

DOKUZ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAREKETLİ KONVEYOR ÜZERİNDE KAMERA
GÖRÜNTÜSÜ İLE NESNE TANIMA VE
NESNELERİ YERİNE KOYMA UYGULAMASI

Turgay ERDOĞAN

Şubat, 2012
İZMİR

HAREKETLİ KONVEYOR ÜZERİNDE KAMERA GÖRÜNTÜSÜ İLE NESNE TANIMA VE NESNELERİ YERİNE KOYMA UYGULAMASI

**Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
Mekatronik Mühendisliği Anabilim Dalı**

Turgay ERDOĞAN

**Şubat, 2012
İZMİR**

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

TURGAY ERDOĞAN tarafından **YRD.DOÇ.DR. AHMET ÖZKURT** yönetiminde hazırlanan “**HAREKETLİ KONVEYOR ÜZERİNDE KAMERA GÖRÜNTÜSÜ İLE NESNE TANIMA VE NESNELERİ YERİNE KOYMA UYGULAMASI**” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir yüksek lisans tezi olarak kabul edilmiştir.

Yrd.Doç.Dr. Ahmet ÖZKURT

Yönetici

Doç.Dr. Zeki KIRAL

Jüri Üyesi

Doç.Dr. Mehmet ÇAKMAKÇI

Jüri Üyesi

Prof.Dr. Mustafa SABUNCU

Müdür

Fen Bilimleri Enstitüsü

TEŞEKKÜRLER

Yüksek lisansımın her aşamasında, akademik gelişimime katkıda bulunan ve bu tezi hazırlamamda emeği geçen değerli hocam Yrd.Doc.Dr. Ahmet ÖZKURT'a; değerli hocam Yrd.Doc.Dr. Nalan ÖZKURT'a; değerli hocam Prof.Dr. Erol UYAR'a; ayrıca yüksek lisans yapmamda en büyük desteği veren sevgili eşim Serpil ERDOĞAN ve biricik kızım Ece Naz ERDOĞAN'a teşekkür ederim.

Turgay ERDOĞAN

HAREKETLİ KONVEYOR ÜZERİNDE KAMERA GÖRÜNTÜSÜ İLE NESNE TANIMA VE NESNELERİ YERİNE KOYMA UYGULAMASI

ÖZ

Günümüzde, görüntü işleme çalışmaları arttıkça uygulama alanları da hızla gelişmektedir. Kamera gibi görüntü araçları görüntü işleme algoritmaları eklenerek bir çeşit gelişmiş sensör ve tek başına sistem ve makine olarak kullanılmaya başlanmıştır. Bu gelişmeden yola çıkarak bu tezde düşünülen, bant sistemlerinin üzerinden paketlemeye kadar olan süreçte malzeme tespiti ile ilgili bir çalışmanın başarısını görmektir. Bu tespitin hem hatalı ürünü ayırt edecek hem de ürünün doğru sınıflandırmasını yapacak özellikleri taşıması beklenmektedir.

Bu tezde, nesnelerin görüntü içerisindeki yerine, açısına bağlı kalmadan tanınması, sınıflandırılması ve robot kol ile yerine konması amaçlanmıştır. Bu amaca yönelik deneysel bir düzenek oluşturulmuştur. Bu deneysel düzende aktif bir kamera sistemi, nesnelerin özelliklerinin çıkartılmasında kullanılan görüntü işleme algoritması, bu özellikleri kullanarak nesnelerin tanınmasını sağlayan bir Yapay Sinir Ağı (YSA) algoritması, tanınan nesnelerin yerine konması için kontrol ünitesi ve bir robot kol kullanılmıştır.

Bu sistemde konveyör bant hız kontrolü, haberleşme ve robot kol açı değerlerinin sağlanması için mikroişlemcili kontrol ünitesi hazırlanmıştır. Robot kol üç eksenli olarak rc tip servo motorlu yapıdadır. Robot kol gövdesi lazer kesim alüminyumdan yapılmıştır. Konveyör bant üzerine yerleştirilen pc kamerası ile makine görmesi sağlanmıştır.

Bu tez çalışmasının üretimin ihtiyacı olan bu sistemlerin gelişmesinde kullanılması ve bu sistemlerin üretim hatlarındaki yerlerini alması umulmaktadır.

Anahtar Sözcükler: Nesne tanıma, robot kol, görüntü işleme, yapay sinir ağı

OBJECT RECOGNITION WITH CAMERA IMAGES ON MOVING CONVEYOR AND PICK AND PLACE APPLICATION THE OBJECTS

ABSTRACT

Nowadays, image processing applications are increased and as paralel, the application areas of image processing implementations are developed in wide areas. Embedded imaging tools like cameras and the program algorithms had been begun to use like a sensor and astandalone machine. The goal in this study, is to evaluate the success of the study concerning the determination of the material on conveyor from the production machine to packaging. It is expected that this determination should have some features like determination the faulty material and recognition the correct product.

In this thesis, it is aimed to recognize, classify without regarding its location, orientation in an image and to pick and place the target parts with the robot arm. For this aim, an experimental system is designed and implemented. This system includes an active camera system, the image processing algorithm which is used in feature extraction, an Artificial Neural Network (ANN) which is used for recognition and classification, and a robot arm and its control unit for pick and place application.

In the developed system, a speed control circuit for conveyor, a control unit with a microcontroller for communication with computer and a robot arm have been applied. The serial robot manipulator arm has three axis and uses RC type servo motors. The robot arm body is laser-cut aluminium material. The machine vision has been provided using a webcam camera which is placed in the conveyor box.

It is hoped that application of this thesis can be used in development of similiar systems and these type systems can be used in the production lines.

Keywords: Object recognition, robot arm, image processing, neural network

İÇİNDEKİLER

Sayfa

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU.....	ii
TEŞEKKÜRLER.....	iii
ÖZ.....	iv
ABSTRACT.....	v

BÖLÜM BİR – GİRİŞ.....1

1.1. Projenin Amacı.....	1
1.2. Nesnelerin Birbirinden Ayırt Edilmesi.....	2
1.3. Daha Önce Yapılan Çalışmalar ve Yayınlanan Makaleler.....	5
1.4. Tez Aşamaları.....	8

BÖLÜM İKİ – TEORİK ALTYAPI.....9

2.1 Kamera Özellikleri.....	9
2.2 Görüntü işleme teknikleri ve özellik çıkartma.....	10
2.2.1 Eşik Belirleme.....	10
2.2.2 Kenar Belirleme.....	10
2.2.3 Bölge Büyütme.....	11
2.2.4 Filtreleme.....	12
2.2.5 Özellik Çıkartma.....	13
2.3 Yapay Sinir Ağları.....	14
2.4 Robot Kol Kinematikı.....	18
2.4.1 Hız ve ivme Denklemleri.....	20
2.5 Robot Kol Dinamik Analizi.....	22

BÖLÜM ÜÇ - DENEYSEL ÇALIŞMA.....26

3.1 Genel Yapı	26
3.1.1 Nesne Tanıma Programı ve Kullanıcı Arayüzü.....	29
3.1.1.1 Görüntü İşleme.....	30
3.1.1.2 Teach Butonu, Tool Menüsü Teach Butonu ve İşlevi.....	31
3.1.1.3 Learn Butonu, Tool Menüsü Learn Butonu ve İşlevi.....	34
3.1.1.4 Start Butonu, Tool Menüsü Start Butonu ve İşlevi.....	35
3.1.1.5 Stop Butonu, Tool Menüsü Stop Butonu ve İşlevi.....	36
3.1.1.6 Load Database Butonları ve İşlevi.....	36
3.1.1.7 Tanıma.....	36
3.1.2 DC motor pwm sürücü ve haberleşme elektronik kartı.....	37
3.1.3 Konveyör bant ve mekanik gövde.....	38
3.1.4 Robot Kol Manipülatörü Kinematik ve Dinamik Hesapları.....	40
3.1.5 PC Kamerası.....	45
3.2 Deneysel Veriler.....	46

BÖLÜM DÖRT – SONUÇ VE ÖNERİLER.....49

KAYNAKLAR52

EKLER57

BÖLÜM BİR

GİRİŞ

1.1 Projenin Amacı

Günümüzde, görüntü işleme ve nesne tanıma çalışmaları ve araştırmaları arttıkça uygulama alanları da hızla gelişmektedir (Wahi, Ravi ve Saranya, 2010), (Wahi, Athiq ve Palasinamy, 2008), (Fuente, Echanobe, Campo, Susperregui ve Murtua, 2010), (Yang, Zhu, Lu ve Yu, 2010). Kamera gibi görüntü araçlarının sağladığı görüntüyü ve görüntü işleme algoritmalarını kullanarak karar verebilen yapısı ile nesneleri tanıyan ve taşıma, yer değiştirme yapan robotik sistemler geliştirilmektedir (Declercq ve De Keyser, 1999), (Johan ve Prabuwo, 2011), (M.P.Cabrera, Juarez, R.R.Cabrera, Osorio ve Gomez, 2010), (Fuente, Echanobe, Campo, Susperregui, Murtua, 2010). Bilgisayar ve makine görmesi kavramı bu gelişmelerle ortaya çıkan, gören makinelerin teknolojisidir (Johan ve Prabuwo, 2011). Bu tür sistemlerde yaygın olarak yapay sinir ağı (YSA) tekniği kullanılmaktadır (Wang ve Shang, 2010), (Declercq ve De Keyser, 1999), (Johan ve Prabuwo, 2011), (Yishuang, Hong ve Gang, 2010), (M.P.Cabrera, Juarez, R.R.Cabrera, Osorio ve Gomez, 2010), (Onis, Garcia, Sanson ve Dugelay, 2009).

Bu gelişmelerden yola çıkarak bu tez çalışmasında amaç, bant sistemlerinin üzerinden paketlemeye kadar olan süreçte malzeme tespiti ve ayırma ile ilgili bir sistem tasarlamak ve başarısını görmektir. Bu tespitin hem hatalı ürünü ayırt edecek hem de ürünün doğru sınıflandırmasını yapacak özellikleri taşıması beklenmektedir.

Mevcut üretim makinelerinin birçoğunda üretim bandı bulunduğu ve çıkan ürünlerin hata kontrolü ve ayrıştırılması işlemlerinin uzman operatörler tarafından yapıldığı görülmektedir. Şimdiye kadar makineleştirilmeye yeni yeni başlanan ve yaygınlaştırılamayan bu tür sistemlerin üretim için ihtiyaç olduğu ve bu sistemlerin geliştirilerek üretim hatlarındaki yerlerini alması gerekmektedir.

Hareketli objelerin belirlenmesi ve tanınması işi çevre şartlarına bağlı olarak zordur (Syed, Zafar ve Moeen, 2009). Nesne tanımadaki başarı oranları; kullanılan modellere, görüntü işleme ve öğrenme algoritmasına, çevre şartlarına ve donanım özelliklerine bağlı değiştiği için değişkenlik göstermekte ve benzer çalışmalar ile detaylı karşılaştırma yapmayı zorlaştırmaktadır (Johan ve Prabuwno, 2011).

Bu projede kullanılan yazılım, kamera görüntüsü sınırları içinde kalmak şartı ile her türlü malzemenin tanıtılabileceği ve sistemin tanıyabileceği şekilde kullanılabilir. Sistem veritabanı genişletilebilir yapıdadır. Kullanılan malzemeler piyasada kolay bulunabilen ve uygun maliyetli çözüm sunabilen bir yapıdadır.



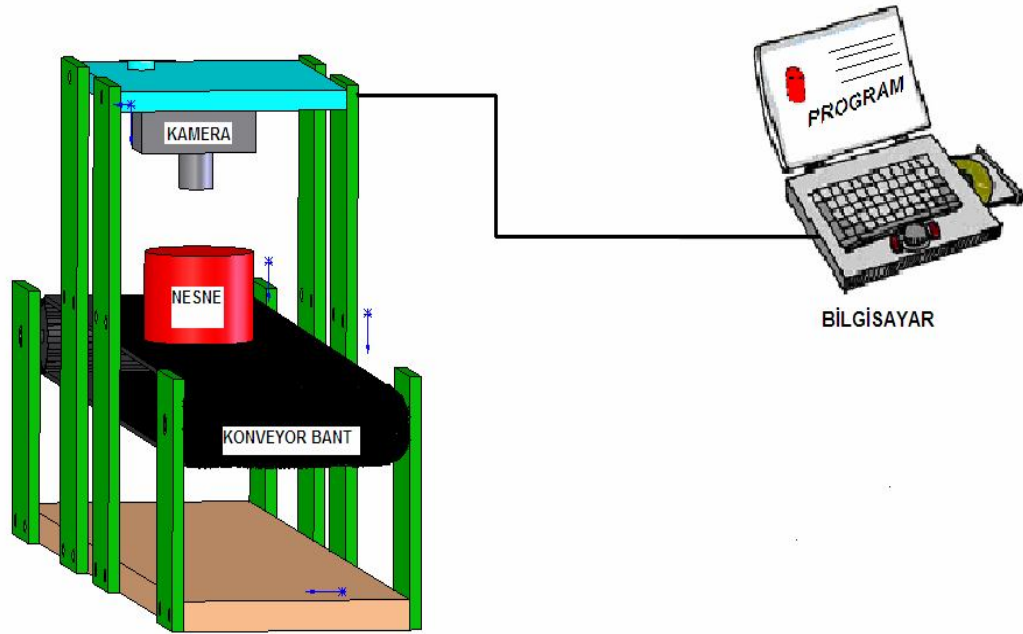
Şekil 1.1 Dört kollu, kameralı delta robot

1.2 Nesnelerin Birbirinden Ayırt Edilmesi

Nesnelerin kamera kullanılarak birbirinden ayırt edilmesi, nesnenin görsel özellikleri olan rengi, şekli, çevre uzunluğu, köşe sayısı, kenar uzunluğu, ışık yansıtması, histogram gibi özelliklerinin akıllı sistemler ile bir algoritmada işlenmesi ile olmaktadır (Türkoğlu, 2003). Ayırt edilecek nesnelerin özellikleri birbirinden çok zıt özellikler taşıyor ise akıllı sistem (Yapay Sinir Ağı, Fuzzy Lojik vs.) ihtiyacı oluşmadan basit seçme algoritması ile de nesne tanıma yapılabilir. Nesne özelliklerinin birbirine yakın olduğu uygulamalarda nesneye ait özel ağırlıklı özellik ve karakterlerini değerlendirecek akıllı sistemlere ihtiyaç oluşmaktadır. Eğer nesneler birbirine çok benzer ise ayırtetme çok daha zor olur (Declercq ve De Keyser, 1999). Yapay sinir ağı tekniği nesne tanıma ve robot uygulamalarında geniş bir şekilde

kullanılmaktadır (Moreno, Alarcon, Salvador ve Riesgo, 2009), (Johan ve Prabuwno, 2011).

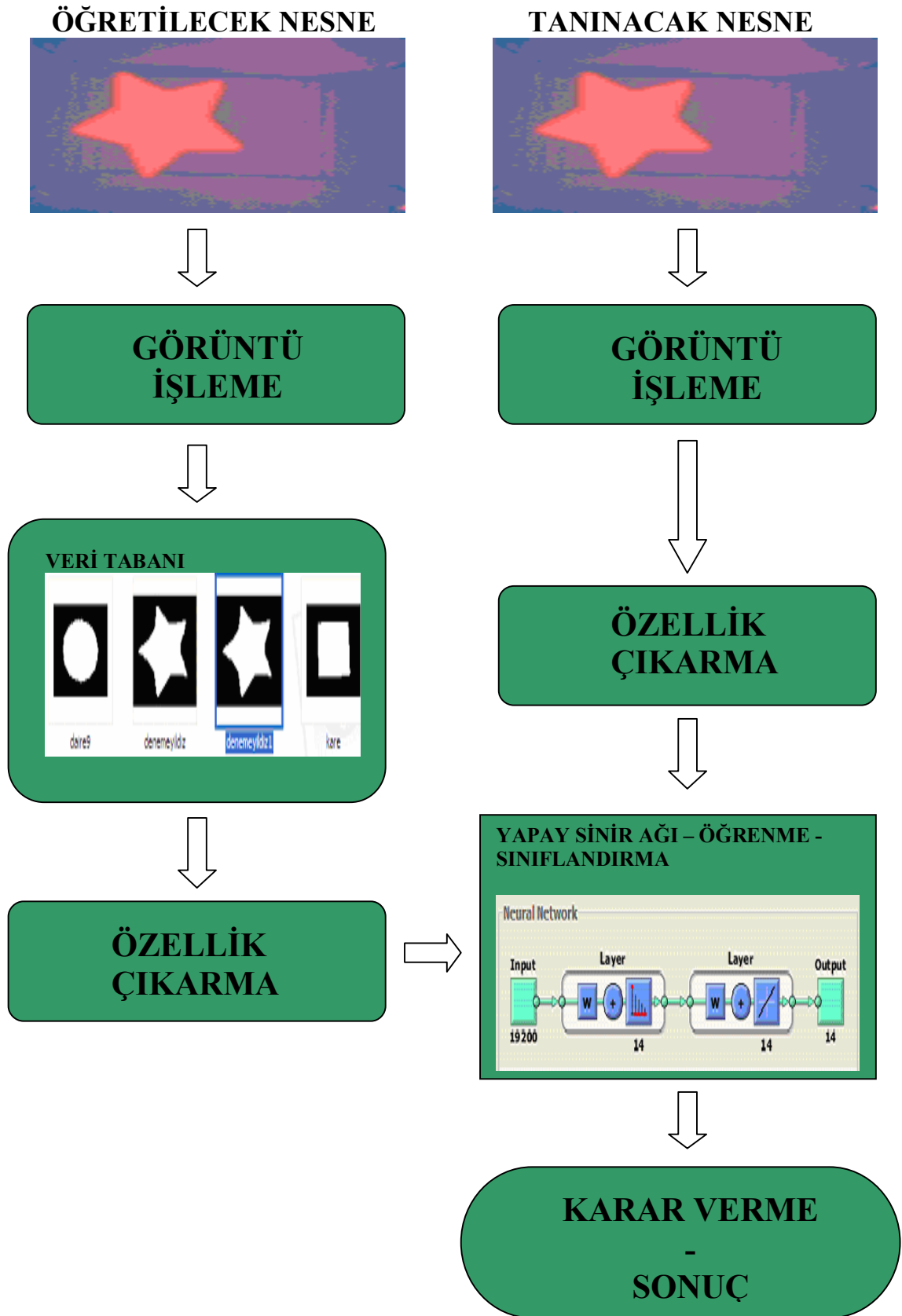
İnsanların nesneleri tanınması, insanların nesnelere ait özellikleri algılamaları ve bu özellikleri daha önce öğrendikleri bilgiler ile karşılaştırmaları ve benzetme ile olur. Nesne tanıma sistemlerinde de benzer yöntem söz konusudur. Önce tanınacak nesneler için veritabanı oluşturulur. Bu veritabanı sisteme öğretilir ve karşılaştırma algoritması ile nesne, daha önce öğrendiği ile kıyas yaptırılıp nesnenin tanınması sağlanır (Johan ve Prabuwno, 2011).



Şekil 1.2 Nesne tanıma sistemi genel şeması

Nesne tanıma sistemlerinde en önemli parça sistem yazılımıdır. Yazılımın çevre girdilerinin de iyi olması gerekir ki sistemin nesne tanınması daha iyi gerçekleşebilsin. Çevre girdisinde ışıklandırma ve kamera özellikleri yazılıma giden görüntü kalitesini etkileyen en önemli unsurlardır (Ali, M.F ve Tayyab, 2009), (Johan ve Prabuwno, 2011).

Nesne tanıma algoritması genel yapı olarak aşağıdaki adımlardan oluşur.



Şekil 1.3 Sistem akış şeması

Öğretilecek nesne algoritmasında, örnek görüntü pc kamerası aracılığı ile renkli 160x120x3 formatında digital olarak alınır. Algoritmada renkli görüntü kullanılmadığı için siyah beyaz hale getirilen görüntü, görüntü işleme adımları ile işlenerek belli bir dosyada kaydedilir. Diğer öğretilecek nesnelere ait görüntüler de aynı dosyaya aynı yöntem ile işlenerek kaydedilir. Kaydedilen bu görüntülere ait şekil ve piksel özellikleri seçilen yapay sinir ağı girişine verilerek bu özellikleri ağın öğrenmesi sağlanır.

Tanınacak nesne algoritmasında, kamera görüntüsü ile gerçek zamanlı görüntü pc kamerası aracılığı ile renkli 160x120x3 formatında digital olarak alınır. Algoritmada renkli görüntü kullanılmadığı için siyah beyaz hale getirilen görüntü, görüntü işleme adımları ile işlenir, özellikleri çıkartılır ve daha önce öğretilen nesnelere ait özellikleri yapay sinir ağı karşılaştırarak benzetme yapar ve bir sonuca ulaşır.

1.3 Daha Önce Yapılan Çalışmalar ve Yayınlanan Makaleler

Konveyör bant üzerinde ürün tanıma, ayırıştırma ile ilgili deneme çalışmalarına rastlanmaktadır (Declercq ve De Keyser, 1999), (Johan ve Prabuwo, 2011). Bunun yanında yapay sinir ağı kullanılarak yapılan birçok nesne (ürün) tanıma makale ve çalışmaları da bulunmaktadır. Bu yoğun ve çok sayıda çalışmalardan bir kısmının özeti aşağıda verilmiştir.

A. D. Kulkarni, Al. C. Yap ve P. Byars tarafından 1990 yılında yayınlanan makalede geri yayınlı ve rekabetçi öğrenme algoritması ve iki katmanlı sigmoid fonksiyonlu ağ ile yapılan çalışmada örneklerin scale farklılıkları sebebiyle iyi sonuç alınamamıştır (Kulkarni, Yap ve Byars, 1990).

Dr. Amitabh Wahi , F. Mohamed Athiq ve C. Palanisamy tarafından 2008 yılında yayınlanan makalede döndürülmüş nesne için yeni hibrit özellik çıkarımı amaçlanmış, 2D-DFT Discrete Fourier Transformation (DFT) ve 2D-DFT + 2D-DWT Discrete Wavelet Transformation (DWT) metodları ile nesne görüntülerinden özellikler çıkartılmıştır. Bir giriş katmanlı, iki gizli katmanlı, bir çıkış katmanlı, geri yayınlı,

ileri beslemeli 128:68:35:10 ağ mimarisi ile tanıma başarımı %95.5 ve 160:85:55:10 ağ mimarisi ile tanıma başarımı %98.33 şeklinde gerçekleşmiştir (Wahi, Athiq ve Palanisamy, 2008).

