



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



GÖRÜNTÜ İŞLEME TABANLI
ENDÜSTRİYEL ROBOT KOL İLE ÇİKOLATA
TOPLAMA VE PAKETLEME SİSTEMİ

Muhammed Ali YILMAZ

YÜKSEK LİSANS

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ağustos-2023
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Muhammed Ali YILMAZ tarafından hazırlanan “GÖRÜNTÜ İŞLEME TABANLI ENDÜSTRİYEL ROBOT KOL İLE ÇİKOLATA TOPLAMA VE PAKETLEME SİSTEMİ” adlı tez çalışması 10/08/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Dr. Öğretim Üyesi Hakan TERZİOĞLU

.....

Danışman

Prof. Dr. Cemil SUNGUR

.....

Üye

Doç. Dr. Akif DURDU

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mevlüt UYAN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Muhammed Ali YILMAZ

Tarih: 10.08.2023

ÖZET

YÜKSEK LİSANS

GÖRÜNTÜ İŞLEME TABANLI ENDÜSTRİYEL ROBOT KOL İLE ÇİKOLATA TOPLAMA VE PAKETLEME SİSTEMİ

Muhammed Ali YILMAZ

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı**

Danışman: Prof. Dr. Cemil SUNGUR

2023, 75 Sayfa

**Jüri
Prof. Dr. Cemil SUNGUR
Doç. Dr. Akif DURDU
Dr. Öğretim Üyesi Hakan TERZİOĞLU**

Gıda sektöründe paketleme işleminde en önemli iki konu hijyen ve hız olarak görülmektedir. Paketleme işlemi hijyen kurallarına uygun ve hızlı bir şekilde gerçekleştirilirken aynı zamanda kalitenin de sağlanması gerekmektedir. Paketlemede hijyen, hız ve kalitenin bir arada sağlanması ancak bu işleme uygun makineleşme ile gerçekleşebileceği düşüncesi ön plana çıkmaktadır. Bu makineleşme işlemi sayesinde gıda paketlemede üretimde insan faktörü azaltılarak hijyen, hız ve kalitenin artırılması hem kaliteli ürün hem de ticari karın artırılmasının mümkün olacağı ön görülmüştür.

Bu tez çalışmasında gıda paketlemede insan faktörünü azaltarak üretimi hızlandırırken hijyeni ve kaliteyi artıran bir çikolata paketleme sistemi tasarlanmış ve imalatı gerçekleştirilmiştir. İmalatı gerçekleştirilen bu sistemde bugüne kadar manuel veya kısmi olarak robotlarla yapılan paketleme işlemi endüstriyel robot kol ve görüntü işleme yöntemi kullanılarak yapılmıştır. Bu sistemde Robot kolun çikolatayı toplayabilmesi için insan gözü yerine bir endüstriyel kamera kullanılmış ve hattan akan farklı çikolatalar görüntü işleme yardımı ile tespit edilmesi amaçlanmıştır. Üretim hattında hareket halindeki çikolatalara ait görüntünün özellikleri robot kola gönderilmiştir. Robot kol gelen bilgileri kullanarak tespit edilen çikolataları konveyörü durdurmadan toplamıştır. Görüntü işleme aşamasında USB kamera kullanılmıştır. USB kameradan alınan görüntüler Python ortamında OpenCV kütüphanesi kullanılarak işlenmiş, tespit edilen her bir çikolataya ait merkez X ve merkez Y konumu ile açısı ve tür bilgileri Modbus TCP/IP haberleşmesi ile önce PLC'ye (Programlanabilir Lojik Kontrolör) ve oradan da panel üzerinden robot kola aktararak robot kolun ürünü toplaması sağlanmıştır.

Yapılan bu çalışmalarda manuel olarak yapılan paketleme işlemleri ile yeni geliştirilen sistem Hijyen, hız ve kalite olarak karşılaştırılmıştır. Geliştirilen robotik sistemde elde edilen ürünlerin manuel olarak üretilenlere göre hijyen açısından %90, hız açısından %23 ve kalite açısından %30 daha üstün olduğu gözlemlenmiştir. Böylece robotik sistemin hem insan sağlığına hem de firmaya ticari olarak olumlu katkı sağladığı görülmüştür.

Anahtar Kelimeler: Çikolata, Görüntü işleme, OpenCV, Python, Robot kol

ABSTRACT

MS THESIS

IMAGE PROCESSING BASED CHOCOLATE PICKING AND PACKAGING SYSTEM BY INDUSTRIAL ROBOTIC ARM

Muhammed Ali YILMAZ

**Konya Technical University
Institute of Graduate Studies
Department of Electrical-Electronics Engineering**

Advisor: Prof. Dr. Cemil SUNGUR

2023, 75 Pages

Jury

**Prof. Dr. Cemil SUNGUR
Assoc. Prof. Dr. Akif DURDU
Asst. Prof. Dr. Hakan TERZİOĞLU**

The two most important issues in the packaging process in the food sector are seen as hygiene and speed. While the packaging process is carried out in accordance with the hygiene rules and quickly, quality must also be ensured. The idea that providing hygiene, speed and quality together in packaging can only be realized with mechanization suitable for this process comes to the fore. Thanks to this mechanization process, it is foreseen that it will be possible to increase hygiene, speed and quality by reducing the human factor in production in food packaging, and to increase both quality products and commercial profits.

In this thesis, a chocolate packaging system has been proposed that increases hygiene and quality while speeding up production by reducing the human factor in food packaging. In this system, an industrial camera was used instead of the human eye so that the robot arm could collect the chocolate, and it was aimed to detect different chocolates flowing from the line with the help of image processing. The features of the image of the chocolates in motion on the production line were sent to the robot arm. Using the incoming information, the robotic arm collected the detected chocolates without stopping the conveyor. USB camera was used in image processing. The images taken from the USB camera were processed using the OpenCV library in the Python environment, the center X and center Y position, angle and type information of each detected chocolate firstly to the PLC (Programmable Logic Controller) and then to the robot arm via Modbus TCP/IP communication. The robot arm was transferred to collect the product.

In these studies, manual packaging processes and the newly developed system were compared in terms of hygiene, speed and quality. It has been observed that the products obtained in the developed robotic system are 90% better in terms of hygiene, 23% in terms of speed and 30% in terms of quality compared to those produced manually. Thus, it has been seen that the robotic system makes a positive contribution to both human health and the company commercially.

Keywords: Chocolate, Image processing, OpenCV, Python, Robotic arm

ÖNSÖZ

Bu tezin her aşamasında emeđi geen, bilgi, tecrbe ve yardımlarını esirgemeyen danıřman hocam Sayın Prof. Dr. Cemil SUNGUR'a ve tm ğrenim hayatım boyunca benden desteklerini hibir zaman esirgemeyen aileme teřekkr bir bor bilirim.

Muhammed Ali YILMAZ
KONYA-2023



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ	1
1.1. Endüstrinin gelişimi	2
2. KAYNAK ARAŞTIRMASI	7
3. MATERYAL VE YÖNTEM.....	12
3.1. Materyal	12
3.1.1. Endüstriyel robot.....	12
3.1.2. Programlanabilir lojik kontrolör (PLC)	15
3.1.3. İnsan-Makine ara yüzü.....	17
3.1.4. Motor sürücü.....	20
3.1.5. Enkoder	22
3.1.6. İki parmaklı tutucu.....	23
3.1.7. Konveyör	24
3.1.8. Endüstriyel haberleşme yöntemleri	25
3.1.9. Görüntü işleme.....	30
3.2. Yöntem.....	31
3.2.1. Renk dönüşümü	31
3.2.2. Eşikleme.....	32
3.2.3. Kenar tespiti.....	32
3.2.4. Basit kapalı alan tespiti (Simple blob detection)	33
3.2.4. PLC Programı	33
4. DENEYSSEL ÇALIŞMA VE TARTIŞMA.....	36
4.1. Görüntü İşleme Sonuçları	37
4.2. Kamera ve Robot Kalibrasyonu.....	41
4.3. Robot ve Enkoder Kalibrasyonu.....	43
4.4. Python-robot kol haberleşmesi	43
4.5. Arayüz tasarımı	44
4.6. Deneysel çalışma	45
5. SONUÇLAR VE ÖNERİLER.....	47
5.1 Sonuçlar	47
5.2 Öneriler	50

KAYNAKLAR	52
EKLER	55



SİMGELER VE KISALTMALAR

Kısaltmalar

PLC: Programlanabilir Lojik Kontrolör

HMI: İnsan Makine Arayüzü



1. GİRİŞ

Gıda üretim proseslerinde üretim işlemi tamamlandıktan sonra ürünün pazarlanması aşamasına geçilmektedir. Ürünün pazarlama aşamasında tüketiciyi etkileyen önemli faktörlerin hijyen ve kalite olduğu görülmektedir. Bunun bilincinde olan üretici için de hijyen; kalite ve paketleme işleminin hızı olarak kabul görmektedir. Bu durumda hem tüketici hem de üreticinin memnuniyetini sağlamak için gerekli gözlemler ve incelemeler yapılmıştır.

Bu gözlemlerden elde ettiğimiz bilgilere göre gıda üretimin son aşaması olan paketlemenin öncelikli parametrelerinden bir tanesi insan sağlığını olumsuz yönde etkileyen faktörlerin ortadan kaldırılmasıdır. Bu olumsuz faktörlerin başında insan sağlığı içinde çok önemli olan hijyen gelmektedir. Gıda proseslerinde hijyen sağlanabilmesi için yapılacak işlemlerden en önemlisi üretilen gıda ile insan temasının kesilmesidir. Bunun için insan eli değmeden üretim yapabilen birçok üretim hattı geliştirilmesi gerekmektedir. Ancak geliştirilen bu üretim hatları da beraberinde bazı problemler getirmişlerdir. Bu problemlerin başlıcaları ise elde edilen ürünün kalitesinin kontrolünün zorluğu ve paketleme hızının yavaş olduğu görülmüştür. Günümüzde hâlâ insanlar tarafından paketlemenin yapıldığı üretim hatları mevcuttur. Bu üretim hatlarında makineleşmeye gidildiği zaman makineleşme masrafını hattın çok kısa sürede amorti edeceği de bilinen bir gerçektir.

Paketleme tesislerinde insanlarla yapılan işlemlerin hijyen problemi yakın geçmişte yaşadığımız ve etkileri günümüzde de devam eden pandemi dönemi koşullarıdır. Bu süreçte birçok üretim tesisi insan gücü eksikliğinden dolayı üretime ara vermiştir. Pandemi sürecinde yaşadıklarımız bize insan gücüne dayalı üretim yapmayan firmaların ne kadar isabetli bir karar verdiklerini göstermiştir. Bu canlı örnek üretimde makineleşmenin çok önemli olduğu göstermektedir.

Gıda endüstrisinde seri üretimde ürün toplama ve paketleme işlemleri genel olarak insanlar ile yapıldığı görülmektedir. İnsan çalışan sistemlerde verim sürekli yüksek olmamakla birlikte insanın hata yapma olasılığı da yüksektir (Burak, 2021). Bu nedenle seri üretimler için bir otomatik toplama ve paketleme sisteminin gerekliliği kaçınılmazdır. Paketleme sisteminin tam otomatik yapılması problemin tamamen çözümü sağlamış olmamakta, bu otomasyon sisteminin optimize şekilde çalıştırılması için kontrol ve takibinin de yapılması gerekmektedir. Bunun için paketleme sisteminin üretim prosesini en başından takip ediyor olmalıdır. Bu işlem ise üretim esnasında

konveyör üstünde bulunan ürünün hangi aşamada olduğunun bir sonraki hatta bildirilmesi bir haberleşme ağı ile sağlanabilmektedir. Bu haberleşme ağı üretim aşamasında kullanılan sürücüler, PLC'ler ve operatör panelleri vb. üzerinden gerçekleştirilebilmektedir. Tüm bu haberleşme ağı nesnelere interneti teknolojisi ile eş zamanlı takip edilebilmektedir. Üretimi en başından beri takip edebilen, paketleme hızını ve verimliliğini diğer kontrolcülerle haberleşerek ayarlayabilen ve çalışmasını raporlandırarak analizini yapabilen bu sistemler Endüstri 4.0 olarak anılmaktadır.

1.1. Endüstrinin gelişimi

Fabrikalar son yüzyıllarda üretim süreçlerini ve ekonomik yapıları yeniden şekillendiren teknolojik ilerlemelerle önemli dönüşümler geçirdi. Endüstri 1.0'dan Endüstri 4.0'a olan yolculuk, otomasyonu, dijitalleşmeyi ve bağlantıyı sanayi işlemlerine ilerleyici bir şekilde entegre eden devrimler dizisi olarak tanımlanabilir.

Endüstri 1.0, İlk Sanayi Devrimi olarak da bilinir ve tekstil üretiminin mekanizasyonu ile su ve buharla çalışan makinelerin tanıtılmasıyla başlamıştır. Bu dönemde, manuel emekten mekanize süreçlere geçiş sağlanmış olup, verimlilik önemli ölçüde artırıldı. James Watt'ın 1765 yılında buhar makinesini icat etmesi, bu devrimi sürdürmede önemli bir rol oynamıştır (Uslu, 2022).

İkinci Sanayi Devrimi olan Endüstri 2.0, elektrik enerjisinin ve montaj hattının gelişimi ile karakterize edilmektedir. Bu dönemde, kitlesel üretimin yükselişi gözlemlenmiş olup, üretimde artan standartlaşma ve verimlilik sağlanmıştır. Telefon, telegraf ve içten yanmalı motor gibi yenilikler, sanayi büyümesini daha da hızlandırmıştır (Uslu, 2022).

Endüstri 3.0 veya Üçüncü Sanayi Devrimi, bilgisayarlaştırma ve otomasyonun yükselişi ile tanımlanmaktadır. PLC'ler ve bilgisayar kontrollü makineler, üretim süreçlerini önemli bir şekilde değiştirmiştir. Elektronik ve bilgi teknolojisinin entegrasyonu, daha büyük hassasiyeti, azalan insan müdahalesini ve geliştirilmiş kalite kontrolü sağlamıştır (Uslu, 2022).

Endüstri 4.0, fiziksel ve dijital teknolojilerin birleşmesini temsil eden mevcut endüstri evrimini temsil etmektedir. Bu çağda, Nesnelere İnterneti (IoT), yapay zeka (AI), büyük veri ve bulut bilişimini üretim süreçlerine entegre etme prosesleri öne çıkmaktadır. Siber-fiziksel sistemler, makineler arasında gerçek zamanlı veri alışverişini mümkün kılarak otonom karar verme ve uyarlanabilir üretim sistemlerini kolaylaştırmaktadır (Uslu, 2022).

Endüstri 4.0, şunları içerir (Shrouf ve ark., 2014):

Akıllı Fabrikalar: Gerçek zamanlı izleme ve üretim süreçlerini optimize etme yeteneği sunan bağlı sensörler ve cihazlarla donatılmış fabrikalar.

Dijital İkiz: Fiziksel varlıkların, ürünlerin veya süreçlerin sanal kopyaları, simülasyon, analiz ve kestirimci bakım için.

Merkezsiz Karar Verme: İnsan müdahalesi olmaksızın veri analizi ve karar alma yeteneğine sahip otonom sistemler, daha büyük verimlilik ve azalan hatalar sağlamaktadır.

Özelleştirme ve Esneklik: Hızlı üretim süreçleri sayesinde özelleştirilmiş ürünlerin verimli bir şekilde üretilebilme yeteneği.

Veri Odaklı İçgörüler: Büyük veri analitiği ve yapay zekâ kullanarak işlem iyileştirmesi ve yenilik için uygulanabilir içgörülerin elde edilmesi.

Endüstri 4.0 bilişim ve endüstri faaliyetlerinin ortak çalışması olarak açıklanmakta olup son zamanlarda popüler olmuş bir terimdir. Endüstri 4.0'ı bir fabrikada gerçekleştirebilmek için bilişim teknolojilerine ihtiyaç vardır. Bilişim teknolojileri sinyal ve görüntü işleme, otomasyon, Scada ve yapay zekâ vb. teknolojilerden oluşmaktadır. Günümüzde kullanım alanı iyice genişleyen yapay zekâ sistemler endüstride üretim yapan makinelere entegre edilerek makineler daha "akıllı" bir hale getirilmektedir. Bu sayede üretim yapan makineler geliştirilerek değişime uğramaya başlamıştır. Değişim ilk olarak makine otomasyon sistemleri ile başlamıştır. Otomasyon sistemleri akıllı sensörler, akıllı sürücüler ve insansız robotlar ile gelişimini ilerletmeye devam ettirmektedir. Fabrikalarda yer alan akıllı otomasyon sistemleri, yine fabrikalarda insana özgü görevleri yapan ve birbirleriyle etkileşime geçen akıllı robotlar teknolojinin ilerlemesini sağlamaktadır. Bu sayede üretim verimi artırılmaktadır. Üretim veriminin artırılmasında robot kollar çok büyük bir öneme sahiptir.

Robotlar sabit ve hareketli robotlar olmak üzere iki türe ayrılmaktadır. Hareketli robotlar çok daha geniş alanda işlevlerini yerine getirirken sabit robotlar bir yere monte edilmiş olup kendilerine tanımlanmış görevleri gerçekleştirir. Sabit robotlara kartezyen, küresel, delta, silindirik, Scara, 6 eksen ve kolaboratif robotlar örnek verilebilir. Şekil 1.1'de örnek robot türleri görülmektedir.



Şekil 1.1. Soldan sağa doğru Delta Robot, Scara Robot, Endüstriyel Robot ve Kolaboratif Robot

Şekil 1.1’de görülen sabit robotlar da mekanik ve tekrarlayan görevleri yerine getirebilir, önceden programlanmış hareketlerle nesnelere yönetebilir. Verileri toplayıp, işleyebilir ve makineye girebilir, işi optimize edebilir ve yapay zekâ sayesinde davranışını çevre ortamına uyarlayabilirler. Endüstride daha çok gövdesi bir yere sabitlenmiş robot kolları kullanılmaktadır. Endüstride en çok tercih edilen robot türü 6 eksen robot kollarıdır. Bunun sebebi 6 eksen endüstriyel robotların x, y, z eksenlerinde düzlemsel ve döner hareketleri yapabilmeleridir. Endüstriyel robotlar çoğunlukla paletleme, istifleme, kaynak, kolileme, sınıflandırma, al-bırak, paketleme, dizme, taşıma ve montaj uygulamalarında kullanılmaktadır.

Pandemi dönemi ile insansız ve robotlu üretim hatlarının kullanılması daha önemli bir duruma gelmiştir. Özellikle gıda üretim hatlarının insansız üretime uygun hâle getirilmesi sağlık açısından büyük önem taşımaktadır. Tez çalışmasında paketleme işlemi Konya Organize Sanayi Bölgesi’nde bulunan bir firmada gerçekleştirilmiştir. Firma ağırlıklı olarak çikolata üreten makine sistemleri üretmektedir. Gıda piyasasında çikolata çok sık tüketilen bir üründür. Çikolata üretimi ilk olarak akışkan hâlde bulunan çikolata ile başlamaktadır. Tasarlanan çikolata kalıbına, akışkan hâlde bulunan çikolata depozitörler ile istenilen gramaj ve çikolata şekline göre basılır. İçleri çikolata dolu kalıplar soğutma tüneline geçerek çikolataya kalıbın şekli aldırılır. Soğutma tüneline kaplanmaya hazır şekilde çıkan çikolatalar konveyör sistemleri ile çikolata şelalelerine doğru taşınır. Çikolata şelalesinden çıkan çikolatalar ovalama bölümünde işlenerek dış şekli verilir. Bu bölümden çıkan çikolatalar üstüne kaplanacak olan ürünler ile kaplanır. Üretimin bu safhasından sonra çikolataların üzerine artık gıda maddesi eklenmez. Çikolatalar tekrar soğutma tünelleri içinden geçirilerek üretim tamamlanmış olur. Üretim hattının sonuna konumlandırılan metal detektörü sayesinde içinde metal

kalıntısı bulunan çikolatalar tespit edilerek ayrıştırılır. Geriye sadece içinde tüketilebilir ve paketlenmeye hazır olan çikolatalar kalır.

Tez çalışmasında insan sağlığı için çok önemli olan hijyenin sağlanması için gerekli tedbirlerin alındığı ayrıca üretici için de kalitenin ve üretim sürecinde birim zamanda yapılan işin aratılarak maliyetin düşürülmesi konusunda çalışmalar yapılmıştır. Çikolataların paketlenmesi için insan eli yerine 6 eksen endüstriyel robot kol, insan gözü yerine ise bir adet kamera konularak sistem otomatik hâle getirilmiş ve hızı artırılmıştır. Yapılan çalışmada insan eli değmeden, ürün kalitesi, çeşitliğine dikkat edilerek hızlı çalışan bir üretim hattı tasarlanmıştır. Tasarlanan üretim hattında endüstriyel robot kol, kamera, PLC, arayüz tasarımının bulunduğu panel ve konveyör hattı kontrolü için inverter kullanılmıştır. Bu donanımın çalışması için gerekli yazılımlar (Görüntü işleme yazılımı, robot kol yazılımı, arayüz yazılımı, PLC yazılımı) gerçekleştirilmiştir.

