



Fenerbahçe Üniversitesi
BLM 102 – Algoritmalar ve Programlama II
Görüntü İşleme Yöntemleri ile Üretim Hattı Analizi
Proje İçeriği

Veriliş Tarihi: 10.05.2020

Teslim Tarihi ve Yeri: 10.06.2020, Ders Saatlerinde, Elektronik olarak

1. Tanım:

Fabrikalarda üretim hatlarından çıkan ürünlerin analizi yüksek çözünürlüklü, yüksek FPS (Frame per Second) özellikli kameralar ile yapılmaktadır. Bir üretim hattından çıkan üç farklı tür ürünün, video kayıtları boyunca kaç'ar adet üretildiğinin tespit edileceği bir sistem geliştirilecektir. OpenCV (Open Computer Vision) kütüphanesi kullanılacaktır.

2. Proje Ekibi:

Proje 4 kişiye kadar olan ekiplerden oluşacaktır. Her bir proje ekibinin bir sorumlusu olacaktır. Öğrenciler 1, 2, 3 veya 4 kişilik kendi proje ekiplerini ve proje sorumlusunu belirlemelidirler.

Ekiplerin kurulması ve proje sorumlusunun belirlenmesi en geç **12.05.2020** tarihine kadar tamamlanmalıdır. Ekip sorumluları, LMS'te açılmış olan "Proje Ekip Sorumluların Takımlarını Bildirmesi" başlığının altına, ekip üyelerinin isimlerini göndermelidirler.

LMS adresi: <http://levent.tc/lms/>

3. Kullanılacak Araçlar:

Proje kapsamında, ödev ve LAB'larda kullanılan Microsoft'un derleyicisi olan Visual Studio Community kullanılacaktır.

4. Tasarım Gereksinimleri

Proje kapsamında bir üretim hattında akan 3 farklı ürünün, verilen video süresince kaç'ar adet üretildiğinin tespit eden bir sistem geliştirilecektir. Görüntü işleme teknikleri uygulanacaktır. Görüntü işleme çalışmaları için sıklıkla tercih edilen açık kaynak kodlu bir kütüphane olan OpenCV kütüphanesi kullanılacaktır.

Visual Studio ile C++ dilinde OpenCV kütüphanesinin kullanılması için öncelikle kütüphanenin indirilmesi gerekmektedir. OpenCV'nin birçok versiyonu bulunmaktadır. Projede 2.4.13.6 versiyonu kullanılacaktır. Aşağıdaki linkten indirilebilir.

<https://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.4.13/opencv-2.4.13.6-vc14.exe/download>

İndirilen dosya açıldığında kütüphaneyi çıkartacağı bir lokasyon soracaktır. Herhangi bir yere çıkartabilirsiniz. Çıkarılan dosyaların olduğu klasörde build ve sources klasörleri bulunmaktadır. Sources klasörü OpenCV'nin kaynak

Öğretim Elemanı: Dr. Vecdi Emre Levent, emre.levent@fbu.edu.tr, İzinsiz Kopyalanamaz

kodlarının bulunduğu klasördür. Build klasörü ise Windows x84 ve x64 mimarileri için derlenmiş olan dosyalar bulunmaktadır. OpenCV kaynak kodları derlenme ihtiyacı bulunmamaktadır. Dolayısıyla sadece build klasörünün içerisindeki dosyalar kullanılacaktır.

