüphanelerin kullanılabileceği hakkında araştırma yaptım.

Görüntü işleme, görüntüyü dijital form haline getirip bazı işlemler gerçekleştirerek spesifik görüntü elde etmemizi sağlar. Böylece görüntünün ilstenilen kısmını ele alarak yararlı bilgiler çıkarabiliriz. İstediğimiz bir görüntü üzerinde değişiklikler yapabiliriz, kalite düşüşlerini indirgeyip görüntüyü iyileştirebiliriz. Örneğin uydudan alınan görüntülerini bu teknikle anlaşılır hale getirebiliriz.

Görüntü işleme tekniği aşağıda belirtilen sistemlerde yaygın olarak kullanılabilmektedir:

- Yüz tanıma ve güvenlik sistemleri.
- Demografik bilgi analizi.
- Trafik, astroloji, radar ve fotoğraf endüstrisi uygulamaları.
- Uydu görüntülerinden yararlanma.
- Radyoloji alanı. (ultrason, tomografi vb.)
- Askeri endüstri.
- Tıp alanı. (çeşitli organ yapılarının ayırt edilmesi gibi.)

Görüntü işlemenin amacı 5 grupta ele alınır:

- Görselleştirme
- Görüntü keskinleştirme ve restorasyon
- Görüntü alımı
- Desen tanıma
- Görüntü tanıma

Görüntü işleme de; c++, c#, Java veya Python dilleri kullanılabilir. Her dil için gerekli kütüphaneler farklıdır. Başlangıç için Python dilinde yazılması ve yüksek performans elde etmek için OpenCV kütüphanesinin kullanılması daha iyi bir seçim olacaktır. Anaconda ön yükleyicisinden yararlanarak PyCharm veya Visual Studio Code ortamlarını kullanabiliriz.

Yapılan Çalışmanın Konusu : Görüntü işleme alanları hakkında araştırma.

Bugün görüntü işleme alanlarının neler olduğu ve ne amaçla kullanıldığı araştırdım. Bunlardan bazıları aşağıda yer almaktadır.

- **Yüz tanıma:** Görüntünün analizi sonucunda desen karşılaştırması yapılarak belirli bir kişiyi tanımlayan biyometrik uygulamadır. Genellikle güvenlik alanında kullanılır. Ayrıca aynı mantıkta nesne tanıma insan bedenlerinin tanınması gibi uygulamalarda yapılabilir.
- **Trafik ve radar:** Otoyollarda ve kavşak kesişimlerinde bulunan kameralardan elde edilen video görüntülerinden araç sayımı, ihlal ve plaka tespiti, sinyalizasyon sistemlerinin yönetimi gibi uygulamalar yapılmaktadır.
- **Uydu:** Uydudan alınan görüntülerin iyileştirilip daha net görüntüler elde edilebilir. Uydu görüntüler ile nüfus yoğunluğu tespiti gibi uygulamalar yapılmaktadır.
- **Radyoloji ve Tıp:** Tibbi görüntüleme yöntemleri ile tomografi, ultrason, MR ile çekilmiş görüntüler üzerinden hastalıkların teşhisi için görüntü işleme kullanılır. Çeşitli organ yapılarının ayırt edilmesinde yardımcıdır.
- **Askeri endüstri:** Hem su altı görüntülerini hem de uygun görüntülerinin anlaşılmasıında kullanılır. Hedef tanıma ve izleme de kullanılır.

Yapılan Çalışmanın Konusu : Kullanılacak programların araştırılması.

Bugün görüntü işlemede kullanabileceğim Python, PyCharm, Visual Studio Code, Anaconda ve OpenCV hakkında araştırma yaptım.

Python: Yüksek seviyeli, nesne yönelimli, modüler etkileşimli yorumlanabilir bir programlama dilidir. Çok sayıda işi az kod satırında yapabilmeyi sağlar. Girintilere dayalı basit sözdizimi, dilin öğrenilmesini ve akılda kalmasını kolaylaştırır. Bu nedenle görüntü işleme alanında Python dilini kullanmanın başlangıç için avantaj sağlayacağımı düşündüğüm için bu dili kullanacağım.

PyCharm: JetBrains tarafından geliştirilen Python geliştirme ortamıdır.

Visual Studio Code: Microsoft tarafından geliştirilmiş güçlü, kolay kurulumlu ve paketlere ulaşım rahatlığıyla kullanıcıları bir ortamdır. C++, C#, Java, Python gibi programlama dillerini destekler. Ayrıca Windows, macOS ve Linux işletim sistemlerinde kullanılabilen bir yazılım geliştirme ortamıdır.

Anaconda: Python, R programlama dili, very bilimi, bilimsel araştırmalarla uğraşmak isteyenler için geliştirilen bir ön yükleyicidir.

OpenCV: Açık kaynak kodlu görüntü işleme ve makine öğrenimi için geliştirilmiş bir yazılım kütüphanesidir. OpenCV kütüphanesi içerisinde görüntü işleme ve makine öğrenimine yönelik 2500'den fazla algoritma bulunduğu belirtilmektedir. Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma, plaka tespiti vb. Birçok işlem yapılabilir.

Numpy: Python'da bilimsel hesaplama yapmak için kullanılan bir pakettir. Çok boyutlu diziler ve matrisler üzerinde işlem yapmayı sağlar. Matematiksel, mantıksal, sıralama, seçme, giriş/çıkış gibi çeşitli yordamlar sağlayan bir Python kütüphanesidir.

Tarih : 27 / 08 / 2020	 EMELTEK BIYOMEDİKAL SİSTEMLER YAZ. BİLGİ İMZALELE ELETRONİK SİSTEMLER AR-GE DAN. SAN. VE TİC. LTD. ŞTİ. Fırat Üniversitesi Fırat Teknokent Ek Binası No: 203 Hazar V.D.: 333 072 5787 ELAZIG	Sayfa No : 4
------------------------	--	--------------

Yapılan Çalışmanın Konusu : Örnek çalışmaların incelenmesi.

Bugün görüntü işleme kullanılarak yapılan çalışmaları inceledim. Yüz tanıma, nesne tespiti, trafikte kullanımı, askeri endüstri alanındaki çalışmalar, uzay görüntülerinden yararlanma, tip alanında kullanım gibi alanlarda görüntü işlemenin kullanılması ne gibi faydalalar sağlar ve ne tür projeler yapılabilir ile ilgili araştırmalar yaptım.

Yüz tanıma alanındaki kullanımın güvenlik sistemleri için yararlı olacağını öğrendim. Örneğin kurum ve kuruluşlardaki çalışanları tespit etme, giriş çıkışların kontrolü gibi uygulamalarda kullanılabiliyor. Trafikte kural ihlallerinin tespiti, plaka tanıma, trafik yoğunluğunun kontrolünün sağlanması vb. uygulamalarda kullanılarak ulaşımın kontrol altında tutulmasına fayda sağlıyor. Tip alanında kullanımıyla birçok hastalığın tanısında yardımcı oluyor. Bununla ilgili kanser tanısı ve tadavisinde kullanılan çeşitli uygulamalar mevcut. Görüntülerdeki karmaşıklık, bozukluk gibi sorunların çözümünde kullanılabilmektektir. İnsan bedenlerini tespit ederek bir bölgedeki yoğunluğu, kişi sayısını ve giriş çıkışları kontrol edebilmek için faydalı.

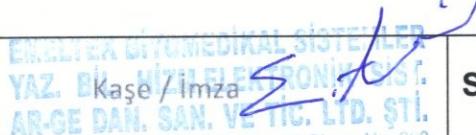
Bugün ki araştırmamda bunlar gibi birçok uygulamayla görüntü işlemenin hayatımızın önemli birçok alanında kolaylık sağladığını ve bu konuda birçok proje geliştirebileceğini öğrendim.

Yapılan Çalışmanın Konusu : Toplantı.

Bugün staj yaptığım kurumda bir toplantı gerçekleştirdik. Hafta boyunca yaptığım araştırmaları ve neler öğrendiğimi anlattım.

Bundan sonraki günlerde neler çalışacağımı konuştuk. Çalışmalarımda PyCharm ortamı kullanmam istendi. Görsel okuma ve görsel üzerinde değişiklik yapma, renk değiştirme, görselden insan tespit etme gibi uygulamaların kodunu araştıracağım ve bu kodları çalıştırıp denemem gerektiği konuşuldu. Daha sonrasında video üzerinden insan tespiti, kaemeradan canlı görüntü alma, alınan görüntünün kaydedilmesi, boyutlandırılması gibi çalışmalar yapacağım ve filtrelemeleri kullanacağım konusunda planlama yaptık.