Endüstriyel robotlar hızlı çalışan, yorulmayan, 24 saat çalışabilen ve bir ücret beklentisi olmayan işçi makine olarak tanımlanabilir. Günümüzde endüstriyel alandaki robotlar, insan işçilerin sıkıcı, kötü, tehlikeli veya çok ince görülmesi gereken işlerde yerini almak üzere tasarlanmaktadır. Robot hareketleri insan hareketlerini taklit etmektedir, fakat insanın düşünce ve hareket tarzını kopyalayamazlar. Bazen insanların sahip olmadıkları metotlar ve özellikler yardımıyla bir insanın gösterebileceği performansın üzerine çıkarlar. Bir robot kızılötesi ve ultrasonik sensörler kullanabilir ve görüşü insandan daha iyi düzeyde olabilir. Göz olarak kullandığı sensörleri insandakinden farklı yerlerde, örneğin parmağında olabilir, böylelikle insanın göremediği şeyleri görebilir ve daha titiz bir çalışma yapabilir. İnsanların çalışmadıkları koşullar altında elektromanyetik özelliklere, özel tutma-kavrama cihazlarına ve değinilen tüm diğer özelliklere sahip bir robot rahatlıkla çalışabilir, çoğu zaman da daha ucuza daha iyi performans gösterebilir. Robotlar kartezyen robot, silindirik robot, küresel robot, delta robot, Scara robot, 6 eksen robot ve işbirlikçi robot olmak üzere birçok türe ayrılmaktadır. Görüntü işleme aşamasında kullanılan görüntüler USB kamera üzerinden alınarak yapılmıştır. Konveyörü kontrol etmek için PLC ve inverter kullanılmıştır. PLC, otomasyon sistemlerinde kumanda ve kontrol devrelerini yazılımsal olarak gerçekleştirilmesini sağlayan, belirli sayıda giriş ve çıkış birimlerine sahip olan endüstriyel bir bilgisayardır. İlk başta röleli kumanda sistemleri yerine düşünülmüş olup, ticarileşmesi 1969 yılında Modicon firması tarafından gerçekleştirilmiştir. Ticarileştiği ilk yıllarda sadece mantıksal işlemler ile çalışmakta

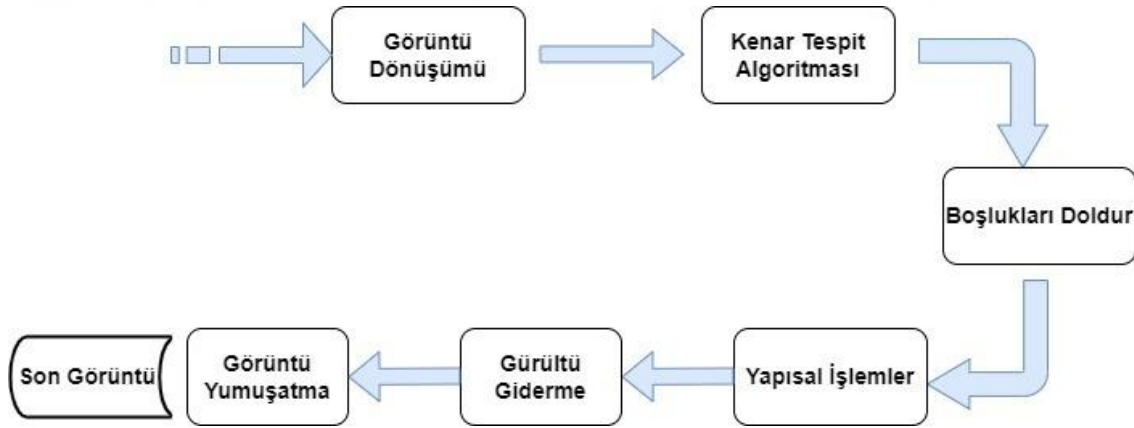
iken ilerleyen zamanlarda birçok firma yüksek performanslı PLC üretimini gerçekleştirmiştir (Bingöl, 2010). PLC girişine sensör ve buton gibi elemanlar bağlanabilirken; çıkış birimine röle, kontaktör gibi devre elemanları bağlanabilir. PLC ile otomasyon sistemlerinde birçok cihaz kontrol edildiği için sistemin beyni olarak görülmektedir (Çolak, 2007). PLC tarafından kontrol edilen konveyöre hareket vermek için asenkron motor kullanılmıştır. Asenkron motorlar az bakım isteyen, düşük maliyete sahip, güvenli ve sağlam olması gibi birçok avantaja sahiptir. Asenkron motorun sayılan birçok artı yönüne rağmen hız kontrolü uygulamalarında DC motorlar daha çok tercih edilmekteydi. Ancak, günümüzde yarı iletken ve mikroişlemci teknolojisinin gelişmesiyle birlikte güç elektroniği devreleri yüksek verimlilikle tasarlanabilir hale gelmiştir. Bu da asenkron motor sürücü devrelerinin tasarlanabilir olmasını sağlamıştır. Asenkron motor sürücü tasarım çeşitliliği arttıkça asenkron motor kullanım oranı da artmaya başlamıştır (Alkan, 2021). İnverterler piyasada kullanım alanı geniş olan güç elektroniği çeviricileri arasında yer almaktadır. İnverterler istenilen genlik ve frekansta gerilim üretebilen cihazlardır. Öncelikli olarak otomasyon sistemlerinde kullanılan inverterler birçok alanda önemli bir yere sahiptir. Tek faz-çok faz dönüşümü, çok faz-çok faz dönüşümü, şebeke gerilimi düzeltme gibi birçok uygulamada inverter kullanılmaktadır (Akhan, 2022).

Hareket eden konveyör üzerinde asenkron motorun konumunu tespit etmek amacıyla enkoder (sinyal üretici) kullanılmıştır. Enkoder bir motor milinin dönme ya da ötelenme hareketine karşılık, sayısal bir elektrik sinyali üretecek şekilde tasarlanmış kontrolöre geri bildirim sinyali gönderen bir elektromekanik bir cihazdır. Hem manyetik hem de optik olarak çalışabilmektedir. Sinyal oluşturmak için enkoderlerde farklı türde teknolojiler kullanılmaktadır. Bu teknolojiler optik, manyetik ve dirençli olmakla birlikte en sık kullanılanı optik teknolojidir. Enkoderin gönderdiği geri bildirim sinyali ile bir cihazın konumu, hızı ve yönü bulunabilir. Enkoderler endüstriyel kontrol işlemlerinde, robotlarda, servo motorlarda, CNC tezgahlarında, ölçme araçlarında, çizici ve bölücülerde, levha işleme makinelerinde, anten ve teleskoplarda, baskı ve paketleme makinelerinde, medikal cihazlarda ve birçok farklı cihazda kullanılmaktadır.

2. KAYNAK ARAŞTIRMASI

Bu tez çalışmasında kullanılacak olan materyal ve metotları literatürde kullanan çalışmalar konusunda araştırmalar yapılmış, makale ve tezler incelenerek tez konusuyla uyumlu olanlar aşağıda sıralanmıştır.

Elbehiery ve ark. (2005) seramik karo imalatında kalite kontrol aşamasında görüntü işleme yöntemini kullanmışlardır. Homojenliği iyileştirmek için paketleme işleminden önce görüntü işleme ve morfolojik yöntemler tekniklerini kullanarak seramik karolarda renk analizi, boyut doğrulama ve yüzey kusur tespiti yapmışlardır. Önerilen yöntemin akış şeması Şekil 2.1'de görülmektedir.



Şekil 2.1. Önerilen yöntemin akış şeması

Kale ve Kulkarni (2013) hareket eden bir sistem içinde akan nesnelerin gerçek zamanlı kalite kontrolünü yapmıştır. Çalışmada Logitech C270 webcamden alınan görüntüler gerçek zamanlı işlenerek bir karar algoritması ile al-bırak uygulaması gerçekleştirilmiştir. Nesneler renk ve boyutlarına göre incelenerek hattan alınarak önceden belirlenen noktalara konulmuştur. Çalışmada görüntü işlemek için Matlab yazılımı ile robot hareketini sağlamak için ise Arduino kiti kullanılmıştır.

İnan (2013) çalışmasında iki eksenli kartezyen robot yapmış olup, kartezyen robot kontrolü görüntü işleme ile kontrol edilmiştir. Kameradan alınan görüntü işlenerek kartezyen robot hedef noktaya gönderilmiştir. Görüntü işleme programı olarak iki boyutlu çapraz korelasyon yöntemi ile hedef nesne tanınmıştır. Görüntü işleme algoritması MATLAB programında yapılmıştır.

Parlak (2018) çalışmasında, bir taşıyıcı kayış üzerinden geçen ürünleri saymak için görüntü tabanlı bir sistem geliştirmiştir. Kamera kullanılarak hem hatalı ürünü ayırt

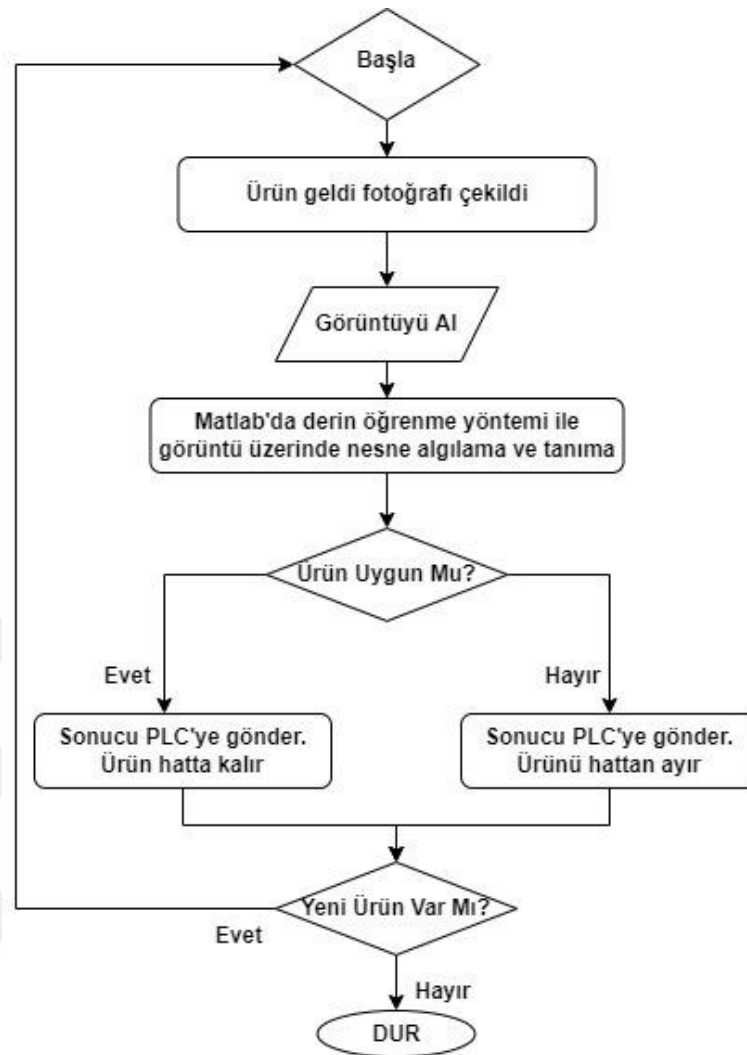
edecek hem de ürünün doğru sınıflandırmasını yapacak özellikleri taşıyan bir sistem kurulmuştur. Çalışmada, görüntü-tabanlı bir yaklaşım kullanarak, dolayısıyla imge ve video işleme tekniklerinin dahil edilmesi ile, yalnızca kamera kullanılarak bir taşıyıcı kayış üstünden geçen ürünlerin sayımı araştırılmıştır.

Kılıç (2018) çalışmasında, dijital görüntü işleme yöntemi ile parça boyutu analizi, patlatma sonrası oluşan yığından veya bant konveyörden dijital olarak görüntü alınması, bu görüntünün uygun yazılımla analiz edilmesi ve sonucun parça boyut dağılımının alan hesabıyla sunmuştur. Çalışmada, Imagej dijital görüntü işleme yazılımı kullanılarak Limak Çimento'ya ait kalker açık ocak işletmesinde parça boyut dağılımı saptanarak, patlatma verimlilik analizi yapılmıştır.

Zheng ve ark. 2018 yılında yaptıkları çalışmada otomobil parçası kaynağı esnasında sistemin kaynak başarısını arttırmak için görüntü işleme kullanarak kaynak yapılacak alanları yüksek hassasiyet ile tespit etmiştir. Sistem Çin'de bulunan bir otomobil fabrikasındaki kaynak robotuna entegre edilerek test edilmiş ve yüksek performans sağlanmıştır.

Savran ve Kumbasar (2018) yaptığı çalışmada gaz kaçak testi için görüntü işleme tabanlı ABB IRB 140 robot uygulaması gerçekleştirilmiştir. Çalışmada bakır boru üzerindeki kaynak noktaları görüntü işleme algoritması ile tespit edilmiştir. Tespit edilen noktalar robot konum bilgisine çevrilmiş ve gerçek zamanlı uygulamalarda pozisyon sapmaları incelenmiştir. Görüntü işleme tarafında dijital görüntü işleme algoritmaları kullanılmıştır.

Özdemir ve Koç (2019) derin Öğrenme yöntemi kullanarak akıllı fabrika prototipinde kalite kontrol uygulaması yapmışlardır. Yapılan çalışmada akıllı fabrika modelinde montaj hattına koyulan bir kamera ile görsel kalite kontrol otomasyonu önerilmiştir. Otomasyon sistemi Şekil 2.2'de sunulmuştur.



Şekil 2.2. Önerilen yöntemin akış şeması

Barstuğan ve ark. (2020) hidrolik pres üreten Hürsan Pres şirketinde yaptıkları Ar-Ge çalışmasında 5 farklı kaynak ustasının yaptığı farklı kaynak dikişleri üzerinden aldıkları 2250 görüntüyü 3 sınıfa (iyi kaynak, gözenekli kaynak ve süreksizlik) ayırmıştır. Bu sınıflandırma 2 aşamada yapılmıştır. İlk aşamada kaynak bölgesi tespit edilmiş, sonraki aşamada yapay sinir ağları yöntemi ile kaynak dikişinin özellikleri sınıflandırılmıştır. Eğitim aşamasında 1500 görüntü kullanılmış ve %94,03 eğitim performansı elde edilirken, test aşamasında 750 görüntü kullanılmış ve %94,31 sınıflandırma doğruluğu elde etmişlerdir.

Tarı (2020) çalışmasında görüntü işleme teknikleri kullanarak enjektör (şırınga) üretimi yapan bir işletmenin enjektör paketleme aşamasında kalite kontrol işlemini gerçekleştiren personele yardımcı olacak sistem tasarlamıştır. Çalışmada görüntü almak için aydınlatma sistemiyle birlikte bir platform oluşturulmuş ve 12.2 MP endüstriyel

kamera kullanılmıştır. Problemlerin çözümüne yönelik görüntü işleme tekniklerinden sıkça kullanılan şablon eşleştirme, haar-cascade sınıflandırıcısı, renk filtreleme, morfolojik işlemler gibi yöntemler birleştirilmiştir. Enjektör ve iğne ucu paketlenmesi aşamasında paket görüntüsü içinde enjektör/iğne ucu nesnelere yönelik var/yok/adet/yabancı madde tespit-kontrol üzerine çalışılmıştır.

Han ve ark. (2019) hareket eden bir konveyör üzerinden Scara ve Delta robotlar ile ürün toplama işlemini gerçekleştirmiştir. Çalışmada ürün toplama aşamasında robot kollar konveyörde bulunan ürünü takip ederek konveyör durdurulmadan ürün banttan ayrılmıştır. Robot kolun ürünü takip ederek al bırak işlemini en kısa yoldan yapması ile %10-40 arasında verim elde edilmiştir.

Mnyusiwalla ve ark. (2020) bir kasa içerisinde rastgele konumlarda bulunan meyve sebzeler kameradan alınan konumlar yardımıyla robotik kol ile toplanarak başka bir kasaya sıra ile dizilmiştir. Ürün toplama aşamasında dört farklı tutucu el çeşidi kullanılarak tutucu ellerin her birinin tutma performansı karşılaştırılmıştır. Görüntü işlemek için 3-boyutlu görsel çıkarıcı ve alınan görüntüde her bir pikselin derinlik bilgisini ve ürün pozisyonunu robota ileten 3-boyutlu bir kamera kullanılmıştır.

Premachandra ve ark. (2020) sıralı optimizasyon yöntemini kullanarak nesnelere renklerine ve boyutlarına göre ayırmışlardır. Nesnelere ayrılması ve dizilmesi için OpenCV kütüphanesi kullanılmıştır. Ürün ayırma ve sıralama aşamasında robotun en kısa yolu kullanması için genetik algoritma yöntemi uygulanmıştır. Genetik algoritma çalışma süresini kısaltarak al-bırak işlemini maksimum performans ile gerçekleştirmiştir.

Yang ve ark. (2021) havaalanı bagaj konveyörleri üzerinde akan valizlerin konumunu ve açısını Lidar sensör ile takip etmiş ve bir robot kol ile doğru açı ve konumda yaklaşarak konveyörü durdurmadan valizi alarak bir sonraki taşıma konveyörüne doğru açı ile bırakan bir çalışma gerçekleştirmiştir. Çalışmada sistem performansını ölçmek amacıyla bir deney düzeneği kurulmuştur. Deney düzeneğinden alınan verilere göre robot kolun ürünleri doğru şekilde bırakırken yaptığı hata ortalama 15.17mm olarak bulunmuştur. Ayrıca, 20 saniyede bir valiz dizme performansına ulaşılmıştır. Transfer aşamasında rastgele konumlarda ve açılarda gelen ürünler başarılı bir şekilde dizilmiştir.

Yıldız ve ark. (2021) yaptıkları çalışmada paketlenmiş çikolataların çıktığı hattın ambalajı bozuk ürünleri görüntü işleme ile tespit etmiştir. Tespit sonucunda ambalajı sağlam veya hatalı şekilde sınıflandırma işlemi yapılmıştır. Konveyör üzerinde akan

paketlenmiş ikolatalardan hatalı ambalaja sahip olanlar robot kol ile toplanmış ve sistemden ayrılmıştır. Görüntü işleme aşaması Python üzerinde gerçekleştirilmiş ve %100 sınıflandırma doğruluğuna ulaşılmıştır.



3. MATERYAL VE YÖNTEM

Tez çalışması Konya Organize Sanayi'sinde gıda makineleri üreten bir firmada bulunan materyaller ile gerçekleştirilmiştir. Gerçekleştirilen çalışma iki aşamadan oluşmakta olup ilk aşamada yer alan otomasyon sistemi güncel endüstriyel otomasyon ürünlerini içermektedir. İkinci aşama olan görüntü işleme aşamasında ise ürün tespiti için görüntü işleme yöntemleri kullanılmıştır.

3.1. Materyal

Gerçekleştirilen çalışmanın ilk aşaması olan otomasyon aşamasında Mitsubishi ürünleri olan FX5U PLC, GS2107 HMI, RV-2FD Endüstriyel Robot Kol ve FR-D700 motor sürücü kullanılmıştır. Ayrıca kurulan otomasyon sisteminde Pepperl+Fuchs ürününü olan enkoder ile Schunk marka iki parmaklı elektrikli tutucu da bulunmaktadır.

3.1.1. Endüstriyel robot

Bu çalışmada Şekil 3.1'de örnek görseli sunulan Mitsubishi marka RV-2FD model 6 eksen endüstriyel robot kol kullanılmıştır.



Şekil 3.1. Mitsubishi Electric 6 eksen robot

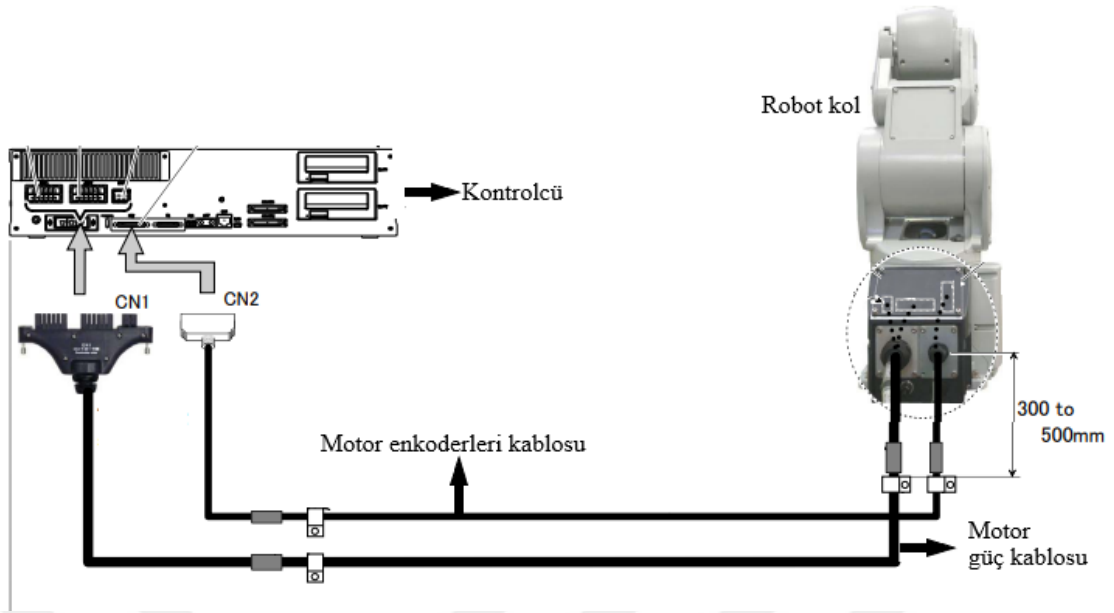
Robot kol 2kg taşıma kapasitesine sahip olup her bir eksenin dönüş açısı Çizelge 3.1'de sunulmuştur.

Çizelge 3.1. Robot kol dönüş aralıkları

Eksen	Dönüş Aralığı (Derece)
J1	±190
J2	-90~+150
J3	-10~+157.5
J4	±200
J5	±200
J6	±360

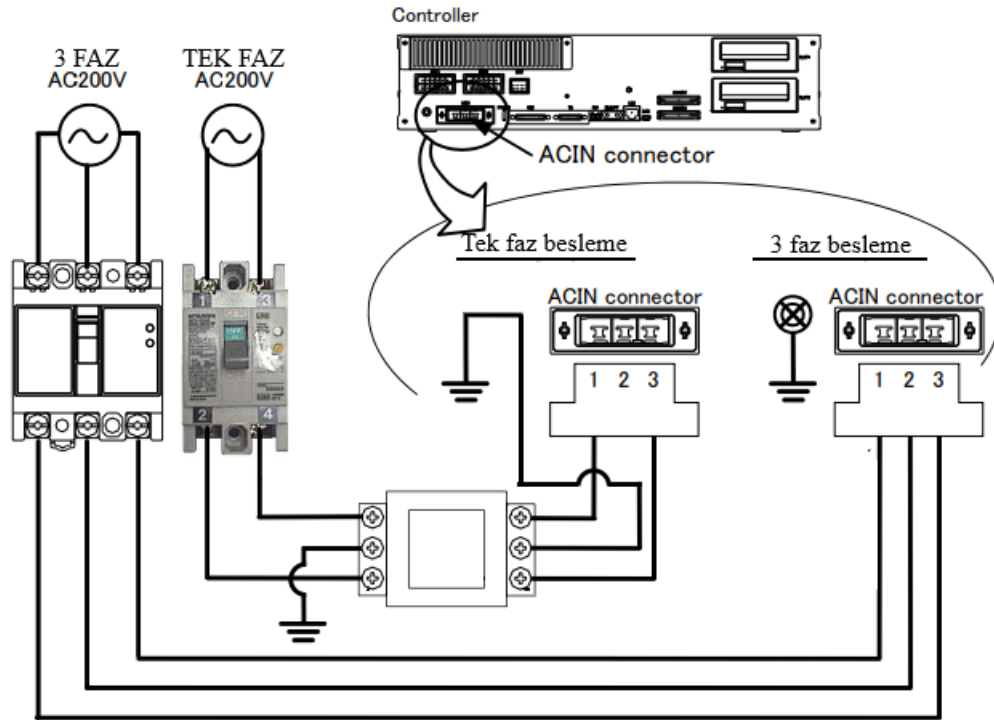
Endüstriyel robotlar, üç ana bileşenden oluşur: mekanik yapı, elektronik kontrol sistemi ve yazılım. Mekanik yapı, robotun hareketini sağlayan ve üretim sürecindeki farklı görevleri yerine getirmek için tasarlanan kollar, eklemliler mekanizmalar, sensörler ve araçları içerir. Elektronik kontrol sistemi, robotun hareketini kontrol eden ve sensörlerden gelen verileri işleyen bilgisayar tabanlı bir sistemdir. Yazılım, robotun görevleri tamamlamasını sağlayan programları içerir. Endüstriyel robotlar, görevlerini yerine getirmek için farklı yöntemler kullanırlar. Bazıları, insanların yapabileceği gibi doğrudan nesnelere kavrar ve taşırken, bazıları nesnelere vakum veya manyetik kuvvetlerle taşır. Bazıları da görüntü işleme teknolojisi kullanarak nesnelere tanıyabilir ve onlarla etkileşime geçebilir. Endüstriyel robotların avantajları arasında daha hızlı, daha doğru ve daha verimli bir şekilde çalışmaları, işçilerin monoton ve tehlikeli görevlerden kurtulmaları ve üretim sürecindeki kalitenin artması yer alır. Endüstriyel robotların çeşitleri arasında taşıma robotları, kaynak robotları, montaj robotları, paletleme robotları, paketleme robotları, boyama robotları ve hassas işleme robotları bulunur.