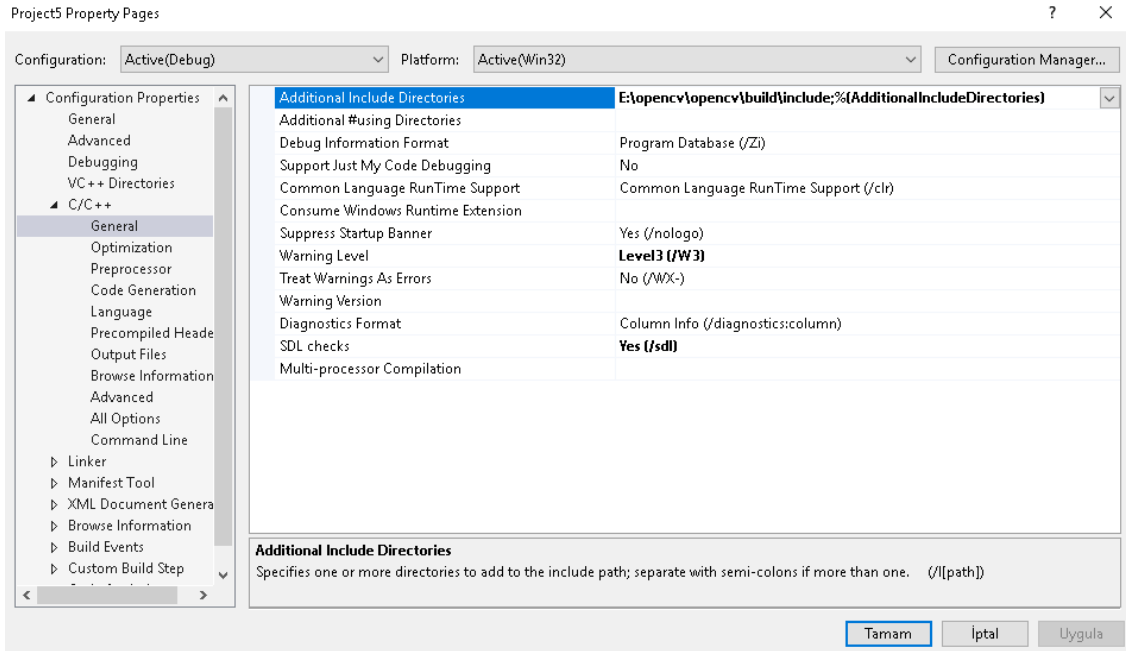
Visual Studio'da boş bir C++ projesi oluşturularak, aşağıda verilen başlangıç kodu projeye ekleyiniz.

<http://levent.tc/courses/blm102/projeler/proje3/main.cpp>

Kodda kullanılan blob.jpg resmini aşağıdaki linkten indirebilirsiniz. Kodun içerisinde kullanılan blob.jpg adresini bilgisayarınızdaki lokasyon için düzenleyiniz.

<http://levent.tc/courses/blm102/projeler/proje3/blob.jpg>

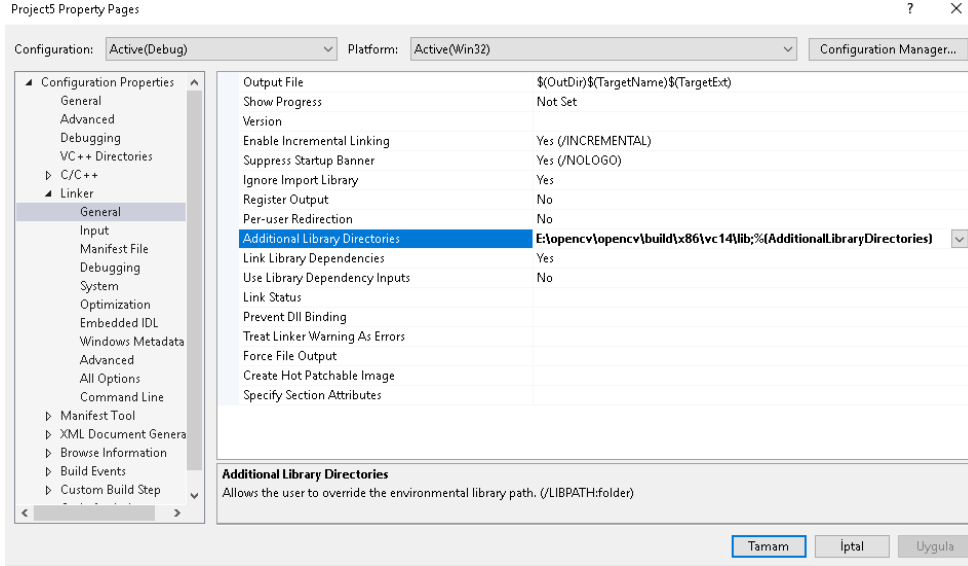
Bu aşamada projedeki OpenCV kütüphaneleri tanıtılmadığı için hata verecektir. OpenCV kütüphanelerini projeye eklemek için sağ tarafta bulunan Solution Explorer penceresinden oluşturulan projenin ismine (solution değil) sağ tıklayıp properties açılmalıdır. Açılan pencerede C/C++ bölümünden General seçilir. General penceresi Şekil 1'de verilmektedir.