Bunları araştırip kodları denemek için Udemy uygulamasından ders aldım. Ayrıca diğer platformlardaki kodları da inceleyip PyCharm ortamında yazarak deneyeceğim. Alacağım hataların çözümlerini araştırip gidereceğim. Çıktıları kontrol edip tekrar hata olup olmadığını denetleyeceğim.

Tarih: 31 / 08 / 2020	 ERZİTER EĞİTİM MEDİAL SİSTEMLERİ YAZ. BİL. UYD. ELEKTRONİK SİST. AR-GE DAN. SAN. VE TİC. LTD. ŞTİ. Kaşe / İmza	Sayfa No: 6
-----------------------	---	-------------

Yapılan Çalışmanın Konusu: Python, PyCharm, Anaconda ve OpenCV kurulumu.

Bugün görüntü işleme için kullanacağım programları indirdim. Önce Python'un 3.8.5 sürümünü yükledim ve kurulumunu yaptım. Daha sonra PyCharm 2020.2.1 sürümünü indirip kurdum. Anaconda'nın ise Python 3.8 sürümlerini destekleyen son sürümünü indirdim. (anaconda3)

- Python için; <https://www.python.org/>
- PyCharm için; <https://www.jetbrains.com/pycharm/>
- Anaconda için; <https://www.anaconda.com/>
- OpenCV için; <https://opencv.org/>

Resmi web sitelerini kullandım.

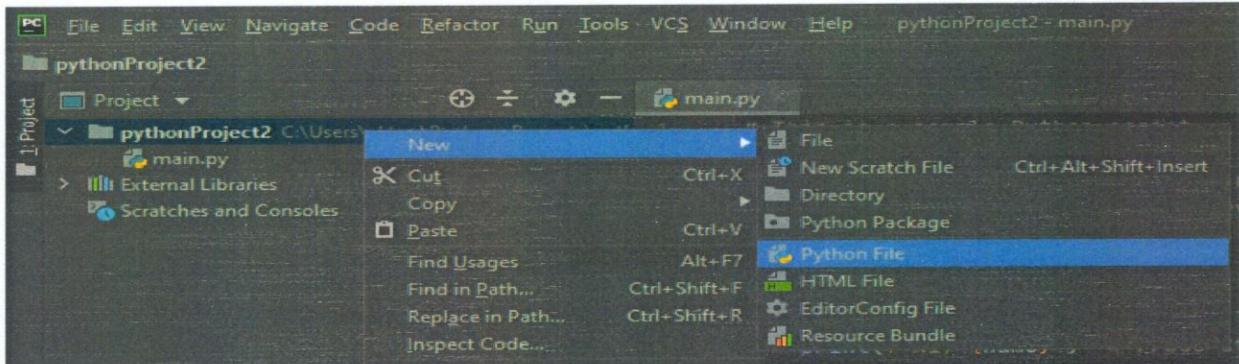
Tüm kurulumları sorunsuz hallettikten sonar hepsini tek tek açarak denedim ve hata olup olmadığını kontrol ettim. *Python IDLE* ile küçük bir kod satırı yazarak çıktı verip vermediğini kontrol ettim. *Anaconda navigator*'ü açıp desteklediği programların doğru yükleyip yüklenmediğini kontrol ettim. *PyCharm*'da ise yeni proje sayfası açıp proje dosyalarının nerden eklendiği, kodun nerden çalıştırıldığını ve yapılan çalışmaların nasıl ve nereye kaydedildiğini inceledim.

PyCharm ortamında daha önce çalışmadığım için bu konu hakkında araştırma yapmam gereki. Kodlar nerede eve nasıl yazılır? Nasıl çalıştırılır? Kuralları nelerdir? Kütüphaneler nasıl eklenir? Sorularının cevabını almış olduğum kurslardan ve internet üzerinden araştırarak gösterilen adımları denedim.

Tarih : 01 / 09 / 2020	EMELTEK DİYOMEDİKAL SİSTEMLER YAZ. BİLF. HİZMETLERİNİN ELEKTRONİK SİSTEMLERİ AR-GE DAN. SAN. VE TİC. LTD. ŞTİ. Fırat Üniversitesi Fırat Teknokent Ek Binası No: 203 Hazar V.D.: 333 072 5787 ELAZIG	Sayfa No : 7
------------------------	--	--------------

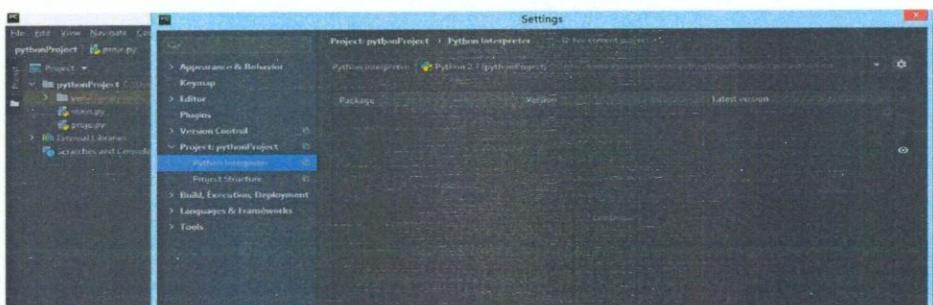
Yapılan Çalışmanın Konusu : Kurulan programların incelenmesi.

Bugün PyCharm'da kodların nerede yazıldığı ve dosyaların nereden eklendiğini araştırdım. FILE-New Project butonunu seçip yeni proje dosyası oluşturduğum ve dosyama isim verdim. Şekil-1'de proje klasörümé python dosyasını nasıl eklediğimi gösterdim.

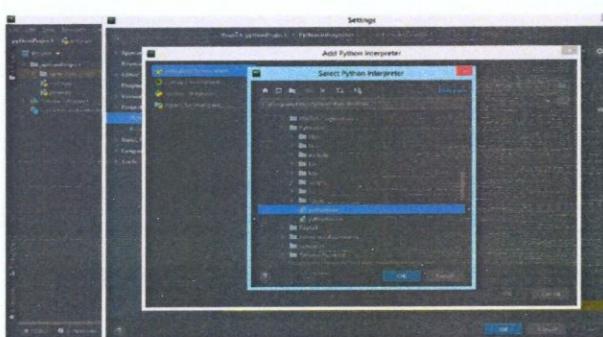


Şekil-1

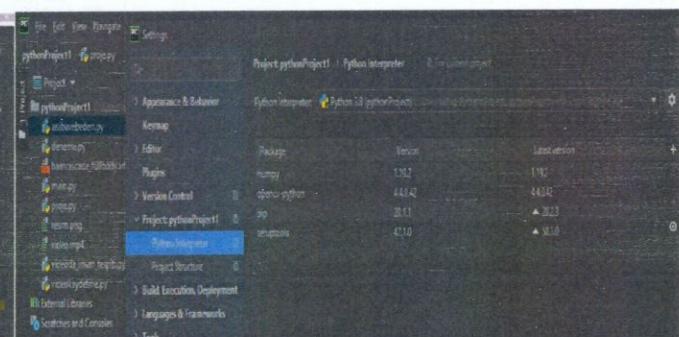
Eklediğim python dosyasına önce OpenCV ve diğer kütüphaneleri import edebilmek için Setting-Python Interpreter butonlarını tıklayıp Şekil-2'de açılan pencereden Şekil-3'de göründüğü gibi python.exe dosyasının dosya yolunu ekledim. Daha sonra Şekil-4'de göründüğü gibi kullanacağım kütüphaneleri eklemiş oldum.