Robot kol kontrolü bir kontrol ünitesi üzerinden yapılmaktadır. Robot için yazılan programlar ve parametreler kontrol ünitesi içerisinde tutulmaktadır. Robot kol ve kontrolcü bağlantısı Şekil 3.2’de görülmektedir.



Şekil 3.2. Robot kol ve kontrolcü

Robot kontrol ünitesi 3 faz veya tek faz olmak üzere iki farklı besleme yöntemi ile beslenebilmektedir. Besleme gerilimi eğer 3x200 VAC ise 3 fazlı, faz nötr arası gerilim 200 VAC ise tek fazlı beslenmektedir. Şekil 3.3'te kontrol ünitesinin enerji bağlantısı görülmektedir.



Şekil 3.3. Robot kol kontrol ünitesi besleme bağlantısı

Bağlantıları yapılan robot kol ve kontrol ünitesi program yazılmaya hazır duruma gelmiş olmaktadır. "RT Toolbox 3" arayüzü aracılığıyla robot kol içine

program yazılabilir. Kontrolcü bağlantısı kontrolcü üzerinde bulunan ethernet veya mini USB soketi ile yapılmaktadır.

3.1.2. Programlanabilir lojik kontrolör (PLC)

Tez çalışmasında tüm sistemi kontrol etmek için ana istasyon olarak PLC kullanılmıştır. Çalışmada kullanılan PLC Mitsubishi FX5U model olup kompakt bir yapıya sahiptir. Üzerinde motor sürücü ile haberleşebilmesi için dahili RS485 haberleşme modülü bulunmaktadır. İnsan-Makine ara yüzü ile haberleşebilmesi için Ethernet (RJ45) girişi de bulunmaktadır. Kullanılan PLC'nin görüntüsü Şekil 3.4'te sunulmuştur.



Şekil 3.4. Sistemde kontrolör olarak kullanılan PLC

PLC fabrika otomasyonlarında robot, motor gibi makinaların kontrolünü sağlayıp, sistemin beyni olarak yer alan bir otomasyon cihazıdır. Sisteme ait giriş verilerini çok hızlı bir şekilde işledikten sonra çıkış bilgisini oluşturur. PLC'nin en büyük avantajı gürültü, sıcaklık ve darbelere karşı dayanıklı üretilmesidir. PLC üretimde oluşacak hatayı minimize eder ve insan kullanımını en aza indirger. Böylece oluşturulan ürünün kalitesi önemli ölçüde artar. Bir PLC, Merkezi İşlem Birimi (CPU), Bellek Birimi (RAM, ROM, PROM vb.), Giriş Birimi (IN), Çıkış Birimi (OUT) olmak üzere dört ana bölümden oluşur.

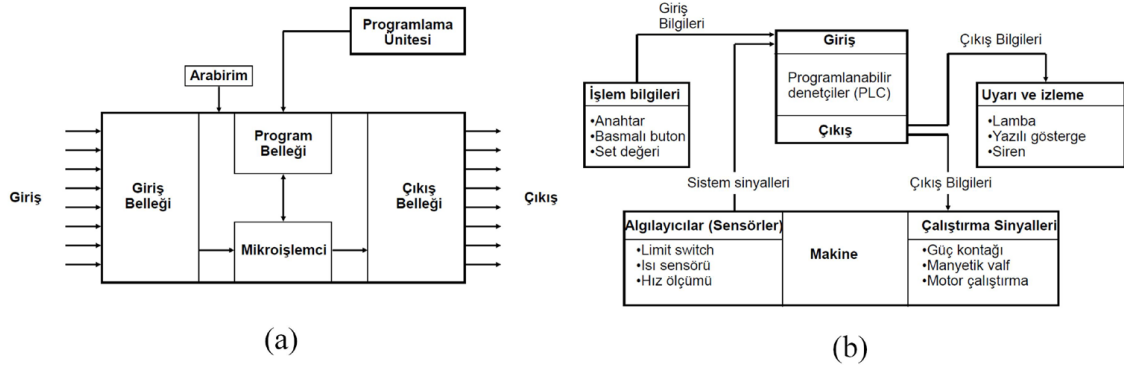
Merkezi İşlem Birimi (CPU): PLC'nin lojik, aritmetik gibi işlemlerini gerçekleyen birimdir. PLC, CPU sayesinde bu işlemleri yapar ve buna göre çıkış bilgisi üretir. CPU, PLC'nin çalışmasını düzenleyen en önemli birimdir.

Bellek Birimi (RAM, ROM, PROM, vb.): Bellekler CPU biriminden sonraki en önemli ikinci birim olarak geçerler. Programlar hafızada saklanır. Hafızadan CPU'ya gönderilir. Farklı çeşit bellekler bulunur ve her birinin farklı işlevi bulunmaktadır. Örneğin, EPROM (Eresable Programmable Read Only Memory), salt okunur, silinebilir ve programlanabilir hafıza olarak geçer. Silinebilir, üzerine yazılabilir ve programlanabilir hafızadır. RAM geçici bellek olup, ROM sadece okunabilir hafıza olarak geçer.

Giriş Birimi (IN): Analog (AI) ve Dijital (DI) olarak iki farklı giriş çeşidi bulunmaktadır. Giriş birimine gelen değerler (0 veya 1) inputlar vasıtasıyla alınır. PLC'nin anlayacağı şekilde lojik gerilim seviyelerine dönüştürülürler. Genellikle anahtarlar, röleler, sensörler giriş elemanı olarak kullanılabilir.

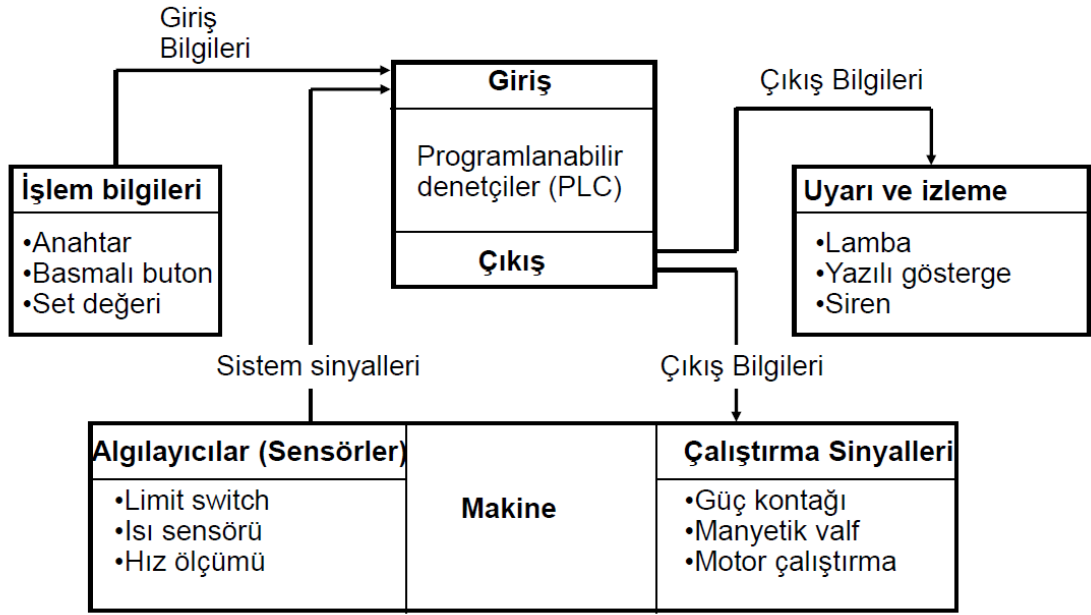
Çıkış Birimi (OUT): Analog (AO) ve Dijital (DO) olarak iki farklı çıkış çeşidi bulunmaktadır. PLC, hesaplanan çıkış için üretilen lojik gerilimi kontaktör, röle vb. elemanlar için elektriksel işaretlere çevirir. Yüksek hızlı sayıcılarda ise transistörlü çıkış birimleri kullanılmaktadır. Çıkış elemanı olarak genellikle fan, motor, röle vb. elemanlar kullanılmaktadır.

Dört ana birimden oluşan PLC'nin şematik gösterimi (a) ve çalışma şekli (b) Şekil 3.5'te görülmektedir.



Şekil 3.5. (a) PLC'nin şematik gösterimi, (b) PLC'nin çalışma şekli

PLC içindeki program taranırken ilk olarak PLC çıkışları sıfırlanır. Daha sonra, girişte bulunan sinyaller ve durumları işlem kaydedicisine alınır. Program 1. satırdan n. satıra kadar taranmaya başlanır. Her bir satırda işlem sonuçları kaydedilir. İşlem sonuçları belli oldukça çıkış durumları fiziksel çıkışlara aktarılır. Bu işlem dış dünyadan herhangi bir müdahale (elektrik kesintisi, acil durdurma işlemi vb.) olmadığı sürece devam eder. PLC'nin çalışmasının akış şeması Şekil 3.6'da görülmektedir.



Şekil 3.6. PLC'nin çalışma sistemi

3.1.3. İnsan-Makine ara yüzü

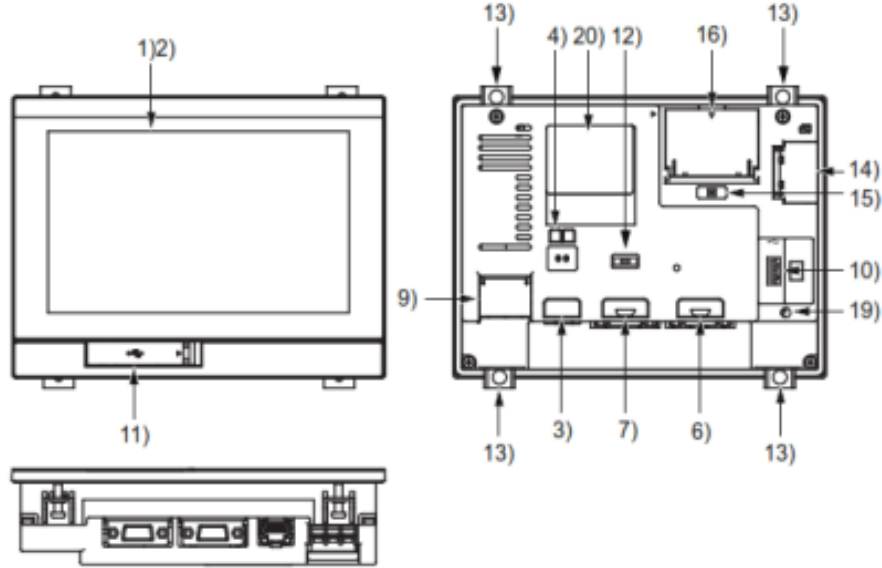
Sistemin çalışmasını kontrol etmek amacıyla bir ara yüz hazırlanmıştır. Arayüz üzerinden motor hız kontrolü, sistem çalıştır ve sistem durdur özellikleri eklenmiştir. Tasarlanan ara yüz Mitsubishi GS2107 ekrana yüklenmiştir. Ara yüz üzerinden sistemin kontrolü görselleştirilerek kolay kontrol edilebilir hâle getirilmiştir. Şekil 3.7'de kullanılan GS2107 ekran sunulmuştur. GS2107 ekran GT Designer3 adlı program tarafından programlanmaktadır. Mitsubishi Electric'in kendi bünyesinde çıkardığı bu yazılım sayesinde ekran tasarımı yapılabilmektedir. Ekran üzerinde Ethernet, RS422 ve seri haberleşme bulunmaktadır. Bu sayede ekran Siemens, Omron, Yaskawa gibi farklı marka ve model ürünlerle de çeşitli yöntemlerle haberleşebilmektedir.



Şekil 3.7. Sistem ara yüzünün yüklendiği ekran

3.1.3.1 GS2071 serisi ekranın terminal bağlantıları

GS2107 modelin terminal bağlantıları Şekil 3.8’de numaralandırılmış şekilde sunulmuştur.



Şekil 3.8. GS2107 modelin terminal numaraları

Şekil 3.8’de verilen terminal bağlantıları numaralandırılmıştır. Bu bağlantılar numaralarına göre açıklanırsa:

1 numara: Görüntü ekranı.

2 numara: Dokunmatik ekran.

3 numara: Kontrolör veya kişisel bilgisayar ile iletişim kurmaya yarayan ethernet portu (RJ-45 tip ethernet kablosu).

4 numara: Ethernet iletişimi durum ledi. SD ve RD LED olmak üzere 2 led bulunur. (Şekil 1.7’de görüldüğü gibi yan yana duran ledler) Eğer:

- SD / RD LED yanıyorsa data yollandı veya alındı anlamına gelir.
- SD / RD LED yanmıyorsa bilgi yollanamadı veya alınamadı anlamına gelir.
- Speed LED (SD) yanıyorsa 100Mbps hız ile haberleşme yapılıyor anlamına gelir.
- Speed LED (SD) yanmıyorsa 10Mbps hız ile haberleşme yapılıyor ya da bağlantı kopmuş anlamına gelir.

6 numara: Kontrolör ya da kişisel bilgisayar ile haberleşmeyi sağlayan RS-232 portu.

7 numara: Kontrolör ile haberleşmeyi sağlayan RS-422/485 portu.

9 numara: Güç bağlantısı terminali, “+”, “-“ 24 VDC güç terminali

10 numara: USB mouse, klavye ya da barkod okuyucu ve data transfer ya da datayı kaydetmek için kullanılan TYPE-A USB portu.

11 numara: Kişisel bilgisayarı ekrana bağlamak için kullanılan Mini-B USB portu

12 numara: RS-422/485 portu için sınırlama direnci. 330 ohm ya da 110 ohm olarak ayarlanabilir. (Fabrika çıkışı 330 ohm ayarında).

13 numara: Ekranın kasaya montaj noktaları.

14 numara: SD kart slotu.

15 numara: SD kart erişilebilme LEDi. Eğer;

- Yanıyorsa SD kart takılmıştır.
- Yanıp sönüyorsa SD karta erişim sağlanmıştır.
- Yanmıyorsa ya takılmamıştır ya da takılmış fakat SD karta erişilemiyordur.

16 numara: Batarya alanı.

19 numara: Kablo kelepçesi montaj deliği. USB kablosunun çıkmaması için alınabilecek önlem, kablo kelepçesi ile kabloyu bağlamaktır.

20 numara: Plaka.

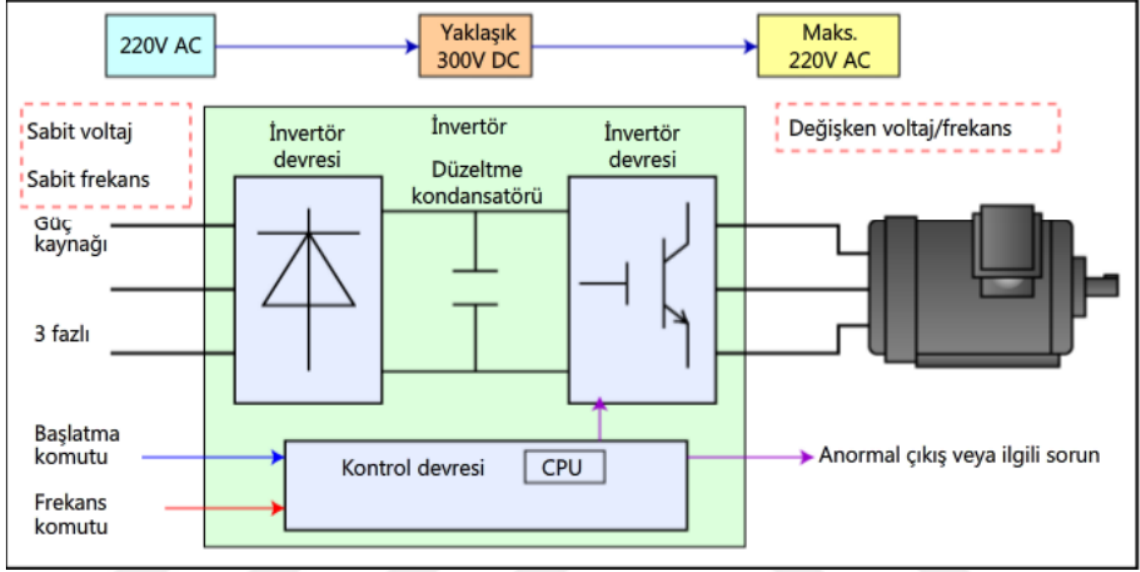
3.1.4. Motor sürücü

Bu tez çalışmasında, tasarlanan sistemde konveyörün hareketini sağlamak amacıyla motor sürücüsü kullanılmıştır. Asenkron motorların güçleri, farklı boyut ve uygulama gereksinimleri için geniş bir aralıkta değişebilir. Ancak genellikle, endüstriyel uygulamalar için kullanılan asenkron motorların gücü 0,18 kW'tan başlar ve 1000 kW'a kadar çıkabilir. Bu çalışmada, 180W asenkron motor kullanılmıştır. 180W asenkron motoru sürmek için kullanılan motor sürücünün maksimum çıkış akımı 2.5A iken, maksimum çıkış frekansı 400Hz'dir. Kullanılan motor sürücü Şekil 3.9'da görülmektedir.



Şekil 3.9. Konveyör kontrolü için kullanılan motor sürücü

İnverter 4 bölümden oluşmaktadır. Bunlar; konvertör devresi, düzeltme kondansatörü, inverter devresi ve kontrol devresidir. Bunlar aşağıda tek tek açıklanmıştır. Konvertör devresi, AC'yi DC'ye dönüştürürken; düzeltme kondansatörü, konvertör devresi tarafından dönüştürülen DC voltajın kalitesini yükseltmek için çalışır. İnverter devresi ise DC gerilimi AC gerilime çeviren bölümdür. İşlev olarak konvertör devresinin tersi olarak görev yapmaktadır. Kontrol devresi ise inverter devresini kontrol eden bir yapıya sahiptir. Şekil 3.10'da inverterin yapısı görülmektedir.



Şekil 3.10. Konveyör kontrolü için kullanılan motor sürücü

3.1.4.1 Motor sürücü parametreleri

Motor sürücülerin modeline ve serisine göre parametreler değişkenlik göstermektedir. Bu inverter serisinde sık kullanılan parametreler Şekil 3.11'de görülmektedir.

Pr.	Parametre adı	Ayar aralığı	Fabrika değeri	Set aralığı	Parametre açıklaması
1	Maksimum frekans	0,01Hz	120Hz	0-120Hz	İstenen minimum çıkış frekansı limiti
2	Minimum frekans	0,01Hz	0Hz		İstenen maksimum çıkış frekansı limiti
3	Baz frekans	0,01Hz	50Hz	0-400Hz	Motor etiketinde yazan frekans değeri girilir
4	Çoklu hız ayarı	0,01Hz	RH	50Hz	Değişik hızlarda çalışmak istendiğinde ve ilgili teminaller kullanıldığında belirlenmiş çıkış frekans değerleridir.
5			RA	30Hz	
6			RL	10Hz	
7	Hızlanma zamanı	0,1s	5/10/15s	0-3600s	Hızlanma ve yavaşlama sürelerinin girildiği parametrelerdir.
8	Yavaşlama zamanı		5/10/15s		
9	Elektronik termik röle	0,01A	Inverterin etiketinde yazan çıkış akımı	0-500A	Motor koruma fonksiyonu, motor etiket akımı girilir
71	Motor tipi	1	0	0/1/3/13/23/40/43/50/53	Uygulamada kullanacağınız motor tipini giriniz, standart tip motorlar için 3 girilebilir, özel motor kullanıyorsa manueli inceleyiniz.
77	Parametre yazma koruması	1	0	0/1/2	Bu parametre 0 yapılırsa inverter çalışmadığı zamanlarda parametre değiştirmek mümkün olur, 1 yapılırsa parametreleri hiçbir şekilde değiştirmek mümkün olmaz, 2 yapılırsa inverter çalışırken veya dururken parametreleri değiştirmek mümkün olur.
79	Çalışma modu seçimi	1	0	0/1/2/3/4/6/7	Start bilgisinin ve çalışma frekansının veriliş şekli ile ilgili lokasyonun seçildiği parametre
80	Motor etiketindeki güç değeri	0,01kw	9999	0,1 ile inverter gücü arası / 9999	Motor etiketindeki güç bilgisi girilir.
81	Motor kutup sayısı	1	9999	2/4/6/8/10	Motor etiketindeki kutup sayısı girilir
83	Motor etiketindeki voltaj değeri	0,1VAC	200VAC/400VAC	0VAC ile 1000VAC arası	Motor etiketindeki ve inverterin ön yüzündeki etiketteki voltaj bilgisi.
84	Motor etiketindeki frekans değeri	0,01Hz	50Hz	10 ile 120Hz arası	Motor etiketindeki frekans değeri girilir
96	Auto tuning tipi	1	0	0/11/21	Yapmak istediğimiz Autotuning tipi girilir. Bu parametre ile ilgili Autotuning prosedürünü inceleyiniz.
125	Frekans ayarı kazanç frekansı	0,01Hz	50Hz	0-400Hz	Terminal 2
126					Terminal 4

Şekil 3.11. Sık kullanılan inverter parametreleri

3.1.5. Enkoder

Endüstriyel enkoderler, hareketli parçaların pozisyonunu veya hareketini belirlemek için kullanılan cihazlardır. Bu cihazlar, endüstriyel otomasyon, robotik, CNC makineleri, rüzgâr türbinleri, motorlar, makine aletleri ve diğer birçok endüstriyel uygulamalarda yaygın olarak kullanılır. Endüstriyel enkoderler, bir mil üzerine monte edilir ve bir dönen disk veya şaftın hareketini okumak için kullanılır. Bu dönen disk veya şaft, hareketli parçaların hareketlerini takip etmek için manyetik, optik veya kapasitif sensörlerle okunur. Okunan sinyaller, belirli bir sayıda darbe veya yön değişikliği şeklinde işlenir ve daha sonra hareketin hızını, yönünü ve konumunu belirlemek için kullanılır. Endüstriyel enkoderlerin birçok avantajı vardır. Bunlar arasında yüksek doğruluk, yüksek çözünürlük, hızlı yanıt süresi, güvenilirlik,

dayanıklılık ve ölçüm aralığının genişliği yer alır. Bu çalışmada kullanılan enkoder Pepperl+Fuchs marka olup bir tur dönüşte 1024 darbe üretmektedir. Üretilen bu darbeler sayesinde asenkron motorun hangi konumda olduğu bilinmektedir. Bu bilgi kullanılarak robotun her zaman aynı aralıkta fotoğraf çekmesi sağlanmıştır. Kullanılan enkoder Şekil 3.12’de görülmektedir.