Şekil 1. C/C++ Ayarları

Bu sayfadaki "Additional Include Directories" bölümüne, indirilen OpenCV klasörünün içerisindeki build\include klasörünün yolu eklenmelidir.

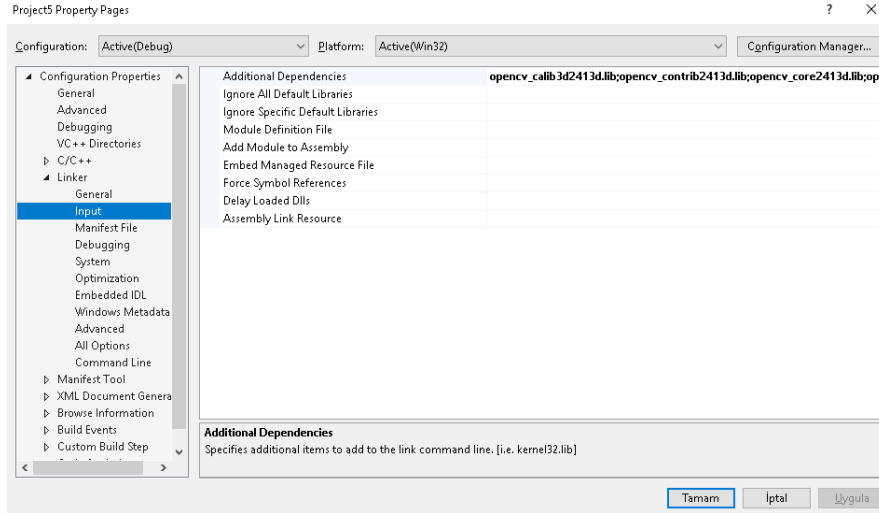
Linker başlığında general'in altındaki "Additional Library Directories" bölümüne, indirilen OpenCV klasörünün içerisindeki build\x86\vc14\lib klasörünün yolu eklenmelidir. Eklenmiş örnek giriş Şekil 2'de verilmektedir.



Şekil 2. Kütüphanelerin Eklenmesi

Şekil 3'te linker başlığının alındaki Input bölümündeki "Additional Dependencies" bölümüne çeşitli girişler yapılmıştır. Bu girişler aşağıda verilmektedir.