Dr. Amitabh Wahi, Priyadharshini Ravi ve M. Saranya tarafından 2010 yılında yayınlanan makale de DWT metodu ile 5, 10, 15, 20, 25 derecelerde döndürülmüş görüntüler ile geri yayınlı Widrow-Hoff öğretme kurallı çok katmanlı ağ ile yapılan tanıma çalışmasında 5 derecede %95,9, 25 derecede %56,67 başarı elde edilmiştir (Wahi, Ravi ve Saranya, 2010).

Yuhua Zheng, Yan Meng ve Yaochu Jin tarafından 2010 yılında yayınlanan makale de iki katmanlı aşağı ve yukarı yol birleştirmeli sinir ağı (fused bottom-up and top-down pathways, FBTP-NN), FBTP-NN ağ ve üç katmanlı ileri beslemeli ağ (FF-NN) ile yapılan tanıma çalışmasında %90 - %98 arasında başarılı tanımlar elde edilmiştir (Zheng, Meng ve Jin, 2010).

Abhijit Neogy, S.N.Balakrishnan, ve Cihan H. Dagli tarafından 1992 yılında yayınlanan makalede IBM 7535 robotu ile 13cm*9cm kamera görüntüsü içinde 3cm/s hızla başarılı tanıma elde edilmiştir (Neogy, Balakrishnan ve Dagli, 1992).

Ji Ming ve Shen Zhenkang tarafından 1992 yılında yayınlanan makale de karışık uzay sinir ağ mimarisi (a Space-Perturbation Neural Network, SPNN architecture) ile yapılan çalışmada 4000 uçak veritabanı görüntüsü ile eğitildiğinde gürültüsüz görüntüler ile %99,8 , 0.6 gürültü oranı ile %95,4 tanıma oranı gerçekleştirilmiştir (Ming ve Zhenkang, 1992).

B. Chen ve L. L. Hoberock, tarafından 1996 yılında yayınlanan makale de yeni bulanık yapaysinir ağı (the new fuzzy neural network fuzamp) ile tanıma yüksek doğrulukla elde edildi (Chen ve Hoberock, 1996).

Hong Pan ve Liang-Zheng Xia tarafından 2008 yılında yayınlanan makalede dalgacık yapay sinir ağı (wavelet neural network, WNN), dalgacık dönüşümlü yapay

sinir ağı (wavelet transform artificial neural network, WT-ANN), en yakın K komşu (K-nearest-neighbor, k-NN), şekil bağlamı (shape context, SC-KNN) ve eğrilik derecesi uzaysal temsili (curvature scale space representation, CSS-KNN), destek vektör makinesi (the support vector machine, SVM), geleneksel fourier tanımlamaları (the traditional fourier descriptors, FD-SVM), en büyük dalgacık modüllü yapay sinir ağı (the wavelet modulus maxima artificial neural network, WMM-ANN) ile yapılan tanıma çalışmalarında %82,57 - %99,28 oranında doğru tanıma sonuçlarına ulaşılmıştır (Pan ve Xia, 2008).

Yuhua Zheng ve Yan Meng tarafından 2011 yılında yayınlanan makale de alt ve üst yol parçalarını birleştirmeli modüler yapay sinir ağı (modular neural network system with fusing both bottom-up and top-down pathways, FBTP-MNN) ve ileri beslemeli modüler sinir ağları (feed-forward modular neural Networks, FF-MNN), sadece pozitif öğrenme stratejisi (the positive-only, PO training strategy), pozitif ve negatif öğrenme ağları (positive and negative PN training Networks) ile yapılan tanıma %42 - %97 başarı ile neticelenmiştir (Zheng ve Meng, 2011).

N.G.Chitaliya ve A.I.Trivedi tarafından 2010 yılında yayınlanan makale de temel bileşen analizi (Principal Component Analysis, PCA), dalgacık – temel bileşen analizi (wavelet-PCA) ve doğrusal ayırım analizi (Linear Discrimination analysis) ile özellik çıkartımı ve sinir modeli (Neuron Model, tansig, logsig, purelin), ileri besleme sinir ağı (Feedforward Neural network) ile yapılan çalışma sonuçlarında %77,7 - %100 arasında başarılı tanımlar gerçekleştirilmiştir (Chitaliya ve Trivedi, 2010).

Wenming Cao, Fei Lu ve Shoujue Wang tarafından 2006 yılında yayınlanan makale de desen tanıma için öncelik sıralı yön esas fonksiyonlu sinir ağı (the priority order direction basis function neural network for pattern recognition PO DBFNN) ile yapılan çalışmada %98,5 doğruluk oranı elde edilmiştir (Cao, Lu ve Wang, 2006).

Patil P.M, Manish Deshmukh, P.V.Bonde, Dhabe P.S ve T.R. Sontakke tarafından 2004 yılında yayınlanan makale de 3 katmanlı ileri beslemeli fourier fuzzy sinir ağı

(feed forward Fourier fuzzy neural network, FFNN) ile yapılan tanıma çalışmasında %83 başarılı tanıma gerçekleştirilmiştir (Patil, Deshmukh, Bonde, Dhabe ve Sontakke, 2004).

Teuku Muhammad Johan ve Anton Satria Prabuwno tarafından 2011 yılında yayınlanan makalede yapay sinir ağı kullanılarak konveyör bant üzerinde cıvata ve somun tanıma çalışması yapılmış ve %92 tanıma başarısı elde edilmiştir (Johan ve Prabuwno, 2011).

1.4 Tez Aşamaları

Birinci bölümde tezin amacı, nesne tanıma ve robot uygulamaları hakkında bilgi ve daha önce yapılan benzer çalışma ve makale özetleri verilmiştir.

İkinci bölümde, sistemde kullanılan görüntü işleme, nesne tanıma, yapay sinir ağı, robot kol kinematiği metod ve yöntemleri teorik altyapı olarak anlatılmıştır.

Üçüncü bölümde, gerçekleştirilen sistemde bulunan kullanıcı arayüzü, mekanik gövde ve konveyör bant, haberleşme ve sürücü elektronik kartları ve robot kol kinematiği hakkında bilgiler verilmiştir.

Dördüncü bölümde, gerçekleştirilen sistemin iş yapabilme kabiliyeti ve geliştirilebilecek yönleri ile ilgili sonuç değerler ve öneriler verilerek tez sonuçlandırılmıştır.

BÖLÜM İKİ

TEORİK ALTYAPI

Bir görüntü $f(x,y)$ şeklindeki iki boyutlu bir fonksiyon şeklinde tanımlanabilir. Burada x ve y koordinat bileşenleri olup bu koordinatlara bağlı olarak f fonksiyonunun değeri ise yoğunluk (intensity) veya diğer bir deyişle grilik seviyesi (gray level) olarak isimlendirilir. f fonksiyonunun x,y ve yoğunluk değerlerinin tümü sonlu ise bu görüntü “sayısal görüntü” (digital image) olarak adlandırılır (Gonzales ve Woods, 2002).

Modelleme ve canlandırma için kullanılan sentetik görüntüler bilgisayarlar vasıtasıyla oluşturulurlar. Dijital görüntüleri elde etmek için iki öge gereklidir. İlki, görüntüde istediğimiz obje tarafından ışını yayan enerjiye duyarlı fiziksel bir aygıttır. İkincisi, dijitalleştirici adlandırılan, fiziksel algılama aygıtının çıkışını digital forma çevirmek için bir aygıttır (Gonzalez ve Woods, 2002).

2.1 Kamera Özellikleri

Kamera özellikleri olarak çözünürlük, fps (frame per second-saniyede çektiği fotoğraf), yakınlaşma – uzaklaşma (zoom) ve odaklama sayılabilir. Kamera çözünürlüğü, kameranın yakalayabildiği ayrıntı miktarı olarak adlandırılır. Kamera içinde yer alan CCD yada CMOS resim algılayıcısının sahip olduğu piksel sayısı kameranın çözünürlüğünü belirler. Kamera çözünürlüğü ne kadar yüksek ise o kadar iyi görüntü detayı alınabilir. Fps, kamera içinde bulunan görüntü işleme işlemcisinin özelliğidir. Hareketli görüntü çekebilme özelliği hareketli nesne çekimlerinde çok önemli bir özelliktir ve fps değerinin yüksek olması nesnenin ayırt edici özelliğinin hareket sebebiyle farklılaşmasını ve kaybolmasını engeller. Zoom, hedefe yakınlaşma veya hedeften uzaklaşma değeridir. Kameralarda ya optik zoom ya dijital zoom yada her ikisi de kullanılır. Optik zoom da lens düzeneği içindeki merceklerin konumu değiştirilerek gerçek büyütme ve küçültme işlemi yapılır. Dijital zoom işlemi resmin aralarına piksel yerleştirme veya resimden piksel alma şeklinde dijital ortamda yapılır. Gerçek büyüme veya küçülme değildir. Büyütme

oranı ölçeğinde resim kalitesi düşer (Mesleki eğitim ve öğretim sisteminin güçlendirilmesi projesi- Megep, 2007).

2.2 Görüntü İşleme Teknikleri ve Özellik Çıkartma

Algılayıcılar vasıtasıyla, bilgisayara sayısal olarak alınmış olan görüntü, daha basit bir şekle getirmek için bir dizi görüntü işleme sürecinden geçirilir. Kullanılan önemli görüntü işleme ve özellik çıkartma teknikleri aşağıda verilmiştir.

2.2.1 Eşik Belirleme

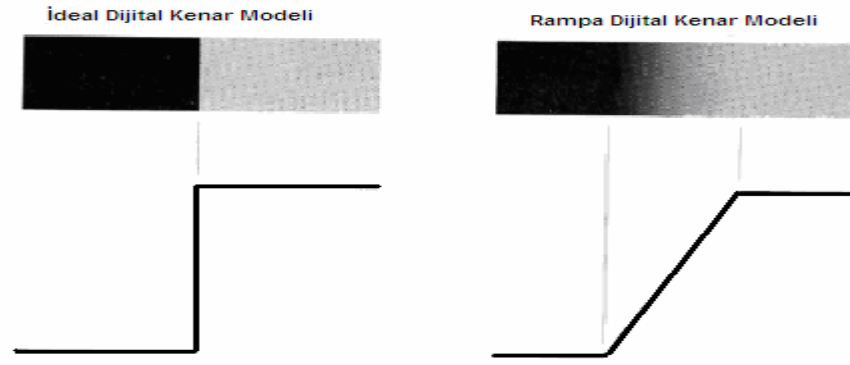
Görüntü işlemede en çok tercih edilen yöntemlerden biridir. Özellikle arka plan ve nesneyi birbirinden ayırt etmek için kullanılır (Gonzales ve Woods, 2002).

Sayısal bir görüntünün eşikleme işlemine tutulmasındaki amaç, nesne görüntüsünün özelliklerini belirlemede kolaylık sağlamaktır. Eşikleme işlemiyle, görüntü iki renkle ifade edilebilir biçime getirilir. Görüntüyü eşikleme işlemine tabi tutmadan önce bir eşik değeri saptanır. Eşik değerinden daha yüksek gri seviye değerine sahip olan piksellere “1” değeri, daha küçük değerlere sahip olan piksellere ise “0” değer ataması yapılarak görüntü daha basit bir biçime (siyah-beyaz) getirilmiş olur (Türkoğlu, 2003).

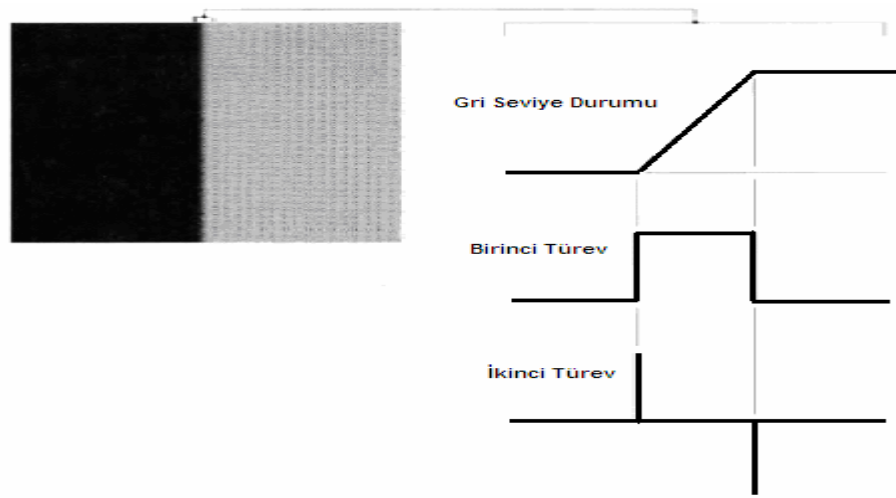
2.2.2 Kenar Belirleme

Kenar belirleme, gri seviyede anlamlı süreksizlikleri belirlemek için en yaygın yaklaşımdır. Kenar, iki bölge arasında sınırda uzanan bağlantılı piksellerin kümesidir (Gonzales ve Woods, 2002).

Kenar çıkarmadaki amaç, görüntünün içerdiği bilgiyi değerlendirip, gereksiz ve tanıma işlemlerinde zaman kaybettiren bilgiyi eleyerek yeterli düzeye indirgemektir (Türkoğlu, 2003).



Şekil 2.1 Görüntü işlemede dijital kenar modelleri (Gonzales ve Woods, 2002)

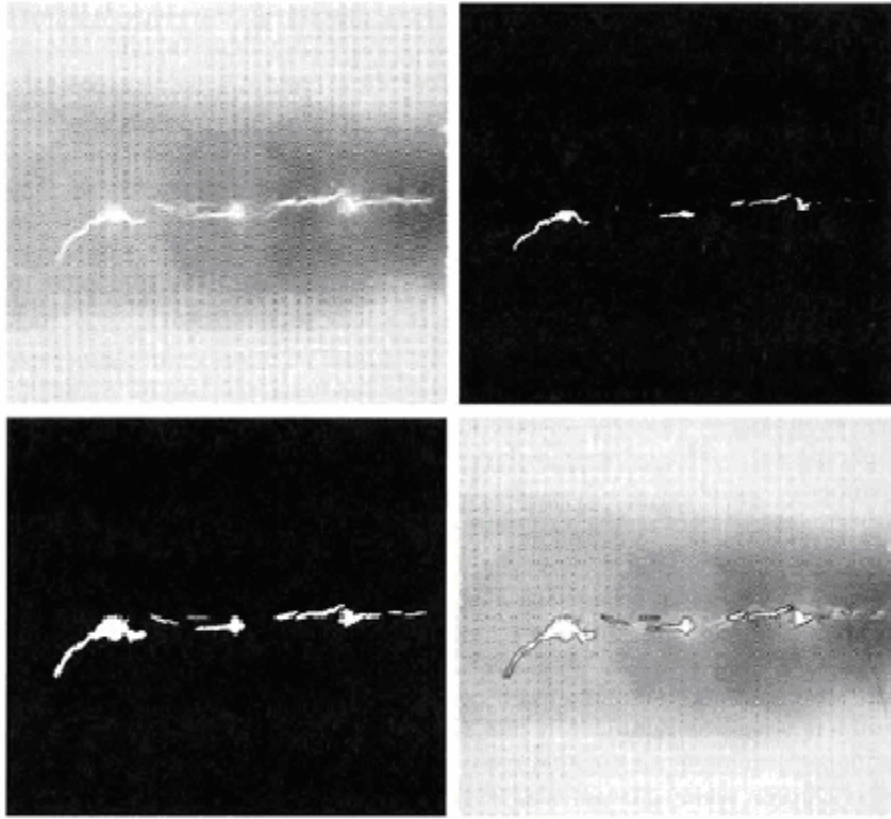


Şekil 2.2 Görüntü işlemede dijital kenarın türev alma ile değişimi (Gonzales ve Woods, 2002)

Birinci türevin genliği, görüntüde bir noktada kenarın bulunmasını belirlemek için kullanılabilir. Benzer şekilde ikinci türevin işareti ya karanlıkta yada kenarın ışık tarafında uzanan kenar piksellerini belirlemek için kullanılabilir (Gonzales ve Woods, 2002).

2.2.3 Bölge Büyütme

Bölge büyütme, büyütme için önceden belirlenmiş kriterlere bağlı olarak daha geniş bölgeler içindeki pikselleri gruplayan bir prosedürdür. Temel yaklaşım, belli noktaların kümesi ile başlamaktır. Bu noktalardan her bir belli komşu noktaya yaklaşıma ile bölgeler büyür (Gonzales ve Woods, 2002).



Şekil 2.3 Kaynak görüntüsü ile bölge büyütme çalışması (Gonzales ve Woods, 2002)

2.2.4 Filtreleme

Dijital görüntüler söz konusu olduğunda görüntü frekansı terimi, görüntü düzleminde piksellerin gri ton değerlerinin değişimini ifade eder. Frekans, görüntünün bir pikselinden diğerine geçildiğinde parlaklık değerindeki değişim ölçüsüdür. Gri ton değerlerinin küçük farklarla değiştiği bir görüntüde alçak frekanslar ağırlıktadır. Ton farkı büyük bölgeler veya ince detaylar içeren görüntülerde yüksek veya alçak frekansları öne çıkarmak veya bastırmak mümkündür. Yüksek frekansları geçiren filtreler görüntünün alçak frekanslı bileşenlerini zayıflatır veya ortadan kaldırırken yüksek frekanslı bölgeleri öne çıkarır. Yüksek frekanslı bileşenler genellikle bölge kenarlarını veya diğer keskin detayları işaret ettiğinden, yüksek geçiren filtre görüntüyü keskinleştirir (Çayırılı, 2006).

Filtre tipleri olarak düzgünleştirme (smoothing), ortalama (average), median, gauss filtreleri sayılabilir.



Şekil 2.4 Görüntüye düzgünleştirme filtresi etkisi (Gonzales ve Woods, 2002)

2.2.5 Özellik Çıkartma

Özellik çıkartma görüntü tanımda önemlidir. Görüntü sınıfları arasında ayrımı gerçekleştirmek için görüntü özelliklerinin çıkarılması gerekir. Günümüzde, çok başarılı sonuçlar veren görüntü sınıflandırıcı türleri mevcuttur. Özellik çıkartmanın ana sebepleri:

1. Ölçüm veya görüntü uzayından daha küçük boyuta dönüşmeyi sağlamaktır. Bu sınıflandırıcının küçük hatalar ile eğitimi ve karar aşamasının daha kısa sürede gerçekleşmesi demektir.

2. Boyut olarak daha düşük olan özellik uzayını sınıflandırıcının daha az parametre ile öğrenmesini mümkün hale getirecektir. Bunun yararı görüntü uzayı ile karar uzayı arasındaki dönüşüm aşamasının daha kısa sürede gerçekleşmesidir.

3. Durağan olmayan zaman serilerinde olduğu gibi karmaşık görüntülerin tanımlayıcı karakteristiklerini bulabilmek için özellik çıkartımı şarttır. Böylece karar aşamasının güvenilirliği artacaktır.

4. Görüntü sınıflandırma sisteminin, sistem içi veya dışındaki kontrolsüz girişimlerden etkilenmemesini sağlayacak bir özellik çıkarımı kararlı bir yapının oluşmasında etken olacaktır. Bu tür kararlı özellikler, sınıflandırıcının genelleme ve ayırma yeteneğinin yüksek olmasında önemlidirler.

Görüntü özelliklerini belirlemede ana işlem verilen esas örüntüden en iyi özellikleri seçmektir (Türkoğlu, 2003). Seçilen bu özellikler vektöre dönüştürülüp yapay sinir ağında kullanılabilir.

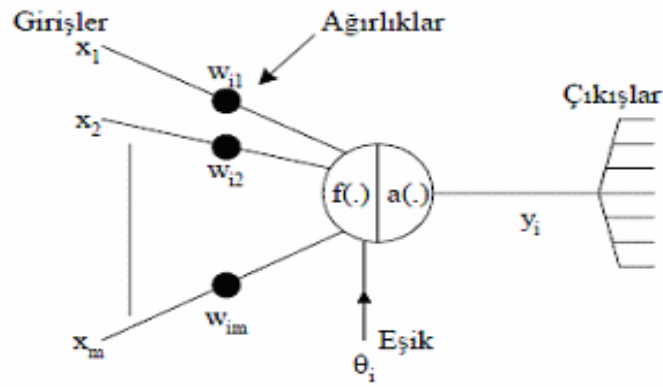


Şekil 2.5 Görüntü ve özellik vektörü

2.3 Yapay Sinir Ağları

Yapay zeka, doğal canlılarda görüldüğünde zeka belirtisi olarak algılanan yetenekleri analiz ederek bunların benzeri yapay yönergeleri geliştirmeye çalışmak olarak tanımlanabilir ancak yapay zeka'nın tam bir tanımını yapmak bu kadar kolay değildir. Çünkü yapay zeka çok kapsamlı bir konu olup pek çok alt alan içermektedir ve kendi alt alanı olarak düşünülemeyecek pek çok alanla da girift bir biçimde ilintilidir. Yapay zeka, gerek gelişim süreci sırasında farklı uzmanlık alanlarının ortaya çıkmasından, gerekse insan zekası çözümlenirken geliştirilecek yaklaşımların birden fazla bilimsel disiplini ilgilendirmesinden ötürü çok sayıda alt alana ayrılmıştır. Bu alt alanların temel hedefleri net olarak belirtilebilse dahi bazı alanların kapsamı/etki alanları arasındaki sınırlar çok kesin değildir. Yapay zeka bilimi içerisinde yer alan alt alanlardan birisi yapay sinir ağlarıdır (www.yapay-zeka.org, bt).