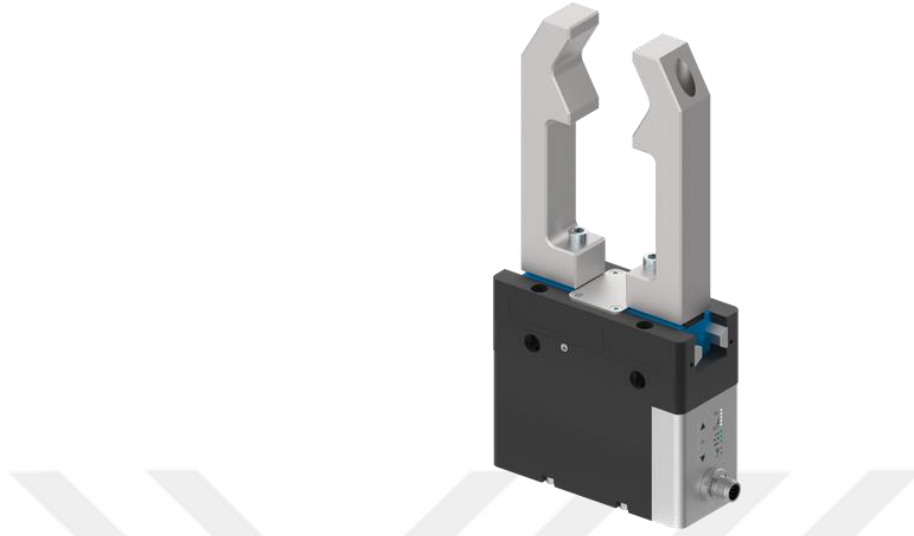


Şekil 3.12. Kullanılan enkoder

3.1.6. İki parmaklı tutucu

Endüstriyel tutucular, farklı uygulamalarda malzemeleri veya parçaları tutmak için kullanılır. Tutucular genellikle endüstriyel robotlar veya otomasyon sistemleri içinde normal ve hassas tutma gerektiren birçok uygulamada kullanılırlar. Endüstriyel tutucular çeşitli tiplerde gelir. Bunlar arasında mekanik tutucular, pnömatik tutucular ve hidrolik tutucular bulunur. Mekanik tutucular, yaylar ve dişli mekanizmalar gibi basit mekanik prensipler kullanarak çalışır. Pnömatik tutucular, basınçlı hava kullanarak hareket eder ve genellikle hızlı ve doğru hareket gerektiren uygulamalarda kullanılır. Hidrolik tutucular ise hidrolik sistemler kullanarak çalışır ve daha yüksek güç gerektiren uygulamalar için idealdir. Endüstriyel tutucular, farklı boyutlarda, şekillerde ve malzemelerde mevcuttur. Bunlar, uygulamanın gereksinimlerine bağlı olarak seçilebilir. Örneğin, parçaların tutulmasında kullanılan basit bir mekanik tutucu, 0,5 kg ağırlığındaki bir parçayı tutabilirken, hidrolik bir tutucu 1000 kg'dan fazla ağırlıktaki parçaları bile tutabilir. Endüstriyel tutucular, imalat, otomotiv, elektronik, gıda ve ambalaj endüstrileri gibi birçok endüstride kullanılır. Bu çalışmada, ürünleri toplamak amacıyla Zimmer marka 2 parmaklı elektrikli bir tutucu el robotun altıncı eksenine takılmış ve ürün toplama işlemi gerçekleştirilmiştir. Kullanılan tutucu el 8mm harekete

sahip olup, minimum tutma gücü 54N iken maksimum tutma gücü 215N'dur. 30 milyon defa açma-kapama özelliğine sahiptir. Şekil 3.13'te tutucu elin görseli sunulmuştur.



Şekil 3.13. Kullanılan tutucu el

3.1.7. Konveyör

Konveyörler, genellikle fabrika, depo ve endüstriyel tesislerde kullanılan bir taşıma sistemidir. Malzeme taşımak veya ürünleri taşımak için tasarlanmıştır. Malzemeler, çoğunlukla ürünlerin taşınması, paketlenmesi, sevkiyatı veya üretim hattı içindeki farklı işlem adımları arasında hareket ettirilmesi için kullanılır. Konveyörler, birçok farklı türde gelir ve birçok farklı malzeme ile yapılabilecekleri gibi, özel amaçlar için de tasarlanabilirler. Bazı yaygın konveyör tipleri şunlardır:

Bant Konveyörleri: Malzemeleri bir yerden diğerine taşımak için bantlar kullanır. Bu konveyörler, birçok farklı endüstriyel uygulamada kullanılır.

Rulo Konveyörleri: Rulolar, ağır yükleri taşımak için birbirine bağlanmıştır. Bu konveyörler, ağır yüklerin taşınması gereken endüstriyel ortamlarda kullanılır.

Zincir Konveyörleri: Zincirli konveyörler, zincirlerin hareketiyle malzemeleri taşır. Bu konveyörler genellikle ağır malzemelerin taşınması için kullanılır.

Hava Konveyörleri: Basıncı hava kullanarak malzemeleri taşır. Bu tür konveyörler, hafif malzemelerin taşınması veya hassas malzemelerin taşınması gerektiği yerlerde kullanılır.

Spiral Konveyörleri: Spiral konveyörler, ürünleri yukarı veya aşağı taşımak için bir spirale sahiptir. Bu tür konveyörler, genellikle depolama sistemlerinde veya yükleme istasyonlarında kullanılır.

Konveyörlerin birçok avantajı vardır. Malzemelerin elle taşınmasından daha hızlı ve verimli olmalarının yanı sıra, birçok konveyör sistemi, malzemelerin taşınması sırasında işçi yaralanmalarını önleyen güvenlik önlemleri içerir. Ayrıca, konveyörler, üretim hatlarının daha verimli ve düzenli çalışmasını sağlayarak, üretim sürecinde zaman ve maliyet tasarrufu sağlarlar.

Konveyör hareketi için çeşitli türlerde motorlar kullanılır. Bu motorlar konveyör miline bağlanır. Bağlantı esnasında motor devrini düşüren bir redüktör kullanılır. Konveyör sistemlerinde, redüktörlerin kullanımı oldukça önemlidir. Redüktörler, konveyörün hareketini kontrol etmek ve işletme süresince yüksek torklu hareketi sağlamak için kullanılan mekanik cihazlardır. Redüktörler, konveyörün motorundan aldığı yüksek devirli hareketi, düşük devirli yüksek torklu harekete dönüştürür. Bu dönüşüm, konveyör bandının, rulolarının ve diğer hareketli parçalarının daha yavaş hareket etmesine izin verir. Ayrıca, redüktörler, konveyörün hareketini daha istikrarlı ve düzenli hale getirir, böylece malzemenin taşınması sırasında olası titreşimleri ve hasarları azaltır. Redüktörler aynı zamanda, konveyörün işletme maliyetlerini de azaltabilir. Redüktörler, konveyör motorunun daha yüksek hızlarda çalışmasını önleyerek enerji tasarrufu sağlar. Daha düşük motor hızları, konveyör bandının daha yavaş hareket etmesine ve daha az güç tüketmesine izin verir. Bu çalışmada kullanılan bantlı konveyör sistemi Şekil 3.14'te görülmektedir.



Şekil 3.14. Kullanılan bantlı konveyör

3.1.8. Endüstriyel haberleşme yöntemleri

Endüstriyel haberleşme, endüstriyel cihazlar ve sistemler arasında veri ve bilgi paylaşımını sağlayan bir iletişim yöntemidir. Endüstriyel haberleşme yöntemleri,

endüstriyel kontrol sistemlerinde önemli bir rol oynar ve endüstriyel süreçlerin yönetiminde kullanılır. Çeşitli haberleşme yöntemleri aşağıda sunulmuştur:

Ethernet: Ethernet, TCP/IP protokolü kullanarak bir ağ üzerinden cihazlar arasında veri iletişimi sağlayan bir teknolojidir. Ethernet, hızlı veri iletimi sağlar ve endüstriyel otomasyon uygulamaları için yaygın bir haberleşme yöntemidir.

Modbus: Modbus, seri haberleşme protokolü kullanarak cihazlar arasında veri paylaşımı sağlayan bir endüstriyel haberleşme protokolüdür. Modbus, endüstriyel otomasyon cihazlarının kontrolü için yaygın olarak kullanılır.

Profibus: Profibus, endüstriyel otomasyon cihazları arasında veri iletimi için kullanılan bir seri haberleşme protokolüdür. Profibus, çok düşük gecikme süreleri ve yüksek hızlı veri iletimi sağlar.

AS-i: AS-i (Actuator Sensor Interface), endüstriyel otomasyon cihazları arasında veri paylaşımı için kullanılan bir haberleşme protokolüdür. AS-i, dijital sinyallerin ve gücün tek bir kabloyla taşınmasına olanak tanır.

HART: HART (Highway Addressable Remote Transducer), endüstriyel sensörler ve cihazlar arasında dijital ve analog iletişimi sağlamak için kullanılan bir haberleşme protokolüdür. HART, sürekli işlem kontrolü ve otomatik regülasyon gibi uygulamalarda yaygın olarak kullanılır.

Bu çalışmada Modbus haberleşme yöntemi kullanılmıştır. Modbus, 1979 yılında Modicon tarafından PLC'ler ile kullanılmak için geliştirilen bir seri haberleşme protokolüdür. Modbus TCP/IP, Modbus mesaj yapısının verilerini uyumlu cihazlar arasında taşımak için TCP/IP ve Ethernet kullanır. Yani, Modbus TCP/IP fiziksel bir ağı (Ethernet), bir ağ standardıyla (TCP/IP) verileri temsil eden standart bir yöntemdir. Modbus protokolünde cihazlar arası haberleşme sağlamak için Master / Slave tekniği kullanılır. Bu ise Modbus protokolünü kullanan herhangi bir uygulamanın bir Modbus Master ve en az bir Modbus Slave'i olduğu anlamına gelmektedir. Bir Modbus Master genellikle bir veya daha fazla Modbus Slave cihazı ile haberleşecek bir denetleyici bilgisayardır. Modbus TCP/IP haberleşme yönteminde gönderilen mesajlar iki ana bölümden oluşur. MBAP ve PDU bölümleri mesajın bütünlüğünü oluşturan kısımlardır (Modbus TCP/IP, 2023). MODBUS mesaj bilgisi Şekil 3.15'te görülmektedir.



Şekil 3.15. MODBUS TCP/IP haberleşme yapısı

MBAP (Modbus Application Protocol Header – MODBUS Uygulama Protokol Başlığı) Bölümü

Modbus TCP/IP çerçevelemesinde MBAP başlığı dört bölümden oluşur. MBAP başlığının uzunluğu 7 byte olup MBAP bölümü haberleşme için kullanılır. Bu bölüm Master ve slave birimlerinin birbiri ile haberleşebilmesi için bazı bilgileri taşır.

İşlem Tanımlayıcı

Master ve slave noktalarını birbirine ilişkilendirerek iki cihazın birbiriyle haberleşebilmesini sağlayan bölümdür. İşlem tanımlayıcı bölümü 2 Byte uzunluğundadır.

Protokol Tanımlayıcı

Çoklu sistemler için düşünülmüştür. Modbus için 0 değerini alır. Gelecekte kullanılabileceği düşünülerek bu alan ayrılmış olup protokol tanımlayıcı bölümü 2 Byte uzunluğundadır.

Uzunluk

Uzunluk bölümü, birim tanımlayıcı bölümü ve Modbus TCP/IP PDU bölümlerini kapsayan veri uzunluğunu byte olarak belli eder ve 2 Byte uzunluğundadır.

Birim Tanımlayıcı

Bu alan sistem için yönlendirme amacıyla kullanılır. Ağ üzerinde bulunmayan uzak birimi belirleme amacıyla kullanılır. Birim tanımlayıcı bölümü 1 Byte uzunluğundadır.

Modbus TCP/IP PDU Bölümü

Modbus TCP/IP PDU bölümü fonksiyon kodu ve data olmak üzere 2 bölümden oluşmaktadır. Modbus TCP/IP çerçevelemesinde Modbus TCP/IP PDU bölümü temel olarak işleve yönelik kodları içerir.

Fonksiyon Kodu

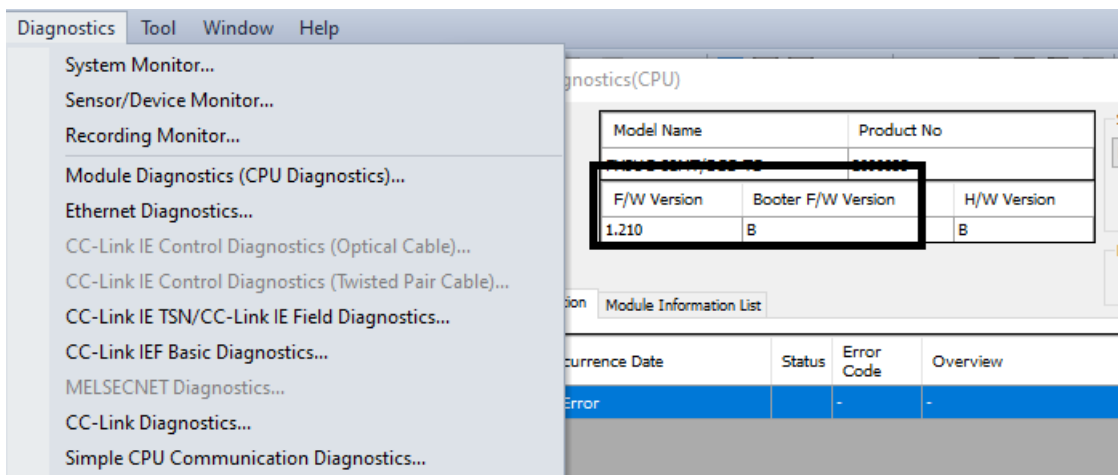
Fonksiyon kodu bölümü, master ünitelerden slave üniteye mesaj gönderiminde yapılması talep edilen işleve yönelik koddan oluşur. Slave ünitelerden master üniteye cevap gönderiminde ise talebe karşılık gelen fonksiyon kodundan oluşur. Fonksiyon kodunun uzunluğu 1 byte'dır.

Data Bölümü

Data bölümü master ünitelerden slave üniteye mesaj gönderiminde yapılması talep edilen işleve yönelik özel veriler içerir. Slave ünitelerden master üniteye cevap gönderiminde ise talebe karşı gönderilecek cevap verileri veya hata kodlarından oluşur. Data bölümü değişken uzunluktadır.

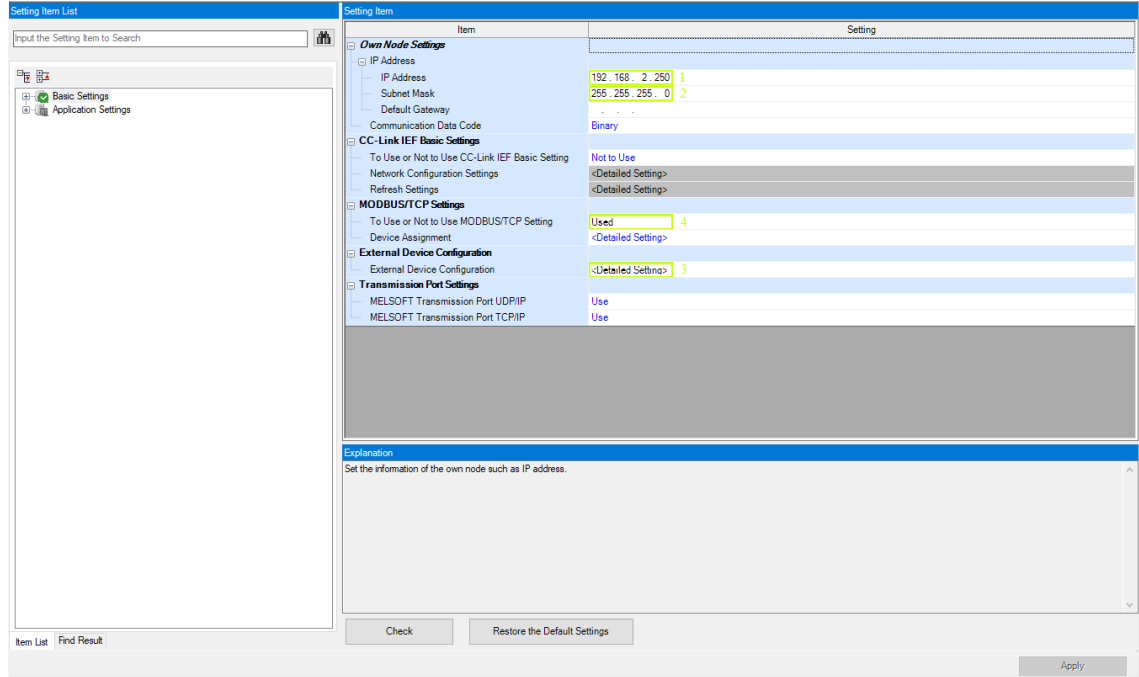
PLC Modbus TCP/IP Haberleşme Ayarı

PLC ve Python programı MODBUS TCP üzerinden haberleştirilmiştir. FX5U PLC'nin MODBUS TCP üzerinden haberleşmesi için donanım versiyonunun 1.060'ın üzerinde olması gerekmektedir. PLC sürümüne Şekil 3.16'da görüldüğü üzere "Diagnostics→Module Diagnostics (CPU Diagnostics)" yolu ile ulaşılabilmektedir. Bu pencerede "F/W Version" adı altında donanım versiyonu görüntülenebilmektedir.



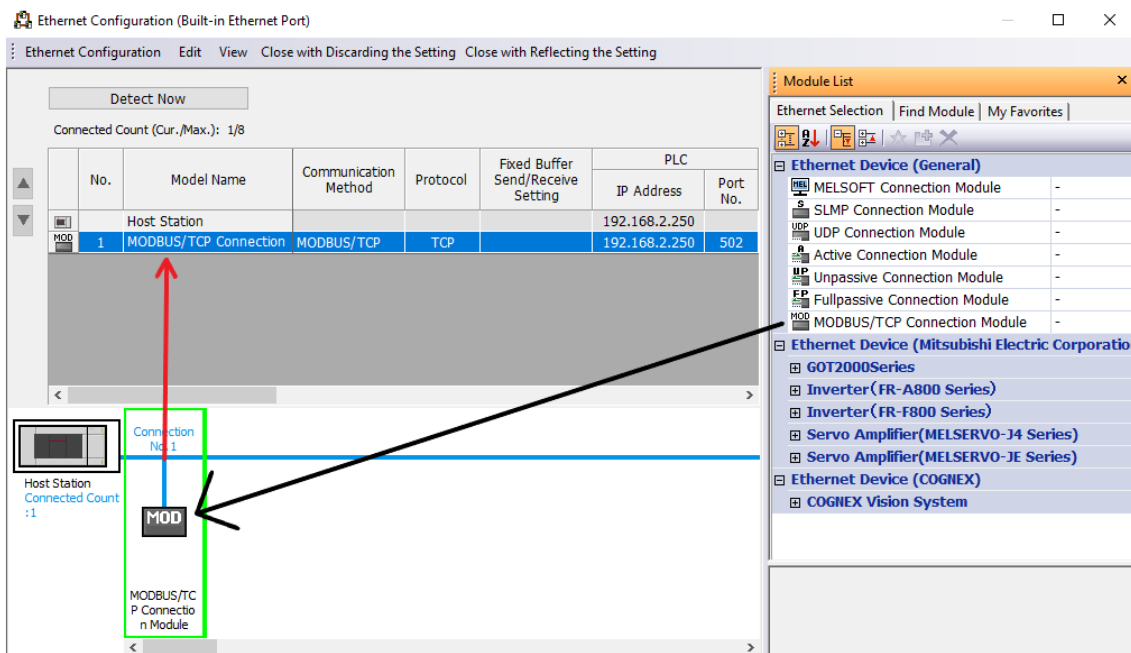
Şekil 3.16. Donanım versiyon kontrolü

Donanım versiyonu 1.060'ın üzerinde olan PLC'nin haberleşme ayarlarını yapmak amacıyla arayüzde bulunan "Navigation-Parameter-FX5UCPU-Module Parameter-Ethernet Port" ayarları açılır. Açılan sayfa Şekil 3.17'de görülmektedir.



Şekil 3.17. Ethernet port ayarları

PLC'nin ethernet portu üzerinden haberleşebilmesi için ethernet ayarlarının yapılması gerekmektedir. Şekil 3.17'de görülen 1 numaralı kısma PLC'nin IP adresi tanımlanmalıdır. Ağ içerisinde bulunan cihazların (PLC, arayüz, robot, bilgisayar) IP adresinin ilk 3 bloğunu aynı ve son bloğunun farklı olmasına dikkat edilmelidir. 2 numaralı "Subnet Mask" kısmına alt ağ geçiş maskesi olarak da bilinen "255.255.255.0" değerinin tanımlanması gerekmektedir. Bu işlemin ardından 3 numaralı kısımdan Şekil 3.18'de görülen pencere açılmalıdır.



Şekil 3.18. FX5U CPU'nun port configuration ayarları

Şekil 3.18'de siyah ok ile gösterilmiş olan “MODBUS/TCP Connection Module” sürüklenip siyah okun ucundaki alana bırakılmalıdır. Modül bu kısma eklenince kırmızı ok ile gösterilmiş olan satır ortaya çıkar. Ayar tamamlandıktan sonra sayfa kapatılırken yapılan ayarların kaydedilir ve MODBUS/TCP ayarları tamamlanmış olur.

