

ROBOTİK LABORATUVARI EKLENTİ KİTAPLARI

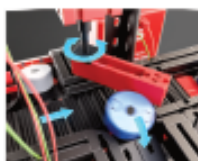
LİSE
LE1

Programlama Dilleri:

C, C++

python

OPEN
ROBERTA

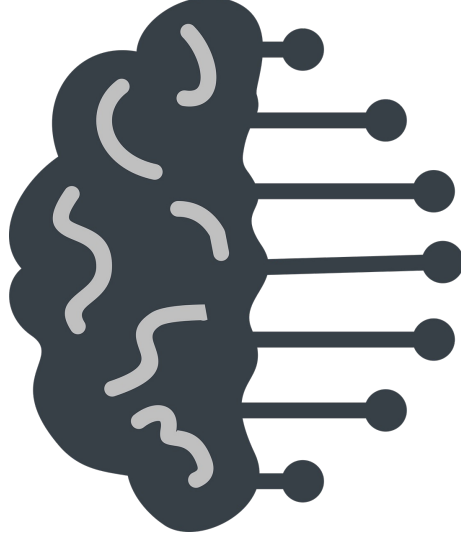


AI YAPAY ZEKÂ

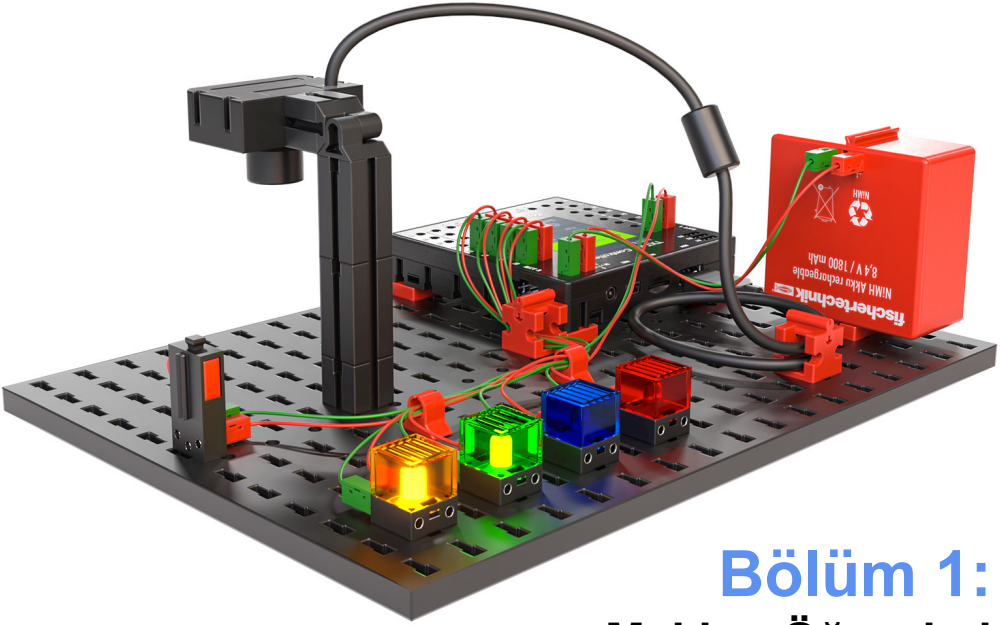


STEM MATEMATİK
INFORMATİK
NATURWISSENSCHAFT
TECHNIK

İçindekiler

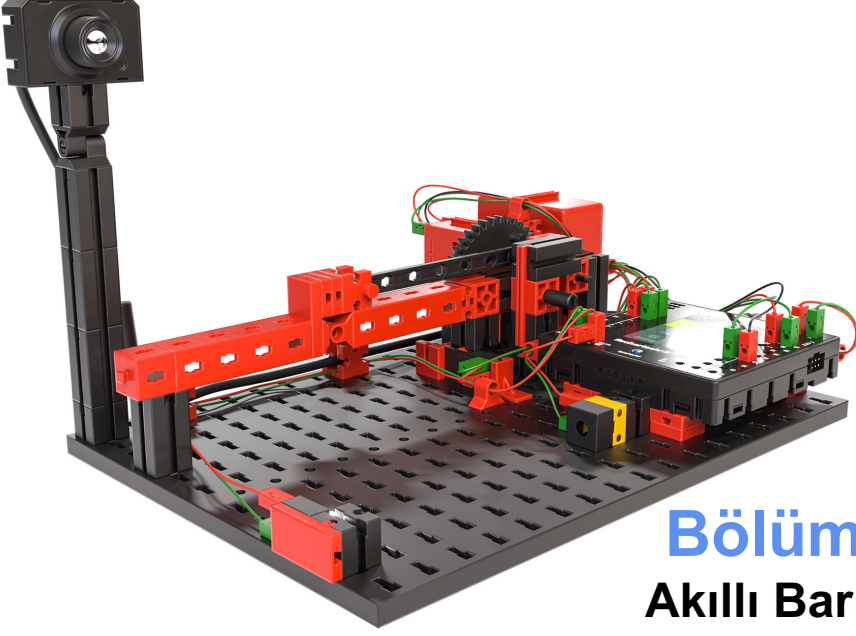


Giriş:
Yapay Zekâ
Sayfa: 4

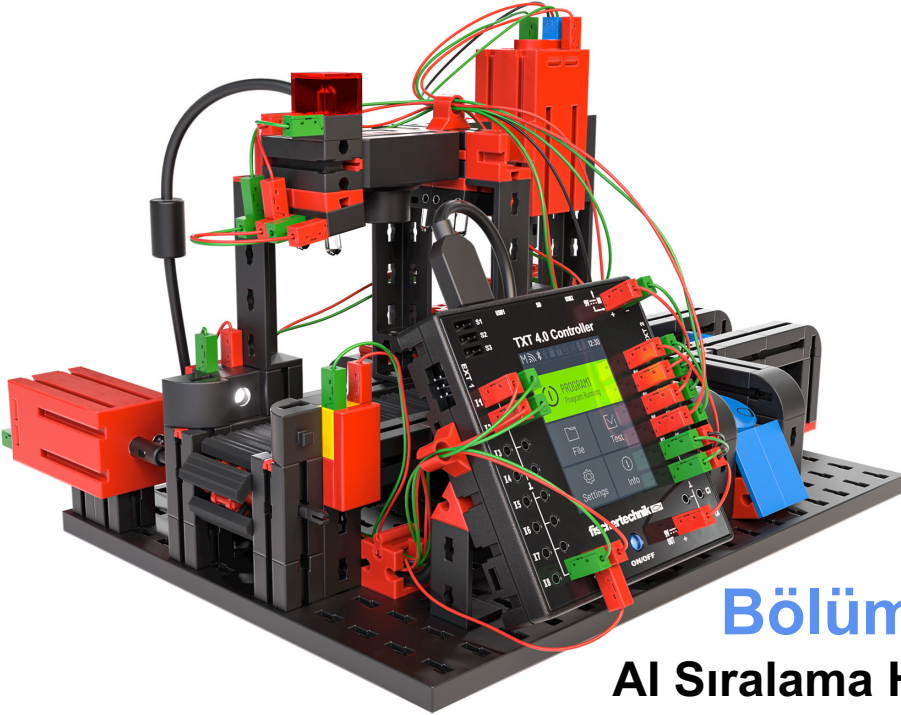


Bölüm 1:
Makine Öğrenimi
Sayfa: 12

İçindekiler

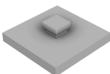
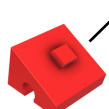
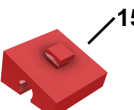
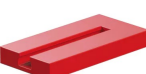
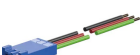
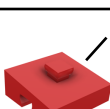
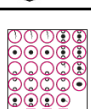


Bölüm 2: Akıllı Bariyer Sayfa: 20



Bölüm 3: AI Sıralama Hattı Sayfa: 30

Parça Listesi

	31 060 1x		32 881 7x		36 950 1x		36 921 4x
	31 330 2x		35 049 3x		38 263 4x		31 011 3x
	31 762 49x		35 064 3x		38 266 1x		36 920 17x
	31 412 5x		32 880 3x		38 267 2x		181 583 9x
	32 330 2x		35 077 1x		36 334 4x		181 584 10x
	36 248 98x		35 078 1x		36 532 1x		36 922 7x
	38 464 2x		130 593 1x		168 961 1x		172 542 10x
	31 982 13x		174 622 5x		174 623 5x		32 879 17x
	174 626 5x		174 627 5x		32 071 6x		197 417 1x
							162 135 4x