Yapay sinir ağıları biyolojik nöron hücresinin yapısı ve öğrenme karakteristiklerinden esinlenerek geliştirilmiş bir hesaplama sistemi ve görüntü tanımda çok kuvvetli sınıflandırıcılardır. Yapay sinir ağıları, özellikle bilgisayar teknolojisinin gelişmesi ile mühendislik sahasında çok geniş bir uygulama alanı bulmuştur. Yapay sinir ağıları birbirine bağlanan çok fazla sayıdaki işlem elemanlarından oluşur. Bağlantı ağırlıkları ile bilgiyi tutarlar. Bir işlem elemanı giriş uyarılarına dinamik olarak tepki verebilir ve tepki tamamen yerel bilgilere bağlıdır (ilgili işlem elemanını etkileyen bağlantılar ve bağlantı ağırlıkları yoluyla gelen giriş sinyali). Giriş verisi ile ayarlanan bağlantı ağırlıkları sayesinde öğrenme, hatırlama ve genelleme yeteneklerine sahiptir (Türkoğlu, 2003).



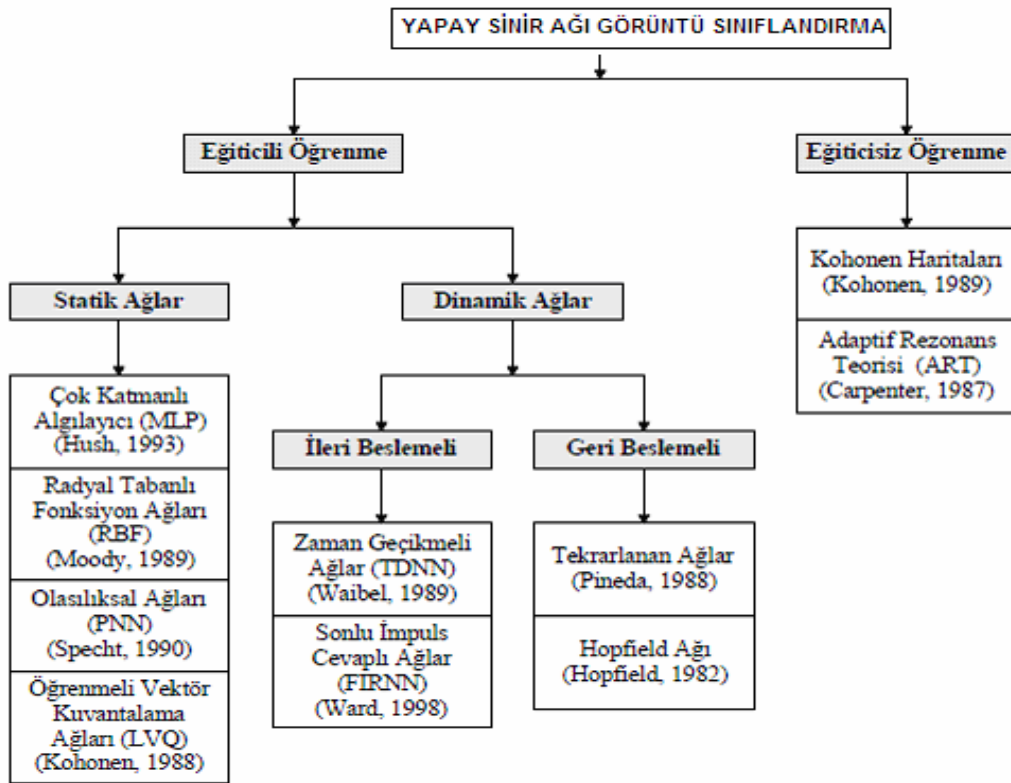
Şekil 2.6 Yapay sinir ağı nöron hücresi matematiksel modeli
(Türkoğlu,2003)

$$y(t+1) = a \left(\sum_{j=1}^m w_{ij} x_j(t) - \theta_i \right) \quad (2.1)$$

Burada $a(.)$ etkinleştirme fonksiyonu, θ_i ise i . işlem elemanının eşik değeridir. İşlem elemanlarının bilgi işlemleri iki kısımdan oluşur. Bunlar giriş ve çıkıştır. Bir işlem elemanı dışardan almış olduğu x_j giriş bilgilerini bağlı bulundukları w_{ij} ağırlıkları üzerinden birleştirerek bir net değeri üretir. i . işlem elemanının net değeri aşağıdaki denklem ile hesaplanır (Türkoğlu, 2003).

$$f_i = \text{net}_i = \sum_{j=1}^m w_{ij} x_j - \theta_i \quad (2.2)$$

Her bir işlem elemanı ikinci adımda, net değerini bir $a(.)$ etkinleştirme fonksiyonundan geçirerek çıkış değerini bulur. Etkinleştirme fonksiyonları işlem elemanlarının çok geniş aralıktaki çıkışını belli aralıklara çekerler ve böylece her bir işlem elemanının tepkisi yumuşak olur ve bağlantı ağırlıklarının değişimlerinin de daha küçük değerlerde olması sağlanır. Dolayısıyla yapay sinir ağının eğitiminde, hata değişiminin ıraksaması engellenerek kararlılığa ulaşmasına yardımcı olunur. Yaygın olarak kullanılan bazı etkinleştirme fonksiyonları, birim basamak, lineer, hipertanjant, tek ve çift yönlü sigmoid'dir (Türkoğlu, 2003).

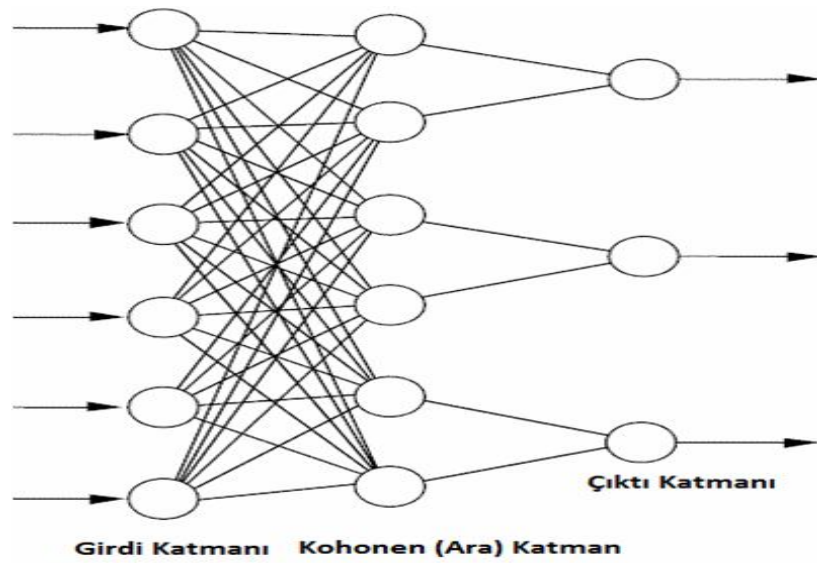


Şekil 2.7 Görüntü sınıflandırmada kullanılan yapay sinir ağları (Türkoğlu, 2003).

Bu tablodan kurguladığımız sistem için eğitici öğrenme sınıfında statik ağlardan öğrenmeli vektör kuvantalama ağı (lvq) seçilmiştir.

Lvq, çıkış noktası n boyutlu bir vektörü elde bulunan hazır vektör seti ile ifade edebilmektir. Bu hazır vektör setlerine referans vektörleri denmektedir. Lvq ağı destekleyici öğrenme türünden ağlardır. Destekleyici öğrenmede sisteme sadece girdiler verilir. Bu girdilere karşılık sistem bir çıktı üretecektir. Üretilen bu çıktıya bakılıp doğru yada yanlış sinyal üretilecek olup bu sinyale göre vektörler yeniden güncellenecektir (Adak, bt).

Lvq ağ yapısına bakacak olursak şekildeki gibi bir yapı görürüz.

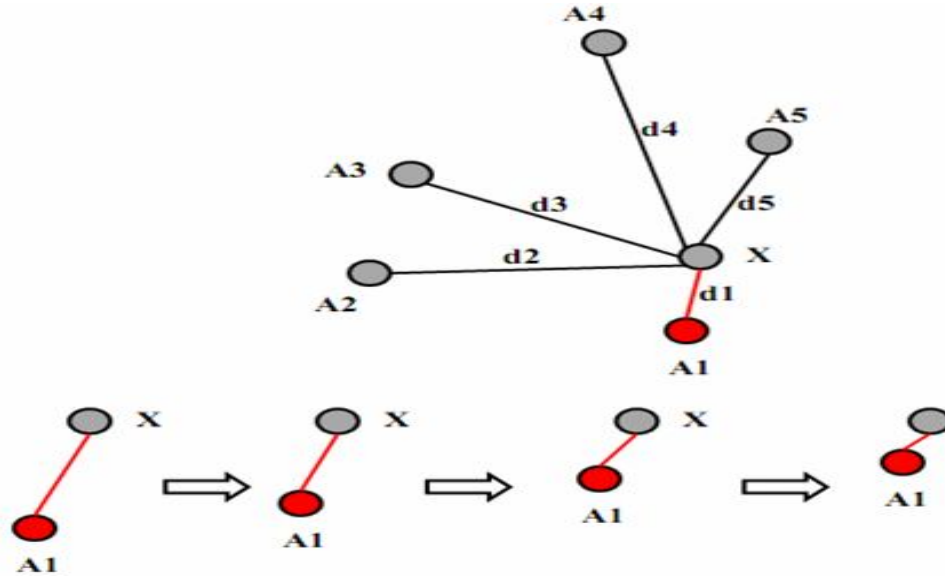


Şekil 2.8 Lvq ağ yapısı (Adak, bt)

Ağdaki çıktılarından sadece biri 1 diğerleri 0 değerini alır. Bu da şu demektir. Sadece bir kazanan vardır. Eğer kazanan eleman doğru sınıftaysa kazanan vektör referans vektörlerine daha çok yaklaştırılır ki tekrar aynı örnek ağa sunulursa kazanma ihtimali artsın yok eğer kazanan vektör yanlış sınıftaysa bu sefer de tam tersi referans vektöründen uzaklaştırılır. Dolayısıyla aynı örnek tekrar sunulduğunda kazanma ihtimali azaltılır. Her bir örnek ağa sunulduğunda bu örnek ile diğer tüm referans vektörleri arasındaki uzaklığa bakılır ve uzaklığın en az olduğu referans vektörünün sınıfında olduğu sonucu üretilir. Uzaklığı bulmak için aşağıdaki formül (öklid bağıntısı) kullanılmaktadır (Adak, bt).

$$d = |A_i - X| = \sqrt{\sum_j (A_{ij} - X_j)^2} \quad (2.3)$$

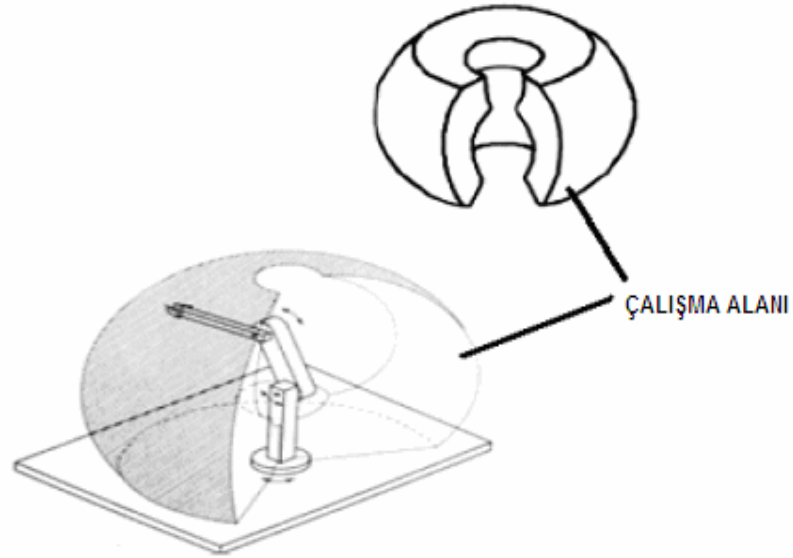
Bu formülde; d, girdi vektörü X ile referans vektörü A arasındaki mesafedir. j, girdi vektörünün j. değeridir.



Şekil 2.9 Lvq ağı öğrenme benzetmesi (Adak, bt)

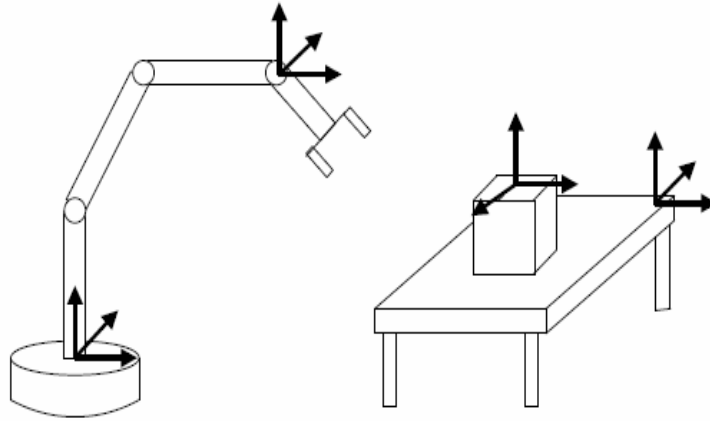
2.4 Robot Kol Kinematiği

Bir robot kolu, birbirine seri ya da paralel olarak bağlı eklemlerden oluşur. Robot kolunun her bir eklemine, belirli açılar verilerek, robotun uç noktasının gideceği konumu belirlemek için kullanılan problemlere ileri kinematik problemler denir. Bunun tam tersi olarak, robot kolunun uç noktasının, istenilen yere gitmesi için eklemlerin alması gereken açıların bulunmasında kullanılan problemlere de ters kinematik problemler denir. Robotun ileri kinematiği, birbirine bağlı her kol çifti için, uç noktanın konfigürasyonunu gösterir. Genellikle ters kinematik çözümler iki sınıfa ayrılabilir. Bunlar kapalı form çözümleri ve sayısal çözümlerdir. Kapalı form çözümleri, istenen bir uç nokta konfigürasyonu için hızlı ve etkin bir hesaplama sağlar. Sayısal çözümler için etkileşimli bir işlem sırası uygular. Bu işlemler, bir küme doğrusal olmayan aritmetik eşitliği, geometrik ve aritmetik özellikler kullanarak çözmek içindir (Bayrak, Sarıtaş, 2007).



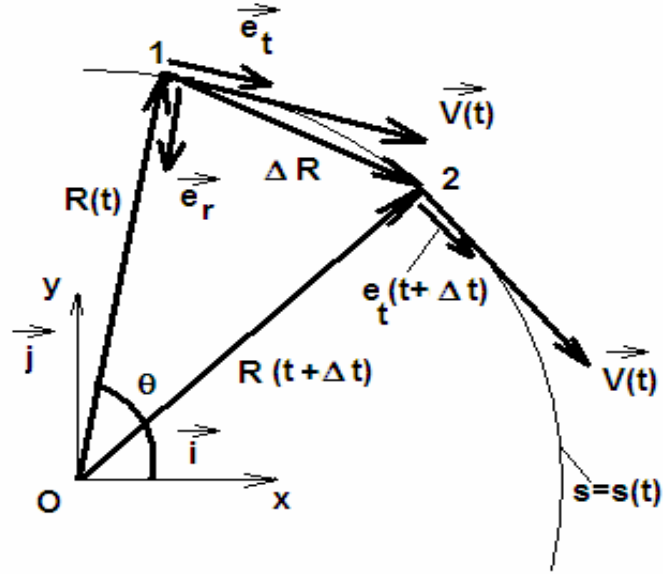
Şekil 2.10 Robot kol çalışma alanı (Kayserilioğlu, 2009)

Robot kol üç boyutlu bir çalışma uzayında hareket etmektedir. Bu durum, robotların ve çalışma uzayındaki nesnelerin birbirlerine göre koordinat ve yönelim tanımlamaları ihtiyacını doğurmaktadır. Bunu yapabilmek için de robotun her eklemine ve nesnelere bir koordinat sistemi yerleştirilir. Bütün bu koordinat sistemleri, evrensel çerçeve üzerinde bulunur ve herhangi bir noktanın konum veya yönelimi evrensel çerçeveye veya dünya çerçevesine göre yapılmaktadır.



Şekil 2.11 Koordinat sistemi yerleşimi (Bayrak, Sarıtaş, 2007).

2.4.1 Hız ve İvme Denklemleri



Şekil 2.12 Hız vektörel gösterimi (Uyar, 2009).

$$\vec{R}(t) = \vec{R}_x(t) + \vec{R}_y(t) \quad (2.4)$$

Kartezyen koordinatlarda gösterimi;

$$\vec{R}(t) = R_x(t) \cdot \vec{i} + R_y(t) \cdot \vec{j} \quad (2.5)$$

Polar koordinatlarda gösterimi;

$$\vec{R}(t) = R(t) \cdot e^{j\theta} = R(t) [\cos\theta + j\sin\theta] \quad (2.6)$$

burada $j = \sqrt{-1}$

Hız denklemleri gösterimi;

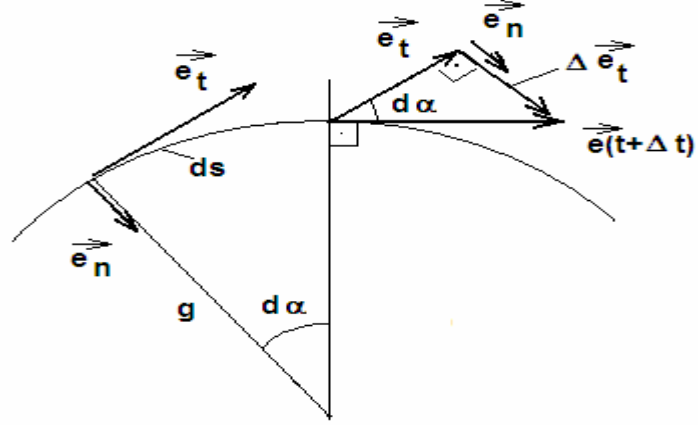
$$\vec{V}(t) = \frac{d\vec{R}(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\vec{R}(t + \Delta t) - \vec{R}(t)}{\Delta t} = \dot{\vec{R}}(t) \quad (2.7)$$

$$V = \left| \vec{V}(t) \right| = \dot{s} = \left| \frac{d\vec{R}(t)}{dt} \right| \quad (2.8)$$

Tangential birim vektör;

$$\vec{e}_t = \frac{\vec{V}(t)}{V(t)} \Rightarrow \vec{V}(t) = V(t) \cdot \vec{e}_t \quad (2.9)$$

İvme denklemleri;



Şekil 2.13 İvme vektörel gösterimi (Uyar, 2009).

$$\vec{a}(t) = \frac{d\vec{V}(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\vec{V}(t + \Delta t) - \vec{V}(t)}{\Delta t} = \dot{\vec{V}}(t) = \ddot{\vec{R}}(t) \quad (2.10)$$

$$\vec{V}(t) = V(t) \cdot \vec{e}_t \quad (2.11)$$

$$\vec{a}(t) = \frac{dV(t)}{dt} \cdot \vec{e}_t + V(t) \cdot \frac{d\vec{e}_t}{dt} \quad (2.12)$$

$$\dot{\vec{e}}_t = \frac{d\vec{e}_t}{dt} = \frac{d\vec{e}_t}{ds} \frac{ds}{dt} = \frac{d\vec{e}_t}{ds} V(t) = \frac{V^2}{g} \vec{e}_n \quad (2.13)$$

$$\Delta \vec{e}_t = \Delta \alpha \cdot \vec{e}_n \quad (2.14)$$

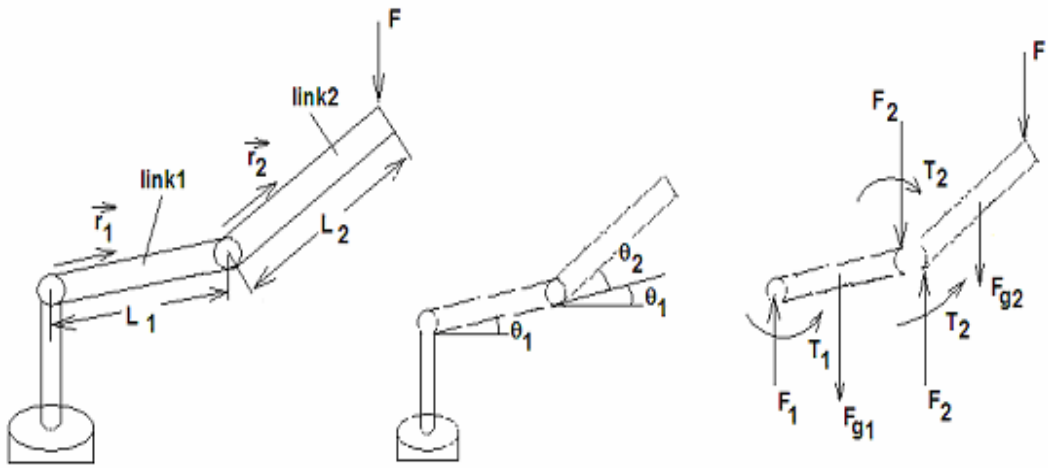
$$d\vec{e}_t = d\alpha \cdot \vec{e}_n \quad (2.15)$$

$$\frac{d\vec{e}_t}{ds} = \frac{d\alpha \cdot \vec{e}_n}{ds} = \frac{d\alpha \cdot \vec{e}_n}{g d\alpha} = \frac{1}{g} \vec{e}_n \quad (2.16)$$

$$\vec{a}(t) = \dot{\vec{V}}(t) \vec{e}_t + \frac{V^2}{g} \vec{e}_n = \vec{a}_t + \vec{a}_n \quad (2.17)$$

2.5 Robot Kol Dinamik Analizi

Bir manipulatörün doğru kontrolü her bir bağlantı noktasının tam kontrolünü gerektirir. Bağlantı noktalarının kontrolü, bağlantılar üzerinde rol oynayan eylemsizlik ve güçleri bilmeye bağlıdır (Uyar, 2009).



Şekil 2.14 Robot kola etki eden kuvvetlerin vektörel gösterimi (Uyar, 2009).