3.1.9. Görüntü işleme

Görüntü işleme, endüstriyel üretim süreçlerinde sıklıkla kullanılan bir teknolojidir. Görüntü işleme, endüstride üretim süreçlerinin daha hızlı, daha doğru ve daha verimli hale getirilmesine yardımcı olur. Görüntü işlemenin endüstride kullanıldığı alanlar aşağıda sunulmuştur:

Kalite kontrolü: Endüstride üretilen ürünlerin kalite kontrolü, görüntü işleme teknolojisi kullanılarak yapılabilir. Görüntü işleme algoritmaları, ürünlerin boyut, şekil, rengi ve diğer özelliklerini kontrol ederek hataları tespit eder.

Otomatik tanıma: Görüntü işleme, endüstride kullanılan parçaların ve ürünlerin otomatik tanınmasına yardımcı olabilir. Örneğin, bir üretim hattında farklı parçaların tanınması için görüntü işleme teknolojisi kullanılabilir.

Robotik görüş: Görüntü işleme, endüstride kullanılan robotik sistemlerin yönlendirilmesinde de kullanılır. Robotlar, üretim sürecindeki nesneleri algılamak ve takip etmek için görüntü işleme teknolojisi kullanılabilir.

Otomatik sınıflandırma: Görüntü işleme, endüstride üretilen ürünlerin otomatik olarak sınıflandırılmasına yardımcı olur. Örneğin, bir üretim hattında farklı ambalajların otomatik olarak sınıflandırılması için görüntü işleme teknolojisi kullanılabilir.

Belge işleme: Görüntü işleme, endüstride kullanılan belgelerin dijitalleştirilmesinde de kullanılabilir. Örneğin, bir üretim hattında kullanılan fatura ve sevkiyat belgeleri gibi kâğıt tabanlı belgelerin dijitalleştirilmesi için görüntü işleme teknolojisi kullanılabilir.

Bu uygulamaların yanı sıra, görüntü işleme teknolojisi endüstride malzeme kontrolü, tıp, güvenlik, trafik yönetimi ve daha birçok alanda da kullanılmaktadır.

3.2. Yöntem

Tez çalışmasının ikinci aşaması olan görüntü işleme aşamasında Logitech C270 marka kameradan alınan görüntüler bilgisayar ortamında işlenmiştir. Görüntü işleme aşamasında renk dönüşümü, eşikleme, kenar tespiti algoritmaları Python dili ve OpenCV kütüphanesi kullanılarak işlenmiştir. Python günümüzün en popüler programlama dillerinden biridir ve birçok farklı amaç için kullanılmakta olup açık kaynak bir dildir. Genel amaçlı bir dildir ve birçok farklı alanda kullanılır. Örneğin, web geliştirme, veri bilimi, yapay zeka, oyun geliştirme ve sistem yönetimi gibi alanlarda kullanılabilir. Kolay okunabilen ve anlaşılabilir bir sözdizimine sahiptir. Bu nedenle, öğrenmesi diğer dillere göre daha kolaydır. Nesne yönelimli bir programlama dili olarak tasarlanmıştır. Bu sayede yazılan kod yeniden kullanıma, bakım yapmaya ve hata ayıklamaya yardımcı olur. Python, geniş bir kütüphane ve modül yelpazesine sahiptir. Bu kütüphaneler sayesinde grafik oluşturma, web geliştirme, bilimsel hesaplama ve makine öğrenmesi vb. birçok işlev kolayca gerçekleştirilebilir. Python, birçok farklı platformda çalışabilir. Windows, Linux ve MacOS gibi işletim sistemleri ile uyumludur.

3.2.1. Renk dönüşümü

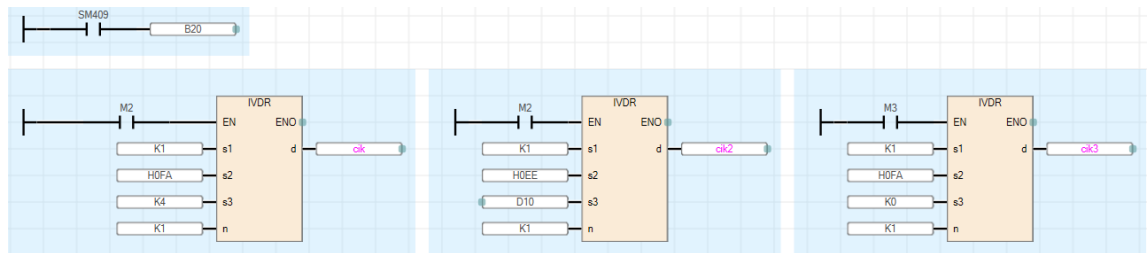
Kameradan alınan görüntüde renkler üzerinden daha kolay işlem yapabilmek için BGR (Blue Green Red) renk uzayından HSV (Hue Saturation Value) renk uzayına geçiş yapılmıştır. HSV renk uzayı sırasıyla renk özü, doygunluk ve parlaklık terimlerini ifade eder. Renk özü, rengin baskın dalga uzunluğunu belirler. Doygunluk, rengin canlılığını belirler. Parlaklık ise rengin aydınlığın yani beyaz oranını belirler. Şekil 3.19'da HSV renk uzayının gösterimine yer verilmiştir (Chang ve ark, 2022).

3.2.4. Basit kapalı alan tespiti (Simple blob detection)

Basit kapalı alan, bir görüntüde benzer özellikleri taşıyan birbirine bağlı piksellerden oluşan bir gruptur. Opencv kütüphanesi içinde bulunan kapalı alan tespiti fonksiyonu birkaç parametre ile kontrol edilebilmektedir. Bu parametreler dairesellik, dışbükeylik, eşik sınırı ve alan parametreleridir. Dairesellik parametresi ile daire şeklindeki kapalı alanlar bulunabilirken dışbükeylik parametresi ile çok kenarlı alanlar tespit edilebilmektedir. Eşik sınırı ve alan parametreleri ile kapalı alanın piksel bazında özellikleri kullanılarak kapalı alan tespiti yapılabilmektedir (Nguyen and Chung, 2009).

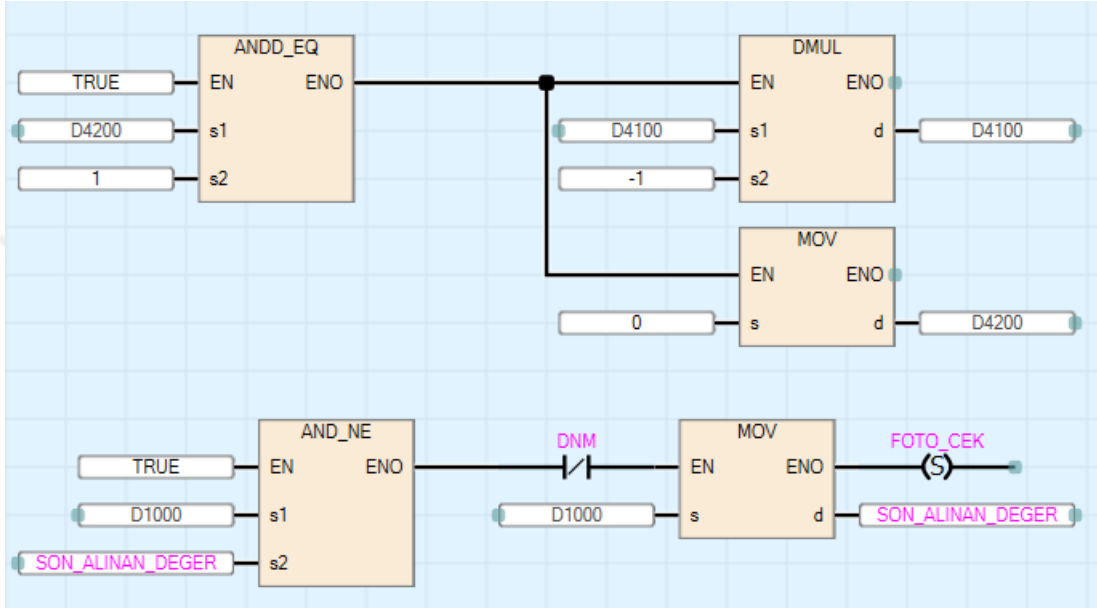
3.2.4. PLC Programı

Tez çalışmasında tasarlanan sistemde bütün kontroller PLC üzerinden sağlanmıştır. PLC içinde çeşitli programlama yöntemleri bulunmaktadır: Ladder, FBD, ST. Ladder dili PLC Programlama yapılırken en yaygın kullanılan dillerden biridir. Elektrik devrelerini temel alan programlama dilidir. Programlama yapılırken yazılan komutlar alt alta yazıldığı için merdiven görünümü oluşturmaktadır. Bu yüzden Ladder (Merdiven) programlama dili denir. FBD sinyal ve veri akışını fonksiyon blokları üzerinden anlatmak için kullanılan grafiksel bir dildir. FBD, kontrol sistemleri algoritmaları ile Mantık kapıları arasındaki ara bağlantıyı ifade etmek için çok kullanışlıdır. ST ise yapısal programlamayı teşvik eden üst düzey bir programlama dilidir. PASCAL'a çok benzer bir yapısı (sözdizimi) vardır. Yani daha çok programlamayı bilenleri hedef almaktadır. Tez çalışmasında FBD programlama dili kullanılmıştır. FBD dili ile konveyör kontrolü, görüntü işleme kontrolü ve robot kol hareketi PLC üzerinden yapılmıştır. Yazılan kod bu bölümde ayrı ayrı açıklanmıştır. Yazılan PLC kodunda ilk olarak haberleşme hızı ve konveyör kontrolü için gerekli kısım yazılmış olup Şekil 3.21'de görülmektedir.



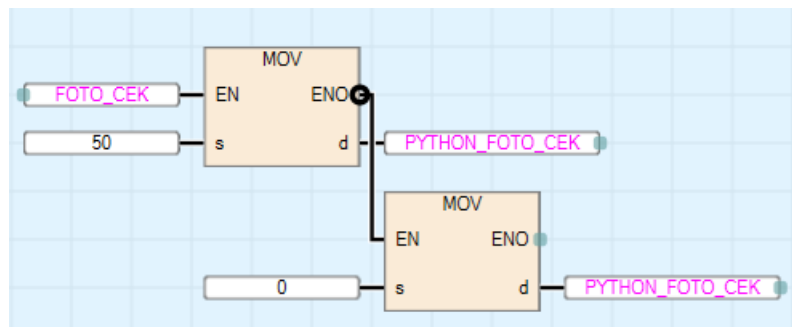
Şekil 3.21. Haberleşme hızı ve inverter kontrolü

Şekil 3.21’de SM409 biti ile haberleşme hızı ayarlanmıştır. SM409 biti 1KHz’lik darbe üreterek her yükselen kenarda veri aktarımını sağlamıştır. Bu sayede 1ms’de bir veri aktarımı gerçekleştirilmiştir. IVRD fonksiyon bloğu inverter kontrolü için kullanılmış olup M2 butonu ile konveyör çalışmakta, M3 butonu ile konveyör durmaktadır. Şekil 3.22’de robot kol tarafından Python’a gönderilen görüntü işleme yapması amacı ile “Fotoğraf Çek” komutunun gönderilmesi aşaması görülmektedir.



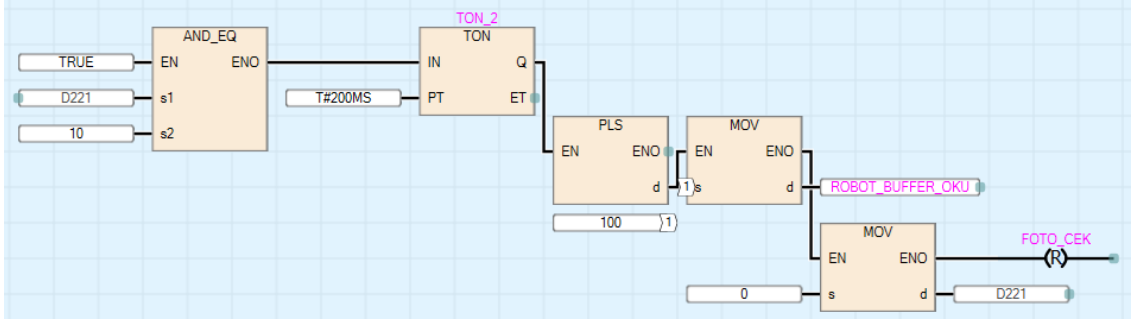
Şekil 3.22. Robot kol tarafından gönderilen fotoğraf çek komutu

Şekil 3.23’te fotoğraf çek komutu geldikten sonra Python programına Modbus TCP/IP üzerinden 50 değeri gönderilerek görüntü işleme yapılması sağlanır. Görüntü işleme bittikten sonra görüntü işleme yap talimatı otomatik olarak sıfırlanır. Bunun sebebi sistemin sadece bir kez görüntü işleme yapması gerektiği içindir.



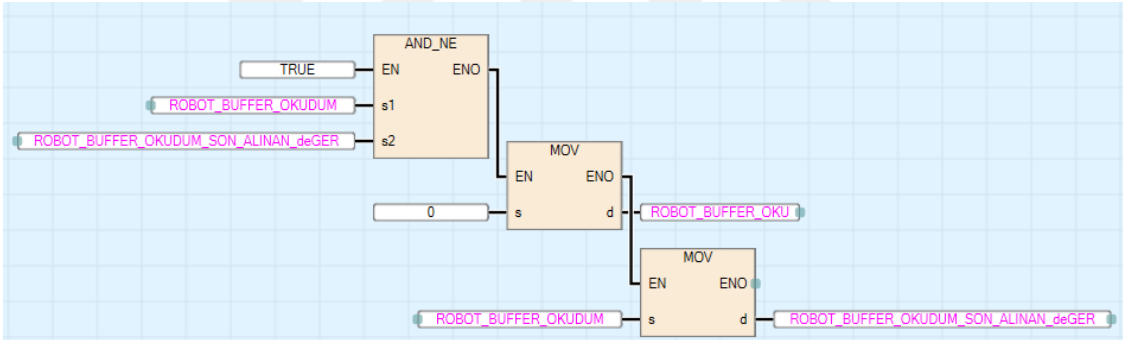
Şekil 3.23. Sadece bir kez yapılan görüntü işleme

Görüntü işlemeden elde edilen bilgiler PLC üzerinden robot buffer adreslerine yazılır. Şekilde fotoğraf çekme işlemi bittikten sonra robota gönderilen verilerin “Robot Buffer Oku” biti üzerinden okuma işlemi yapılmaktadır.



Şekil 3.25. Robotun gelen bilgileri okumasını sağlayan kod parçacığı

Robot buffer okuma aşamasında bir önceki değerlerin değiştiğinin kontrolünün yapılması gerekmektedir. Bir önceki buffer değeri ile anlık buffer değeri arasında fark olmadığı durumda buffer okunmamakta, farklı bir ürün bilgisi geldiğinde buffer okunarak robotun doğru ürüne gitmesi sağlanmaktadır.



Şekil 3.26. Buffer son alınan ürün bilgisi kontrolü

4. DENEYSSEL ÇALIŞMA VE TARTIŞMA

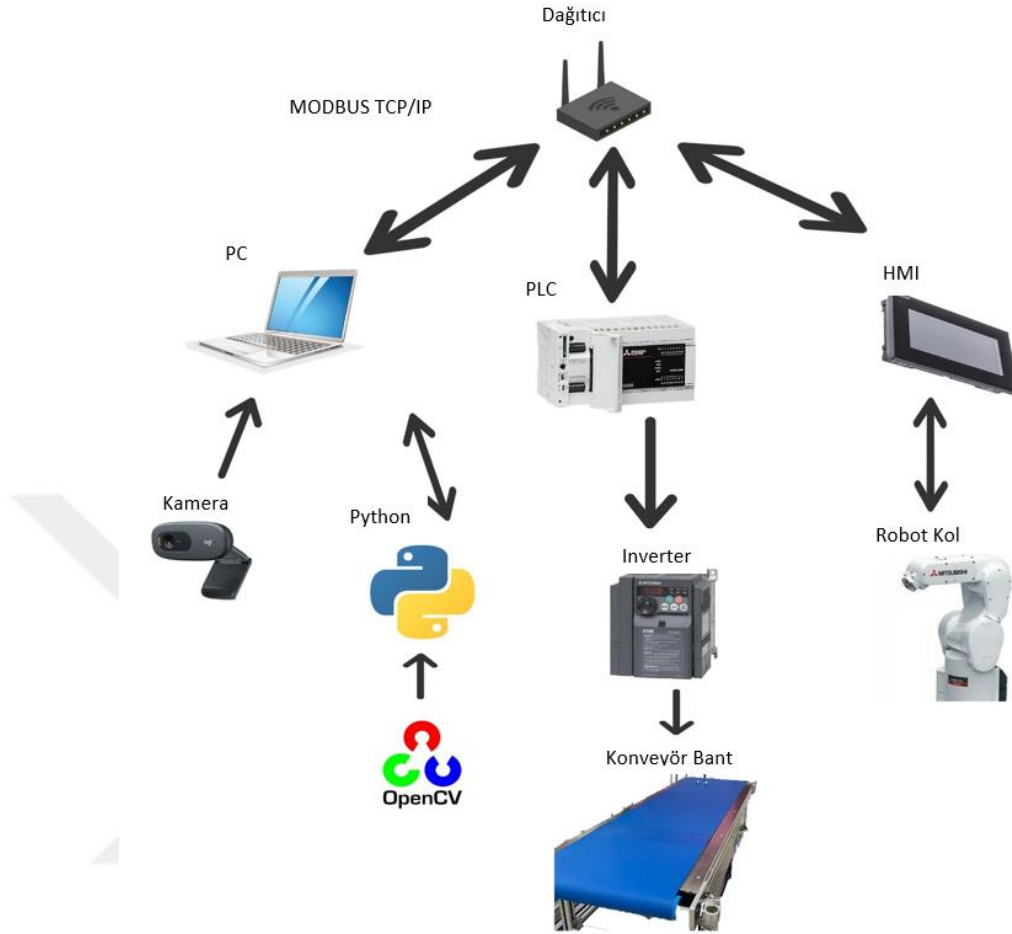
Tez çalışmasında, çalışma konusuna uygun olan, Konya Organize Sanayi Bölgesi'nde çikolata üreten ve paketleyen makinaları üreten bir firmada, gerçek çalışma ortamına uygun bir deneysel sistem kurulmuştur. Bu deney sisteminde birer adet PLC, HMI, konveyör, Inverter, asenkron motor, robot kol, kamera ve bilgisayar kullanılmıştır. Kurulan deneysel sistemde hem manuel hem otomatik paketleme sistemi deneyleri gerçekleştirilmeye çalışılmıştır.

Bu deneyler sonucunda manuel sistemde bir işçinin maksimum performans ile toplayabileceği çikolata sayısı çeşitli ortamlarda ve farklı zaman dilimlerinde sayılarak kayıt altına alınmıştır. Aynı şekilde, otomatik sistemde de aynı şartlar altında robot kolun toplayabileceği çikolata sayısı kaydedilmiştir.

Manuel deney sisteminde bir işçi ilk yarım saatte çok yüksek hızda çikolata toplayabilirken çalışmaya başladıktan bir saat sonra çikolata toplarken hatalar yapmaya başlamıştır. Bu hatalar çikolatanın pakete tam yerleştirilememesi, çikolatanın banttan alırken fazla sıkıldığından ötürü şeklinin deforme olması, alınmaması gereken kırık hatalı üretilmiş çikolataların alınması vb. hatalar olarak kaydedilmiştir. Ayrıca, insan elinin sıcaklığının dokunduğu çikolata üzerinde deformasyona sebep olduğu gözlenmiştir. Bir işçi çalışmaya başladıktan iki saat sonra ise hatalar iyice artmaya başlamış ve üretim kalitesinin düştüğü görülmüştür. Manuel sistemde bir işçi bir mesai saati içinde ortalama 11700 çikolata paketleyebiliyor olmakla birlikte paketlenen çikolatalar içindeki hatalı çikolata sayısının mesai sonuna doğru çok daha fazla arttığı görülmüştür. Bu sayı içerisindeki çikolataların da hatalı ürün oranı işçinin performansına göre %10-20 arasında değiştiği gözlenmiştir.

Robotlu deney sisteminde ise robot çalışma hızı zamandan bağımsız olarak hep aynı kalmıştır. Robot çalışma süresince çikolata paketlerken herhangi bir hata yapmadan çikolatayı doğru şekilde paketlediği, çikolatayı tutma sertliği hep aynı olduğu için çikolatayı deforme etmediği ve kameradan gelen bilgiye bağlı olarak hatalı çikolataları almadığı gözlemlenmiştir. Robotlu deney sisteminde robotun çalışma süresince 14400 çikolata paketleyebildiği ve paketler içerisinde hatalı çikolata olmadığı gözlemlenmiştir. Robotun optimum çalıştığı bu sistemde robotu etkileyen parametreler deneysel olarak değiştirilmiş ve en uygun parametreler ile deneysel sistem testleri yapılmıştır. Bu parametreler konveyör ve robot hızı olarak tespit edilmiş ve deneysel değerler kaydedilerek elde edilen sonuçlar değerlendirilerek sonuçlar bölümünde tablo halinde sunulmuştur.

Deneysel paketleme sisteminin akış ve haberleşme şeması Şekil 4.1’de görülmektedir.