Şekil-2



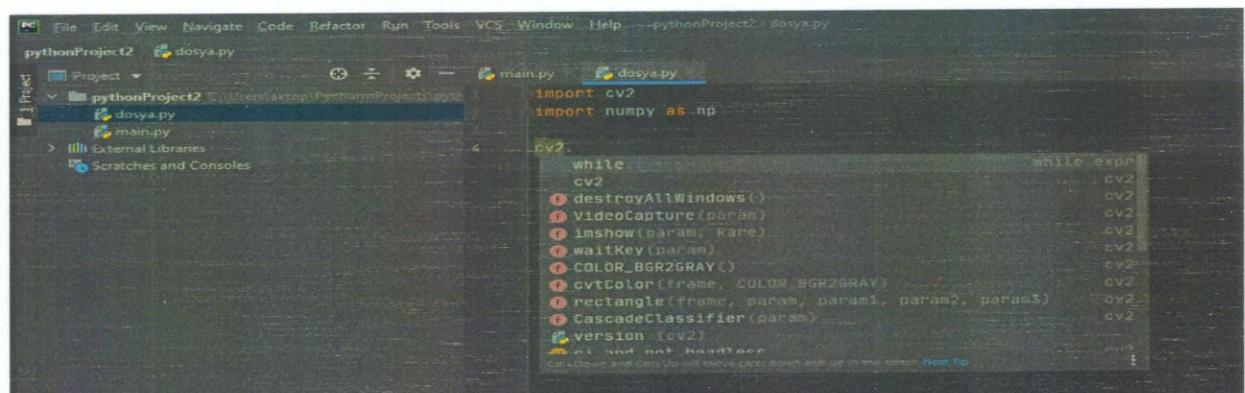
Şekil-3



Şekil-4

Yapılan Çalışmanın Konusu : PyCharm'da kodların yazılması ve çalıştırılması.

Elendegim Python dosyasının içerisinde **import cv2** ve **import numpy** yazdıktan sonra opencv metotlarının sorunsuz eklenip eklenmediğini denemek için Şekil-5'deki gibi **cv2**. Yazdığımızda fonksiyonların görünmesi gereklidir.



The screenshot shows the PyCharm interface with a project named 'pythonProject2'. In the code editor, the file 'dosya.py' is open, displaying the following code:

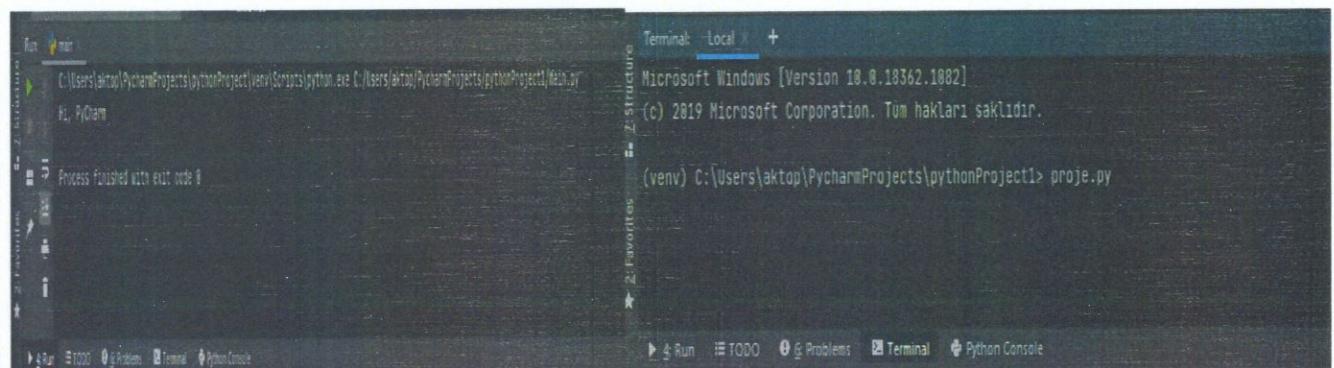
```
import cv2
import numpy as np

cv2.
    while
        cv2
            destroyAllWindows()
            VideoCapture(param)
            imshow(param, kare)
            waitKey(param)
            COLOR_BGR2GRAY()
            cvtColor(frame, COLOR_BGR2GRAY)
            rectangle(frame, param, param1, param2, param3)
            CascadeClassifier(param)
            version(cv2)
```

A code completion dropdown menu is open over the 'cv2.' part of the code, listing various cv2 module functions such as 'destroyAllWindows()', 'VideoCapture()', 'imshow()', etc.

Sekil-5

Yazdığım kodu çalıştırmak için çalıştır butonuna tıklandıktan sonra *run* bölümünde şekil-6'daki gibi bir bildirim gelmektedir. Bu bildirim kodun sorunsuz çalıştığını gösterir. Daha sonra şekil-7'deki gibi Terminal bölümüne **python dosyaadi.py** yazıp *Enter* tuşuna bastığımızda çıktıyi görebiliyorum.



The screenshot shows the PyCharm interface with a terminal window. The terminal output shows:

```
Process finished with exit code 0
```

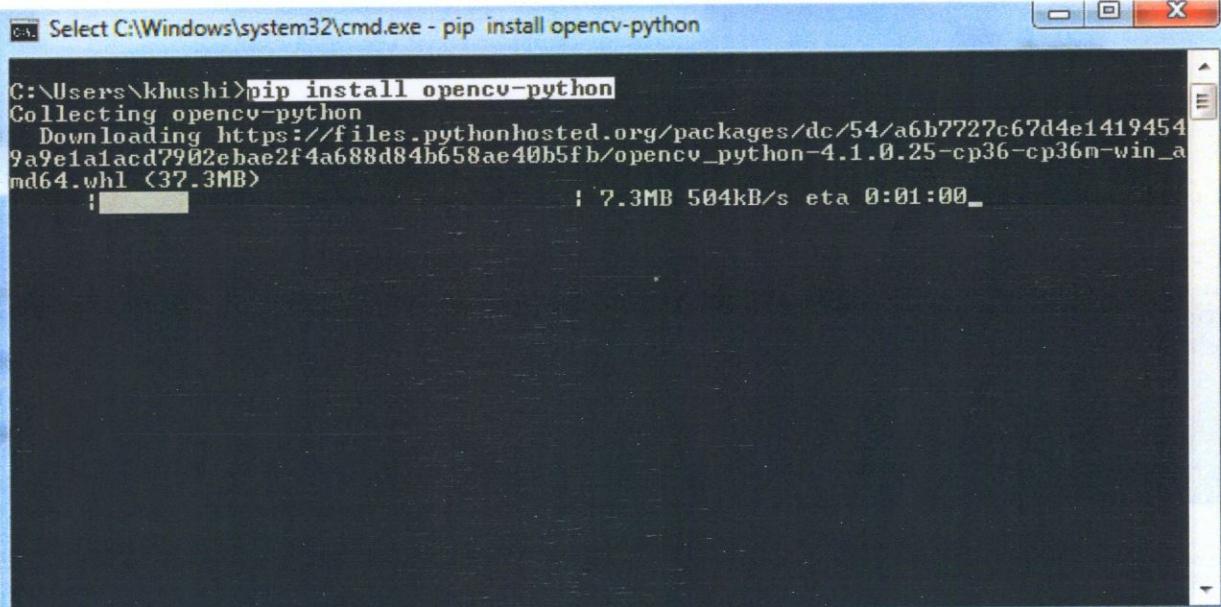
This indicates that the code ran successfully.

Sekil-6

Sekil-7

Yapılan Çalışmanın Konusu : No module cv2 hatası.

OpenCV import edilmiş olduğu halde python yorumlayıcı çalıştırıldığında ModuleNotFoundError: No module named 'cv2' hatası veriyor. Bu durumda yapılması gereken bilgisayarın komut isteminden pip install opencv-python yazarak opencv kütüphanesini python'a eklemiş olmamız gerekiyor. Böylece kod tekrar çalıştırıldığında bu sorun ortadan kalkacaktır. Şekil-8'de bunun için yaptığım işlem görünmektedir. Yükleme tamamlandıktan sonra opencv aktarılmış oluyor.