Güç eşitlikleri, F kuvvetini yenmek için T_1 ve T_2 bağlantı noktası torkları hesabı;

Link1 için;

$$\vec{F}_1 - \vec{F}_2 = 0 \quad (2.18)$$

Link2 için;

$$\vec{F}_2 - \vec{F} = 0 \quad (2.19)$$

$$\vec{F}_1 = \vec{F}_2 = \vec{F} \quad (2.20)$$

Link1 için;

$$\vec{T}_1 = \vec{T}_2 + \vec{r}_1 \cdot \vec{F} \quad (2.21)$$

Link2 için;

$$\vec{T}_2 = \vec{r}_2 \cdot \vec{F} \quad (2.22)$$

$$\vec{T}_1 = \left(\vec{r}_2 \cdot \vec{F} \right) + \left(\vec{r}_1 \cdot \vec{F} \right) = \left(\vec{r}_1 + \vec{r}_2 \right) \cdot \vec{F} \quad (2.23)$$

T_{G1} ve T_{G2} gravity torkları güç denge eşitliklerini kullanarak her bir link üzerinde gravity tarafından meydana gelen güçleri ilave etme ile hesaplanabilir. $\vec{F}_{G1}$ ve $\vec{F}_{G2}$ gravity güç vektörleri her bir linkin merkezinde rol aldığı farzedilebilir. Burada linkler m_1 ve m_2 kütlelerine sahiptir (Uyar, 2009).

Güç eşitlikleri;

$$F_{G2} = m_2 \cdot g = F_2 \quad (2.24)$$

$$F_{G1} = m_1 \cdot g \quad (2.25)$$

$$F_1 = F_2 + F_{G1} = F_{G2} + F_{G1} \quad (2.26)$$

Tork eşitlikleri;

$$\vec{r} = \vec{r}_1 + \vec{r}_2 \quad (2.27)$$

$$\vec{r}_2 = \vec{r} - \vec{r}_1 \quad (2.28)$$

$$\vec{T}_{G2} = \frac{\vec{r}_2}{2} \cdot \vec{F}_{G2} \quad (2.29)$$

$$\vec{T}_1 = \vec{T}_{G2} + \frac{\vec{r}_1}{2} \cdot \vec{F}_{G1} + \vec{r}_1 \cdot \vec{F}_{G2} \quad (2.30)$$

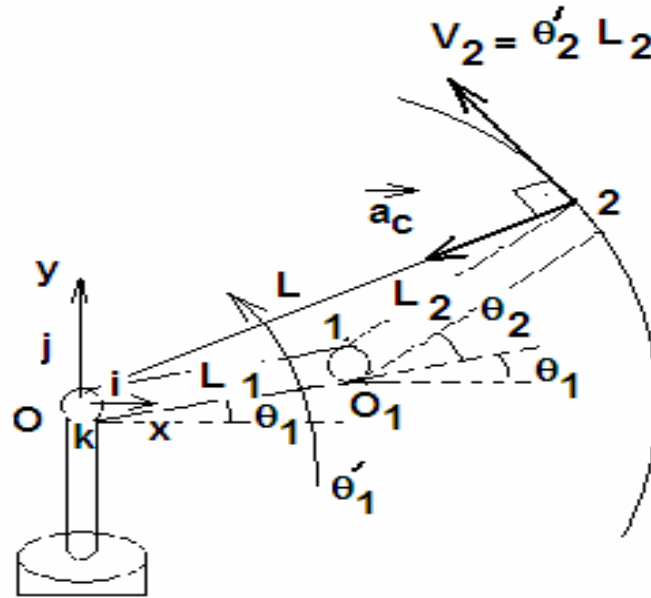
$$r_{1x} = r_1 \cdot \cos \theta_1 \quad (2.31)$$

$$r_{2x} = r_2 \cdot \cos(\theta_1 + \theta_2) \quad (2.32)$$

$$T_{G2} = r_{2x} \cdot F_{G2} = \frac{L_2 \cos(\theta_2 - \theta_1)}{2} \cdot m_2 \cdot g \quad (2.33)$$

$$T_{G1} = L_1 \cos \theta_1 \cdot g \cdot \left(\frac{m_1}{2} + m_2 \right) + \frac{L_2 \cdot m_2 \cdot g}{2} \cdot \cos(\theta_2 - \theta_1) \quad (2.34)$$

Coriolis ivmesi;



Şekil 2.15 Hız ve coriolis torku gösterimi (Uyar, 2009).

$$\vec{a}_c = -2 \vec{\omega} \cdot \vec{v} \quad (2.35)$$

$$v_{2O1} = \vec{\omega}_2 \cdot \vec{L}_2 = \omega_2 \left[L_2 \cdot \cos(\theta_1 + \theta_2) \vec{i} + L_2 \cdot \sin(\theta_1 + \theta_2) \vec{j} \right] \quad (2.36)$$

$$\left(\vec{a}_c \right)_O = -2 \cdot \omega_1 \cdot \vec{k} \{ v_2 \}_{O1} \quad (2.37)$$

$$\vec{v}_{2O1} = \omega_2 \cdot L_2 \cos(\theta_1 + \theta_2) \vec{i} + \omega_2 L_2 \sin(\theta_1 + \theta_2) \vec{j} \quad (2.38)$$

$$\left(\vec{a}_c\right)_0 = -2\omega_1 \vec{k} \left(\omega_2 L_2 \cos(\theta_1 + \theta_2) \vec{i} + \omega_2 L_2 \sin(\theta_1 + \theta_2) \vec{j} \right) \quad (2.39)$$

$$\left(\vec{a}_c\right)_0 = -2\omega_1 \omega_2 L_2 \cos(\theta_1 + \theta_2) \vec{j} + 2\omega_1 \omega_2 L_2 \sin(\theta_1 + \theta_2) \vec{i} \quad (2.40)$$

Coriolis ivmesi genelde uç noktayı O sabit orjinine birleştiren $\vec{R} = \vec{L}$ vektörü doğrultusunda ve uçtan O orjinine doğru yöndedir (Uyar, 2009).

Coriolis torku, aşağıdaki eşitlik ile hesaplanır.

$$\vec{T}_O = m_2 \cdot \vec{a}_c \cdot \vec{L} \quad (2.41)$$

BÖLÜM ÜÇ

DENEYSEL ÇALIŞMA

3.1 Genel Yapı

Görüntü işleme teknikleri kullanılarak endüstriyel uygulama, taşıyıcı bant üzerinden gelen karışık malzemelerin birbirinden ayırt edilerek ayrı sınıflandırılması, hatalı veya bilinmeyen malzemelerin tespit edilmesi için kullanılan ve tespit edilen malzemelerin yerlerine konmasını sağlayan mekatronik bir sistemdir. Sistemde kullanılan malzemeler kolay bulunabilen ve kolay uygulanabilen bir yapıdadır. Kullanılan malzeme ve özellikleri aşağıda verilmiştir.

- Malzeme taşımada kullanılan ve maksimum hızı 10 cm/sn olan konveyor bant ve mekanik gövde
- 1/4 CMOS 640x480 piksel görüntü sensörüne, f=4,9mm lense, USB 2.0 bağlantılı, otomatik odaklamalı, 30fps@320x240-160x120 ve 15fps@640x480-800x600 çerçeve oranına, 58 derece görüş açısına sahip pc kamerası
- 4 adet silikonlu led lamba
- 12 volt 1500 rpm redüktörlü DC motor
- 2 adet reflektörlü fotoelektrik hareket sensörleri (VL180-2P41131 SICK)
- Pic Basic Pro ile programlanan 16F877 mikroişlemci ile yapılan DC motor pwm sürücü ve haberleşme elektronik kartı
- 2.26 GHz 2 Cpu 2GB Ram özelliklerinde bilgisayar

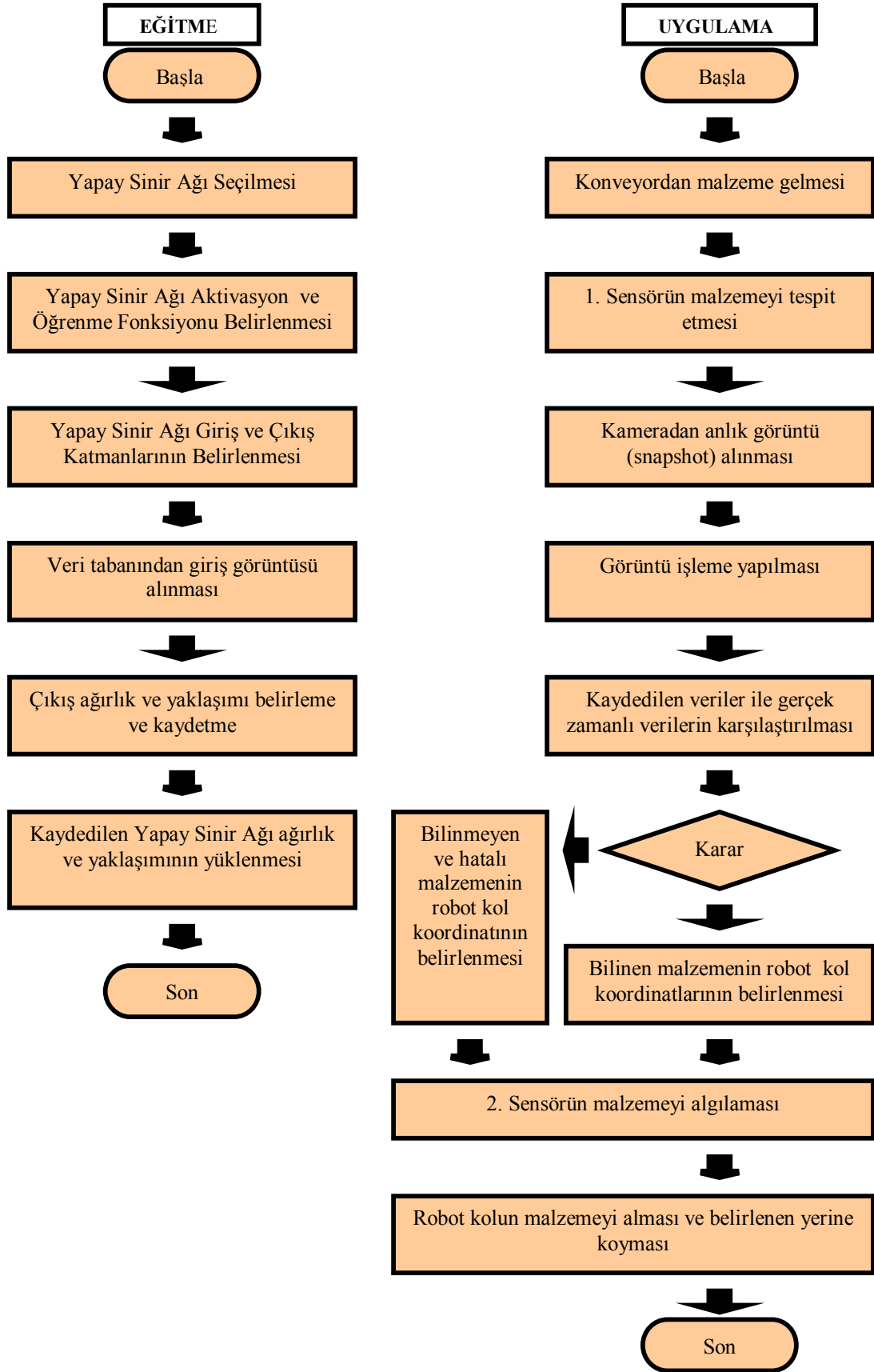
- Matlab programı ile hazırlanan nesne tanıma algoritması ve kullanıcı arayüzü
- Parallax firmasının ürettiği Parallax Servo Controller (#28823) ile yapılan robot kol servo motorlarını sürme ve haberleşme devresi
- Robot kol manipülatörü

Bu proje aşağıdaki blok şemada verilen bileşenlerden oluşmaktadır. Kurulan sistem ile tanınması istenen malzemeler önce eğitme algoritması ve yapay sinir ağı kullanılarak sisteme öğretilir. Öğrenme prosesi boyunca sistem verilen tüm öznitelik verilerini kullanır ve öğrenme veritabanını kaydeder.

Gerçek zaman uygulaması algoritmasında konveyör üzerinden gelen malzeme birinci hareket sensörü tarafından algılandığında kamera tarafından anlık görüntü alınır. Alınan görüntü, görüntü işleme algoritması ile işlenerek görüntünün öznitelikleri çıkartılır. Çıkartılan öznitelik ile veritabanına kaydedilen öznitelikleri yapay sinir ağı karşılaştırarak bir sonuca ulaşır.

Ulaşılan sonuca göre robot kol pozisyon bilgisi robot kol sürücü kartına gelir ve robot kol malzemeyi ikinci sensör gördükten sonra alarak belirlenen yerine bırakarak başlangıç pozisyonuna geri döner. Malzeme yeri, dört adet öğretilen malzeme için ve bir adet de bilinmeyen ve hatalı malzeme için toplam beş adet hazırlanmıştır. Konveyör bant hızı hız sınırları içinde kalmak şartı ile istenilen hıza kullanılan mikroişlemci tarafından oluşturulan pwm (pulse with modulation) sinyali ve kart üzerindeki potansiyometre ile ayarlanabilir.

Sistem blok şeması aşağıda verilmiştir.



Şekil 3.1 Sistem blok şeması

3.1.1 Nesne Tanıma Programı ve Kullanıcı Arayüzü

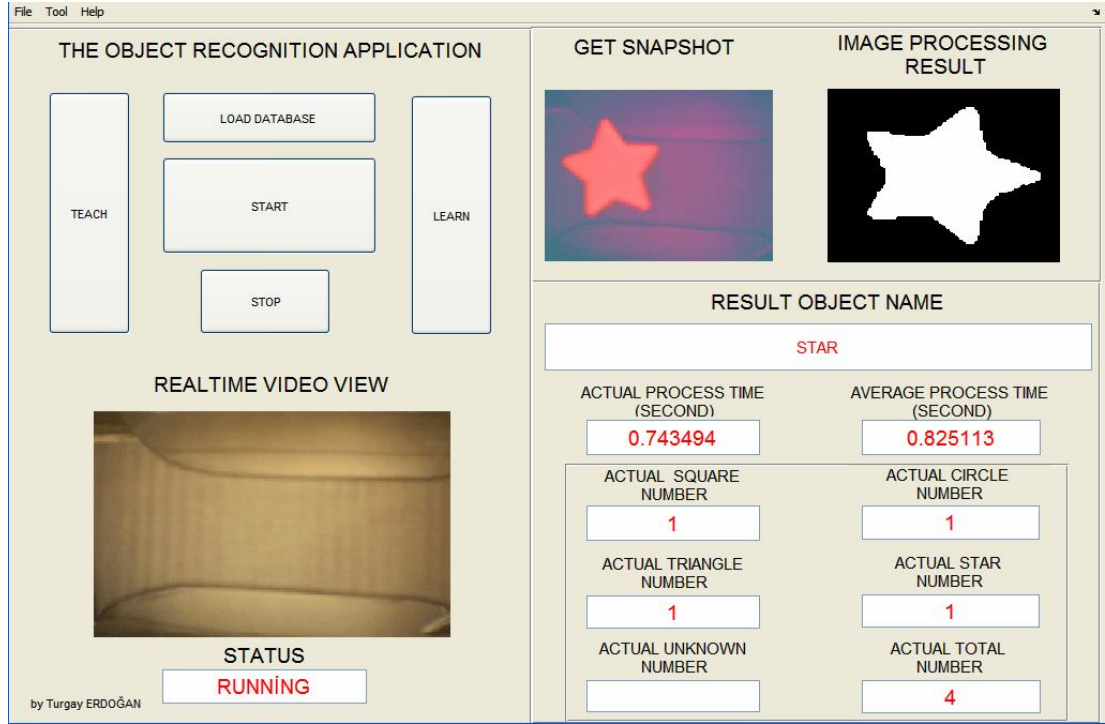
Sistemin program algoritması, Matlab programı yardımı ile hazırlanmıştır. Program, kameradan görüntü alınması, görüntünün işlenerek özniteliklerinin çıkartılması, özniteliklerin YSA ve koşul algoritması ile sınıflandırılması, veritabanı yükleme, veritabanı hazırlama, model öğretme ve YSA öğrenme ve karar verme algoritmalarını kapsamaktadır.

Sistemin kullanıcı arayüzü, sistemin işlevlerini, durum takibini kullanıcıya doğrudan sunan parçasıdır. Bu arayüzü kullanarak bir kullanıcı, sisteme yeni model öğretebilir, veritabanı değiştiğinde sistemin yeni veritabanını öğrenmesini sağlayabilir, en son öğretilen veritabanının özelliklerini yükleyebilir, programı çalıştırabilir, ekrandan mevcut durumu takip edebilir ve programı durdurabilir.

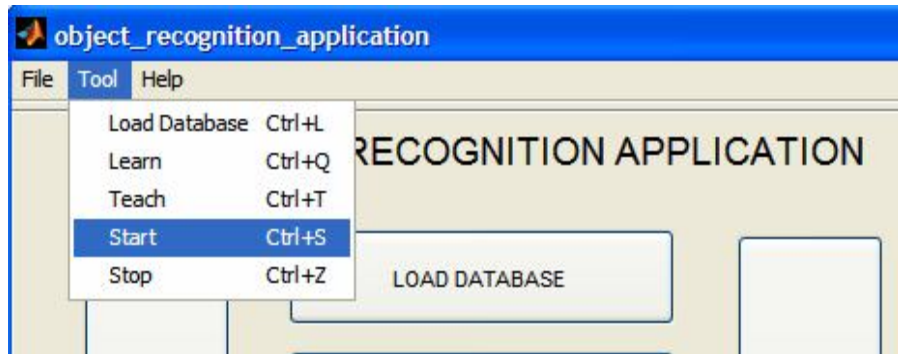
Arayüz üzerinde üç adet görüntü ekranı vardır. Birinci ekran gerçek zamanlı konveyör bant durumunu göstermektedir. İkinci ekran birinci sensör gördükten sonra kamera tarafından alınan anlık görüntüyü (snapshot) göstermektedir. Üçüncü ekran görüntü işleme son durumunu göstermektedir.

Arayüz üzerinde ayrıca tanınması istenen her malzemenin ve bilinmeyen malzemelerin sayısını belirleyen sayaç ve toplam malzeme sayısını belirten sayaç bulunmaktadır.

Arayüz üzerinde işlem zamanını izlemek için zaman sayacı ve ortalama zaman sayacı da bulunmaktadır.



Şekil 3.2 Kullanıcı arayüz programı



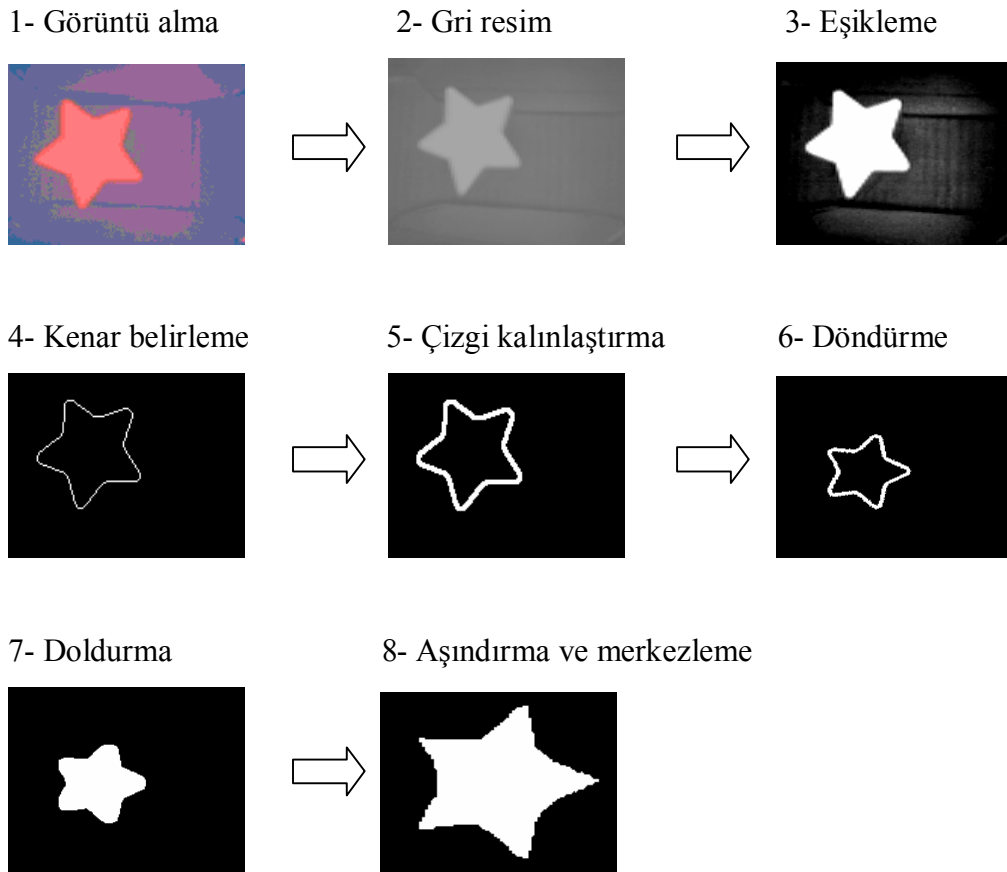
Şekil 3.3 Kullanıcı arayüz programı menüleri

3.1.1.1 Görüntü İşleme

Nesne tanıma ve ayrıştırma projesinde, taşıyıcı bant üzerinden gelen karışık malzemelerin birbirinden ayırt edilerek ayrı sınıflandırılması, hatalı veya bilinmeyen malzemelerin tespit edilmesi için kullanılan yazılım, Matlab programı ile hazırlanmıştır. Matlab programı ile pc kamerası üzerinden örnek alınan görüntü, görüntü işleme algoritmasında işlenmektedir.

Görüntü işleme ilk adımı olarak programda pc kamerasından anlık fotoğraf (snapshot) alınarak rgb (red gren blue) 160x120x3 formatındaki resim görüntü işleme algoritmasına aktarılmaktadır. Bu görüntü, görüntü işleme adımları kullanılarak tanıma için kullanılacak ve öznitelikleri alınacak istenilen şekle getirilir.

Aşağıda görüntü işleme adımları gösterilmiştir.



Görüntü işleme son adımı olarak programda son görüntü kaydedilir. Bu aşamaya kadar olan bölüm öğretim ve tanıma algoritmalarının ortak işlemidir.