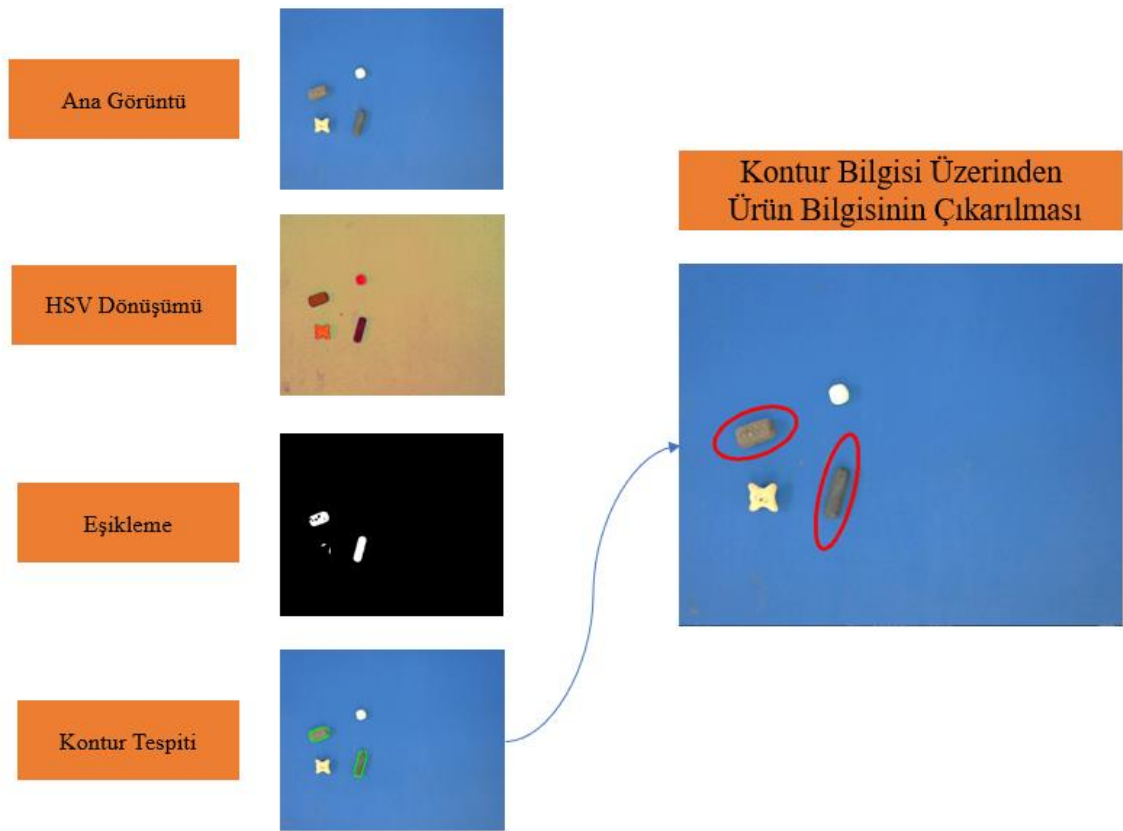
Şekil 4.1. Kurulan sistemin yapısı

Kurulan sistemde PLC, HMI, Robot ve bilgisayar bir dağıtıcı üzerinden haberleşmektedir. Kameradan alınan görüntüler bilgisayar üzerinde Python yazılım dili içerisinde OpenCV kütüphanesi ile işlenerek elde edilen sonuçlar dağıtıcı üzerinden PLC'ye gönderilmektedir. PLC bu bilgileri tekrar dağıtıcı üzerinden HMI'a göndermekte ve oradan bilgiler robota aktarılmaktadır. Konveyör kontrolü PLC'nin RS485 portuna bağlanmış olan motor sürücü ile yapılmaktadır.

4.1. Görüntü İşleme Sonuçları

Tez çalışmasının görüntü işleme ile ürün bilgilerinin çıkarılması aşamasında 3 yöntem kullanılmış ve elde edilen sonuçlar karşılaştırılmıştır. İlk yöntemde HSV dönüşümü, eşikleme ve kontur tespiti yapılmıştır. İkinci yöntemde OpenCV kütüphanesinin kapalı alan tespiti (blob detection) yöntemi kullanılmıştır.

Birinci yöntemde, kameradan alınan görüntü ilk olarak HSV dönüşümüne tabi tutulmuştur. HSV dönüşümü sonucunda elde edilen görüntü eşiklenmiştir. Siyah, beyaz ve krem renkli çikolata için üç ayrı eşik değeri tanımlanmıştır. Tanımlanan 3 eşik değeri hiçbir zaman diğer çikolatayı içermemektedir. Eşiklenen görüntüye daha sonra erozyon işlemi uygulanmış ve küçük noktalar silinmiştir. Elde edilen kalan görüntüde 0'dan büyük tüm piksellere 255 değeri atanarak görüntü siyah beyaz görüntüye dönüştürülmüştür. Siyah beyaz görüntü üzerinde kontur tespiti yapılmış ve kontur bilgisi üzerinden ürünün merkez X ve Y koordinatı, tipi ve açısı tespit edilmiştir. Yapılan işlemin akış şeması Şekil 4.2'de görülmektedir.



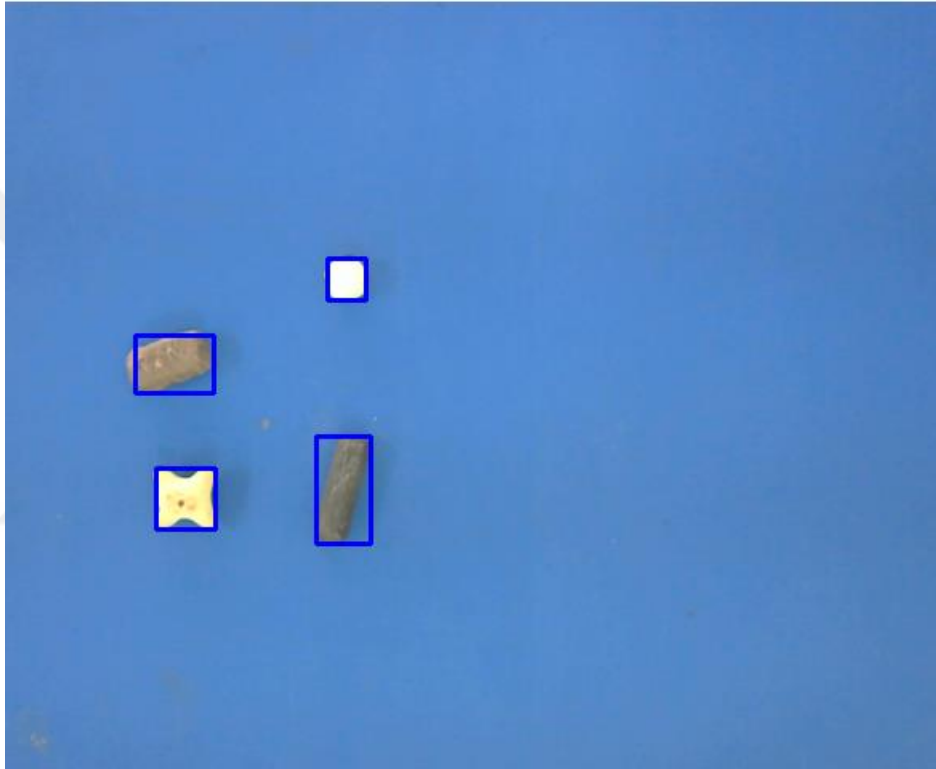
Şekil 4.2. Kahverengi çikolata için elde edilen sonuçlar

Şekil 4.2'de görülen akış yıldız şekilli krem renkli ürün ve beyaz çikolata için de uygulanmış ve sonuçlar elde edilmiştir. 3 farklı çikolata için elde edilen HSV eşikleme değerleri Çizelge 4.1'te sunulmuştur.

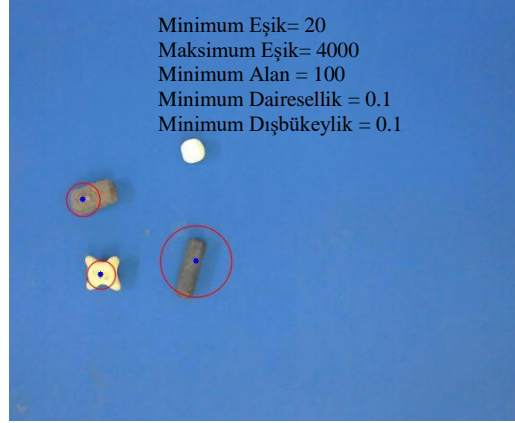
Çizelge 4.1. Her bir ürün için ayrı ayrı eşik değerleri

Ürün	Alt Eşik Sınırı (H, S, V değerleri)	Üst Eşik Sınırı (H, S, V değerleri)
Siyah Çikolata	0,0,0	255,255,160
Beyaz Çikolata	0,0,0	255,50,255
Krem Çikolata	0,0,173	255,127,255

Çizelge 4.1’de görülen değerler tek bir görüntüye sıra ile uygulandığında bir görüntüde var olan 3 farklı ürün elde edilebilmektedir. Bu durum Şekil 4.3’te sunulmuştur.

**Şekil 4.3.** Üç farklı eşik değeri uygulandığında ürünlerin tespit edilmesi

İkinci yöntemde, kameradan alınan görüntü içindeki nesnelere bulunmuştur. Nesne tespiti aşamasında OpenCV kütüphanesinin “SimpleBlobDetector” fonksiyonu kullanılmıştır. Fonksiyon parametreleri değiştirilerek en uygun sonucu veren parametreler elde edilmiştir. Parametre olarak “minThreshold”, “maxThreshold”, “minArea”, “minCircularity” ve “minConvexity” parametreleri kullanılmıştır. Kullanılan parametreler ve elde edilen tespit sonuçları Şekil 4.3-4.6 arasında sunulmuştur.



Şekil 4.4. Belirlenen parametreler ile elde edilen sonuçlar



Şekil 4.5. Belirlenen parametreler ile elde edilen sonuçlar



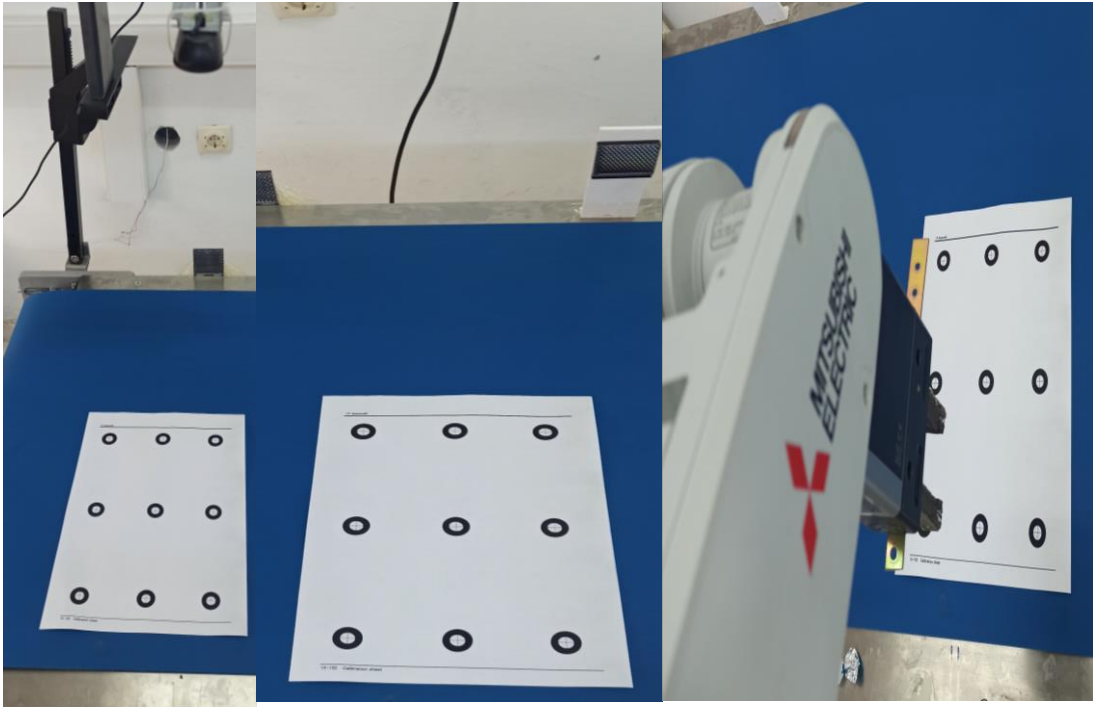
Şekil 4.6. Belirlenen parametreler ile elde edilen sonuçlar

Şekil 4.3-4.6 arasında görüleceği üzere parametreler değiştikçe tespit edilebilen ürün sayısı değişmektedir. SimpleBlobDetection fonksiyonunda tespit edilen ürünlerin hangi sınıfa ait olduğu bilinmemektedir. Bu durum bu fonksiyon için bir dezavantajdır. Ayrıca, şekiller incelendiğinde kahverengi çikolataların merkez tespitinin doğru yapılamadığı görülmektedir. Bu şekilde robot ürünü tutsa bile taşırken ürünün düşme riski ortaya çıkmaktadır. Ancak, birinci yöntemde her bir ürüne özel eşik seviyesi bilindiği için ürünler tam olarak bulunabilmektedir. Bundan dolayı, ürüne özel ayarların

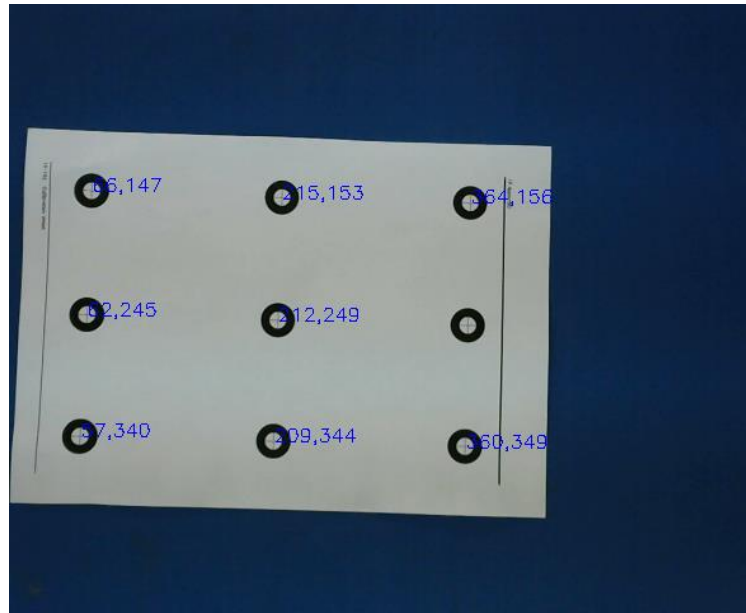
bulunarak yazılımın bu ayarlar üzerinden çalıştırılması en doğru yol olarak görünmektedir.

4.2. Kamera ve Robot Kalibrasyonu

Tez çalışmasında robot ile ürün toplama işlemine başlamadan önce ilk olarak robot-kamera ve enkoder kalibrasyonu yapılması gerekmektedir. Bunun için kalibrasyon kâğıdı kullanılmaktadır. Kalibrasyon kâğıdı ilk önce kameranın altına konular ve üzerindeki noktaların pozisyonları alınır işaretlenir. Daha sonra, konveyör hareket ettirilerek kalibrasyon kâğıdı robot kolun önüne getirilir ve konveyör durdurulur. Bu işlem Şekil 4.7’de görülmektedir.

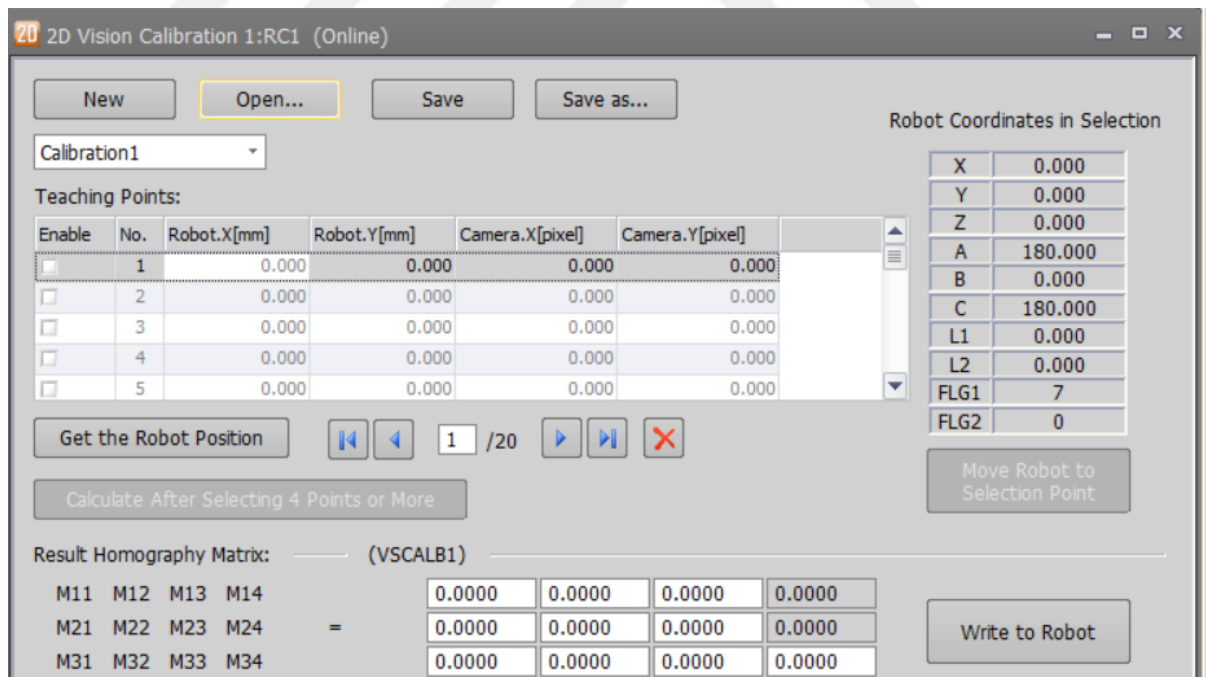


Şekil 4.7. Kamera altında noktaları kaydedildikten sonra robot altına getirilen kalibrasyon kâğıdı



Şekil 4.8. Kamera altında alınan noktalar

Kamera altında işaretlenen her bir noktaya robot kol sıra ile gönderilerek konumları RT Toolbox3 programı içinde “2D Vision Calibration” sekmesi içinde bulunan bölüme girilir. Bu bölüm Şekil 4.9’da görülmektedir.



Şekil 4.9. Kamera robot kalibrasyonu aşaması

Kamerada X ve Y piksellerine denk gelen yerlerde robot kolun X ve Y mm değerleri kaydedilerek bir matris hesaplatılır. Bu matris kameradan gelen herhangi bir X ve Y konumu için robot kolun gideceği X ve Y konumlarını hesaplar ve robotun doğru

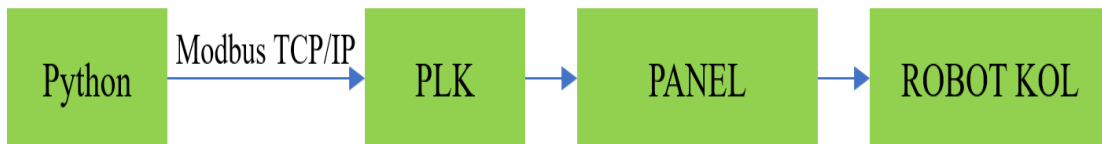
konuma gitmesini sağlar. Kalibrasyon işlemi kameradan alınan konum ile robotun gittiği konumun eşleştirilmesi olarak adlandırılabilir.

4.3. Robot ve Enkoder Kalibrasyonu

Kamera ve robot kalibrasyonu tamamlandıktan sonra enkoder ve robot kol kalibrasyonuna geçilir. Kalibrasyon kâğıdı robot kolun altına konular ve kalibrasyon kâğıdı üzerinde herhangi bir noktada enkoder değeri ve robot kol konumu kaydedilir. Daha sonra, kalibrasyon kâğıdı yine robotun erişebileceği bir mesafede olmak üzere konveyör hareket ettirilir. Kalibrasyon kâğıdı üzerinde aynı noktaya robot tekrar götürülür ve enkoder değeri ile robot kol konumu tekrar kaydedilir. İki farklı enkoder değeri arasındaki fark ile robot kolun her bir eksenindeki farklar hesaplanır ve enkoderin ürettiği 1 darbe değerinin her bir ekseninde kaç mm harekete denk geldiği hesaplanır. Bu çalışmada konveyör hareketi düz olduğu için robot kol sadece Y ekseninde hareket etmiş ve 1 darbe değeri 0.07mm olarak bulunmuştur.

4.4. Python-robot kol haberleşmesi

Çalışmada görüntü işlemeden elde edilen bilgilerin robot kola aktarılması aşamasındaki haberleşme ağı Şekil 4.10'da görülmektedir.



Şekil 4.10. Haberleşme akışı

Python'da elde edilen bilgi PLC'ye Modbus TCP/IP haberleşmesi üzerinden aktarılır. PLC'ye gelen bilgi 10 milisaniyede bir otomatik olarak panele ve oradan otomatik olarak robot kola aktarılır. Bunun için gerekli PLC'de okunacak ve robot kolda yazılacak verilerin ara veri aktarıcısı olarak kullanılan paneldeki arayüze yazılma biçimi Şekil 4.11'de görülmektedir.

Device Data Transfer

No.	Device Type	Points	Source Device	Destination Device	Comment	Offset
1	Signed BIN16	1	D4998	@2 U3E0-10019	görüntü işleme bitti	None
2	Signed BIN16	1	D4999	@2 U3E0-10020	çikolata sayısı	None
3	Signed BIN16	1	D5000	@2 U3E0-10021	çikolata X	None
4	Signed BIN16	1	D5001	@2 U3E0-10022	çikolata X	None
5	Signed BIN16	1	D5002	@2 U3E0-10023	çikolata X	None
6	Signed BIN16	1	D5003	@2 U3E0-10024	çikolata X	None
7	Signed BIN16	1	D5004	@2 U3E0-10025	çikolata X	None
8	Signed BIN16	1	D5100	@2 U3E0-10026	çikolata Y	None
9	Signed BIN16	1	D5101	@2 U3E0-10027	çikolata Y	None
10	Signed BIN16	1	D5102	@2 U3E0-10028	çikolata Y	None
11	Signed BIN16	1	D5103	@2 U3E0-10029	çikolata Y	None
12	Signed BIN16	1	D5104	@2 U3E0-10030	çikolata Y	None
13	Signed BIN16	1	@2 U3E1-10010	D1000		None
14	Signed BIN16	1	@2 U3E1-10011	D1001		None
15	Signed BIN16	1	D12	@2 U3E0-10010		None

Şekil 4.11. PLC'den robota panel üzerinden aktarılan veriler

4.5. Arayüz tasarımı

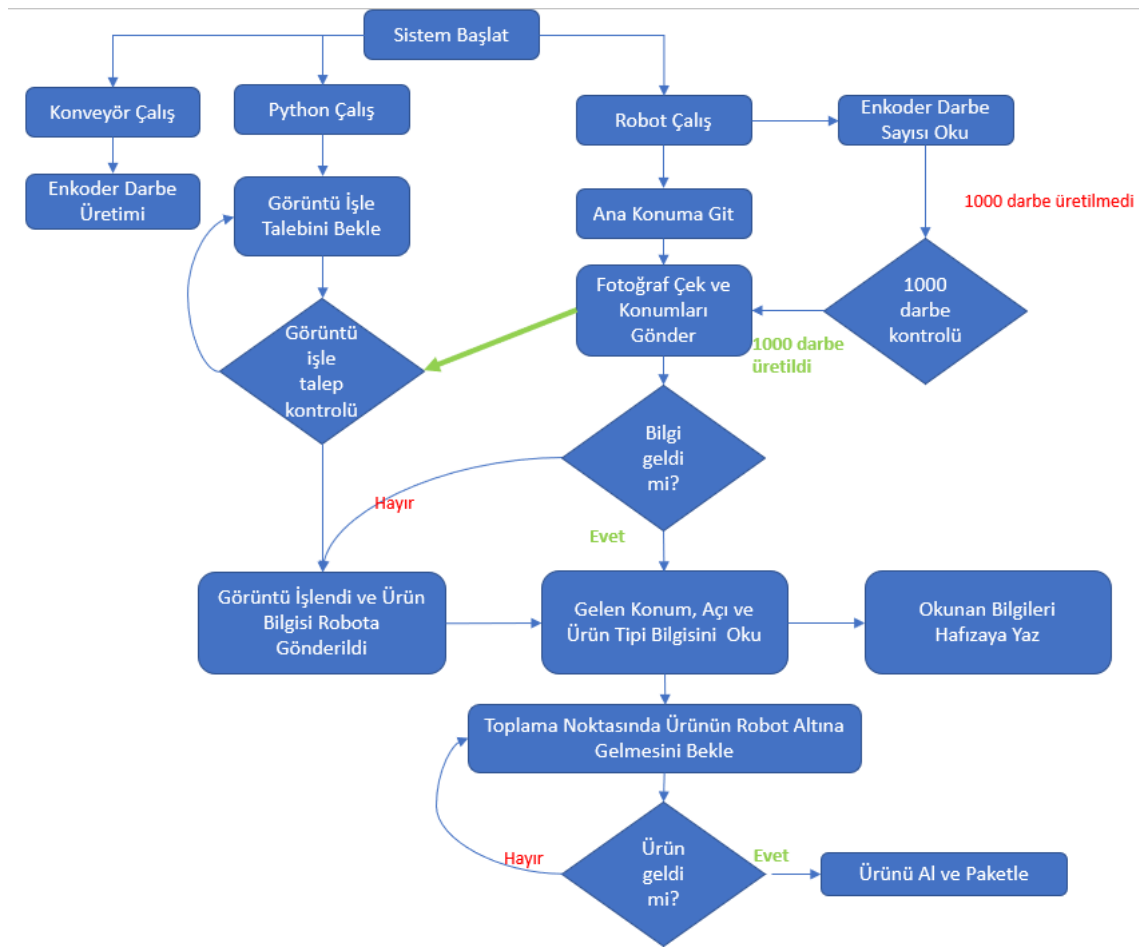
Sistemin çalışması ve durdurulması ile hız kontrolü sağlamak amacıyla bir arayüz tasarımı yapılmış ve sistem arayüz tasarımı üzerinden kontrol edilmiştir. Tasarlanan arayüz Şekil 4.12'de görülmektedir.