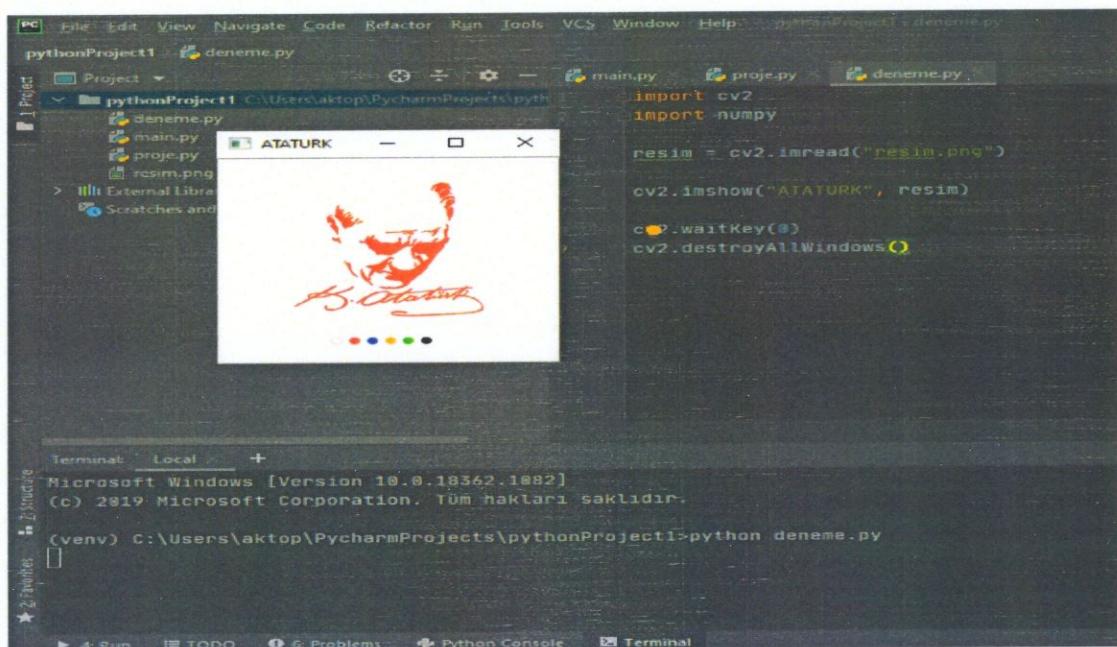
```
Select C:\Windows\system32\cmd.exe - pip install opencv-python

C:\Users\khushi>pip install opencv-python
Collecting opencv-python
  Downloading https://files.pythonhosted.org/packages/dc/54/a6b7727c67d4e14194549a9e1a1acd7902ebae2f4a688d84b658ae40b5fb/opencv_python-4.1.0.25-cp36-cp36m-win_amd64.whl (37.3MB)
    |██████████| 7.3MB 504kB/s eta 0:01:00
```

Şekil-8

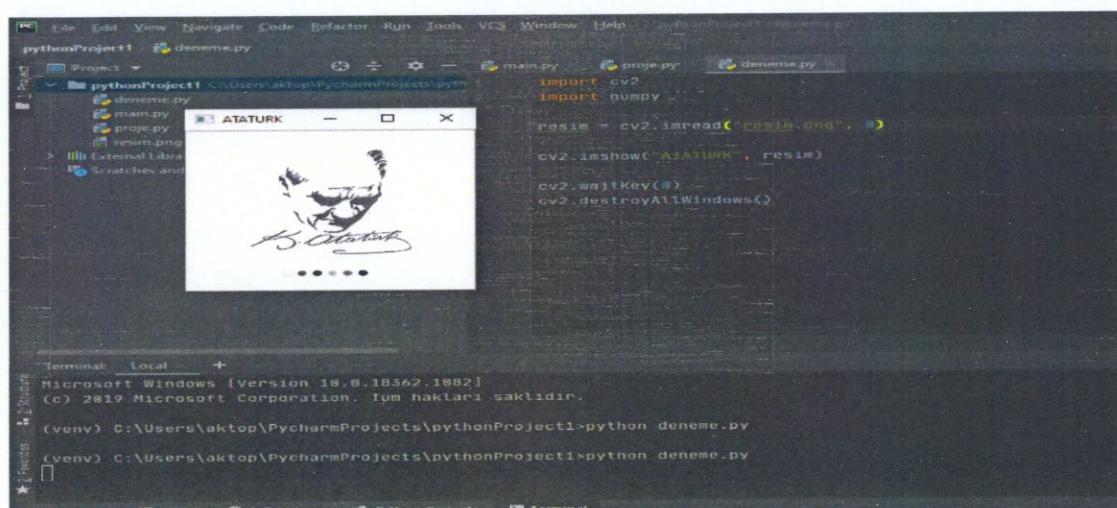
Yapılan Çalışmanın Konusu : Görsel açma, okuma ve yazma.

Bugün PyCharm üzerinden kendi eklediğim bir görseli ekrana yazdırıldım. **imread()** fonksiyonu görüntütüyü okumayı, **imshow()** fonksiyonu ise ekranda görüntülmeyi sağlar. Imread() fonksiyonunun içersine “...” içerisinde görüntünün tam adı yazılarak bir değişkene atanır. Imshow() fonksiyonunun içersine ise (“çıkıya verilecek isim”,değişken) yazılır. Bu durumda çıktı şekil-9'daki gibi olacaktır.



Şekil-9

Resmi grileştirmek ise, **imread()** fonksiyonunun içine resmin yolunu kopyaladıktan sonra 0 yazılır. Burada 0, resmin piksellerini gri tona çevirmemizi sağlar. Böylece imread foksiyonu resmi gri tonda okumuş olacaktır. Çıktı şekil-10'daki gibidir.

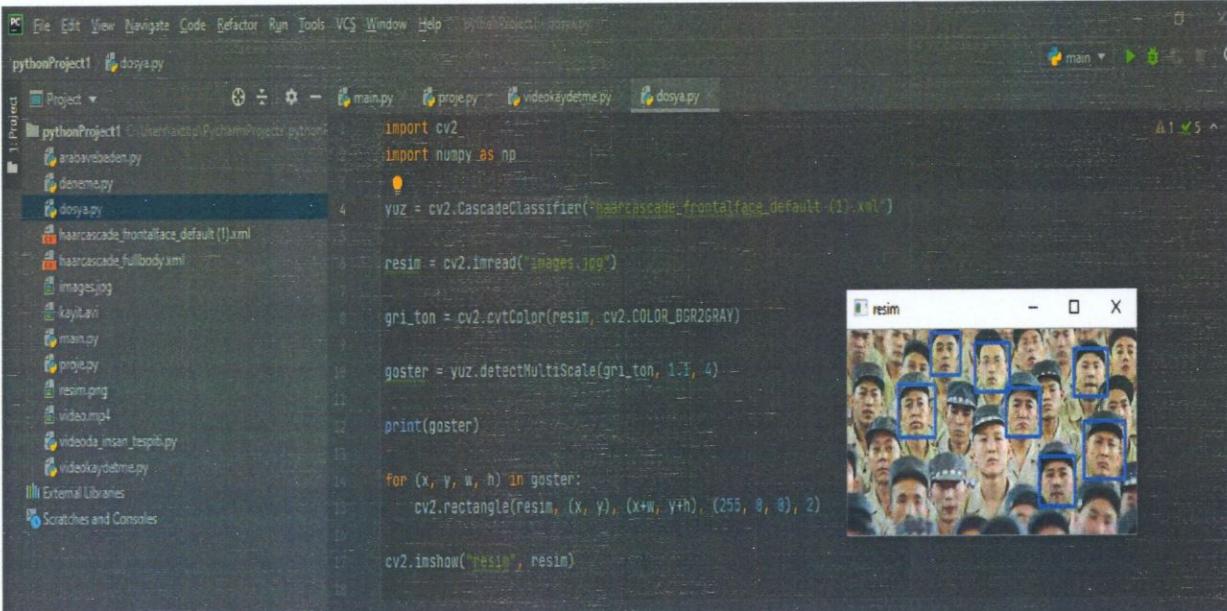


Şekil-10

Yapılan Çalışmanın Konusu : Görsel üzerinden yüz ve göz tespiti.

Daha önce görsel okuma ve ekrana yazdırmayı denemiştim. Bugün ise bir görsel üzerinden yüz ve göz tespiti yapmayı denedim. Bunun için *haarcascade_frontalface_default.xml* dosyasını kullanmam gerekti. Burada önemli olan nokta şu, eğer görselde yüz ve göz tam görünmüyorsa, insanları kafaları sağa, sola, yukarı, aşağı dönük şekilde ise program bu yüzleri tanımıyor. Bunun için farklı xml dosyaları ile kodu değiştirmemiz gereklidir.

Şekil-11'de bunun için yazdığım kod ve çıktısı görülmektedir.