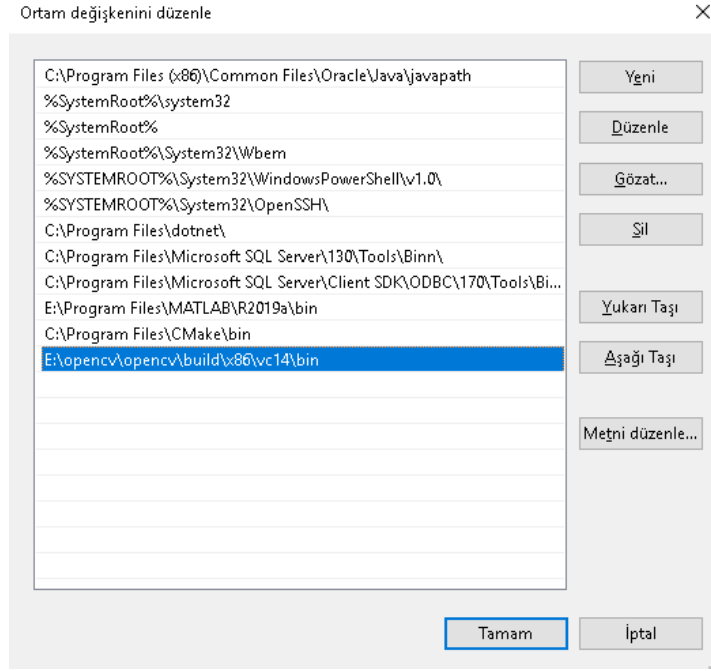
```
opencv_calib3d2413d.lib;opencv_contrib2413d.lib;opencv_core2413d.lib;opencv_features2d2413d.lib;opencv_flann2413d.lib;opencv_gpu2413d.lib;opencv_highgui2413d.lib;opencv_imgproc2413d.lib;opencv_legacy2413d.lib;opencv_ml2413d.lib;opencv_nonfree2413d.lib;opencv_objdetect2413d.lib;opencv_ocl2413d.lib;opencv_photo2413d.lib;opencv_stitching2413d.lib;opencv_superres2413d.lib;opencv_ts2413d.lib;opencv_video2413d.lib;opencv_videostab2413d.lib;%(\AdditionalDependencies)
```



Şekil 3. Kütüphanelerin Eklenmesi

Uygulama bu ayarlar ile derlenebilir durumdadır. Ancak uygulama çalıştırıldığında .dll uzantılı dosyaları bulamadığı hatası verecektir. Bunun için OpenCV dll'eri sistemin Path'ine gösterilmelidir.

Bunun için bilgisayarım'a sağ tıklayıp özellikler açılır. Sol menüde bulunan gelişmiş sistem ayarlarına basılır. Açılan penceredeki en alt bölümde bulunan ortam değişkenlerine basılır. Sistem değişkenleri bölümündeki Path değişkeni bulunur ve düzenle tuşuna basılır. Açılan pencerede yeni tuşuna basılarak indirilen OpenCV klasörünün içerisindeki build\x86\vc14\bin klasörünün yolu eklenir. Şekil 4'te örnek giriş verilmiştir.