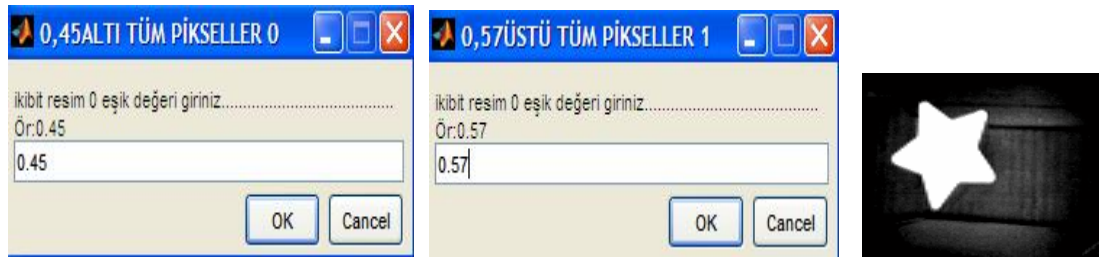
3.1.1.2 Teach Butonu, Tool Menüsü Teach Butonu ve İşlevi

Öğretim işleminde tanıtılacak malzemenin, algorithmada oluşan tüm görüntüleri son görüntü işleme adımıyla kaydedilerek veritabanında kullanılabilir. Ayrıca

kullanıcı arayüzünde TEACH butonu veya menüsü kullanılarak da veritabanı oluşturulabilir. Kullanıcı arayüzü işlem adımları aşağıdaki gibi uygulanır.

Öğretme programı başlatıldığında anlık kamera görüntüsü alınır ve bu renkli görüntüyü ikili görüntüye çevirmek için ekran üzerine gelen iletişim kutusuna görüntü işleme adımlarına göre değerler girilir. Değerler girildiğinde kullanıcı arayüzü üzerinde bulunan İMAGE PROCESSİNG RESULT bölümünde görüntünün son hali görülebilir. Aşağıda görüldüğü gibi diyalog kutuları ve İMAGE PROCESSİNG RESULT bölümü ve işlem adımları takip edilerek öğretme tamamlanır.

1- Eşikleme değerleri diyalog kutularına girilir.



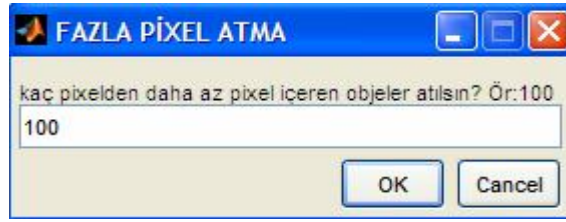
Şekil 3.4 Eşikleme değerleri ve eşikleme uygulanan görüntü

2- Kenar belirleme filtresi değeri diyalog kutusuna girilir.



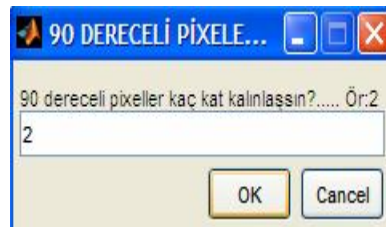
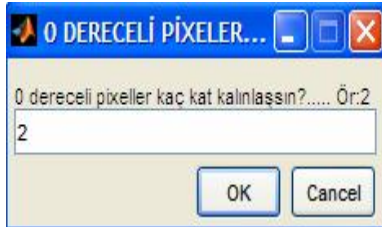
Şekil 3.5 Kenar belirleme filtresi ve kenar belirleme uygulanan görüntü

3- Belirli sayıdaki pikselleri yoketmek için sınır değeri girilir.



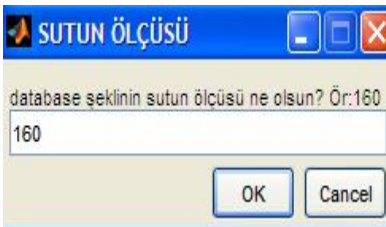
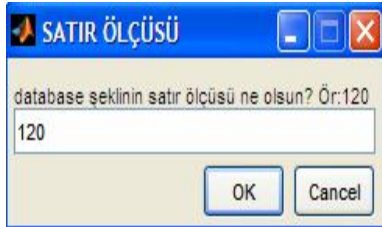
Şekil 3.6 Fazla pikselleri yoketme ve piksel yoketme uygulanan görüntü

4- Sıfır dereceden ve doksan dereceden pikselleri kalınlaştırma değerleri girilir.



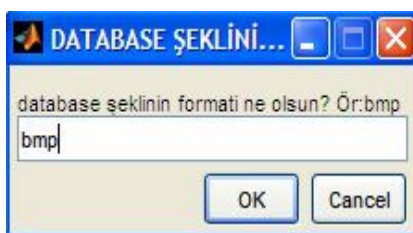
Şekil 3.7 Pikselleri kalınlaştırma ve kalınlaştırma uygulanan görüntü

5- Veritabanına kaydedilecek olan görüntü boyutları girilir.



Şekil 3.8 Veritabanı için görüntü boyutları girilmesi ve boyutların uygulandığı görüntü

6- Kaydedilecek görüntü için isim ve adres girilir ve işlem sonlanır.

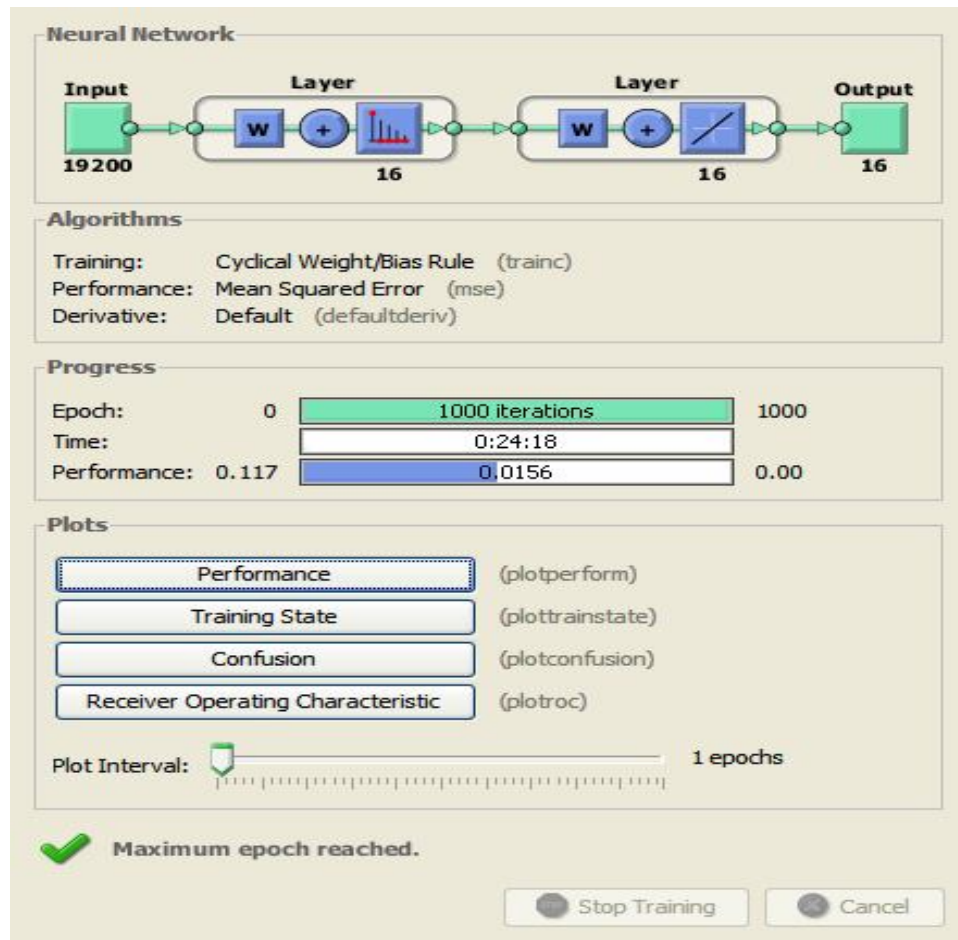


Şekil 3.9 Veritabanı için görüntü isim ve adresinin girilmesi ve uygulanan görüntü

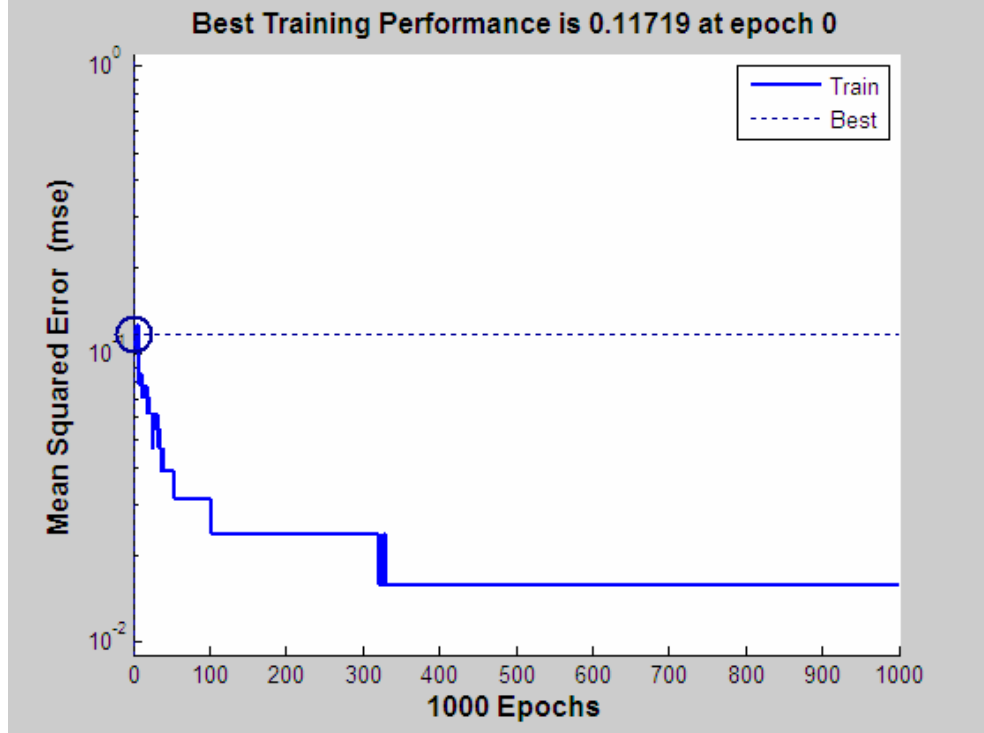
3.1.1.3 Learn Butonu, Tool Menüsü Learn Butonu ve İşlevi

Görüntü işleme aşamasından sonra görüntü yapay sinir ağına verilir. Programda öğrenme için 1988 yılında Kohonen tarafından geliştirilen eğitici öğrenme yapan statik ağlardan LVQ (Öğrenmeli Vektör Kuantalama ağları – Learning Vector Quantization) yapay sinir ağı kullanılmıştır. Yapay sinir ağı öğrenme fonksiyonu trainc seçilmiştir.

Trainc fonksiyonu, bir ağı girişin her bir temsilinden sonra artımlı güncellemeler ile ağırlık ve yaklaşım öğrenme kuralları ile eğitir. Bu fonksiyon ile girişler periyodik sırada temsil edilir. Öğretme yapılacak nesnelerin görüntü işleme ile oluşturulan veritabanı kullanılarak öğrenme işlemi tamamlanır.

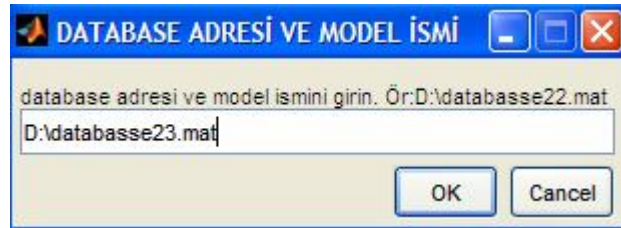


Şekil 3.10 Yapay sinir ağı öğrenme ekranı



Şekil 3.11 Yapay sinir ağı öğrenme performansı

Öğrenme işlemi tamamlandıktan sonra ekrana bir diyalog kutusu çıkar. Bu diyalog kutusuna yapay sinir ağının öğrendiği verileri tuttuğu dosya adresini ve ismini yazarak işlem sonlandırılır.



Şekil 3.12 Öğrenme işlemi adres ve isim girme

3.1.1.4 Start Butonu, Tool Menüsü Start Butonu ve İşlevi

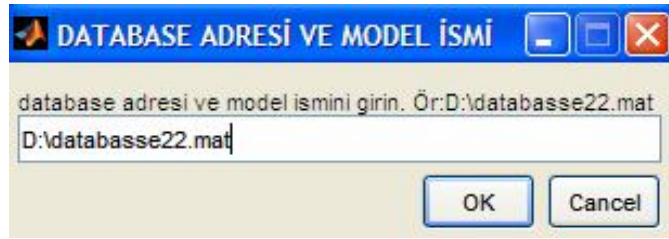
Start butonu, programı robot kol işlemsiz çalıştırır. Tool menüsü içinden açılan Start butonu, programı robot kol ile çalıştırır. Çalışmanın başladığı STATUS durum çubuğu üzerinden takip edilebilir.

3.1.1.5 Stop Butonu, Tool Menüsü Stop Butonu ve İşlevi

Stop butonu, programı durdurur. Çalışmanın durduğu STATUS durum çubuğu üzerinden takip edilebilir.

3.1.1.6 Load Database Butonları ve İşlevi

Load Database butonu, daha önce yapay sinir ağı özellikleri ve görüntü veritabanı seçilerek öğrenilen özniteliklerin tutulduğu veritabanını seçmemizi ve yüklememizi sağlar. Ekrana gelen diyalog kutusuna yüklemek istediğimiz veritabanının adresini girerek veritabanını yükleyebiliriz.



Şekil 3.13 Öğrenme sonuç dosyası yükleme isim ve adresi

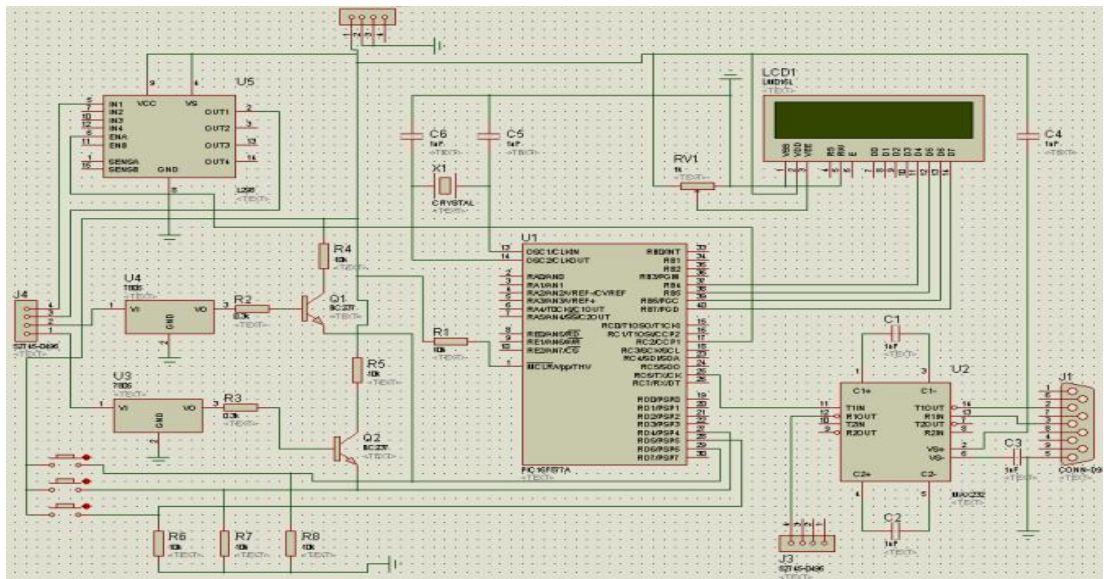
3.1.1.7 Tanıma

Veritabanında kaydedilen modeller ile öğretilen yapay sinir ağı girişine verilen tanınacak malzemenin görüntü işleme adımları ile işlenen görüntüsünü kullanarak yapay sinir ağı bir sonuç üretir. Sonucu, sistem kendi öğrendikleri modeller ile karşılaştırarak oluşturur.

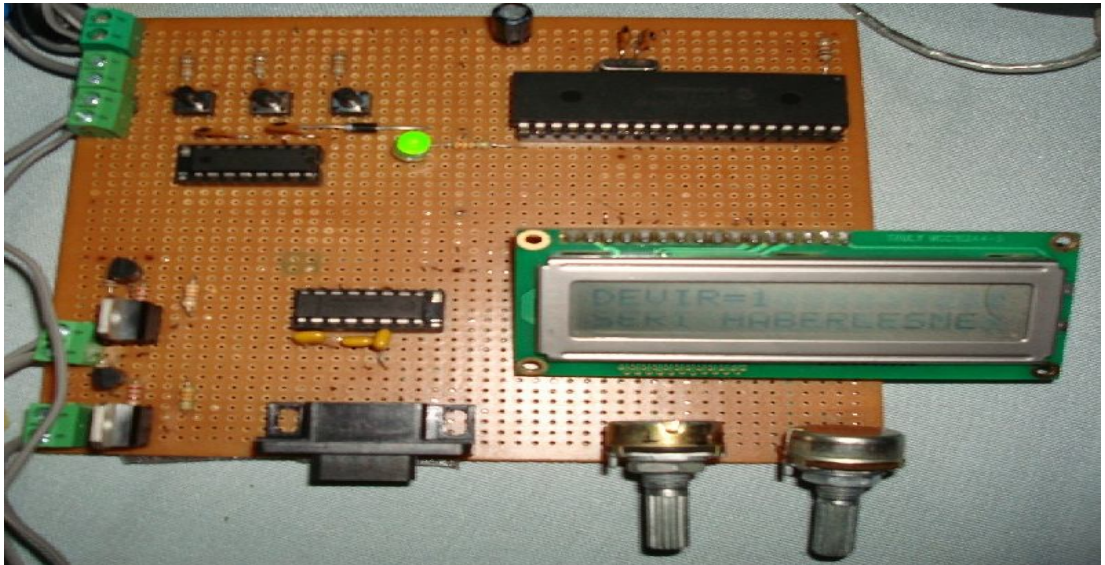
Tanıma algoritmasında şekil özellikleri ve yapay sinir ağı sonucunun tamamının tanınacak nesne özelliklerini karşılaması durumunda nesne tanıma algoritması bir sonuç gösterir. Aksi bir durum olur ise sonuç bilinmeyen malzeme şeklinde belirtilir.

3.1.2 Dc motor pwm sürücü ve haberleşme elektronik kartı

16F877A mikroişlemcisi kullanılarak oluşturulan pwm (pulse width modulation) sinyali ile dc motorun istenilen hızda çalıştırılması 10 K Ω potansiyometre ile ayarlanmaktadır. 16F877A mikroişlemcisi donanımsal seri haberleşme yapısına sahiptir. Konveyör mekanik gövdesi üzerine yerleştirilen reflektörlü sensör üzerinden gelen sinyal ile kameraya anlık görüntü (snapshot) aldırılan bilgi bilgisayara gönderilmektedir.



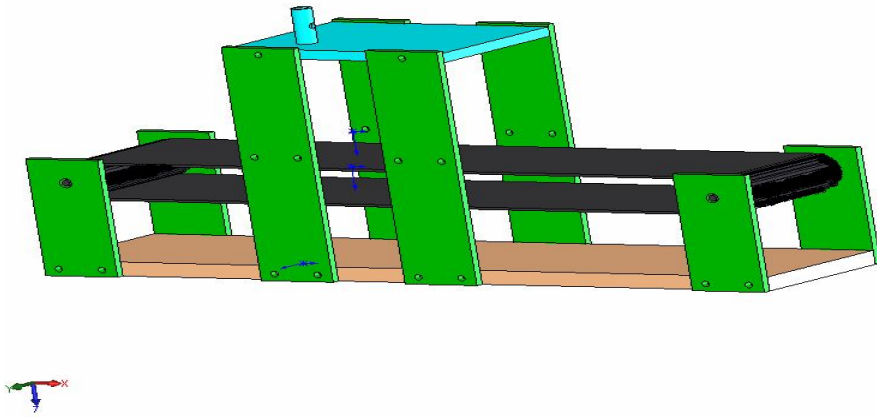
Şekil 3.14 Dc motor pwm sürücü ve haberleşme kartı devre şeması



Şekil 3.15 Dc motor pwm sürücü ve haberleşme kartı

3.1.3 Konveyör Bant ve Mekanik Gövde

Konveyör bant ve mekanik gövde, alüminyum karkas, dökme demirden yataklı dört adet rulman, DC motor, kauçuk bant ve derlin millerden meydana getirilmiştir. Bu parçalar oluşturulmadan önce Solidworks programı kullanılarak prototip çizim çalışması yapılmıştır. İlk model çalışma şekilindeki gibidir.