The screenshot shows the PyCharm IDE interface. On the left, the project structure is visible with files like main.py, videokeydetme.py, and dosya.py. The dosya.py file is open, containing the following Python code:

```
pythonProject1 dosya.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject1 dosya.py
Project 1. Project
pythonProject1 C:\Users\asus\PycharmProjects\pythonProject1
arabesbeden.py
deneme.py
dosya.py
haarcascade_frontalface_default.xml
haarcascade_fullbody.xml
images.jpg
kayitavi
main.py
projek
resim.png
video.mp4
videoda_isinan_tespiti.py
videokaydetme.py
External Libraries
Scratches and Consoles

import cv2
import numpy as np

yuz = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

resim = cv2.imread("images.jpg")

gri_ton = cv2.cvtColor(resim, cv2.COLOR_BGR2GRAY)

goster = yuz.detectMultiScale(gri_ton, 1.1, 4)

print(goster)

for (x, y, w, h) in goster:
    cv2.rectangle(resim, (x, y), (x+w, y+h), (255, 0, 0), 2)

cv2.imshow("resim", resim)
```

The right side of the screen shows the resulting image from the code execution, titled "resim". The image displays a group of people with several blue rectangles drawn around their faces, indicating where the algorithm has detected them.

Sekil-11

Yapılan Çalışmanın Konusu : Videodan insan tespiti.

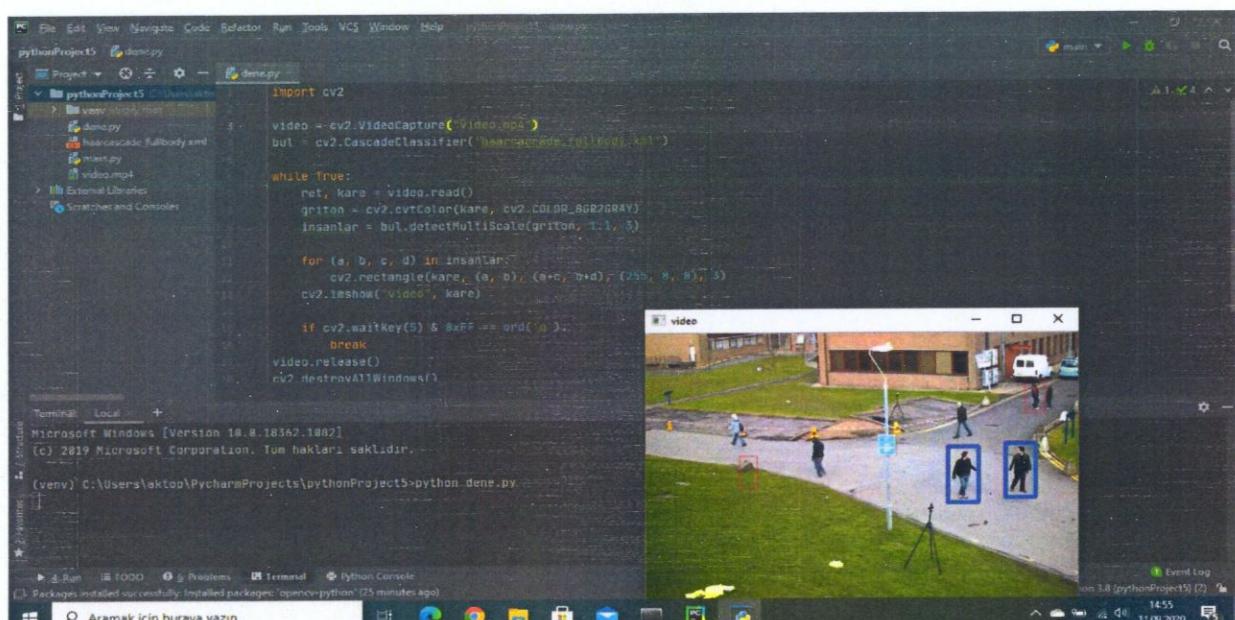
Bugün ki çalışmam da proje dosyamın içerisinde attığım .avi veya .mp4 uzantılı bir videoyu ekran'a nasıl yazdıracağımı ve videodaki insanları tespit etmek için dikdörtgen içeresine almayı öğrendim. Bunun için öncelikle videomu proje dosyamın içine kopyalıyorum. **videoCapture()** fonksiyonu içeresine görüntülemek istedigim videonun adını yazdım ve bunu bir değişkene atadım. Daha sonra kameranın çalışıp çalışmadığını kontrol edip ret değişkeni ile devamlı True dönmeyi sağlıyoruz. İkinci değişkenimiz ise atadığımız videonun okunup görüntüülendiği değişkendir.

İnsan bedenlerini tespit etmek için github'da hazır bulunan xml dosyalarını kullandım. Bu xml dosyaları *cascadeClassifier* sınıfına atanıyor. **cvtColor()** fonksiyonu ise görüntü üzerinde renk değiştirebilmeyi sağlıyor ve iki parameter alıyor. İlk parametresine videoyu atadığımız değişkeni, ikinci parametresine ise resmi çevirmek istedigimiz renklere ait fonksiyon yazılır.

Yüz tanıma işlemi *cascadeClassifier* sınıfındaki **detectMultiScale()** fonksiyonu kullanılıyor. Bu fonksiyon ise 3 parametre alıyor. İlki *cvtColor()* fonksiyonunu atadığımız değişken, ikincisi ölçeklendirme faktörü yani resmin boyutu, üçüncüüsü minimum komşu sayısıdır.

For döngüsünün içerisinde (*koordinatlar, genişlik, yükseklik*) yazılıyor. Bu döngü sayesinde video da birden fazla yüz varsa hepsi dikdörtgen içeresine alır. **Rectangle()** fonksiyonu ise dikdörtgen çizmeye yarar. İlk parametresine videoyu, ikinci parametresine sol üst köşenin koordinatları, üçüncüye sağ alt köşenin koordinatları, dördüncüye dikdörtgenin rengini(BGR ile), son parameter ile ise dikdörtgenin kenar kalınlıkları belirleniyor. Daha sonra ise *imshow()* ile videoyu görüntülüyoruz.

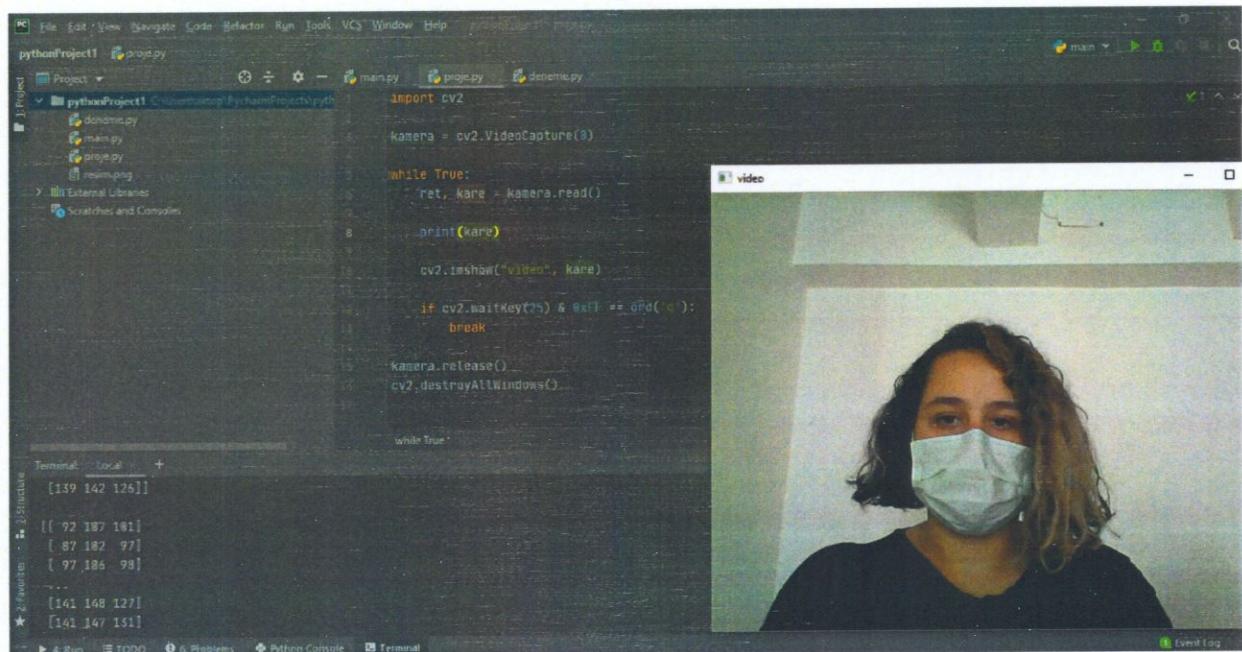
Bu işlemler uygulanıp kod yazıldıktan sonra çıktı **Şekil-12**'deki gibi oluyor.



Şekil-12

Yapılan Çalışmanın Konusu : Kameradan görüntü alma.