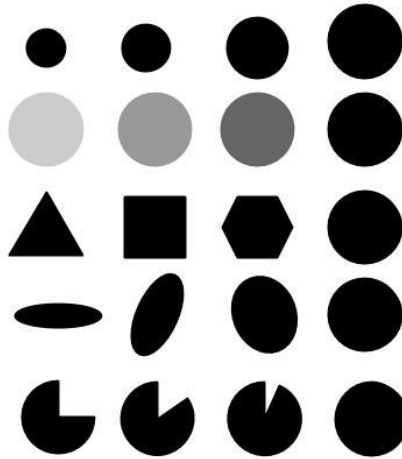


Şekil 4. OpenCV Kütüphane Gösterimi

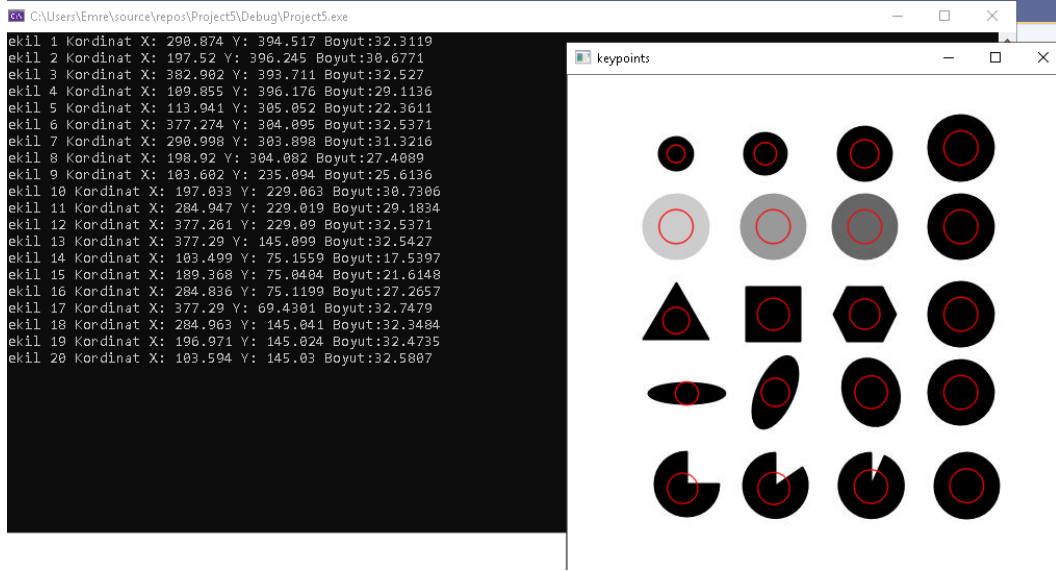
Tüm pencerelere tamam diyerek çıkılır. Bu ayarları Visual Studio ile derlenen programların görebilmesi için Visual Studio açık ise yeniden başlatılması gerekmektedir. Bu aşamada verilen referans tasarım derlenip, çalıştırılabilir olacaktır.

Not: Eklenen x86 klasörleri 32 bit işlemcileri içindir. 64 bit'lik derlemeler yapılacağı zaman, x64 klasörleride yukarıda anlatılan proje ayar sayfalarına aynı şekilde girilmelidir.

Proje Şekil 5'te verilen blob.jpg dosyası ile çalıştırıldığında, Şekil 6'da verilen çıktı görüntü oluşmaktadır.



Şekil 5. Giriş blob.jpg



Şekil 6. Çıktı Görüntü

Konsola blob.jpg'de verilen şekillerin ağırlık merkezlerinin bulunarak çıktı verildiği görülmektedir.

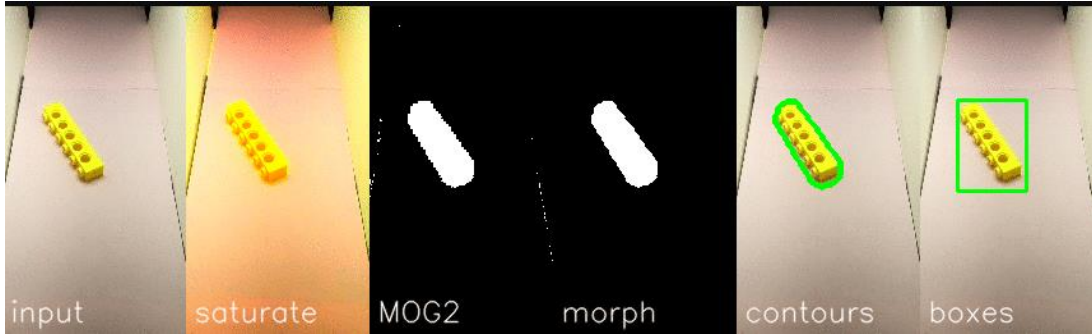
5. Gerçekleme Yöntemi:

Verilen referans koda OpenCV'nin birçok fonksiyonu kullanılmıştır.

Bu projeyi yapmak için birçok yaklaşım olabilir. Kendi yaklaşımınızı geliştirmelisiniz. Aşağıdaki adresten giriş video indirilebilir.

<http://levent.tc/courses/blm102/projeler/proje3/uretimHatti.mp4>

Uygulanan çeşitli test filtrelerine göre çıktılar Şekil 7'de verilmiştir.