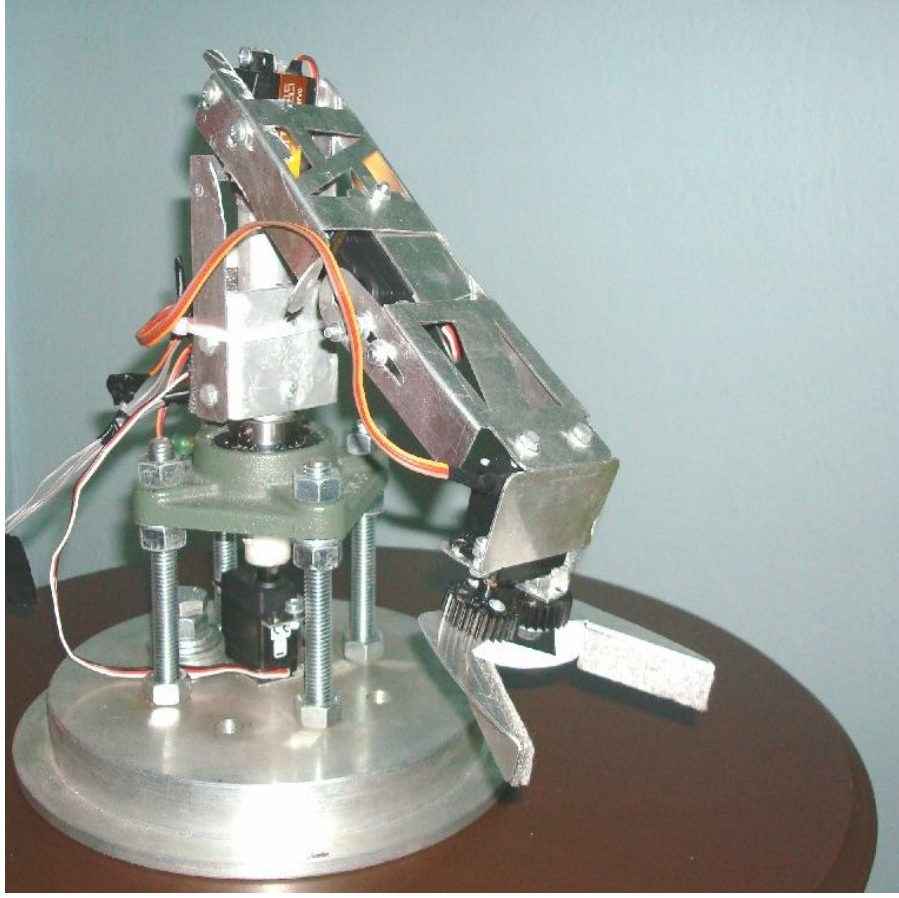


Şekil 3.16 Mekanik gövde ve konveyör bant model çizimi

Bu çalışma sonrası birkaç değişiklik ve ilavelerle konveyör bant, mekanik gövde ve robot kol son halini almıştır.



Şekil 3.17 Mekanik gövde son hali



Şekil 3.18 Robot kol



Şekil 3.19 Sistemin son hali

3.1.4 Robot Kol Manipulatörü Kinematik ve Dinamik Hesapları

Robot kol manipulatörü üç eksen üzerinde belli açılarla, sınırlı hareket edebilen RC tip servo motorlar ile alüminyum karkasdan oluşturulmuştur. Robot kol manipulatöründe dört adet RC tip servo motoru kullanılmıştır. Kullanılan motor kodları ve bu motorlara ait tork değerleri tablodaki gibidir.

Tablo 3.1 Robot kol eklemlerinde kullanılan motorlar

	1.eklem	2. eklem	3.eklem	Gripper
Motor kodu	Sm-s4303b	Sc-1256tg	Sm-s4461m	Hx12k
Motor torku	4.4 kg.cm	20 kg.cm	6.8 kg.cm	10 kg.cm

$$\text{Yörünge çevresi} = 2 \cdot \pi \cdot r = 2 \cdot 3,14 \cdot 0,27 = 1,7 \text{ m}$$

$$1 \text{ derece} = \frac{1,7}{360} \approx 4,5 \text{ mm}$$

$$v = 0,6 \text{ cm/s} \quad (\text{katalog değerinden})$$

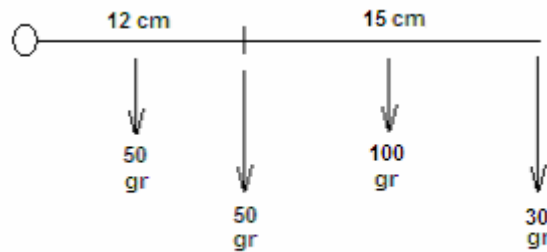
$$\omega = 60 \text{ derece/s} = \pi / 3 \text{ rad/s} \quad (\text{katalog değerinden})$$

$$\omega'_1 = \omega'_2 \approx 0,5 \text{ rad/s}^2 \quad (60 \text{ derecede ortalama değer})$$

Motor tork değerleri hesaplamaları aşağıdaki gibidir.

$$T = T_{\text{gravity}} + T_{\text{payload}} + T_{\text{Damping}} + T_{\text{inertia}} + T_{\text{coriolis}} \quad (3.1)$$

2.eklem için;



Şekil 3.20 İkinci eklem yük gösterimi

$$T_{\text{payload}} = 0,03.10.27 = 8,1\text{Ncm}$$

$$T_{G1} = L_1 \cos \theta_1 \cdot g \cdot \left(\frac{m_1}{2} + m_2 \right) + \frac{L_2 \cdot m_2 \cdot g}{2} \cdot \cos(\theta_2 - \theta_1)$$

$$T_{\text{gravity}} = 12.10.(0,05 + 0,130) + \frac{15.0,130.10}{2} = 21,6 + 9,75 = 31,35\text{Ncm}$$

$$\vec{T}_{\text{coriolis}} = m_2 \cdot \vec{a}_c \cdot \vec{L}$$

$$\vec{a}_c = 2 \vec{\omega} \cdot \vec{v} = 2.1,04.0,6 = 1,248 \frac{\text{cm}}{\text{s}^2}$$

$$\vec{T}_{\text{coriolis}} = m_2 \cdot \vec{a}_c \cdot \vec{L} = 0,02.1,248.27 = 0,673\text{Ncm}$$

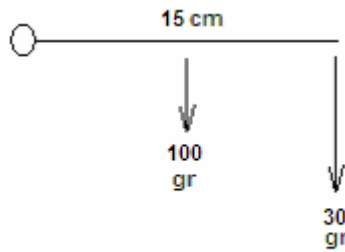
$$T_{\text{inertia}} = \exists_{01} \left(\frac{d\omega_1}{dt} + \frac{d\omega_2}{dt} \right) = m_2 L^2 \cdot (\omega_1' + \omega_2') = 0,130.27^2 \cdot 1 = 94,77\text{Ncm} \quad (3.2)$$

$$T = T_{\text{gravity}} + T_{\text{payload}} + T_{\text{damping}} + T_{\text{inertia}} + T_{\text{coriolis}}$$

$$T_{\text{damping}} = 1\text{Ncm varsayarak,}$$

$$T = 31,35 + 8,1 + 1 + 94,77 + 0,673 = 135,893\text{Ncm}$$

3.eklem için;



Şekil 3.21 Üçüncü eklem yük gösterimi

$$T_{\text{payload}} = 0,03.10.15 = 4,5\text{Ncm}$$

$$T_{G2} = r_{2x} \cdot F_{G2} = \frac{L_2 \cos(\theta_2 - \theta_1)}{2} \cdot m_2 \cdot g$$

$$T_{\text{gravity}} = \frac{15.0,130.10}{2} = 9,75 \text{ Ncm}$$

$$\vec{T}_{\text{coriolis}} = m_2 \cdot \vec{a}_c \cdot \vec{L}$$

$$\vec{a}_c = 2 \vec{\omega} \cdot \vec{v} = 2.1,04.0,6 = 1,248 \frac{\text{cm}}{\text{s}^2}$$

$$\vec{T}_{\text{coriolis}} = m_2 \cdot \vec{a}_c \cdot \vec{L} = 0,02.1,248.27 = 0,673 \text{ Ncm}$$

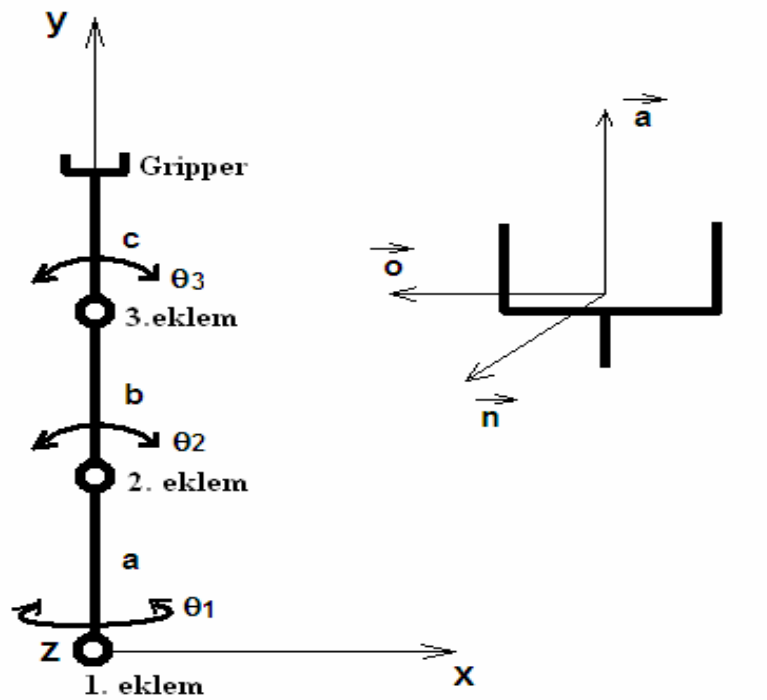
$$T_{\text{inertia}} = \exists_{02} \frac{d\omega_2}{dt} = m_2 L_2^2 \cdot \omega_2' = 0,130.15^2.0,5 = 14,625 \text{ Ncm}$$

$$T = T_{\text{gravity}} + T_{\text{payload}} + T_{\text{damping}} + T_{\text{inertia}} + T_{\text{coriolis}}$$

$$T_{\text{damping}} = 1 \text{ Ncm varsayarak,}$$

$$T = 9,75 + 4,5 + 1 + 14,625 + 0,673 = 30,548 \text{ Ncm}$$

Genel transformasyon matrisi;



Şekil 3.22 Dik konuma getirilmiş robot kol ve yerleştirilen koordinatlar

$$A = \text{Rot}(y, \Theta_1)$$

$$B = \text{Trans}(y, a)$$

$$C = \text{Rot}(x, \Theta_2)$$

$$D = \text{Trans}(y, b)$$

$$E = \text{Rot}(x, \Theta_3)$$

$$F = \text{Trans}(y, c)$$

x yönünde a kadar ilerleme

$$\text{Trans}(x, a) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

y yönünde b kadar ilerleme

$$\text{Trans}(y, b) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

z yönünde c kadar ilerleme

$$\text{Trans}(z, c) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$\cos \Theta = c\Theta$ ve $\sin \Theta = s\Theta$ yazarsak, x ekseninde Θ kadar dönme

$$\text{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

y ekseninde Θ kadar dönme

$$\text{Rot}(y, \theta) = \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

z ekseninde Θ kadar dönme

$$\text{Rot}(z, \theta) = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

$$\vec{T} = A * B * C * D * E * F \quad (3.9)$$

$$\vec{T} = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Burada; $\vec{a}$ yaklaşım vektörü, $\vec{o}$ yönelim vektörü, $\vec{n}$ normal vektörü, $P(x,y,z)$ uç noktanın pozisyonu, a birinci eklem uzunluğu, b ikinci eklem uzunluğu, c üçüncü eklem uzunluğu, Θ_1 birinci eklem dönme açısı, Θ_2 ikinci eklem dönme açısı, Θ_3 üçüncü eklem dönme açısı, $\vec{T}$ transformasyon matrisidir.

Robot kol sürücü kartı olarak, parallax firmasının ürettiği Parallax Servo Controller (#28823) ile yapılan robot kol servo motorlarını sürme ve haberleşme devresi kullanılmıştır.



Şekil 3.23 Parallax Servo Controller #28823
(Parallax, 2010)

3.1.5 PC Kamerası

PC kamerası 1/4 CMOS 640x480 piksel görüntü sensörüne, $f=4,9\text{mm}$ lense, USB 2.0 bağlantılı, otomatik odaklamalı, 30fps@320x240-160x120 ve 15fps@640x480-800x600 çerçeve oranına, 58 derece görüş açısına sahiptir. Fps (frame per second), saniyede çekebildiği fotoğraf sayısıdır.

Projede pc kamerası 30fps@160x120 özellikleri ile kullanılmıştır.



Şekil 3.24 A4 Tech pc kamerası (A4tech, 2010)

3.2 Deneysel Veriler

Öğrenme algoritması için farklı yapay sinir ağı öğrenme yöntemleri denenmiş ve sonuçta lvq yapay sinir ağı kullanılmıştır. Yapılan çalışma ve sonuç tablosu aşağıdadır.

Tablo 3.2 Yapay sinir ağları ve öğrenme fonksiyonları ile yapılan çalışma sonuçları

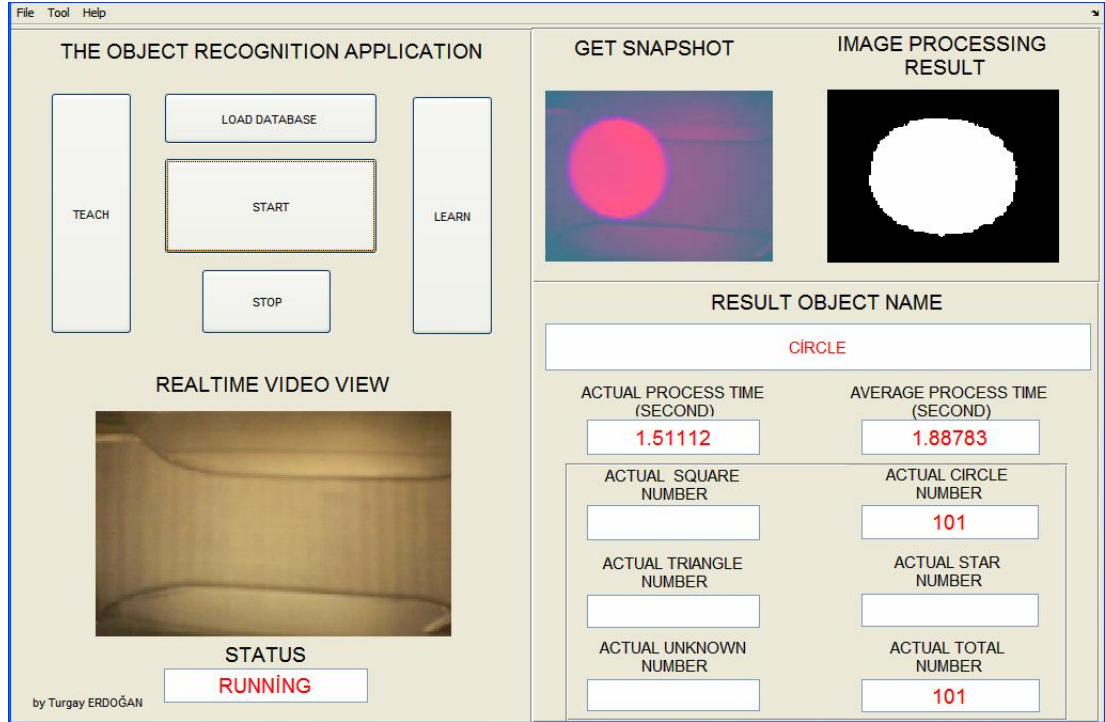
Yapay Sinir Ağı	Öğrenme Fonksiyon	Hız=2,5cm/s				Hız=10cm/s			
		Başarılı Tanıma Oranı				Başarılı Tanıma Oranı			
		Kare	Üçgen	Daire	Yıldız	Kare	Üçgen	Daire	Yıldız
New lvq	Traingd	%29	%54	%29	%54	%26	%52	%26	%51
	Traingdm	%29	%54	%29	%54	%26	%52	%26	%51
	Traingdx	%0	%0	%0	%0	%0	%0	%0	%0
	Trainoss	%0	%0	%0	%0	%0	%0	%0	%0
	Traincgp	%0	%0	%0	%0	%0	%0	%0	%0
	Traincgf	%0	%0	%0	%0	%0	%0	%0	%0
	Traincgb	%0	%0	%0	%0	%0	%0	%0	%0
	Trainscg	%0	%0	%0	%0	%0	%0	%0	%0
	Trainrp	%0	%0	%0	%0	%0	%0	%0	%0
	Trainbfg	Hata	Hata	Hata	Hata	Hata	Hata	Hata	Hata
	Trainbr	Hata	Hata	Hata	Hata	Hata	Hata	Hata	Hata
	Trainlm	Hata	Hata	Hata	Hata	Hata	Hata	Hata	Hata
	Trains	%99	%99	%99	%99	%99	%99	%99	%99
	Trainb	%99	%99	%99	%99	%99	%99	%99	%99
	Trainrv	%75	%75	%75	%75	%75	%75	%75	%75
	Trainrp	%0	%0	%0	%0	%0	%0	%0	%0
	Trainr	%99	%99	%99	%99	%99	%99	%99	%99
	Trainc	%99	%99	%99	%99	%99	%99	%99	%99
Newff	Trainc	%54	%51	%51	%55	%54	%51	%51	%55
	Trainr	%51	%51	%51	%51	%51	%51	%51	%51
	Trainb	%51	%51	%52	%51	%51	%54	%52	%50
	Trains	%66	%64	%55	%58	%66	%64	%55	%58
Newcf	Trains	%0	%0	%0	%0	%0	%0	%0	%0
	Trainb	%0	%0	%0	%0	%0	%0	%0	%0
	Trainr	%0	%0	%0	%0	%0	%0	%0	%0
	Trainc	%0	%0	%0	%0	%0	%0	%0	%0

Tablo 3.2devamı

New elm	Trainc	%0	%0	%0	%0	%0	%0	%0	%0
	Trainr	%0	%0	%0	%0	%0	%0	%0	%0
	Trainb	%0	%0	%0	%0	%0	%0	%0	%0
	Trains	%29	%34	%23	%24	%29	%34	%23	%24
Newp	Trains	Hata	Hata	Hata	Hata	Hata	Hata	Hata	Hata
	Trainr	%75	%75	%0	%75	%75	%75	%0	%75
	Trainc	%22	%26	%22	%22	%20	%21	%20	%19
	Trainb	%18	%22	%21	%22	%17	%22	%21	%22

Oluşturulan sistem ile tanıma başarısı olarak lvq yapay sinir ağı modeli, trainr, trainc, trainb, trains öğrenme fonksiyonları, 1000 iterasyon, 16 hücreli giriş katmanı, 16 hücreli gizli katman ve tek hücreli çıkış katmanı yapısı kullanılarak %98 -%100 arasında değişen tanıma oranı gerçekleştirilmiştir. 14 hücreli giriş katmanı ve 14 hücreli gizli katman yapısı ile yapılan denemelerde aynı ağ yapısı ve öğrenme fonksiyonu ile başarı oranının %80 -%85 arasında değiştiği denemelerde görülmüştür.

Sisteme daire, yıldız, üçgen ve kare olmak üzere dört adet malzeme öğretilmiştir. Programın malzemeleri tanıma hızları her yüz adetlik örnekleme çalışmalarında ortalama denemeler sonucu kare için 1,2 saniye, üçgen için 1,3 saniye, yıldız için 1,3 saniye, daire için 1,8 saniye olarak ölçülmüştür. Daire için sürenin diğerlerinden daha fazla olmasının sebebi piksel ve bit bazında değerlendirme yapan yapay sinir ağının daire vektörlerini diğer malzemelere benzetmeye daha yatkın olmasındandır. Eğri yapı piksel boyutunda incelendiğinde nokta karesel hücrecik yapısı görülür. Bu yapı da her malzemenin piksel yapısında olduğu için tam seçme yapıncaya kadar tüm olasılıkların değerlendirilmeye devam edilmesi süreyi uzatan sebep olmaktadır. Bu sürelerin değerlendirilmesi ile ulaşılan sonuç, maksimum 10 cm/s hızla hareket eden konveyör için malzemelerin tamamını sistemin algılaması için 18 cm de bir malzemenin bant üzerinden gelmesinin gerektiğidir. Daha az mesafe ile gelen malzemeler için sistem program algoritması tanıma işlemini yaparken bekleme konumuna geçmediği için bu kör zamanlarda sistem aradaki malzemeleri kaçırabilir. Sistemin dakikada malzeme belirleme adedi 33'tür. Saatteki belirlediği malzeme adedi 2000'dir. Sistem %99 başarılı tanıma oranı ile saatte 1980 adet doğru malzeme belirlemesi yapabilmektedir.



Şekil 3.25 Kullanıcı arayüzünde daire malzemesinin doğru tanınma sayısı ve tanınma zamanları

Robot kol kullanarak yerine koyma işlemi ile birlikte sistemin bir malzemeyi işleme süresi kare için 10 saniye, daire için 11 saniye, üçgen için 10 saniye, yıldız için 11 saniyedir. Robot kol kullanıldığında malzemelerin arasında en az 110 cm mesafe olması malzemelerin tamamını sistemin algılaması ve malzemeleri yerine koyması için gereklidir. Bu şekilde sistem dakikada 5 adet, saatte 300 adet malzeme işleme kapasitesine sahiptir.

BÖLÜM DÖRT

SONUÇ VE ÖNERİLER

Bu çalışmada, hareketli konveyör üzerinde kamera görüntüsü ile gerçek zamanlı nesne tanıma ve nesneleri yerine koyma uygulamasında yapay sinir ağı kullanılarak gerçek zamanlı tanıma uygulaması geliştirilmiş ve çeşitli öğrenme yöntemleri ile başarımı üzerine çalışmalar yapılmıştır.

Geliştirilen sistem üzerinde yapılan çalışmada, yazılımın önemli bir bileşen ve sistem başarımında önemli bir role sahip olduğu görülmüştür. Yazılım içinde kullanılan görüntü işleme, öğretme, öğrenme algoritmaları uygulamaya ne kadar uygun ve ne kadar başarılı yapılır ise tanıma algoritmasının da başarısını aynı oranda artırdığı gözlenmiştir. Öğretilen nesnelerin veritabanlarını öğrenen akıllı sistemin başarısı, tanıma başarısında önemli rol oynamaktadır.

Başarımda diğer önemli faktör de çevresel şartlardan etkilenmeyen kapalı bir yapı oluşturulmasıdır. Çevre şartlarından etkilenmeyen kapalı yapı, sistem veritabanı için oluşturulan örneklerle en yakın doğrulukta örnekler alınmasını sağlamak ve tanıma doğruluk oranını artırmaktadır. Çevre şartlarına açık durumda bulunan sistem denemelerinde tanıma başarı oranlarının düştüğü görülmüştür.

Sistemin başarısını etkileyen üçüncü önemli faktör aydınlatma ve kamera özellikleridir. Oluşturulan kapalı ortamın aydınlatılması nesne özelliklerini ortaya çıkartmak için görüntü işleme algoritmasına uygun şekilde seçilmesi sistemin doğru tanıma başarımını artırmaktadır. Farklı tip malzeme tanıma işlemlerinde aydınlatma özelliklerinin değiştirilebilir olması avantaj sağlamaktadır. Kullanılan aydınlatma şeklinin ve görüntü işleme algoritmasının sistemde en doğru şekli ile uygulanması hem başarı oranını artıracak hem de işlem süresini azaltan etki yapacaktır. Kullanılan kamera özelliklerinin artırılması ve endüstriyel tip kameralar tercih edilmesi sistemin başarı oranını artıracaktır.