Bugün bilgisayarımın kamerasından canlı görüntü almayı öğrendim. Bunun için yine **videoCapture()** fonksiyonunu kullandım fakat içeriğe 0 yazdım. Daha sonra önceki çalışmam da yazdığım gibi iki değişken kullandım. **ret** değişkeni ile True döndürüp, **kare** değişkeni ile görüntüyü okudum ve print diyip yazdırıldım. Şekil-13'de aldığım görüntü ve yazdığım kod görülmektedir.

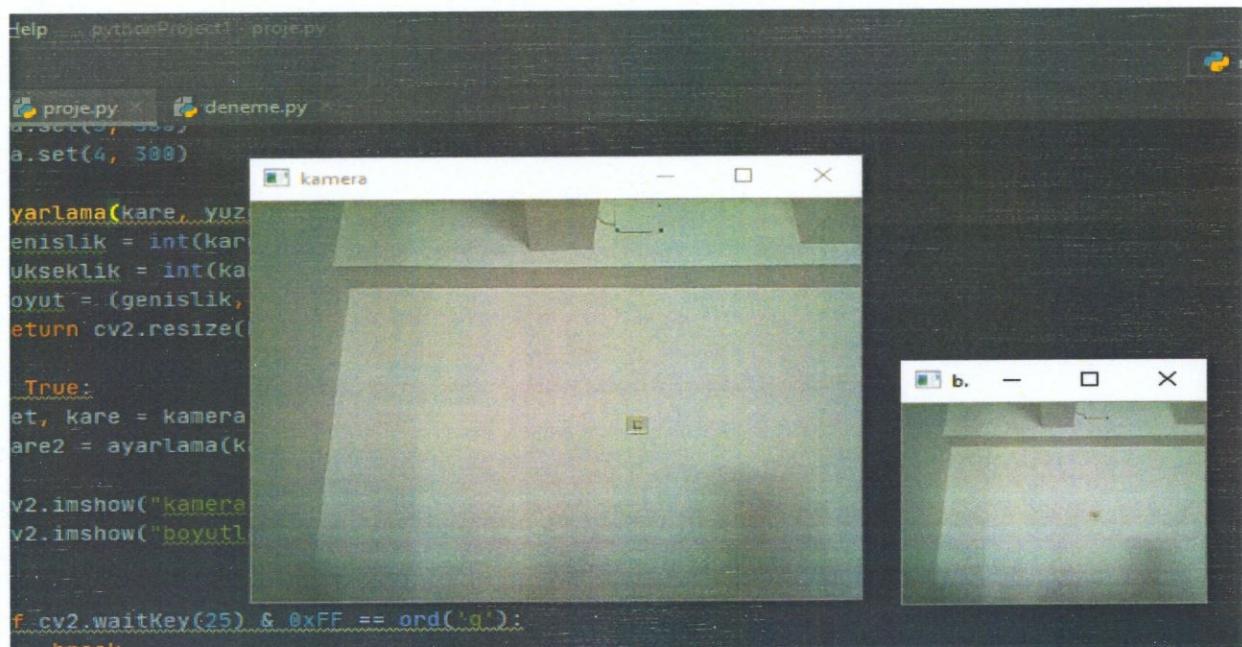


Sekil-13

Burada **if cv2.waitKey(25) && 0xFF == ord('q')** satırı ile yapılmak istenen; waitKey(25) her 25 milisaniyede bir görüntüyü kapat anlamına geliyor ve q'ya basıldığı anda break devreye girip durduruyor. Local'de görünen matrisler ise print(kare) satırı sayesinde görünümeye ve her 25 milisaniyede bir görüntüden bir kare alındığını göstermektedir.

Yapılan Çalışmanın Konusu : Görüntü boyutlandırma.

Kameradan aldığım görüntünün boyutunu belirlemek için görüntüyü aldığımız **değişkenin ismi.set()** fonksiyonunu kullandım. Bu fonksiyon iki parameter alıyor. İlk parametreye *yükseklik(4)* veya *genişliği(3)*, ikinci parametreye ise boyutun ne olacağını yazdım. Ancak bu tür yapılan boyutlandırma orantısal değil boyutsal büyütmedir. Çıktı **Şekil-14**'de görülmektedir. Burada kamera adlı görüntü normal boyutu, boyutlandırılmış adlı görüntü küçültülmüş halini gösteriyor.



Şekil-14

Pikselleri orantısal büyütmek için;

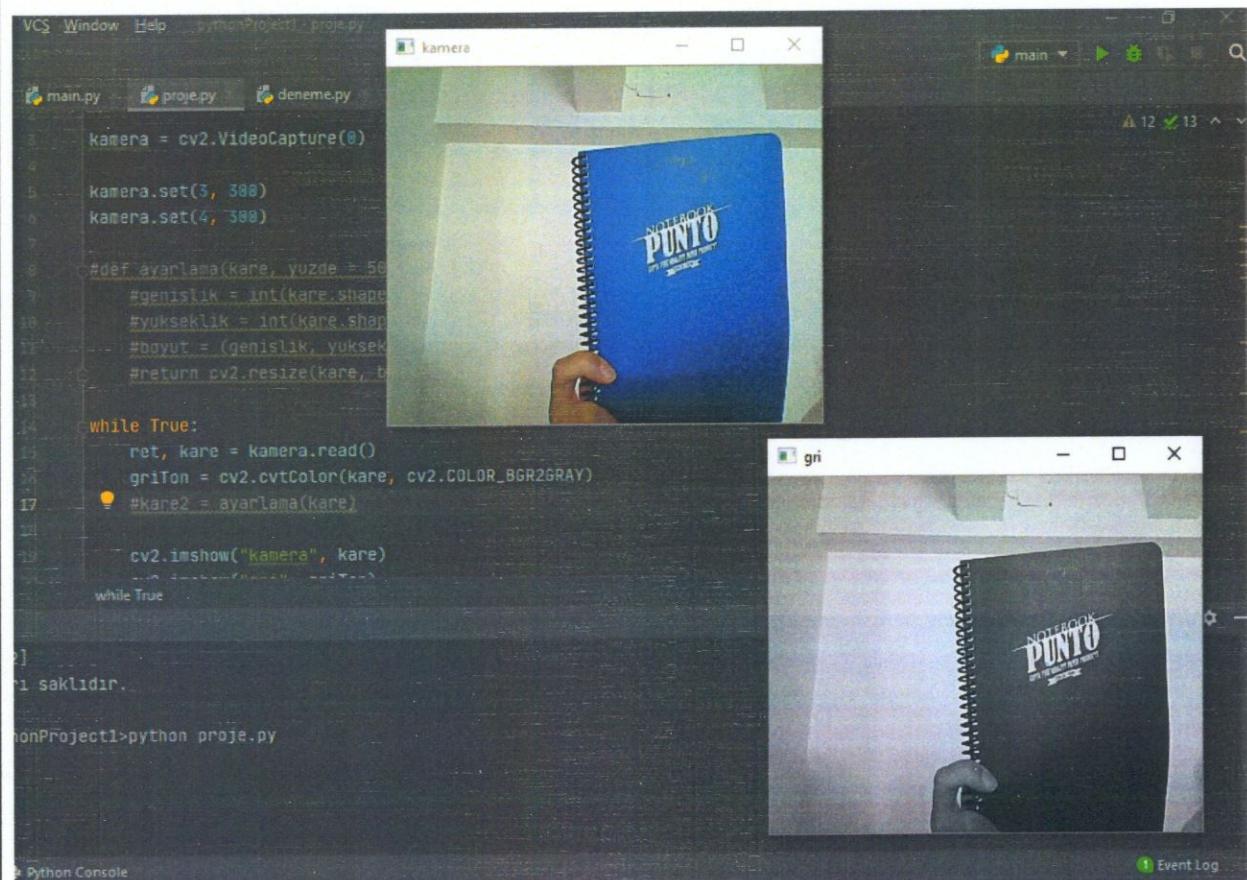
```
Def ayarlama(kare,yuzde=75); //75 oranında ayarla  
    genislik=int(kare.shape[1*yuzde/100]) //1.parametrede dönen genişliğin %75'ini alır  
    yukseklik=int(kare.shape[0*yuzde/100]) //0.parametre yüksekliği dönderir.  
    boyut=(genislik,yukseklik)  
    return cv2.resize(kare,boyut, interpolation=cv2.INTER_AREA)  
    //Burada yapılan iş yeniden ayarlanan yapıyı karesel olarak döndürmek.
```

Kod parçası yazılmıştır.