Şekil 7. Giriş Resmine Göre Uygulanan Farklı İşlemlerin Çıktıları

Örnek yaklaşım:

Referans dosyada tek bir resim karesinin açılarak blobDetection uygulaması yapılmıştır. Bu projede verilen video dosyasını, OpenCV'nin video okuma sınıfları ile gerçekleştirilebilir. Aşağıda video okuma için referans kod parçacığı verilmiştir.

<http://levent.tc/courses/blm102/projeler/proje3/videoCapture.cpp>

Açılan Mat türündeki objeye, her bir döngüde yeni bir kare alınmaktadır. Bu aşamada OpenCV'nin kendi hazır sınıflarından olan `SimpleBlobDetector` sınıfı kullanılarak gruplanmış olan piksel grupları bulunabilir. Burada dikkat edilmesi gereken nokta bulunan şeklin görüntüde aslında bulunmak istenmeyen bir yere ait olabilir. Bunların filtrelenmesi gerekmektedir. Filtreleme işlemi, verilen referans main.cpp dosyasındaki gibi blob bulucu fonksiyona

Öğretim Elemanı: Dr. Vecdi Emre Levent, emre.levent@fbu.edu.tr, *İzinsiz Kopyalanamaz*

parametre olarak verilebilir. Ancak bu yeterli olmayabilir, bunun için blob detection yapmadan önce arka planın silinmesi gerekebilir (Örneğin MOG algoritması).

Her bir karede sadece istenen obje ve onun koordinatları bulunduktan sonra, sayım işlemi yapılabilir. Bunun için 3 değişken açılıp 0'dan başlatılabilir. Karelerde obje bulunduğça hangi obje olduğu tespit edilerek ilgili değişken 1 arttırılabilir. Burada 2 sorun oluşacaktır.

İlk sorun objelerin arasındaki farkın nasıl anlaşılacağıdır. Bu aşamada objelerin renkleri kullanılabilir. Bulunan koordinatlardaki renk bilgisi ile obje'nin ne olduğuna karar verilebilir.

İkinci sorun ise ardışık karelerde aslında çoğu zaman aynı obje geliyor olacaktır. Burada aynı obje için sürekli artış yapılmamalıdır. Buradaki yaklaşım ise, bulunan objenin koordinatları, bir önceki karede bulunan objedeki koordinatlardan farkı alınır ve bu fark az ise, aynı objeler denebilir ve artış yapılmaz.

Video tamamlandıktan sonra bulunan obje sayıları konsol ekranına bastırılacaktır.

Tasarım yapılırken nesneye yönelimli programlama (sınıflar, kalıtım, çok biçimlilik) ve modüler programlama (operatör aşırı yükleme, şablonlar, istisna idaresi) yaklaşımlarını mümkün oldukça kullanınız. Proje teslim raporunuzda geliştirilen tasarımın mimarisini gösteren UML diyagramını veriniz.

Not: Tasarım test edilirken, proje Visual Studio'da geliştirilirken Debug modunda çalıştırıldığında yavaş çalışabilir, hızlandırmak için release x64 modunda çalıştırılabilir.

6. Notlandırma ve Proje Teslimi:

Bu başlık "Görüntü İşleme Yöntemleri ile Üretim Hattı Analizi" projesinin teslimi ve notlandırılması hakkında bilgiler içermektedir.

6.1. Notlandırma:

Projenin iki ana değerlendirme kriteri vardır. Her iki kriter 50 şer puandır.

İlk kriter "Görüntü İşleme Yöntemleri ile Üretim Hattı Analizi" 'nin **doğru çalıştırılması ve objeye dayalı programlama dili yaklaşımlarının kullanılmış olmasıdır.**

İkinci kriter ise **Proje Teslim Dokümanı ve Sunumdur.**

- **Proje Teslim Dokümanı:**

Öğrenciler, proje raporlarını verilen "Proje Teslim Dokümanı" 'nın içerisini doldurarak yapacaklardır.