Genel tanıma algoritmasında tanıma oranını artıran en önemli etkilere birini de konveyör üzerindeki kılavuzlama sistemi oluşturmaktadır. Malzemeler her seferinde aynı çizgi üzerinden geçirildikleri için normalizasyon algoritması işlemlerinin sonucunda daha az olasılık ortaya çıkmakta ve bu da tanıma oranını artırmaktadır. Ayrıca kılavuzlama ve normalizasyon işleminin iyi yapılması sonucu oluşacak olasılıkların daha az olması, veritabanında tutulması gereken örneklerin sayısını da önemli ölçüde azaltmaktadır. Neticede tanıma algoritması daha kısa olmakta ve tanıma hızı da artmaktadır.

Görüntü işleme ve nesne tanıma algoritmalarında her uygulama kendine özel detaylara sahiptir. Her farklı uygulamada bu detaylara göre adaptasyon yapılmaktadır ve bu da sistem başarımını doğrudan etkilemektedir. Seçilen algoritmanın yapılan işe en uygun olanı deneme yapılmadan net olarak görülememektedir ve netleştirilememektedir. Bu sebeple bu çalışmada da birkaç kez deneme sonrasında en iyisi yapılmaya çalışılmıştır.

Uygulaması yapılan bu sistem daha fazla geliştirilebilir, başarı oranı artırılabilir ve tanıma süreleri düşürülebilir yapıdadır. Veritabanına ve tanıma algoritmasına daha fazla örnek eklenerek sistem doğruluğu artırılabilir ancak bu işlemler sistemin öğrenme işlem hızını azaltacaktır. Daha güçlü bilgisayar sistemlerinin kullanımı, işlem süresini bir miktar kısaltacaktır.

Robot kol eklemlerinde kullanılan rc tip servo motorlar yerine daha hızlı ve güçlü olan endüstriyel tip servo motorlar kullanılması, sistemi kuvvetlendirecek ve hızlandıracaktır. Ayrıca robot tipinin, delta tip robot gibi daha hızlı robotlar ile değiştirilmesi de işlem süresini kısaltan bir etki yapacaktır. Robot gövdesi kullanılan yüke orantılı olarak güçlendirilir ve özellikle eklem yapıları için döner mafsallı yapılar kullanılır ise daha kararlı bir çalışma sağlanabilir.

Bu çalışma sonrası geliştirilen görüntü işleme, nesne tanıma ve robotik uygulamaları sistemi ve algoritmaları, gelecekte benzer veya değişik uygulamalar ile geliştirilmeye uygundur. Bu şekilde düzenlenen uygulamalar ile daha ekonomik

özümlel yaratılarak ekonomimize bir katkı saęlayarak toplumsal yararın arttırılması yolunda bir gelişme saęlanabilir.

KAYNAKLAR

- Adak, F. (b.t). *Lvq ağırları*. 06 Eylül 2011, <http://www.fatihadak.com/Default.aspx?makID=97>.
- Bayrak, A. ve Sarıtaş, M. (2007). *Beş eksenli bir robot kolu simülasyonu ve engel/hedef takibi*. 18 Mayıs 2011, http://www.emo.org.tr/ekler/91e044e04955c_ek.pdf.
- Bilişim teknolojileri, görüntü işleme cihazları*, (b.t). 06 Nisan 2011, http://megep.meb.gov.tr/mte_program_modul/modul_pdf/481BB0018.pdf.
- Cabrera, M. P., Juárez, I. L., Cabrera, R. R., Osorio, R. ve Gómez, H. (2010). Real time object recognition methodology. *2010 electronics, robotics and automotive mechanics conference*, 439-444.
- Cao, W., Lu, F. ve Wang, S. (2006). Priority ordered direction basis function neural networks and the application for object recognition. *Granular computing, 2006 ieee international conference*, 229-231.
- Chen, B. ve Hoberock, L.L. (1996). Machine vision fuzzy object recognition and inspection using a new fuzzy neural network. *Intelligent control, proceedings of the 1996 ieee international symposium*, 206-211.
- Çayırılı, M. (2006). *Yanma olayının modellenmesi ve görüntü işleme yoluyla yanma performansının optimizasyonu*. 10 Aralık 2011, <http://tez.sdu.edu.tr/Tezler/TF00909.pdf>.
- Declercq, F. ve Keyser, R.D. (1999). Object recognition and selective handling by a robot. *Journal of Intelligent and Robotic Systems*, 25, 121–132.

- Fuente, M. I., Echanobe, J., Campo, I., Susperregui, L. ve Murtua, I. (2010). Hardware implementation of a neural-network recognition module for visual servoing in a mobile robot. *Database and expert systems applications (DEXA), 2010 Workshop*, 226-232.
- Gonzalez, R.C. ve Woods, R.E. (2002). *Digital image processing se (2)*. New Jersey: Prentice-Hall International Inc.
- Johan, T. M. ve Prabuwno, A. S. (2011). Recognition of bolt and nut using artificial neural network. *Pattern analysis and intelligent robotics (ICPAIR), 2011 international conference, 1*, 165-170.
- Kayserilioğlu, E. (2009). *Otomasyon sistemleri ve otomatik bir paketleme makinesinin tasarımı*. 11 Temmuz 2011, <http://tez2.yok.gov.tr/>.
- Kulkarni, A. D., Yap, A. C. ve Byars, P. (1990). Neural network for invariant object recognition. *Applied computing, proceedings of the 1990 symposium*, 28-32.
- Ming, J. ve Zhenkang, S. (1992). A novel neural network for object recognition with blurred shapes. *Circuits and systems, ISCAS '92, proceedings, 1992 ieee international symposium, 6*, 2901-2904.
- Moreno, F., Alarcón, J., Salvador, R. ve Riesgo, T. (2009). Reconfigurable hardware architecture of a shape recognition system based on specialized tiny neural networks with online training. *Industrial electronics, ieee transactions, 56*, (8), 3253-3263.
- Neogy, A., Balakrishnan, S. N. ve Dagli C. H. (1992). Moving object recognition and guidance of robots using neural Networks. *Aerospace and electronics conference, NAECON 1992, proceedings of the ieee 1992 national, 3*, 883-886.

Onis, S., Garcia, C., Sanson, H. ve Dugelay, J.L. (2009). Object detection with a minimal set of examples using convolutional pca. *Multimedia signal processing, 2009, MMSP '09, ieee international workshop*, 1-4.

Pan, H. ve Xia, L. Z. (2008). Efficient object recognition using boundary representation and wavelet neural network. *Neural networks, ieee transactions*, 19, (12), 2132-2149.

Parallax Servo Controller USB (#28823) datasheet, (b.t). 06 Eylül 2011, www.parallax.com/.../28823-PSCusbRevB.

Patil P.M, Deshmukh, M., Bonde, P.V., Dhabe, P.S ve Sontakke, T.R. (2004). Fourier fuzzy neural network for clustering of visual objects based on their gross shape and its application to handwritten character recognition. *Neural networks, proceedings, 2004, ieee international joint conference*, 3, 2391-2396.

Syed, S.A., Zafar, M.F. ve Moeen, T. (2009). Detection and recognition of human in videos using adaptive method and neural net. *Soft computing and pattern recognition, 2009. SOCPAR '09. International conference*, 604-609.

Teknik bilgiler, (b.t). 09 Eylül 2011, <http://www.a4tech.com/product.asp?cid=77&scid=160&id=532>.

Trivedi, A.I. ve Chitaliya, N.G. (2010). *Feature extraction using wavelet-pca and neural network for application of object classification & face recognition. Computer engineering and applications (ICCEA), 2010 second international conference*, 1, 510-514.

Türkoğlu, İ. (2003). *Örüntü tanıma sistemleri ders notları*. 11 Şubat 2010, www.bm.erciyes.edu.tr/basturk/courses/spring_term/.../69_22586.pdf.

- Uyar, E. (2009). *Robot manipölatörlerin kontrolü ders notları*. Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Anabilim dalı
- Wahi, A., Athiq, F. M. ve Palanisamy, C. (2008). A hybrid feature extraction method-based object recognition by neural network. *Computing, communication and networking. ICCCN 2008, International Conference*, 1-5.
- Wahi, A., Ravi P. ve Saranya, M. (2010). A neural network approach to rotated object recognition based on edge features recognition rate and cpu time improvement for rotated object recognition using dwt. *Computing communication and networking technologies (ICCCNT), 2010 international conference*, 1-6.
- Wang, Y. ve Shang, Y. (2010). Fuzzy clustering rbf neural network applied to signal processing of the imaging detection. *Measuring technology and mechatronics automation (ICMTMA), 2010 international conference*, 2, 321-324.
- Yang, M., Zhu S., Lu, F. ve Yu, K. (2010). Correspondence driven adaptation for human profile recognition. *Computer vision and pattern recognition (CVPR), 2011 ieee conference*, 505-512.
- Yapay Zeka Nedir, (b.t). 14 Şubat 2012, <http://www.yapay-zeka.org/modules/wiwimod>.
- Yishuang, S., Hong, H. ve Gang, C. (2010). Research on the visual image processing based on neural network. *Computer modeling and simulation, 2010. ICCMS 10. second international conference*, 4, 55-58.
- Zheng, Y. ve Meng, Y. (2011). Modular neural networks for multi-class object recognition. *Robotics and automation (ICRA), 2011 ieee international conference*, 2927 – 2932.

Zheng, Y., Meng, Y. ve Jin, Y. (2010). Fusing bottom-up and top-down pathways in neural networks for visual object recognition. *Neural networks (IJCNN), the 2010 international joint conference*, 1-8.

EKLER

PIC BASIC PRO KODLARI

```

TRISA=%111111
TRISB=%00000001
TRISC=%10000000
TRISD=%01110000
TRISE=%111
PORTA=0
PORTB=0
PORTC=0
PORTD=0
PORTE=0
****HPWM AYARLARI****
DEFINE CCP1_REG PORTC    'HPWM
DEFINE CCP1_BIT 2        'CCP1 PINI PORTC.2
*****ADC TANIMLAMALARI*****
DEFINE ADC_BITS    10
DEFINE ADC_CLOCK    3
DEFINE ADC_SAMPLEUS    20
DEFINE LCD_DREG PORTB 'LCD data bacaklar? PortB ye ba?!?
DEFINE LCD_DBIT 4 'LCD data bacaklar? 4. bit den ba?!?yor
DEFINE LCD_RSREG PORTB 'LCD RS Baca?? PortB ye ba?!?
DEFINE LCD_RSBIT 1 'LCD RS baca?? 1. Bite ba?!?
DEFINE LCD_EREG PORTB 'LCD Enable Baca?? PortB ye ba?!?
DEFINE LCD_EBIT 3 'LCD Enable Baca?? 3. bite ba?!?
DEFINE LCD_BITS 4 'LCD 4 bit olarak ba?!?
DEFINE LCD_LINES 2 'LCD 2 s?ra yazabiliyor
DEFINE CHAR_PACING 100
ADC1 VAR WORD
DUTY VAR WORD
ADCON1=%10000010    '10 BIT SONUÇ ALMAK İÇİN
SAYI VAR BYTE
SO VAR PORTC.6    'TX UCU
SI VAR PORTC.7    'RX UCU
'B0 VAR BYTE
DEVIR VAR byte
'-----
LCDOUT $FE,1
PAUSE 10
'-----
BASLA:
HIGH PORTD.2
PAUSE 20
LOW PORTD.2
ADCIN 0,ADC1
DUTY=ADC1/4
HPWM 1,DUTY,1000
LCDOUT $FE, 1, "DEVIR=",#DUTY
LCDOUT $FE,$C0, "SERİ HABERLESME"
PAUSE 3
GOSUB HABER
GOTO BASLA
HABER:
IF PORTD.4=1 THEN
SEROUT SO,2,["1"]
PAUSE 3
LCDOUT $FE,1
PAUSE 3
LCDOUT $FE,$C0, "SERİ4 "

```

```

    PAUSE 3
  ENDIF
  IF PORTD.5=1 THEN
    SEROUT SO,2,["2"]
    PAUSE 3
    LCDOUT $FE,1
    PAUSE 3
    LCDOUT $FE,$C0, "DURACAK "
    PAUSE 3
  ENDIF
  IF PORTD.6=1 THEN
    SEROUT SO, 2, ["3"]
    PAUSE 3
    LCDOUT $FE,1
    PAUSE 3
    LCDOUT $FE,$C0, " ROBOT "
    PAUSE 3
  ENDIF
  RETURN
END

```

MATLAB KODLARI

%START MENÜSÜ KULLANARAK ROBOT KOLLU ÇALIŞTIRMA

```

function Start_menu_Callback(hObject, eventdata, handles)
% hObject    handle to Start_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(instrfindall);
clear handles.s1
handles.s1=serial('COM1');
handles.s1.BaudRate=2400;
handles.s1.Timeout=0.0000000001;
fopen(handles.s1);
karesay=0;
dairesay=0;
ucgensay=0;
yildizsay=0;
toctop=0;
bilinmeyensay=0;
handles.oku=fgetl(handles.s1);
pause(0.01);
stop_run=getappdata(handles.stop_button,'value')
stop_run1=getappdata(handles.Stop_menu,'value')
if stop_run==1
setappdata(handles.stop_button,'value',0);
set(handles.durum,'string','STOPPING')
    stop_run=0
    break
end;
if stop_run1==1
    setappdata(handles.Stop_menu,'value',0);
    set(handles.durum,'string','STOPPING')
    stop_run1=0
    break
end;
set(handles.durum,'string','RUNNING')
    if handles.oku=='1'

tic
    setappdata(handles.start_button,'value',1);
    setappdata(handles.Start_menu,'value',1);

```

```

pause(0.01);
RGB = getsnapshot(handles.image);
axes(handles.snapshot);
% Display image in axesImage1.
imshow(RGB);
fclose(handles.s1);
% Convert to gray scale
if size(RGB,3)==3 %RGB image
grisekil=rgb2gray(RGB);
end
I = IM2UINT8(grisekil);
Imax=max(I);
smax=max(Imax);
if smax<=140
set(handles.edit3,'string','EITHER THERE IS NOT A KNOWN OBJECT
OR THE CAMERA IS BROKENDOWN - !! ERROR !!')
axes(handles.image2);
% Display image in axesImage1.
imshow(grisekil);
else if smax>140
grisekil = imadjust(grisekil,[0.45 0.57], [0 1]);
%kenar bulma
kenarbulma=edge(grisekil,'canny',0.39,1);%standard deviation=0,95
% Remove all object containing fewer than 10 pixels
binarysekil = bwareaopen(kenarbulma,100);
I3 = IM2UINT8(binarysekil);
Imax3=max(I3);
smax3=max(Imax3);
if smax3<=106
set(handles.edit3,'string','EITHER THERE IS NOT A KNOWN OBJECT OR
THE SENSOR-1 IS BROKENDOWN - !! ERROR !!')
axes(handles.image2);
% Display image in axesImage1.
imshow(binarysekil);
else if smax3>106
[m1 n1]=size(binarysekil)
%ilave çizgi ekleme
se0 = strel('line', 2, 0);
se90 = strel('line', 2, 90);
ilavecizgi = imdilate(binarysekil, [se0 se90],'same');

BWfinal =ilavecizgi
s = regionprops(BWfinal, {'Perimeter','centroid','FilledArea','BoundingBox','FilledImage','MinorAxisLength','Orientation','MajorAxisLength','Eccentricity'});
centrox=s(1).Centroid(1)
centroy=s(1).Centroid(2)
majo=s(1).MajorAxisLength
mino=s(1).MinorAxisLength
ori=s(1).Orientation
if (centrox < 20 || centrox >143 || centroy < 10 || centroy > 130 )
set(handles.edit3,'string','THE OBJECT IS OUT OF AREA')
else
theta = -90:89;
[R,yp] = radon(BWfinal,theta);
[R1,r_max] = max(R);
theta_max = 90;
while(theta_max > 50 || theta_max<=50)
[R2,theta_max] = max(R1);
R1(theta_max) = 0;
theta_max = theta_max - 91;
end
[m n]=size(BWfinal)
BWfinal=imrotate(BWfinal,-theta_max);
doldurulumsekil = imfill(BWfinal, 'holes');
BWfinal1=doldurulumsekil
se = strel('octagon', 3);

```

```

BWfinal1 = imerode(BWfinal1,se);
I2 = IM2UINT8(BWfinal1);
Imax2=max(I2);
smax2=max(Imax2);
if smax2<=96
    set(handles.edit3,'string','THE UNKNOWN OBJECT - !! ERROR !! - THE IMAGE PROCESSING FAULT')
    axes(handles.image2);
    % Display image in axesImage1.
    imshow(BWfinal1);
else if smax2>96
    s = regionprops(BWfinal1, {'Perimeter','centroid','FilledArea','BoundingBox','FilledImage','MinorAxisLength','Orientation','MajorAxisLength','Eccentricity'});
    ori3=s(1).Orientation
    majo3=s(1).MajorAxisLength
    mino3=s(1).MinorAxisLength
    centrox3=s(1).Centroid(1)
    centroy3=s(1).Centroid(2)
    xsize=centrox3-1.5*majo3/2
    ysize=centroy3-1.5*mino3/2
    width=1.5*majo3
    height=1.5*mino3
    yenisize1=imcrop(BWfinal1,[xsize ysize width height]);
    model=imresize(yenisize1,[120 160]);
    axes(handles.image2);
    % Display image in axesImage1.
    imshow(model);
    imwrite(model,char('D:\model.bmp'),char('bmp'));
    ss = regionprops(model, {'Perimeter','centroid','FilledArea','BoundingBox','FilledImage','MinorAxisLength','Orientation','MajorAxisLength','Eccentricity'});
    per=ss(1).Perimeter
    fil=ss(1).FilledArea
    maj=ss(1).MajorAxisLength
    min=ss(1).MinorAxisLength
    c=[model(:)]
    Y = sim(net,c);

```

%KARAR VERME VE KARAR VERİLEN DURUMA GÖRE ROBOT KOL KONUM KODLARI

```

p=1
switch p==1
case (per>=300 & per<=340) & (fil>=5800 & fil<=6500) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(1)==1)%*(0.5<Y(1)<40) && (Y(1)>Y(3)
&& Y(1)>Y(4) && Y(1)>Y(5) && Y(1)>Y(6) && Y(1)>Y(7) && Y(1)>Y(8) && Y(1)>Y(9) && Y(1)>Y(10) && Y(1)>Y(11) && Y(1)>Y(12) ))
    set(handles.edit3,'string','SQUARE')
    tim=toc;
    tim
    karesay=karesay+1
    set(handles.square_piece,'string',karesay)
    set(handles.process_time,'string',tim)
    pause(0.01);
    fopen(handles.s1);
    pause(0.01);
    ch=9;

ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
    ii=0;
    for ii=1:1000;
        handles.oku=fgetl(handles.s1);
        if handles.oku=='3'

pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);

```

```

pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=2450;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5)
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);

```

```

fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
end
case ((per>=300 & per<=340) & (fil>=5800 & fil<=6500) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(2)==1 )%(0.5<Y(1)<40) &&
(Y(1)>Y(3) && Y(1)>Y(4) && Y(1)>Y(5) && Y(1)>Y(6) && Y(1)>Y(7) && Y(1)>Y(8) && Y(1)>Y(9) && Y(1)>Y(10) && Y(1)>Y(11) && Y(1)>Y(12) ))
    set(handles.edit3,'string','SQUARE')
    tim=toc;
    tim
    karesay=karesay+1
    set(handles.square_piece,'string',karesay)
    set(handles.process_time,'string',tim)
    pause(0.01);
    fopen(handles.s1);
    pause(0.01);
    ch=9;

ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'

pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=2450;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma

```

```

ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
Pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
fclose(handles.s1);
pause(0.01);
break
end;
pause(0.01);
end;
case ((per>=300 & per<=340) & (fil>=5800 & fil<=6500) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(3)==1) % (0.5<Y(2)<40) &&
(Y(2)>Y(3) && Y(2)>Y(4) && Y(2)>Y(5) && Y(2)>Y(6) && Y(2)>Y(7) && Y(2)>Y(8) && Y(2)>Y(9) && Y(2)>Y(10) && Y(2)>Y(11) && Y(2)>Y(12) ))
set(handles.edit3,'string','SQUARE')
tim=toc;
tim

```

```

        karesay=karesay+1
        set(handles.square_piece,'string',karesay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'

pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5)
ch=1;
ra=9;
pw=2450;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);

```

```

fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
        fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
end;

case ((per>=300 & per<=340) & (fil>=5800 & fil<=6500) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(4)==1 ) %%(0.5<Y(3)<40) &&
(Y(3)>Y(1) && Y(3)>Y(2) && Y(3)>Y(5) && Y(3)>Y(6) && Y(3)>Y(7) && Y(3)>Y(8) && Y(3)>Y(9) && Y(3)>Y(10) && Y(3)>Y(11) && Y(3)>Y(12) ))
    set(handles.edit3,'string','SQUARE')
    tim=toc;
    tim
    karesay=karesay+1
    set(handles.square_piece,'string',karesay)
    set(handles.process_time,'string',tim)
    pause(0.01);
    fopen(handles.s1);
    pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'

pause(1);
ch=14;

```

```

ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=2450;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma

```

```

ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
Pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
        fclose(handles.s1);

        pause(0.01);
        break
    end;
    pause(0.01);
end;

    case ((per>=290 & per<=350) & (fil>=6400 & fil<=6800) & (maj>=95 & maj<=110) & (min>=70 & min<=90) & Y(5)==1) %%(0.5<Y(5)<40) ) &&
(Y(5)>Y(1) && Y(5)>Y(2) && Y(5)>Y(3) && Y(5)>Y(4) && Y(5)>Y(9) && Y(5)>Y(10) && Y(5)>Y(11) && Y(5)>Y(12)))
        set(handles.edit3,'string','CIRCLE')
        tim=toc;
        tim
        dairesay=daresay+1
        set(handles.circle_piece,'string',daresay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'
                pause(1);

ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=2150;%daire koyma
pwk=pw/2;