Yapılan Çalışmanın Konusu : Kameradan alınan görüntüyü grileştirme.

Bugün kameradan aldığım canlı görüntüyü gri ton'da göstermeyi denedim. Bunun için yazılan kod normal görüntü almak için yazılan kodun aynısıdır. Yanlızca ek olarak *While* döngüsü içerisinde kameradan görüntüyü okuduktan sonra *griTon=cv2.cvtColor(kare, cv2.COLOR_BGR2GRAY)* satırını ve *cv2.imshow("gosterilecekisim",griTon)* satırını ekleyip çalıştırırmak yeterlidir.

Şekil-15'de bunun için yazlığım kod ve çıktısı görülmektedir. Anlaşıllır olması için kameradan hem normal görüntüyü hemde gri tonla gösterilen görüntüyü aynı kod içerisinde çalıştırıldım.



Sekil-15

Yapılan Çalışmanın Konusu : Kameradan alınan görüntüyü kaydetme.

Bugün bilgisayarımın kamerasından aldığım görüntüyü kaydetmeyi denedim. Bunun için kameradan görüntüyü aldıktan sonra;

fource=cv2.VideoWriter_fource(*"XVID")

//kaydedilecek videonun formatını belirlemek için.

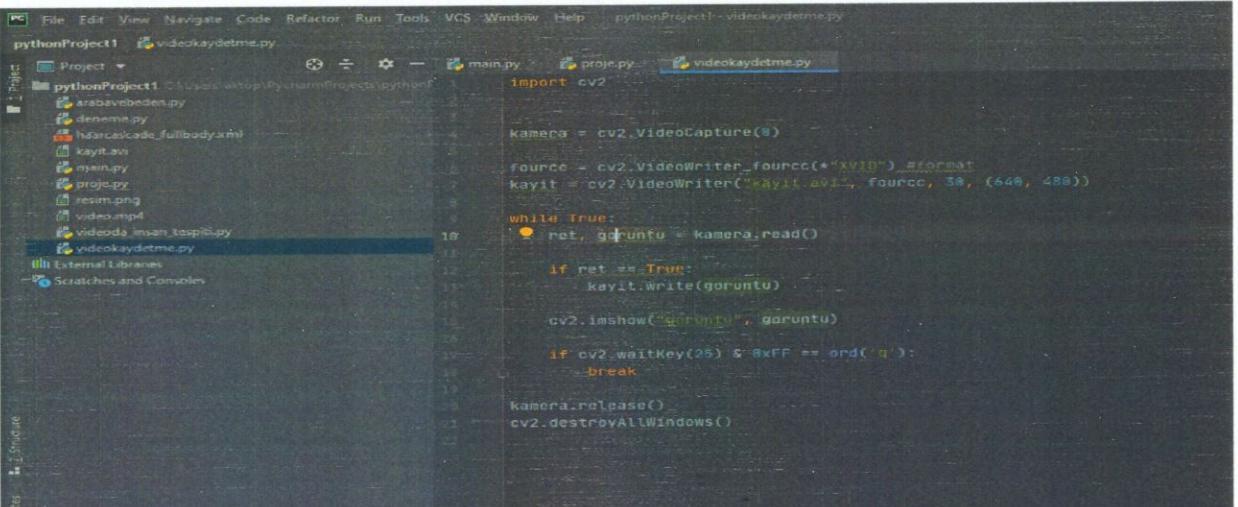
Kayıt=cv2.VideoWriter("kayıt.avi",fource,30,(640,480)) //Bu satır kaydedilecek videonun adını, kayıt formatını,fbs, kaydedilecek videonun piksellerini belirliyor.

Daha sonra while döngüsünün içince;

If ret == True: //kameranın açık olup olmadığından doğrulanması.

Kayıt.write(goruntu) //alinına görüntünün kayıt değişkeninde kaydolduğu satır.

Satırlarının eklenmesi yeterlidir. Kodun tamamı şekil-16'de görülmektedir.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject1--videokaydetme.py
pythonProject1 C:\Users\aktop\Desktop\PycharmProjects\pythonProject1
Project
  pythonProject1
    - arabavbeden.py
    - deneme.py
    - haarcascade_fullbody.xml
    - kayıt.avi
    - main.py
    - proje.py
    - resim.png
    - video.mp4
    - videoda_ismen_tespiti.py
    - videokaydetme.py
  External Libraries
  Scratches and Corrections
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject1--videokaydetme.py
import cv2

Kamera = cv2.VideoCapture(0)

source = cv2.VideoWriter('fource(*"XVID")', format
kayıt = cv2.VideoWriter("kayıt.avi", fource, 30, (640, 480))

while True:
    ret, goruntu = Kamera.read()

    if ret == True:
        kayıt.write(goruntu)

        cv2.imshow("goruntu", goruntu)