Şekil 4.12. Arayüz tasarımı

4.6. Deneysel çalışma

Kamera robot kol ve enkoder-robot kol kalibrasyonu için toplamda 50 farklı deney yapılmıştır. Yapılan deneyler ile kamera-robot kol ve enkoder-robot kol kalibrasyonu tamamlanmış olup görüntü işleme sonucunda gelen bilgiye göre ürün toplama aşamasına geçilmiştir. Bu aşamada tüm sistem robot kol tarafından yönetilmiştir. Sistemin akış şeması Şekil 4.13'te sunulmuştur.



Şekil 4.13. Sistemin akış şeması

Sistemde başlatıldığında konveyör harekete başlar, robot kol bekleme konumuna gider ve görüntü işleme programı çalışır. Bir ürünün kamera alanına girip kamera alanında çıkması 1000 darbe süresi kadar zaman almaktadır. Bundan dolayı aynı ürünü iki veya daha fazla kez saymamak için enkoder darbe sayısı robotta sayılmış ve 1000 darbeye 1 kez fotoğraf çekilerek her ürünün konumu robota sadece bir kez gönderilmiştir. Robot

1000 darbe saydığında PLC üzerinde bir register değerini değiştirmektedir. Bu değer Python tarafından Modbus TCP/IP haberleşmesi üzerinden sürekli okunmakta olup değer değiştiği an fotoğraf çekilmekte ve bilgiler PLC üzerinden panele, oradan da robota aktarılmaktadır. Bilgi aktarımı yapılır yapılmaz register değeri sıfırlanmakta, Python görüntü işleme için bir sonraki register değer değişimini yani robotun fotoğraf çek komutunu beklemektedir. Robot kol bilgileri hafızaya aldıktan sonra ürün toplama noktasına gitmekte ve ürünlerin toplama alanına girmesini beklemektedir. Burada her ürünün fotoğraf çekildiği an hangi enkoder konumunda olduğunu bilmektedir. Ürünlerin toplama alanında nerede bulunduğunu enkoder değeri, X ve Y konumları üzerinden takip edip toplamaya ve bırakma noktasında ürün tipine göre ayırarak bırakmaya başlamaktadır. Robot toplama işlemi yaparken konveyör akmaya devam ettiği için paralel programda fotoğraf çekme ve alınan ürün bilgilerinin hafızaya yazılma işlemi devam etmektedir. Robot hafızasındaki değerleri sırası ile okuyarak her bir ürünü toplamaktadır. Toplama işlemi Şekil 4.14'te görülmektedir.



Şekil 4.14. Farklı çikolataları toplama işlemi

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Endüstride işçilerin hijyenle ilgili sorunları olabilirken, robotlar steril ve hijyenik koşullarda çalışabilirler. Bu nedenle, gıda işleme, ilaç üretimi ve diğer endüstrilerde robotların kullanımı, ürünlerin daha hijyenik bir şekilde üretilmesine yardımcı olabilir. Robotlar, programlanabilir olduklarından ve belirli bir süre içinde aynı görevi tekrarlayabildiklerinden, işletmelerde iş süreçlerini hızlandırabilirler. İşçilerin uzun sürede yaptıkları bir görevi, robotlar daha kısa bir sürede bitirebilirler. Robotlar, programlanabilir olduklarından, belirli bir kalite standardına uygun olarak çalışabilirler. Bu, ürün kalitesinde tutarlılığı artırır ve hatalı üretim miktarını azaltır. Ayrıca, işçilerin fiziksel sınırlamaları nedeniyle bazı işleri tamamlamaları zor olabilirken, robotlar bu görevleri hatasız ve tutarlı bir şekilde tamamlayabilirler.

Çalışmada robot ve konveyör hızı farklı şekillerde değiştirilerek optimum ürün toplama hızları belirlenmeye çalışılmıştır. Yapılan denemelerden elde edilen ürün toplama ve kaçırma oranları Çizelge 5.1’de sunulmuştur.

Çizelge 5.1. Robot ve konveyör hızına bağlı olarak sistem performansı

Robot hızı (Hız yüzdesi)	Konveyör hızı (Sürücü Frekansı)	Gelen Çikolata Sayısı (dakikada)	Toplanan Çikolata Sayısı	Toplanamayan Çikolata Sayısı	Hata Oranı (%)
80	10	300	281	19	6,3
90	10	300	283	17	5,6
100	10	300	282	18	6
80	15	300	287	13	4,3
90	15	300	289	11	3,6
100	15	300	290	10	3
80	20	300	275	25	8,3
90	20	300	276	24	8
100	20	300	282	18	6
80	25	300	263	37	12,3
90	25	300	266	34	11,3
100	25	300	270	30	10

Yapılan deneylerde robot hızı yüksek, konveyör hızı düşük tutulduğunda robotun %5 üzerinde hata ile çalıştığı gözlemlenmiştir. Üretim hızı gereği robotlu toplama hattı öncesi hat hızı 15Hz olduğu için kurulan sistemde toplama hattı hızı 15Hz olarak belirlenmiştir. Burada toplama hattı hızı daha düşük tutulduğunda çikolatalar üst

üste binmekte ve robotun kaçırdığı çikolataların arttığı görülmüştür. Aynı şekilde, toplama hattı hızı artırıldığında ise bu sefer robotların yeterli sayıda çikolata toplayamadığı gözlemlenmiştir. Tüm bu denemeler sonucunda hat hızının 15 Hz, robot hızının %100 olduğu parametreler kullanılmıştır.

Manuel sistemde işleme ürüne el değmekte olup ürünün hijyen ve kalitesinin bozulduğu görülmüştür. El değmeden üretim ve paketleme işlemi için insan faktörünün ortadan kaldırılması ile daha temiz bir üretim ve paketleme işlemi gerçekleştirilebilmektedir. Ayrıca, üretimde insan faktörü buldukça üretim hızının değiştiği gözlemlenmiştir. Çünkü işçinin beklenmedik zamanlarda izin alması, hasta olması, çalışma isteksizliği veya yavaş çalışma, tekrarlayan işleri yaparken hatalı çalışma gibi durumları bulunmaktadır.

Tez çalışmasında robotik bir sistem önerilmiştir. Çalışmada görüntü işleme ile tespit edilen ürünler robot kol ile toplanmış ve paketlenmiştir. Ürün olarak renk ve şekil bakımından farklı üç tür çikolata kullanılmıştır. Kullanılan çikolatalar Python yazılım dili ile OpenCV kütüphanesi kullanılarak tespit edilmiş ve Modbus TCP/IP haberleşmesi ile önce PLC'ye gönderilmiştir. PLC üzerinde toplanan veriler panel kullanılarak robot kola aktarılmıştır. Ürün toplama aşamasında konveyör durdurulmamış, robot kol konveyör üzerindeki ürünleri takip ederek sistemden almış ve ayırmıştır. Bu sayede paketleme işlemi daha temiz ve daha hızlı hâle getirilmiştir.

Tez çalışmasında kurulan sistemde görüntü işleme sayesinde hattan hangi ürünün geldiği otomatik tespit edilmiş ve paketleme ayarları ürüne göre güncellenmiştir. Bu amaçla kullanılan iki farklı yöntemin başarısı tespit edilmiş ve en iyi sonucu veren yöntem kullanılmıştır. Elde edilen sonuçlar Çizelge 5.2'de sunulmuştur.

Çizelge 5.2. Farklı deneyler sonucunda elde edilen ürün tespit başarısı

Yöntem		Parametre	Değer	Ürün Sayısı	Tespit Edilen Ürün Sayısı	Doğruluk (%)
Yöntem 1 - HSV + Eşikleme + Kenar Bulma	Deneme 1	Alt-Üst Sınırlar	Çizelge 4.1'de sunulmuştur	100	100	100
Yöntem 2 - Kapalı Alan Tespit Algoritması	Deneme 1	Minimum Eşik	20	100	92	92
		Maksimum Eşik	4000			
Minimum Alan		100				
Minimum Dairesellik		0.1				
	Deneme 2	Minimum Dışbükeylik	0.1			
		Minimum Eşik	100	100	5	5
		Maksimum Eşik	4000			

	Minimum Alan	100			
	Minimum Dairesellik	0.5			
	Minimum Dışbükeylik	0.1			
Deneme 3	Minimum Eşik	20			
	Maksimum Eşik	4000			
	Minimum Alan	300	100	62	62
	Minimum Dairesellik	0.1			
	Minimum Dışbükeylik	0.1			
Deneme 4	Minimum Eşik	20			
	Maksimum Eşik	4000			
	Minimum Alan	300	100	53	53
	Minimum Dairesellik	0.1			
	Minimum Dışbükeylik	0.01			
Deneme 5	Minimum Eşik	20			
	Maksimum Eşik	4000			
	Minimum Alan	500	100	32	32
	Minimum Dairesellik	0.2			
	Minimum Dışbükeylik	0.01			
Deneme 6	Minimum Eşik	20			
	Maksimum Eşik	4000			
	Minimum Alan	500	100	35	35
	Minimum Dairesellik	0.3			
	Minimum Dışbükeylik	0.1			
Deneme 7	Minimum Eşik	20			
	Maksimum Eşik	4000			
	Minimum Alan	200	100	44	44
	Minimum Dairesellik	0.1			
	Minimum Dışbükeylik	0.1			
Deneme 8	Minimum Eşik	20			
	Maksimum Eşik	4000			
	Minimum Alan	200	100	12	12
	Minimum Dairesellik	0.1			
	Minimum Dışbükeylik	0.7			
Deneme 9	Minimum Eşik	20			
	Maksimum Eşik	4000			
	Minimum Alan	75	100	57	57
	Minimum Dairesellik	0.2			
	Minimum Dışbükeylik	0.1			
Deneme 10	Minimum Eşik	20			
	Maksimum Eşik	4000			
	Minimum Alan	75	100	2	2
	Minimum Dairesellik	0.2			
	Minimum Dışbükeylik	0.8			

Çizelge 5.2'den görüleceği üzere en yüksek doğrulukla çikolata tespit eden yöntem, %100 doğruluk veren Yöntem 1 olarak tespit edilmiştir. Yöntem 2'de 10 farklı deney yapılmış ve 100 çikolata içerisinde en yüksek doğruluk 92 olarak elde edilmiştir. Yöntem 2'de yine aynı şekilde birçok deneme yapılabilir. Yapılan 10 deneyde 100 çikolata içerisinde doğruluk oranı farklı deneylerde %2-92 arasında değişmiştir. Yöntem 1'de ise bütün çikolatalar %100 doğruluk ile tespit edilebilmiştir. Yöntem 2'de minimum dairesellik veya minimum dışbükeylik parametrelerinden bir tanesi 0.5 veya daha yukarıda bir değer olduğunda tespit doğruluğu dramatik şekilde düşmekte ve %10'un altında kalmaktadır. Minimum eşik değeri düşük, maksimum eşik değeri yüksek tutulduğunda doğruluk artmaktadır. Ayrıca minimum alan değeri de düşük tutulduğunda tespit başarısı yükselmektedir. Bundan dolayı en iyi doğruluk Deney 1 ile elde edilmiştir.

Çalışmada, tespit edilen ürünün boyut kontrolü yapılarak görüntü içerisinde ürünün sahip olması gereken alandan %10 daha fazla veya daha az alana sahip olanlar paketleme işlemine tabi tutulmamıştır. Bu durum kırık çikolata olduğunu göstermektedir. Bir çikolata kırıldığında kapladığı alan değişir. %10'luk bir alan farkı insan gözü ile tespit edilemez ancak kamera ve görüntü işleme yazılımları ile küçük kırığa sahip çikolata tespiti insan gözüne göre daha yüksek doğrulukla tespit edilmektedir. İnsan gözü ile ayırt edilemeyecek olan bu kalite kontrol işlemi de doğrudan üretim esnasında gerçekleştirilerek hatasız ve hızlı bir üretim yapılması sağlanmıştır.

5.2 Öneriler

Endüstriyel sistemlerde ürün tespiti için maliyeti yüksek endüstriyel kameralar kullanılmaktadır. Tez çalışmasında bu işlem bir webcam ve bilgisayar kullanılarak gerçekleştirilmiştir. Endüstriyel kameralarda kameraya bir ürün tanıtımı işlemi gerçekleştirilmekte ve kamera bir seferde sadece tek bir ürün tanıyabilmektedir. Hattan farklı ürünler geldiğinde kullanıcı arayüzünden kameraya hangi ürünün geldiği bildirilmekte ve kamera programı değişmektedir. Kamera programını değiştirme işlemi operatöre bağlı olduğu için hataya açık bir konudur. Tez çalışmasında ise açık kaynak kütüphane olan OpenCV kütüphanesi sayesinde aynı anda her bir ürün kontrol edilerek hangi ürünün geldiği otomatik tespit edilmiştir. Bu sayede operatörün sisteme müdahale ettiği noktalardan bir tanesi daha ortadan kaldırılarak sistem bir adım daha

geliştirilmiştir. Ayrıca, OpenCV kütüphanesi ile tek seferde daha fazla ürün tespiti de mümkündür. Ancak, her bir ürünü tespit eden özel parametre ayarı yazılması gerekmektedir. Bu durumun önüne geçmek için yapay zekâ algoritmaları kullanılabilir. Sisteme ilk başta 10 farklı çikolata tanımlandı ve yapay zekâ 10 farklı çikolata ile eğitildi ise sisteme verilen 11. farklı çikolatayı yapay zekânın sorunsuz tespit edebileceği düşünülmektedir. Yapay zekalı çikolata tespit sisteminin geliştirilmesi ile bir çikolata veri seti oluşturulup birçok farklı fabrikada çalıştırılabilecek tek bir sistem kurulabilir.



KAYNAKLAR

- Alkan, O. F., & OY, S. A. (2021) Standart santrifüj pompaların şebekeden bağımsız jeneratör olarak kullanımı ve performans incelemesi. *Politeknik Dergisi*, 1(1), 1-14.
- Akhan, H. (2022) Sanayide enerji yönetimi: Pompa ve fan sistemlerinde verimlilik artırıcı uygulamalar. *Trakya Üniversitesi Mühendislik Bilimleri Dergisi*, 23(1), 11-23.
- Barstugan, M., Ceran, Y. S., Yılmaz, M., & Dundar, N. A. (2020). Detection of Defects on Single-Bead Welding by Machine Learning Methods. *In IOP Conference Series: Materials Science and Engineering* 895(1), 1-5.
- Burak, G. Ü. L. & Toptaş, E. (2021). Tasarım Süreçlerinde Otomasyon Sistemlerinin Optimizasyonu ve Etkilerinin İncelenmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (28), 1534-1539.
- Bingöl, O., Aydoğan, T., Didin, H. R., Yalçiner, A. S., & DUYGULU, K. (2010). Plc Kontrollü Otomatik Katli Otopark Sistemi. *Uluslararası Teknolojik Bilimler Dergisi*, 2(1), 65-76.
- Chang, L., Cheng, L., Huang, C., Qin, S., Fu, C., & Li, S. (2022). Extracting urban water bodies from Landsat imagery based on mNDWI and HSV transformation. *Remote Sensing*, 14(22), 5785.
- Çolak, İ., Bayındır, R., & Kuruşçu, S. (2007). PLC KONTROLLÜ ASANSÖR EĞİTİM SETİ TASARIMI VE UYGULAMASI. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, 23(1), 86-94.
- Elbehiery, H., Hefnawy, A., & Elewa, M. (2005). Surface defects detection for ceramic tiles using image processing and morphological techniques, *World Academy of Science, Engineering and Technology Conference*, 158-162.
- Han, S. D., Feng, S. W., & Yu, J. (2019). Toward fast and optimal robotic pick-and-place on a moving conveyor. *IEEE Robotics and Automation Letters*, 5(2), 446-453.
- İnan, T. (2013). Hareketli hedefi takip eden robot kolu sistemi. *Doktora Tezi*, Marmara Üniversitesi.
- Kale, V. R., & Kulkarni, V. A. (2013). Object sorting system using robotic arm. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(7), 3400-3407.
- Kılıç, M. K. (2018), Dijital görüntü işleme tekniği ile patlatma verimlilik analizi, Yüksek Lisans Tezi, Okan Üniversitesi.
- Modbus TCP/IP, 2023, <https://www.hubbox.io/tr/blog/veri-toplama/modbus-protokolu-ve-tum-ozellikleri-nedir>.

- Mnyusiwalla, H., Triantafyllou, P., Sotiropoulos, P., Roa, M. A., Friedl, W., Sundaram, A. M., ... & Deacon, G. (2020). A bin-picking benchmark for systematic evaluation of robotic pick-and-place systems. *IEEE Robotics and Automation Letters*, 5(2), 1389-1396.
- Nguyen, T. B., & Chung, S. T. (2009, October). An improved real-time blob detection for visual surveillance. In 2009 2nd International Congress on Image and Signal Processing (pp. 1-5). IEEE.
- Ozdemir, R., & Koc, M. (2019). A quality control application on a smart factory prototype using deep learning methods. In *2019 IEEE 14th international conference on computer sciences and information technologies (CSIT)*, 46-49.
- Parlak, M. E. (2018). Matlab yardımıyla konveyör bantlarda kameralı ürün sayımı ve görüntü işleme, Yüksek Lisan Tezi, İstanbul Teknik Üniversitesi.
- Premachandra, H. A. G. C., Herath, H. M. A. N., Suriyage, M. P., Thathsarana, K. M., Amarasinghe, Y. W. R., Gopura, R. A. R. C., & Nanayakkara, S. A. (2020). Genetic Algorithm Based Pick and Place Sequence Optimization for a Color and Size Sorting Delta Robot. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, 209-213.
- Pu, M., Huang, Y., Liu, Y., Guan, Q., & Ling, H. (2022). Edter: Edge detection with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1402-1412.
- Savran, A. İ., & Kumbasar, T. (2018). A Vision Based Positioning Gas Leakage Test Automation System. In *IEEE 2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, 1-6.
- Shrouf, F., Ordieres, J., & Miragliotta, G. (2014). "Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm." In *Industrial Engineering and Engineering Management (IEEM)*, 2014 IEEE International Conference on (pp. 697-701).
- Daha fazla detaylı bilgi için bu referansları araştırabilirsiniz. Eğer spesifik sorularınız veya daha fazla açıklama gereksiniminiz varsa, lütfen bana bildirin!
- Tarı, M. (2020). Görüntü işleme teknikleri ile enjektör üretiminde kalite kontrolü, Lisansüstü Eğitim Enstitüsü, Konya Teknik Üniversitesi.
- Uslu, O. (2022). *Endüstri 1.0'dan endüstri 4.0'a toplumsal yapının dönüşümü: Schumpeterci yaklaşım* (Master's thesis, Başkent Üniversitesi Sosyal Bilimler Enstitüsü).
- Yang, Y., Yang, J., Zeng, H., & Li, J. (2021). A 3D Vision-Based Conveyor Tracking System for Pick-and-Sort Robotic Applications. In *2021 6th International Conference on Robotics and Automation Engineering (ICRAE)*, 231-236.

Yıldız, İ., Kaya, A., Gedik, M. A., & Barstuğan, M. (2021). Separation of Defected Products from Production Line with a Robotic Arm via Image Processing Methods. *In CEUR Workshop Proceedings*, 1-6.

Zheng, Z., Ma, Y., Zheng, H., Gu, Y., & Lin, M. (2018). Industrial part localization and grasping using a robotic arm guided by 2D monocular vision. *Industrial Robot: An International Journal*, 45(6), 794-804.