```

```

pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;

pw=1300;%malzeme alma
ra=9;

pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=14;
ra=10;

pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;

pw=2250;%malzeme alma
ra=9;

pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;

pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;

pw=2250;%malzeme alma
ra=9;

pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=14;
ra=10;

pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;

pw=1350;%malzeme alma
ra=9;

pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
ra=9;

pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);

```

```

pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);

        fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
    end;
    case ((per>=290 & per<=350) & (fil>=6400 & fil<=6800) & (maj>=95 & maj<=110) & (min>=70 & min<=90) & Y(6)==1) % (0.5<Y(6)<40) &&
(Y(6)>Y(1) && Y(6)>Y(2) && Y(6)>Y(3) && Y(6)>Y(4) && Y(6)>Y(9) && Y(6)>Y(10) && Y(6)>Y(11) && Y(6)>Y(12)))
        set(handles.edit3,'string','CIRCLE')
        tim=toc;
        tim
        dairesay=dairesay+1
        set(handles.circle_piece,'string',dairesay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);

        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'

pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pAuse(0.5);
ch=1;
ra=9;
pw=2150;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=14;

```

```

ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
fclose(handles.s1);
pause(0.01);
break
end;
pause(0.01);
end;
case ((per>=290 & per<=350) & (fil>=6400 & fil<=6800) & (maj>=95 & maj<=110) & (min>=70 & min<=90) & Y(7)==1) %%(0.5<Y(6)<40) &&
(Y(6)>Y(1) && Y(6)>Y(2) && Y(6)>Y(3) && Y(6)>Y(4) && Y(6)>Y(9) && Y(6)>Y(10) && Y(6)>Y(11) && Y(6)>Y(12)))
set(handles.edit3,'string','CIRCLE')
tim=toc;

```

```

        tim
        dairesay=dairesay+1
        set(handles.circle_piece,'string',dairesay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);

        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'

pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=2150;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
Pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);

```

```

    pw1=mod(pwk,256);
    fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
    pause(0.5);
    ch=1;
    ra=8;
    pw=1150;%malzeme alma
    pwk=pw/2;
    pwh=floor(pwk/256);
    pw1=mod(pwk,256);
    fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
    pause(0.5);
    ch=5;
    pw=2250;%malzeme alma
    ra=9;
    pwk=pw/2;
    pwh=floor(pwk/256);
    pw1=mod(pwk,256);
    fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
    pause(0.5);
    ch=14;
    ra=10;
    pw=1450;%gripper açık
    pwk=pw/2;
    pwh=floor(pwk/256);
    pw1=mod(pwk,256);
    fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
    pause(0.5);
    ch=5;
    pw=1350;%malzeme alma
    ra=9;
    pwk=pw/2;
    pwh=floor(pwk/256);
    pw1=mod(pwk,256);
    fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
    pause(0.5);
    ch=9;
    ra=9;
    pw=950;%malzeme alma
    pwk=pw/2;
    pwh=floor(pwk/256);
    pw1=mod(pwk,256);
    fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
    pause(0.5);
        fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
    end;
    case ((per>=290 & per<=350) & (fil>=6400 & fil<=6800) & (maj>=95 & maj<=110) & (min>=70 & min<=90) & Y(8)==1) % (0.5<Y(6)<40) &&
(Y(6)>Y(1) && Y(6)>Y(2) && Y(6)>Y(3) && Y(6)>Y(4) && Y(6)>Y(9) && Y(6)>Y(10) && Y(6)>Y(11) && Y(6)>Y(12)))
        set(handles.edit3,'string','CIRCLE')
        tim=toc;
        tim
        dairesay=dairesay+1
        set(handles.circle_piece,'string',dairesay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);

    ch=9;
    ra=7;
    pw=1200;%malzeme alma
    pwk=pw/2;
    pwh=floor(pwk/256);

```

```

pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
    ii=0;
    for ii=1:1000;
        handles.oku=fgetl(handles.s1);
        if handles.oku=='3'
            pause(1);
            ch=14;
            ra=9;
            pw=1100;%griper
            pwk=pw/2;
            pwh=floor(pwk/256);
            pw1=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
            pause(0.5);
            ch=5;
            pw=2250;%malzeme alma
            ra=9;
            pwk=pw/2;
            pwh=floor(pwk/256);
            pw1=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
            pause(0.5);
            ch=1;
            ra=9;
            pw=2150;%daire koyma
            pwk=pw/2;
            pwh=floor(pwk/256);
            pw1=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
            pause(1);
            ch=5;
            pw=1300;%malzeme alma
            ra=9;
            pwk=pw/2;
            pwh=floor(pwk/256);
            pw1=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
            pause(1.5);
            ch=14;
            ra=10;
            pw=1450;%gripper açık
            pwk=pw/2;
            pwh=floor(pwk/256);
            pw1=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
            pause(0.5);
            ch=5;
            pw=2250;%malzeme alma
            ra=9;
            pwk=pw/2;
            pwh=floor(pwk/256);
            pw1=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
            pause(0.5);
            ch=1;
            ra=8;
            pw=1150;%malzeme alma
            pwk=pw/2;
            pwh=floor(pwk/256);
            pw1=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
            pause(0.5);
            ch=5;
            pw=2250;%malzeme alma

```

```

ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
        fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
    end;
    case ((per>=300& per<=360) & (fil>=5200 & fil<=6500) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(9)==1) %(0.5<Y(7)<40) &&
(Y(7)>Y(1) && Y(7)>Y(2) && Y(7)>Y(3) && Y(7)>Y(4) && Y(7)>Y(9) && Y(7)>Y(10) && Y(7)>Y(11) && Y(7)>Y(12)))
        set(handles.edit3,'string','TRIANGLE')
        tim=toc;
        tim
        ucgensay=ucgensay+1
        set(handles.triangle_piece,'string',ucgensay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'
                pause(1);
            end
        end
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);

```

```

fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=1870;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);

```

```

pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
    fclose(handles.s1);
    pause(0.01);
    break
end;
pause(0.01);
end;
case ((per>=300& per<=360) & (fil>=5200 & fil<=6500) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(10)==1) %%(0.5<Y(8)<40) &&
(Y(8)>Y(1) && Y(8)>Y(2) && Y(8)>Y(3) && Y(8)>Y(4) && Y(8)>Y(9) && Y(8)>Y(10) && Y(8)>Y(11) && Y(8)>Y(12)))
    set(handles.edit3,'string','TRIANGLE')
    tim=toc;
    tim
    ucgensay=ucgensay+1
    set(handles.triangle_piece,'string',ucgensay)
    set(handles.process_time,'string',tim)
    pause(0.01);
    fopen(handles.s1);
    pause(0.01);
    ch=9;

ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
    ii=0;
    for ii=1:1000;
        handles.oku=fgetl(handles.s1);
        if handles.oku=='3'

pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=1870;%daire koyma

```

```

pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;

```

```

pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
        fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
end;

        case ((per>=300& per<=360) & (fil>=5200 & fil<=6500) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(11)==1)%%(0.5<Y(9)<40) &&
(Y(9)>Y(1) && Y(9)>Y(2) && Y(9)>Y(3) && Y(9)>Y(4) && Y(9)>Y(5) && Y(9)>Y(6) && Y(9)>Y(7) && Y(9)>Y(8)))
        set(handles.edit3,'string','TRIANGLE')
        tim=toc;
        tim
        ucgensay=ucgensay+1
        set(handles.triangle_piece,'string',ucgensay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'

pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=1870;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);

```

```

pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
fclose(handles.s1);
pause(0.01);
break
end;
pause(0.01);
end;
case ((per>=300& per<=360) & (fil>=5200 & fil<=6500) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(12)==1)% (0.5<Y(10)<40) &&
(Y(10)>Y(1) && Y(10)>Y(2) && Y(10)>Y(3) && Y(10)>Y(4) && Y(10)>Y(5) && Y(10)>Y(6) && Y(10)>Y(7) && Y(10)>Y(8)))

```

```

        set(handles.edit3,'string','TRIANGLE')
        tim=toc;
        tim
        ucgensay=ucgensay+1
        set(handles.triangle_piece,'string',ucgensay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'
                pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        pause(0.5);
ch=1;
ra=9;
pw=1870;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;

```

```

pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
ra=9;
pw=950;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
        fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
    end;
    case ((per>=380 & per<=415) & (fil>=5600 & fil<=6100) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(13)==1) % (0.5<Y(11)<40) &&
(Y(11)>Y(1) && Y(11)>Y(2) && Y(11)>Y(3) && Y(11)>Y(4) && Y(11)>Y(5) && Y(11)>Y(6) && Y(11)>Y(7) && Y(11)>Y(8)))
        set(handles.edit3,'string','STAR')

    tim=toc;
    tim
    yildizsay=yildizsay+1
    set(handles.star_piece,'string',yildizsay)
    set(handles.process_time,'string',tim)
    pause(0.01);
    fopen(handles.s1);
    pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;

```

```

pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
    ii=0;
    for ii=1:1000;
        handles.oku=fgetl(handles.s1);
        if handles.oku=='3'
            pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=1800;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1650;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=9;
ra=9;
pw=1850;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5)
ch=1;

```

```

ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
pw=950;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
fclose(handles.s1);
    pause(0.01);
    break
end;
    pause(0.01);
end;
    case ((per>=380 & per<=415) & (fil>=5600 & fil<=6100) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(14)==1) % (0.5<Y(12)<40) &&
(Y(12)>Y(1) && Y(12)>Y(2) && Y(12)>Y(3) && Y(12)>Y(4) && Y(12)>Y(5) && Y(12)>Y(6) && Y(12)>Y(7) && Y(12)>Y(8)))
    set(handles.edit3,'string','STAR')
    tim=toc;
    tim
    yildizsay=yildizsay+1
    set(handles.star_piece,'string',yildizsay)
    set(handles.process_time,'string',tim)
    pause(0.01);
    fopen(handles.s1);
    pause(0.01);

ch=9;
ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
    ii=0;
    for ii=1:1000;

```

```

        handles.oku=fgetl(handles.s1);
        if handles.oku=='3'
            pause(1);
            ch=14;
            ra=9;
            pw=1100;%griper
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
            pause(0.5);
            ch=5;
            pw=2250;%malzeme alma
            ra=9;
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
            pause(0.5);
            ch=1;
            ra=9;
            pw=1800;%daire koyma
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
            pause(1);
            ch=5;
            pw=1650;%malzeme alma
            ra=9;
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
            pause(1.5);
            ch=9;
            ra=9;
            pw=1850;%malzeme alma
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
            pause(1);
            ch=14;
            ra=10;
            pw=1450;%gripper açık
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
            pause(0.5);
            ch=5;
            pw=2250;%malzeme alma
            ra=9;
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
            pause(0.5);
            ch=1;
            ra=8;
            pw=1150;%malzeme alma
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);

```

```

fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=9;
pw=950;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
fclose(handles.s1);
    pause(0.01);
    break
end;
pause(0.01);
end;
case ((per>=380 & per<=415) & (fil>=5600 & fil<=6100) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(15)==1)% (0.5<Y(12)<40) &&
(Y(12)>Y(1) && Y(12)>Y(2) && Y(12)>Y(3) && Y(12)>Y(4) && Y(12)>Y(5) && Y(12)>Y(6) && Y(12)>Y(7) && Y(12)>Y(8)))
    set(handles.edit3,'string','STAR')
    tim=toc;
    tim
    yildizsay=yildizsay+1
    set(handles.star_piece,'string',yildizsay)
    set(handles.process_time,'string',tim)
    pause(0.01);
    fopen(handles.s1);
    pause(0.01);
    ch=9;

ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13])
ii=0;
for ii=1:1000;
    handles.oku=fgetl(handles.s1);
    if handles.oku=='3'
        pause(1);
ch=14;
ra=9;

```

```

pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=1800;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1650;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=9;
ra=9;
pw=1850;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;

```

```

pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
pw=950;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
    end;
    case ((per>=380 & per<=415) & (fil>=5600 & fil<=6100) & (maj>=95 & maj<=115) & (min>=70 & min<=90) & Y(16)==1)%0.5<Y(12)<40) &&
(Y(12)>Y(1) && Y(12)>Y(2) && Y(12)>Y(3) && Y(12)>Y(4) && Y(12)>Y(5) && Y(12)>Y(6) && Y(12)>Y(7) && Y(12)>Y(8)))
        set(handles.edit3,'string','STAR')
        tim=toc;
        tim
        yildizsay=yildizsay+1
        set(handles.star_piece,'string',yildizsay)
        set(handles.process_time,'string',tim)
        pause(0.01);
        fopen(handles.s1);
        pause(0.01);
        ch=9;

ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
        ii=0;
        for ii=1:1000;
            handles.oku=fgetl(handles.s1);
            if handles.oku=='3'

                pause(1);
            ch=14;
            ra=9;
            pw=1100;%griper
            pwk=pw/2;
            pwh=floor(pwk/256);
            pwl=mod(pwk,256);
            fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);

```

```

pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=1800;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=5;
pw=1650;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1.5);
ch=9;
ra=9;
pw=1850;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(1);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);

```

```

ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=9;
pw=950;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
fclose(handles.s1);
        pause(0.01);
        break
    end;
    pause(0.01);
    end;
    otherwise
        set(handles.edit3,'string','THE UNKNOWN OBJECT')
    tim=toc;
    tim
    bilinmeyensay=bilinmeyensay+1
    set(handles.unknown_piece,'string',bilinmeyensay)
    set(handles.process_time,'string',tim)
    pause(0.01);
    fopen(handles.s1);
    pause(0.01);
    ch=9;

ra=7;
pw=1200;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
    ii=0;
    for ii=1:1000;
        handles.oku=fgetl(handles.s1);
        if handles.oku=='3'
            pause(1);
ch=14;
ra=9;
pw=1100;%griper
pwk=pw/2;
pwh=floor(pwk/256);
pwl=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pwl pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);

```

```

pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=1;
ra=9;
pw=800;%daire koyma
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(1);
ch=5;
pw=1300;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(1.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=1;
ra=8;
pw=1150;%malzeme alma
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=2250;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=14;
ra=10;
pw=1450;%gripper açık
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);
fprintf(handles.s1,['!SC' ch ra pw1 pwh 13]);
pause(0.5);
ch=5;
pw=1350;%malzeme alma
ra=9;
pwk=pw/2;
pwh=floor(pwk/256);
pw1=mod(pwk,256);

```

```
%ÖĞRENME BUTONUNA KOMUTLARI
% --- Executes on button press in learn_button.

kare=imread('D:\database8\kare1.bmp');
kare1=imread('D:\database8\kare3.bmp');
kare2=imread('D:\database8\kare5.bmp');
kare3=imread('D:\database8\kare6.bmp');
daire=imread('D:\database8\daire1.bmp');
daire1=imread('D:\database8\daire2.bmp');
daire2=imread('D:\database8\daire3.bmp');
daire3=imread('D:\database8\daire4.bmp');
ucgen=imread('D:\database8\ucgen2.bmp');
ucgen1=imread('D:\database8\ucgen3.bmp');
ucgen2=imread('D:\database8\ucgen6.bmp');
ucgen3=imread('D:\database8\ucgen7.bmp');
yildiz=imread('D:\database8\yildiz1.bmp');
yildiz1=imread('D:\database8\yildiz2.bmp');
yildiz2=imread('D:\database8\yildiz8.bmp');
yildiz3=imread('D:\database8\yildiz9.bmp');

P=[ [kare(:)] [kare1(:)] [kare2(:)] [kare3(:)] [daire(:)] [daire1(:)] [daire2(:)] [daire3(:)] [ucgen(:)] [ucgen1(:)] [ucgen2(:)] [ucgen3(:)] [yildiz(:)] [yildiz1(:)] [yildiz2(:)] [yildiz3(:)]]

T = [[1;0;0;0;0;0;0;0;0;0;0;0;0;0] [0;1;0;0;0;0;0;0;0;0;0;0;0] [0;0;1;0;0;0;0;0;0;0;0;0;0] [0;0;0;1;0;0;0;0;0;0;0;0;0] [0;0;0;0;1;0;0;0;0;0;0;0;0] [0;0;0;0;0;1;0;0;0;0;0;0;0] [0;0;0;0;0;0;1;0;0;0;0;0;0] [0;0;0;0;0;0;0;1;0;0;0;0;0] [0;0;0;0;0;0;0;0;1;0;0;0;0] [0;0;0;0;0;0;0;0;0;1;0;0;0] [0;0;0;0;0;0;0;0;0;0;1;0;0] [0;0;0;0;0;0;0;0;0;0;0;1;0] [0;0;0;0;0;0;0;0;0;0;0;0;1]]

net=newlvq(P,16,[.0625 .0625 .0625 .0625 .0625 .0625 .0625 .0625 .0625 .0625 .0625 .0625 .0625 .0625 .0625]);
net.trainFcn='trainc'

net.trainParam.epochs =1000
net.trainParam.goal=0.000000001;
```

```

net = train(net,P,T)

adres_model_ismi=inputdlg('database adresi ve model ismini girin. Ör:D:\databases22.mat','DATABASE ADRESİ VE MODEL İSMİ')
save (char(adres_model_ismi))

%ÖĞRETME BUTONU KOMUTLARI
% --- Executes on button press in teach_button.
function teach_button_Callback(hObject, eventdata, handles)
% hObject    handle to teach_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
warning off
RGB = getsnapshot(handles.image);
axes(handles.image2);
imshow(RGB);
% Convert to gray scale
if size(RGB,3)==3 %RGB image
    grisekil=rgb2gray(RGB);
end
ikibitresim0=inputdlg('ikibit resim 0 eşik değeri giriniz. Ör:0.45','0,45 ALTI TÜM PİKSELLER 0 OLUR');
ikibitresim0=str2num(char(ikibitresim0))
ikibitresim1=inputdlg('ikibit resim 0 eşik değeri giriniz. Ör:0.57','0,57 ÜSTÜ TÜM PİKSELLER 1 OLUR');
ikibitresim1=str2num(char(ikibitresim1))
grisekil = imadjust(grisekil,[ikibitresim0 ikibitresim1], [0 1]);
axes(handles.image2);
imshow(grisekil);
canny_degeri=inputdlg('canny filtresi eşik değeri giriniz. Ör:0.39','CANNY EŞİK DEĞERİ');
canny_degeri=str2num(char(canny_degeri))
kenarbulma=edge(grisekil,'canny',canny_degeri,1);
axes(handles.image2);
imshow(kenarbulma);
% Remove all object containing fewer than 10 pixels
pixelat=inputdlg('kaç pixelden daha az pixel içeren objeler atılsın? Ör:100','FAZLA PIXEL ATMA')
pixelat=str2num(char(pixelat))
binarysekil = bwareaopen(kenarbulma,pixelat);
axes(handles.image2);
imshow(binarysekil);
%ilave çizgi ekleme
sifir_ekleme=inputdlg('0 dereceli pixeller kaç kat kalınlaşsın? Ör:2','0 DERECELİ PİXELERİ KALINLAŞTIRMA')
sifir_ekleme=str2num(char(sifir_ekleme))
se0 = strel('line', sifir_ekleme, 0);
doksan_ekleme=inputdlg('90 dereceli pixeller kaç kat kalınlaşsın? Ör:2','90 DERECELİ PİXELERİ KALINLAŞTIRMA')
doksan_ekleme=str2num(char(doksan_ekleme))
se90 = strel('line', doksan_ekleme, 90);
ilavecizgi = imdilate(binarysekil, [se0 se90],'full');
axes(handles.image2);
imshow(ilavecizgi);
BWfinal=ilavecizgi;
s = regionprops(BWfinal, {'centroid','FilledArea','BoundingBox','FilledImage','MinorAxisLength','Orientation','MajorAxisLength','Eccentricity'});
ori=s(1).Orientation
ori=num2str(ori)
theta = -90:89;
[R, xp] = radon(BWfinal,theta);
[R1,r_max] = max(R);
theta_max = 90;
while(theta_max > 50 || theta_max < -50)
    [R2,theta_max] = max(R1);
    R1(theta_max) = 0;
    theta_max = theta_max - 91;
end
majo=s(1).MajorAxisLength
mino=s(1).MinorAxisLength
centrox=s(1).Centroid(1)
centroy=s(1).Centroid(2)
[m n]=size(BWfinal)
%Step 4: Fill Interior Gaps

```

```

doldurulmussekil = imfill(BWfinal, 'holes');
axes(handles.image2);
imshow(doldurulmussekil);
%Step 6: Smoothen the Object
duzeltme=inputdlg('objenin tümünde %kaç düzeltme yapılsın? Ör:3 ve katları','DÜZELTME')
duzeltme=str2num(char(duzeltme))
seD = strel('octagon',duzeltme);
BWfinal = imerode(doldurulmussekil,seD);
axes(handles.image2);
imshow(BWfinal);
s = regionprops(BWfinal, {'centroid','FilledArea','BoundingBox','FilledImage','MinorAxisLength','Orientation','MajorAxisLength','Eccentricity'});
ori3=s(1).Orientation
ori3=num2str(ori3)
majo3=s(1).MajorAxisLength
mino3=s(1).MinorAxisLength
centrox3=s(1).Centroid(1)
centroy3=s(1).Centroid(2)
xsize=centrox3-1.5*majo3/2
ysize=centroy3-1.5*mino3/2
width=1.5*majo3
height=1.5*mino3
yenisize1=imcrop(BWfinal,[xsize ysize width height]);
axes(handles.image2);
imshow(yenisize1);
satir_olcusu=inputdlg('database şeklinin satır ölçüsü ne olsun? Ör:120','SATIR ÖLÇÜSÜ')
satir_olcusu=str2num(char(satir_olcusu))
sutun_olcusu=inputdlg('database şeklinin sütun ölçüsü ne olsun? Ör:160','SUTUN ÖLÇÜSÜ')
sutun_olcusu=str2num(char(sutun_olcusu))
model=imresize(yenisize1,[satir_olcusu sutun_olcusu]);
axes(handles.image2);
imshow(model);
adres_model_ismi=inputdlg('database adresi ve model ismini girin. Ör:D:\database5\yildiz9.bmp','DATABASE ADRESİ VE MODEL İSMİ')
model_formati=inputdlg('database şeklinin formati ne olsun? Ör:bmp','DATABASE ŞEKLİNİN FORMATI')
imwrite(model,char(adres_model_ismi),char(model_formati))

```