        if cv2.waitKey(25) & 0xFF == ord('q'):
            break

Kamera.release()
cv2.destroyAllWindows()
```

Sekil-16

Şekil-17 ise .avi formatında kayıt isimli videonun proje dosyamız içerisinde kaydolduğunu göstermektedir.

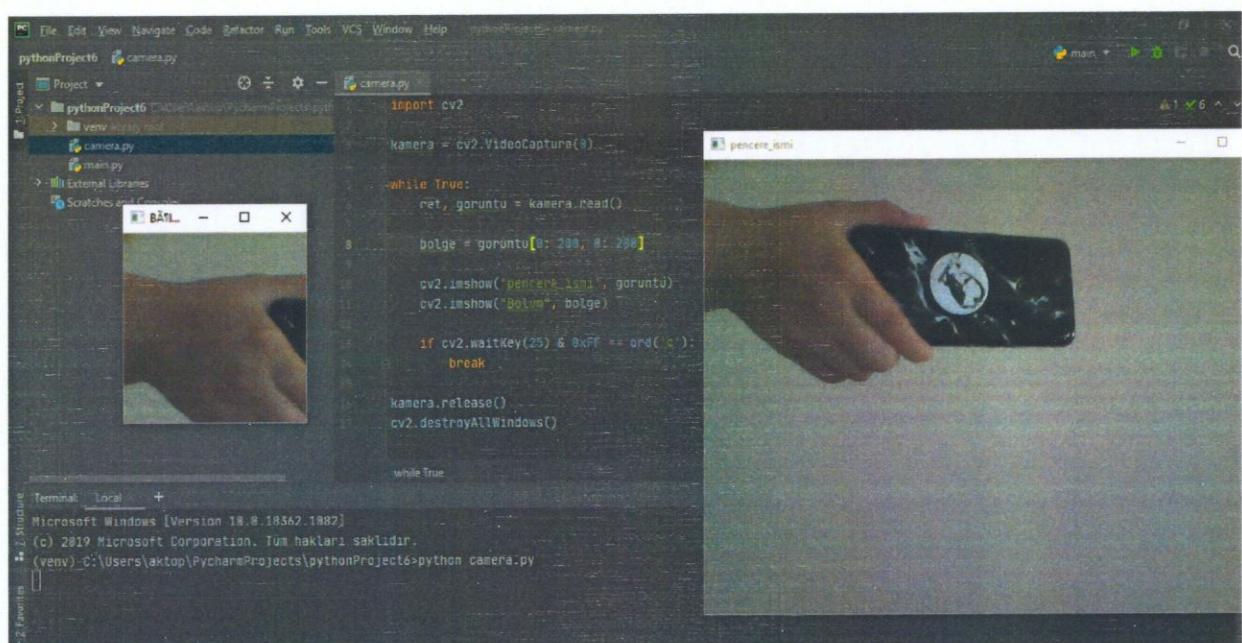


Sekil-17

Yapılan Çalışmanın Konusu : Kameradan alınan görüntünün bir kısmını ayrı pencerede açma.

Bugün yaptığım çalışmada kameradan aldığım görüntünün belirli piksellerdeki bir bölümünü ayrı pencerede açmayı denedim. Bunun için while döngüsü içerisinde bölge isimli bir parametreye ana görüntüden alacağımız bölgenin koordinatlarını yazdım.

Bu işlem; **bolge = goruntu[0:200, 0:200]** satırında gerçekleşti. Daha sonra cv2.imshow diyerek yeni bölgeye vereceğim ismi ve bölgenin değişkenini yazdım. Bu işlem; **cv2.imshow("bölüm", bolge)** Satırında gerçekleşti. Böylece kodun tamamını çalıştırıldığında şekil-18'deki gibi bir görüntü elde ettim.

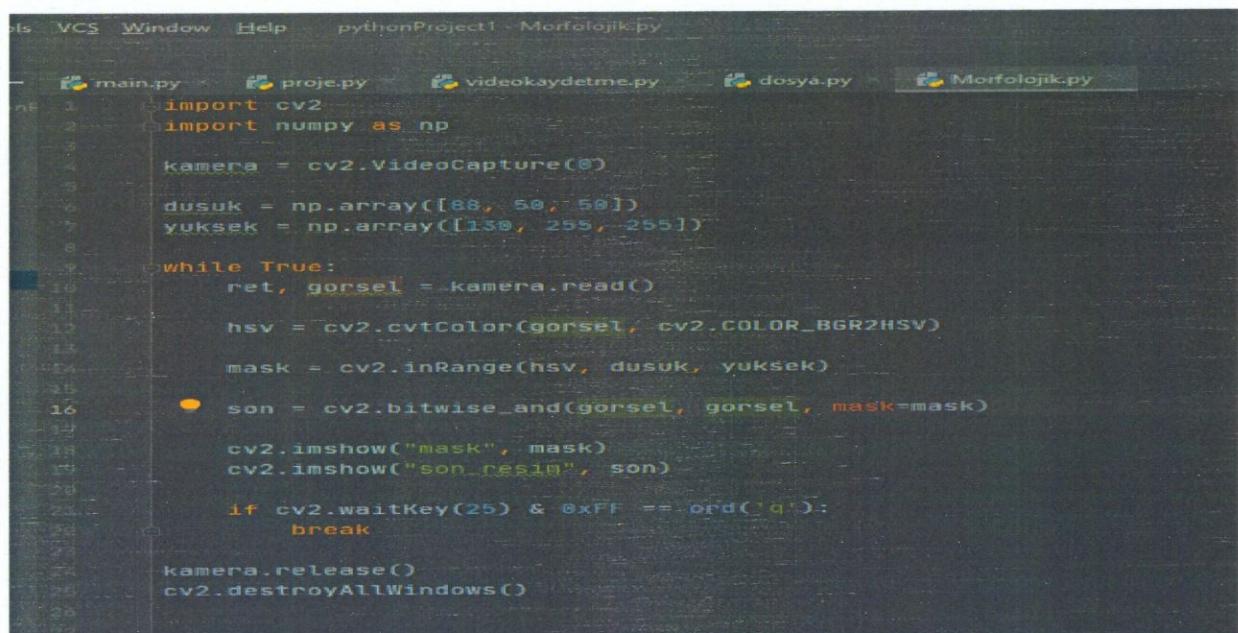


Sekil-18

Yapılan Çalışmanın Konusu : Morfolojik filtreleme.

Bugün morfolojik filtreleme kullanarak kameradan görüntü aldım. Burada numpy sınıfından *array* kullanarak istediğim filtrelemenin en düşük ve yüksek değer aralıklarını belirledim. Benim verdığım değer aralıkları mavi ton için kullanılan aralıktır. Daha sonra diğer yaptığım uygulamalardan farklı olarak BGR yerine HSV kullandım. **HSV** renk uzayıdır ve *Hue*, *Saturation* ve *Value* terimlerine karşılık gelen renkleri içerir. BGR'dan farklı olarak HSV *dolgunluk* ve *parlaklık* içeren renk değerlerini kullanır.

Daha sonra mask değişkeninde **inRange()** fonksiyonunu kullanarak filtrelemek istediğim aralığı belirtiyorum ve bu filtrelenen aralık beyaz renkte görünüyor. Kodun tamamı **Şekil-19**'da görülmektedir.

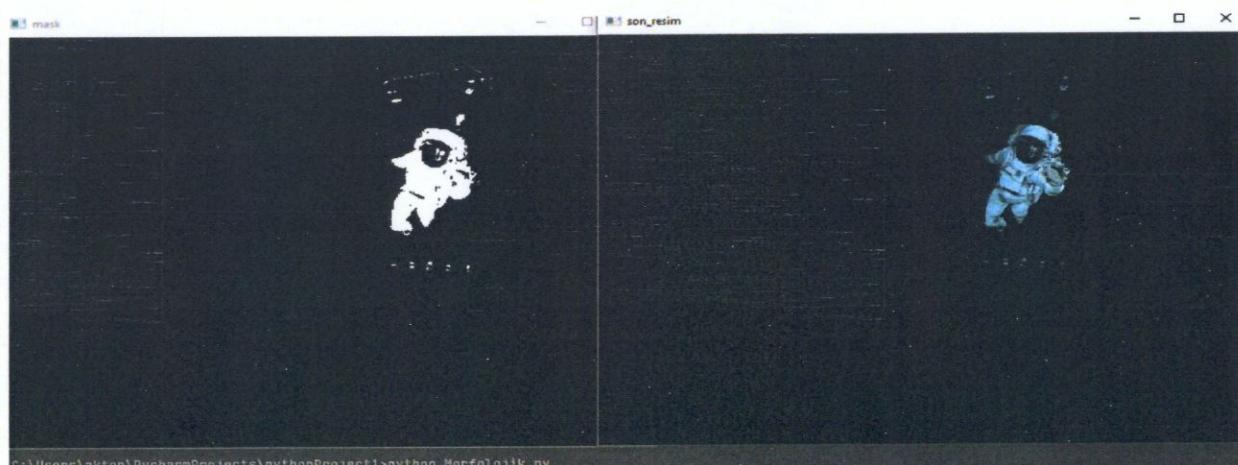


```
VCS Window Help pythonProject1 - Morfolojik.py
main.py proje.py videokaydetme.py dosya.py Morfolojik.py

1 import cv2
2 import numpy as np
3
4 kamera = cv2.VideoCapture(0)
5
6 dusuk = np.array([88, 50, 50])
7 yuksek = np.array([130, 255, 255])
8
9 while True:
10     ret, gorsel = kamera.read()
11
12     hsv = cv2.cvtColor(gorsel, cv2.COLOR_BGR2HSV)
13
14     mask = cv2.inRange(hsv, dusuk, yuksek)
15
16     son = cv2.bitwise_and(gorsel, gorsel, mask=mask)
17
18     cv2.imshow("mask", mask)
19     cv2.imshow("son_resim", son)
20
21     if cv2.waitKey(25) & 0xFF == ord('q'):
22         break
23
24 kamera.release()
25 cv2.destroyAllWindows()
```

Şekil-19

Şekil-20'de ise çıktı görünümektedir. Kameraya telefonumdan gösterdiğim bir görseli mask penceresinde beyaz filter ile gösterip diğer kısımları siyah aldım.



Şekil-20

Yapılan Çalışmanın Konusu : Toplantı.

Bugün stajimin son günü olduğu için yaptığım tüm çalışmanın teslimi ve kontrolü için bir toplantı yaptık. Ve staj süresince yaptığım işlerin bana neler kazandırdığını, neler öğrendiğimi ve öğrendiğim şeyleri ne derece uygulayabildiğini tartıştık.

Ben yaptığım staj süresince daha önce hiç çalışmamışım bir alanda birşeyler öğrenebildiğim için çok mutlu oldum ve burada öğretiklerimi staj bitiminden sonra da devam ettireceğim. Görüntü işlemenin hayatımızın birçok alanında kullanıldığını ve aslında birçok alanda bize kolaylıklar sağladığını öğrendim. Eğlenceli ve kapsamlı bir konu üzerinde çalıştığım için mutluyum.

Resimler, videolar ve kendi kameralardan kaydedebildiğimiz görüntüler üzerinde değişiklikler yapabilmek, nesne ve insan tespiti yapmak, görüntülerimiz renklerini değiştirmek, boyutlarını değiştirmek ve bunları filtrelemek gibi öğrendiğim birçok farklı konu oldu. Ve bunları daha önce hiç çalışmamışım bir ortamda yazarak denedim. Farklı kütüphaneler kullandım.

Kodları yazarken farklı kaynaklardan farklı dersler almam gereki. Karşılaştığım hataları araştırarak düzeltmeyi denedim ve birçogunda başarılı oldum. Kaynaklardan çözümüne erişemediğim hatalar için şirket çalışanlarından yardım aldım.

Staj sürecimin zorlu ancak verimli geçtiğini düşünüyorum ve burada edindiğim bilgileri eğitim hayatımın geri kalanında ve iş hayatımda devam ettireceğim ve bu alanla ilgili daha fazla bilgi edinip daha çok uygulama yapacağım.