EKLER

EK-1 Görüntü işleme ile çikolata tespiti kodu

```

import cv2
import numpy as np
from pyModbusTCP.client import ModbusClient
from time import sleep
import time
from math import atan2, cos, sin, sqrt, pi
from beyaz_cikolata_bul import beyaz_cikolata
from koyu_cikolata_bul import siyah_cikolata
from yildiz_cikolata_bul import yildiz_cikolata

server_host = "192.168.3.250"
server_port = 502
plc = ModbusClient()
plc.close()
sleep(1)

cikolata = cv2.VideoCapture(0)

while True:
    try:
        plc.host(server_host)
        plc.port(server_port)
        plc.open()
    except:
        print("PLC bağlantısı kurulamadı")

    try:
        if not plc.is_open():
            if not plc.open():
                print("PLC'ye bağlanamadı")
    except:
        print("PLC'ye bağlanamadı")

    while plc.is_open():
        ret,frame = cikolata.read()
        frame = frame[:,50:640]
        orj = frame

        islem_bitti = 0
        plc.write_multiple_registers(4998, [islem_bitti])
        goruntu_isle= plc.read_holding_registers(220,1)# D220 değeri 50 olunca görüntü işleme
yapılacak
        d221= plc.read_holding_registers(221,1)# D220 değeri 50 olunca görüntü işleme yapılacak

        if goruntu_isle[0]==50:
            if d221[0] == 0:
                print("görüntü işleme başladı")
                cikolata_x = []
                cikolata_y = []
                cikolata_acilari = []
                cikolata_tipi = []
                yeni_x = [0,0,0,0,0]
                yeni_y = [0,0,0,0,0]

```

```

yeni_aci = [0,0,0,0,0]
yeni_tip = [0,0,0,0,0]

```

```

islem_bitti,beyaz_cikolata_sayisi,beyaz_cikolata_x,beyaz_cikolata_y,beyaz_cikolata_acilari=
beyaz_cikolata(frame)

```

```

    if beyaz_cikolata_sayisi>0:
        for i in range(0,beyaz_cikolata_sayisi):
            cikolata_tipi.append(0)
            cikolata_x.append(beyaz_cikolata_x[i])
            cikolata_y.append(beyaz_cikolata_y[i])
            cikolata_acilari.append(beyaz_cikolata_acilari[i])

```

```

islem_bitti,siyah_cikolata_sayisi,siyah_cikolata_x,siyah_cikolata_y,siyah_cikolata_acilari=
siyah_cikolata(frame)

```

```

    if siyah_cikolata_sayisi>0:
        for i in range(0,siyah_cikolata_sayisi):
            cikolata_tipi.append(1)
            cikolata_x.append(siyah_cikolata_x[i])
            cikolata_y.append(siyah_cikolata_y[i])
            cikolata_acilari.append(siyah_cikolata_acilari[i])

```

```

islem_bitti,yildiz_cikolata_sayisi,yildiz_cikolata_x,yildiz_cikolata_y,yildiz_cikolata_acilari=
yildiz_cikolata(frame)

```

```

    if yildiz_cikolata_sayisi>0:
        for i in range(0,yildiz_cikolata_sayisi):
            cikolata_tipi.append(2)
            cikolata_x.append(yildiz_cikolata_x[i])
            cikolata_y.append(yildiz_cikolata_y[i])
            cikolata_acilari.append(yildiz_cikolata_acilari[i])

```

```

yildiz_cikolata_sayisi = cikolata_sayisi - beyaz_cikolata_sayisi - siyah_cikolata_sayisi

```

```

sayac = cikolata_sayisi
if cikolata_sayisi>5:
    sayac = 5
if cikolata_sayisi==0:
    sayac = 0

```

```

plc.write_multiple_registers(4999, [sayac])

```

```

if cikolata_sayisi == 0:
    for i in range(0,5):
        plc.write_multiple_registers(5000+i, [0])
    for i in range(0,5):
        plc.write_multiple_registers(5100+i, [0])
    for i in range(0,5):
        plc.write_multiple_registers(5200+i, [0])
    for i in range(0,5):
        plc.write_multiple_registers(5300+i, [0])

```

```

else:
    for i in range(sayac):
        plc.write_multiple_registers(5000+i, [cikolata_y[i]])
    for i in range(sayac):

```

```
        plc.write_multiple_registers(5100+i, [cikolata_x[i]])
    for i in range(sayac):
        plc.write_multiple_registers(5200+i, [cikolata_acilari[i]])
    for i in range(sayac):
        plc.write_multiple_registers(5300+i, [cikolata_tipi[i]])

    plc.write_multiple_registers(221, [10]) # PLC içindeki foto çek komutu iptal edilerek
    bir kere işlem yapılıyor
    print("görüntü işleme bitti")

    print("toplam cikolata:",sayac)

    cv2.imshow("esik ",orj)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        cikolata.release()
        cv2.destroyAllWindows()
        plc.close()
        break

cikolata.release()
cv2.destroyAllWindows()
plc.close()
```

EK-2 Beyaz çikolata bulma kodu

```

import cv2
import numpy as np
kernal = np.ones((3,3),"uint8")

def beyaz_cikolata(frame):

    cikolata_x = []
    cikolata_y = []
    cikolata_tipi = []
    cikolata_acilari = []
    orj = frame
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    LowerRegion = np.array([0,0,0],np.uint8)
    upperRegion = np.array([255,50,255],np.uint8)

    cikolata = cv2.inRange(hsv,LowerRegion,upperRegion)
    red = cv2.morphologyEx(cikolata,cv2.MORPH_OPEN,kernal)
    red = cv2.erode(red,kernal,iterations=1)

    res1=cv2.bitwise_and(orj, orj, mask = red)
    res1 = cv2.cvtColor(res1,cv2.COLOR_BGR2GRAY)
    res1[res1>0]=255

    # edged = cv2.Canny(res1, 30, 30)
    # cv2.imshow("esik 2",edged)
    contours, hierarchy
cv2.findContours(res1,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE) =
    try: hierarchy = hierarchy[0]
    except: hierarchy = []

    height, width= res1.shape
    min_x, min_y = width, height
    max_x = max_y = 0
    indeks = 0
    # computes the bounding box for the contour, and draws it on the frame,
    for contour, hier in zip(contours, hierarchy):
        (x,y,w,h) = cv2.boundingRect(contour)
        min_x, max_x = min(x, min_x), max(x+w, max_x)
        min_y, max_y = min(y, min_y), max(y+h, max_y)
        if w <40 and h < 40:
            if w > 20 and h>20:
                indeks = indeks + 1
                # cv2.rectangle(orj, (x,y), (x+w,y+h), (255, 0, 0), 2)
                robot_x = int(np.round(x + w/2))
                robot_y = int(np.round(y+h/2))

                cikolata_x.append(robot_x)
                cikolata_y.append(robot_y)
                ((centx,centy), (width,height), angle) = cv2.fitEllipse(contour)
                if angle>90:
                    angle = angle - 180
                angle = int(angle)
                cikolata_acilari.append(angle)

    cikolata_sayisi = indeks

    islem_bitti = 1
return islem_bitti, cikolata_sayisi,cikolata_x,cikolata_y,cikolata_acilari

```

EK-3 Siyah çikolata bulma kodu

```

import cv2
import numpy as np
kernal = np.ones((3,3),"uint8")
def siyah_cikolata(frame):
    cikolata_x = []
    cikolata_y = []
    cikolata_tipi = []
    cikolata_acilari = []
    orj = frame
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    LowerRegion = np.array([0,0,0],np.uint8)
    upperRegion = np.array([255,255,160],np.uint8)
    cikolata = cv2.inRange(hsv,LowerRegion,upperRegion)
    red = cv2.morphologyEx(cikolata,cv2.MORPH_OPEN,kernal)
    red = cv2.erode(red,kernal,iterations=1)

    res1=cv2.bitwise_and(orj, orj, mask = red)
    res1 = cv2.cvtColor(res1,cv2.COLOR_BGR2GRAY)
    res1[res1>0]=255

    # edged = cv2.Canny(res1, 30, 30)
    # cv2.imshow("esik 2",edged)
    contours, hierarchy
cv2.findContours(res1,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)

    try: hierarchy = hierarchy[0]
    except: hierarchy = []

    height, width= res1.shape
    min_x, min_y = width, height
    max_x = max_y = 0
    indeks = 0
    # computes the bounding box for the contour, and draws it on the frame,
    for contour, hier in zip(contours, hierarchy):
        (x,y,w,h) = cv2.boundingRect(contour)
        min_x, max_x = min(x, min_x), max(x+w, max_x)
        min_y, max_y = min(y, min_y), max(y+h, max_y)

    if w <100 and h < 100:
        if w > 20 and h>20:
            indeks = indeks + 1
            # cv2.rectangle(orj, (x,y), (x+w,y+h), (255, 0, 0), 2)
            robot_x = int(np.round(x + w/2))
            robot_y = int(np.round(y+h/2))

            cikolata_x.append(robot_x)
            cikolata_y.append(robot_y)
            ((centx,centy), (width,height), angle) = cv2.fitEllipse(contour)
            if angle>90:
                angle = angle - 180

            angle = int(angle)
            angle = angle + 360
            cikolata_acilari.append(angle)
    cikolata_sayisi = indeks

    islem_bitti = 1
return islem_bitti,cikolata_sayisi,cikolata_x,cikolata_y,cikolata_acilari

```

EK-4 Yıldız çikolata bulma kodu

```

import cv2
import numpy as np
kernal = np.ones((3,3),"uint8")
def yildiz_cikolata(frame):
    cikolata_x = []
    cikolata_y = []
    cikolata_tipi = []
    cikolata_acilari = []
    orj = frame
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    LowerRegion = np.array([0,0,173],np.uint8)
    upperRegion = np.array([255,127,255],np.uint8)
    cikolata = cv2.inRange(hsv,LowerRegion,upperRegion)
    red = cv2.morphologyEx(cikolata,cv2.MORPH_OPEN,kernal)
    red = cv2.erode(red,kernal,iterations=1)

    res1=cv2.bitwise_and(orj, orj, mask = red)
    res1 = cv2.cvtColor(res1,cv2.COLOR_BGR2GRAY)
    res1[res1>0]=255

    # edged = cv2.Canny(res1, 30, 30)
    # cv2.imshow("esik 2",edged)
    contours, hierarchy
cv2.findContours(res1,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)

    indeks = 0
    # computes the bounding box for the contour, and draws it on the frame,
    for contour in contours:
        (x,y,w,h) = cv2.boundingRect(contour)

        # print(w,h)
        if w <100 and h < 100:
            if w > 35 and h>35:

                indeks = indeks + 1
                # cv2.rectangle(orj, (x,y), (x+w,y+h), (255, 0, 0), 2)
                robot_x = int(np.round(x + w/2))
                robot_y = int(np.round(y+h/2))

                cikolata_x.append(robot_x)
                cikolata_y.append(robot_y)
                ((centx,centy), (width,height), angle) = cv2.fitEllipse(contour)
                if angle>90:
                    angle = angle - 180
                angle = int(angle)
                cikolata_acilari.append(angle)
    cikolata_sayisi = indeks

    islem_bitti = 1
return islem_bitti,cikolata_sayisi,cikolata_x,cikolata_y,cikolata_acilari

```

EK-5 Robot takip kodu

```

If M_Run(2)=0 Then                                'Confirmation of conveyer 1 multitasking
  XRun 2,"CM1MY",1                                'Multitasking setting
  Wait M_Run(2)=1                                  'Wait the task start
EndIf
MGenOvr=100 'Genel hız
Loadset 0,0                                        'Tutucu yük ölçü datası
MUrNo = 1    'Ürün çeşidi numarası
M_Tool=0
'Başlangıç Güvenli Home'a gidiş
'Tracking Parametre Atamaları
  Accel 100,100
  Ovr MGenOvr                                     'Genel hız
  OAdl On                                         'Optimum ivmelenme aktif
  Clr 1                                           'Ne kadar IO lokal global değişken varsa sıfırlanır
  MvTune 2                                        'Yüksek hızlı pozisyonlama modu açılır
  TrClr 1                                         'Buffer içindeki datalar silinir
GoSub *HOME   'Güvenli home'a dönüş
GoSub *PARAMETRE
If MRejWait=0 Then MRejWait=100000&'1000&      '0 ise 10 sn alınır
Def Act 2, M_Timer(7)>MRejWait GoTo *Reject      'Aksiyon tanımlanır
'Alma öncesi bekleme poz. yükseklik değeri alınır
PSave=PSaveMas    'referans psave datasını psave içine atıyor
PWait=PWaitMas    'referans bekleme pozisyonunu pwait pozisyonuna atıyor
*URUNBEKLE
  M_Timer(7)=0    'Ürün bekleme süresi timerı
  Act 2=1        'Ürün bekleme süresini aktif ediyor, süre dolarsa reject'e gidiyor
*LBFCHK
PSave1=PSave
PWait1=PWait
TrkChk 1, PSave, PWait, *LTRGO    'ürün yoksa psave1'de durur, ürün alana girince pwait1'de
durur, robotun çalışma alanına girince *LTRST'ye atlar
  If M_TrkChk(1) <= 1 Then GoTo *LBFCHK    '0 veya 1 ise sonraki bırakma labelına atla, 2 veya 3
ise ürün alabilirsin alta geç
  TrkWait *LBFCHK    'Ürünün alana girmesini bekle süre dolunca yeniden bırakmaya
atla
*LTRGO
  M_Tool=0
  Ovr 100
  Act 2=0    'Ürün bekleme reject aksiyonunu resetler
  Cnt 1,0,0    'Hassas hızlı pozisyonlama
  HClose 1
  GoSub *PARAMETRE
  TrkMv On, PGetUp, 1, *S91STOP    'Tracking yukarı noktasına gider, zorla durma mesafesi
aşılırsa S91STOP'a atlar
  ""HClose 1
  ""Dly 0.1
  Mvs PGet 'M_Out(MVacOffSig)=0 'Tutuş pozisyonu
  HOpen 1
  Dly 0.1
  Mov PGetUp
  TrkMv Off 'tracking sonlandırılır
  Cnt 1,50,50
  Act 2=1
  M_Tool=0
  Ovr MGenOvr
  Mov PBIRAK, -20
  Mvs PBIRAK

```

```

HClose 1
Dly 0.1
Mvs PBIRAK, -20
GoTo *URUNBEKLE 'Toollar bitmişse yeni ürün almaya gider
'PARAMETRE YUKLEMESİ BAŞLANGIÇTA
*PARAMETRE
'Hareket koşullarının belirlenmesi
P_TrkPAcl(1) = (+0.10,+0.10,+0.10,+1.00,+1.00,+1.00,+1.00,+1.00) 'tracking acceleration
P_TrkPDcl(1) = (+0.10,+0.10,+0.10,+1.00,+1.00,+1.00,+1.00,+1.00) 'tracking deceleration
M_TrkBuf(1) = 1 'Buffer Numarası
M_TrkStart(1) = PRng.X 'Robotun takibe başlayacağı noktaya uzaklığı
M_TrkEnd(1) = PRng.Y 'Robotun takibi bırakacağı noktaya uzaklığı
M_TrkStop(1) = PRng.Z 'Robotun takibi iptal edeceği noktaya uzaklığı
M_TrkTime(1) = 360 'Zaman aşımı süresi
PTBASE = P_100(MUrunNo) 'Tanıtılmış referans tutuş pozisyonu
P_TrkBase(1) = PTBASE 'Referans tutuş pozisyonu bu fonksiyon içine atılır
PGet = PTBASE * POffset 'Ürün tutuş pozisyonuna referans nokta ve elle girilen ofset
datalarını içeren poffset pozisyonu eklenir
PGetUp = PGet * PUp1 'Ofsetli pget tutuş pozisyonuna elle z ofseti girilmiş ofset
pozisyonu eklenerek yaklaşma pozisyonuna atanır
'CM1 Çalıştırılması
M_09#=MUrunNo 'Model number specification
Return
'TRACKING STOP
*S91STOP 'Zorla durma noktasına gelirse atlanacak label
TrkMv Off 'tracking işlemini iptal eder
HOpen 1
P91P=P_Fbc(1) 'Anlık pozisyon datası alınır
P91P.Z=PHome.Z 'z datası güvenli yüksekliğe alınır
Mvs P91P Type 0,0 'Kaldırılır
Mov PHome
Act 1 = -1 'Act 1 diye tanımlı bir aksiyon yok
GoTo *LBFCHK 'Bırakma labelına geri döner
Return
'GÜVENLİ HOME YAPMA
*HOME
Ovrd 100 'Yavaş hız olmalı
Accel 20,20
Servo On
P90CURR=P_Fbc
If P90CURR.Z<PHome.Z Then 'anlık yükseklik psave yüksekliğinden küçükse
Ovrd 100
P90ESC=P90CURR 'Onunla aynı yüksekliğe çıkar
P90ESC.Z=PHome.Z
Mov P90ESC
Ovrd MGenOvrd
EndIf
Mov PHome 've safe pozisyona döner
Return
*Reject
Ovrd MGenOvrd
Act 2=0
GoSub *HOME
GoTo *URUNBEKLE
''''''''
End

```

EK-6 Kameradan veri okuma

```

Dim MX(4),MY(4),MT(4),PVS(4),MTR#(8)      'X/Y/C/buffer/encoder data
MTrigSens=10128 'Kameraya çalış diyen ürün sensörü
M01=0
TOPLAMGIREN=0
BUFFER=0
ANLIK=0
'
##### Main processing #####
*S00MAIN
  If M_Run(8)=0 Then          'Confirmation of conveyer 1 multitasking
    XRun 8,"ENCTAKIPA0B",1    'Multitasking setting
    Wait M_Run(8)=1          'Wait the task start
  EndIf
GoSub *DATAAL1               'Kalibrasyon programında toplanan veriler alınıyor
GoSub *DATAAL2               'VS initialization processing
GoSub *DATAAL3               'Condition setting
*L00_00
GoSub *DATABEKLE             'VS recognition check processing
GoTo *L00_00
End
'
##### Data acquisition processing ##### C1 programındaki datalar alınır
*DATAAL1
MWrkNo=1                     'Workpiece number
MEncNo=P_102(MWrkNo).X       'Encoder number
MSkipNo = P_102(MWrkNo).Y    'SKIP input number
MVsLen=P_103(MWrkNo).X       'VS screen size longitudinal distance
MWrkLen=P_103(MWrkNo).Y      'Workpiece size longitudinal distance
'
PTeach=P_100(MWrkNo)         'Position taught to the robot
PVsWrk=P_101(MWrkNo)         'Position recognized by VS
'CCOM$=C_100$(MWrkNo)        'COM port number
'CPRG$=C_101$(MWrkNo)        'Vision program name
Return
##### VS initialization processing #####
*DATAAL2
'Move from the robot coordinate axis (P_Zero position) to the robot origin when the vision sensor
recognizes workpieces
MED1#=M_100#(MEncNo)         'Amount of conveyer movement at calibration between vision
and robot
PRbOrg=P_EncDlt(MEncNo)*MED1# 'Robot origin when the vision sensor recognizes workpieces
'Return a workpiece recognized by the vision sensor to the position taught to the robot
MED2#=M_101#(MWrkNo)         'Amount of conveyer movement from vision sensor recognition
to workpiece teaching
PBack=P_EncDlt(MEncNo)*MED2# 'Calculate the position of the workpiece that the vision sensor in the
robot area recognized
PWrkPos=PRbOrg+PVsWrk+PBack   'Workpiece position recognized by the vision sensor into the
robot area
PVector=(P_Zero/PWrkPos)*PTeach 'Vectors specifying the center of gravity of the vision sensor
and grabbing position
Return
'
##### Condition setting #####
*DATAAL3
MDX = P_EncDlt(MEncNo).X     'Amount of movement per pulse (X)
MDY = P_EncDlt(MEncNo).Y     'Amount of movement per pulse (Y)
MDZ = 0 P_EncDlt(MEncNo).Z   'Amount of movement per pulse (Z)
MD = Sqr(MDX^2+MDY^2+MDZ^2)  'Calculation of the amount of movement per pulse

```

```

MEI#=Abs((MVLen-MWrkLen)/MD)      'Calculation of imaging start setting value
MEDLX#=P_EncDlt(1).X
MEDLY#=P_EncDlt(1).Y
MPDQ#=#Sqr(MEDLX#^2+MEDLY#^2)
Return
'
##### VS recognition check processing #####
*DATABEKLE
MNUM=0
""If M01=0 Then GoTo *DATAAL
If M_In16(10160)=100 Then GoTo *DATAAL      'Comparison between the amount of encoder
movement and the camera startup setting value
GoTo *DATABEKLE
*DATAAL
MNUM=M_In16(10320)
If MNUM=0 Then
    TOPLAMGIREN=TOPLAMGIREN+1
    If TOPLAMGIREN>10000 Then TOPLAMGIREN=0
    M_Out16(10176)=TOPLAMGIREN
    Wait M_In16(10160)=0
    GoTo *DATABEKLE
EndIf
ANLIK=0
"M01=M01+1
*TEKRARDATAAL
ANLIK=ANLIK+1
MCOUT#=# M_Enc(1)
Dly 0.1
MRead#=#M_Enc(1) 'Karşılaştırma için almış
M1#=#M_In16(10320+ANLIK*16) ' x
M2#=#M_In16(10400+ANLIK*16) ' y
"M3#=#Rad(M_In16(10480+ANLIK*16)-360+PDNM.X)
""M3#=#Rad(0) 'Deg(M_In16(10208)) ' aci
M3#=#Rad(M4#)
PVS1.X=M1#
PVS1.Y=M2#
PVS1.C=M3#
MResult#=#M_Enc(1)
""If MNUM=0 Then GoTo *DATABEKLE      'If no workpieces are recognized
""MNUM=0
If MNUM>4 Then MNUM=4      'Set the maximum number (4)
MVsx=PVS1.X      'Position[Pixel] recognized by Vision
MVsy=PVS1.Y
MVsc=Deg(PVS1.C)
PosVS=#PVSCal(1, MVsx, MVsy, MVsc) 'Workpiece position in vision area[mm]
MX(1)=PosVS.X 'PVSx
MY(1)=PosVS.Y
MT(1)=PosVS.C
GoSub *BUFFERDOLDUR      'Tracking data storing processing
Return
'
##### Tracking data storing processing #####
*BUFFERDOLDUR
PSW=P_Zero
PSW=#PRbOrg      'Virtually move the robot close to the vision sensor
PSW.X=PSW.X+MX(1)      'Create the grabbing position
PSW.Y=PSW.Y+MY(1)
PSW.C=PSW.C+MT(1)
PRW=P_Zero
PRW=#PSW*PVector      'Compensate for the error in the calculation value

```

```
'PRW.Z=PZ.Z
PRW.FL1=P_100(MWrkNo).FL1
PRW.FL2=P_100(MWrkNo).FL2
PRW.C=PVS1.C
TrWrt PRW, MCOUT#, MWrkNo,1,MEncNo 'Position, encoder value, model number, buffer number,
encoder number
MCount=MCount+1
If ANLIK<MNUM Then GoTo *TEKRARATAAL
BUFFER=M_TrkBuf(1)
TOPLAMGIREN=TOPLAMGIREN+1
If TOPLAMGIREN>10000 Then TOPLAMGIREN=0
M_Out16(10176)=TOPLAMGIREN
Wait M_In16(10160)=0
Return
```



EK-7 Enkoder takip kodu

```
ENCDEGERI_SON = 0
*DONGU
ENCDEGERI_ILK=M_Enc(1)
If ENCDEGERI_SON - ENCDEGERI_ILK<-1000 Then
  SAYAC=SAYAC+1
  M_Out16(10160)=SAYAC
  ENCDEGERI_SON = ENCDEGERI_ILK
EndIf
If SAYAC >1000 Then SAYAC=0
GoTo *DONGU
```