Proje Teslim Dokümanı: http://levent.tc/courses/blm102/BLM102_proje_teslim_dokumani.doc

Proje teslim dokümanı tamamlanıp PDF'e dönüştürüldükten sonra, PDF'in sonuna Latex (Genellikle bilimsel çalışmaların yazımında kullanılan, görsel olarak Word ile yazılmış dokümanlara göre çok daha başarılı olan bir doküman oluşturma dilidir) ile tasarlanmış özgeçmişler eklenecektir. Özgeçmişin minimumda kullanılan diller, yabancı diller, projeler (kısa açıklamaları ile), kullanılan kütüphaneler (varsa, örn. OpenCV), not ortalaması, github adresi, youtube adresi bilgileri verilmelidir.

Latex ile özgeçmiş oluşturma: http://levent.tc/courses/blm102/BLM102_latex_ozgecmis_olusturma.pdf

Latex ile özgeçmiş oluşturulduğunda PDF uzantılı dosyalar elde edilecektir. Proje teslim dokümanının PDF'inin arkasına eklemek için PDF birleştirici araçları (https://www.ilovepdf.com/merge_pdf gibi) kullanılmalıdır. İşlem sonucunda tek bir PDF dosyası elde edilecektir.

- **Proje Sunumu:**

Powerpoint üzerinde ortalama 5 dakika (4-6 dakika arası) sürecek bir sunum hazırlayarak kayıt etmelidirler. Kayıt işlemi, cep telefonu veya bilgisayar ekran kayıt yazılımları (Screen-Recorder, Bandicam vb...) ile yapılabilir.

Sunum, ekip üyeleri içinden biri tarafından, projenin nasıl yapıldığı, nasıl çalıştığı vb.. konularının powerpoint slaytları üzerinden anlatılırken kaydedilmesi ile olmalıdır. Sunum video'sunda powerpoint slaytları okunabilir ve konuşmacının sesinin anlaşılır olması gerekmektedir. Powerpoint slayt görünüm tasarımı istenildiği gibi yapılabilir.

Proje ekibinin tamamı, notlarını bu değerlendirmeye göre alırlar.

6.2. Teslim:

Projenin teslimi için aşağıdaki adımların gerçekleştirilmesi gerekmektedir. İstenen dosyaları sadece proje ekip sorumlusunun getirmesi, LMS ve Github (Çok yaygın bir açık kaynak kod paylaşım platformudur)'a yüklemelidir.

Proje Teslim Dokümanının, ders sınıfı ve saatinde, çıktılarının alınarak teslim edilmesi gerekmektedir.

Ayrıca LMS'te açılmış olan "Proje Teslim" sayfasına aşağıdaki dosyaların yüklenmesi gerekmektedir.

- Proje kaynak kodları
- Hazırlanan powerpoint sunum dosyası (.ppt uzantılı dosya)
- Proje Teslim Dokümanı (Word formatında yüklenmelidir)
 - Dokümanın alt başlıkları doldurulmalıdır. (Örnek proje dokümanı: http://levent.tc/courses/blm103/projeler/2019-2020-tamamlananlar/BLM103_yapay_zeka_rapor_ozlem_deniz.pdf)
 - Kaydedilen powerpoint sunum video'su youtube'a yüklenip, adresi, dokümanın sonuçlar bölümündeki açılmış yere link'i yazılmalıdır (Video'nun herkes'e görünür olmamasını istiyorsanız, youtube'a yükledikten sonra liste dışı seçeneğini seçerek, sadece link'e sahip olan kişilerin görmesini sağlayabilirsiniz).
 - LMS'e yüklenen tüm dosyalar (Proje kaynak kodları, ppt uzantılı sunum dosyası ve Proje Teslim Dokümanını (PDF formatında)), github.com sitesine üye olup, yüklenip, dokümanın sonuçlar bölümündeki yere link'i yazılmalıdır.