Kablo Bağlantıları



Hazır kablolar

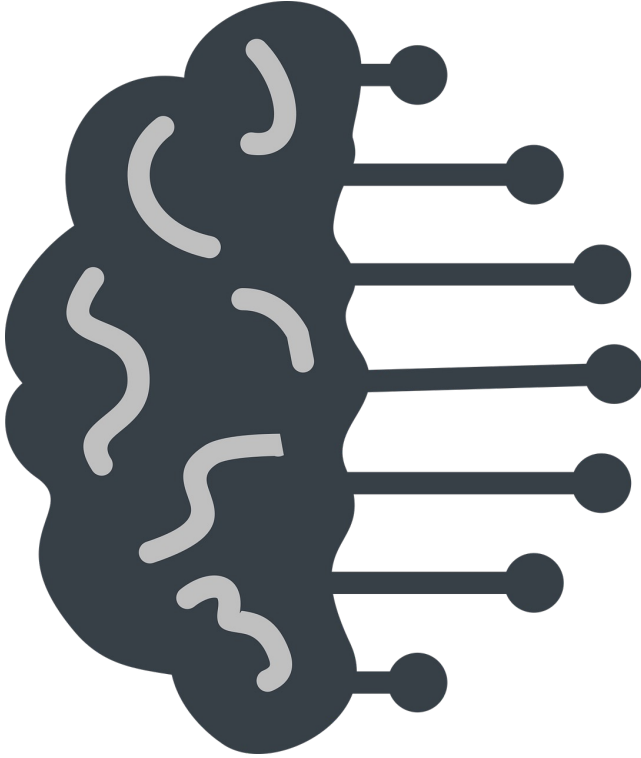
2 X 10 cm

2 X 40 cm



GİRİŞ

AI – Yapay Zekâ



Yapay Zekâya Giriş

Geleceğin Teknolojisini Keşfet!

Yapay zekâ, geleceğin en önemli teknolojilerinden biridir. Bu eğitim seti, öğrencilerin yapay zekâyâ erken yaşta ilgi duymasını sağlar ve onları bu alandaki meslekler için hazırlar. Öğrenciler, bu teknolojiye ait temel kavramları eğlenceli bir şekilde öğrenir ve yapay zekânın nasıl çalıştığını yakından tanıma fırsatı bulur.

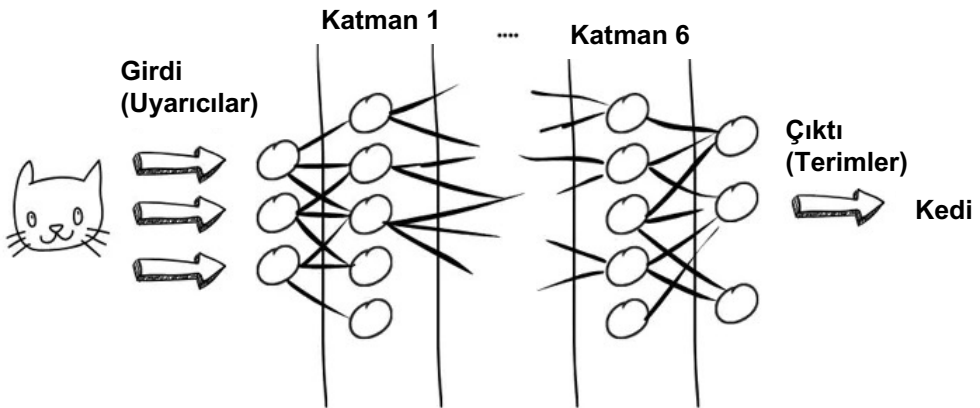
Setin içinde, farklı zorluk seviyelerine sahip üç model yer alır. Bu modeller, yapay zekâ uygulamalarının ne kadar çeşitli olduğunu gösterir ve bu alana giriş yapmak isteyenler için ideal bir başlangıç sunar. Ayrıca, sınıf ortamında kullanılmak üzere hazırlanmış öğretmen destekli öğretim materyalleriyle yapay zekâ hakkındaki bilgi daha da pekiştirilir.

Yapay Zekâ Nedir?

Basitçe söylemek gerekirse, yapay zekâ (YZ), bir programın belirli bir görevi zeki bir şekilde yerine getirmesi anlamına gelir. Bu görev; bir görseli tanımak, bir metni çevirmek ya da müzik üretmek gibi çok farklı alanlarda olabilir.

Yapay zekânın birçok farklı türü vardır. Ancak günümüzde YZ denildiğinde genellikle "yapay sinir ağları" kastedilir. Bunlar, beynimizdeki yapıları taklit eden sistemlerdir. Sinir hücreleri (nöronlar), diğer hücrelerle bağlantı kurarak öğrenme yeteneği kazanan ağlar oluşturur.

Farklı görevler için farklı sinir ağı türleri kullanılır. Örneğin, beynimizde dil işleme merkezi görsel hafızamızdan farklı şekilde çalışır. Yapay zekâda da durum aynıdır: Her görev için uygun sinir ağı yapısı seçilmelidir.



Bir Yapay Sinir Ağı Nasıl Çalışır?

Aşağıdaki açıklama, bir yapay sinir ağının temel yapısını tanıtır:

Sinyaller, ağın sol tarafında yer alan giriş katmanından alınır. Bu katman, duyumlarımızı algılayan hücrelere benzetilebilir. Duyusal hücreler, nöronlarla bağlantı kurar ve bu bağlantının gücü, nöronun gelen uyarıya ne kadar tepki vereceğini belirler.

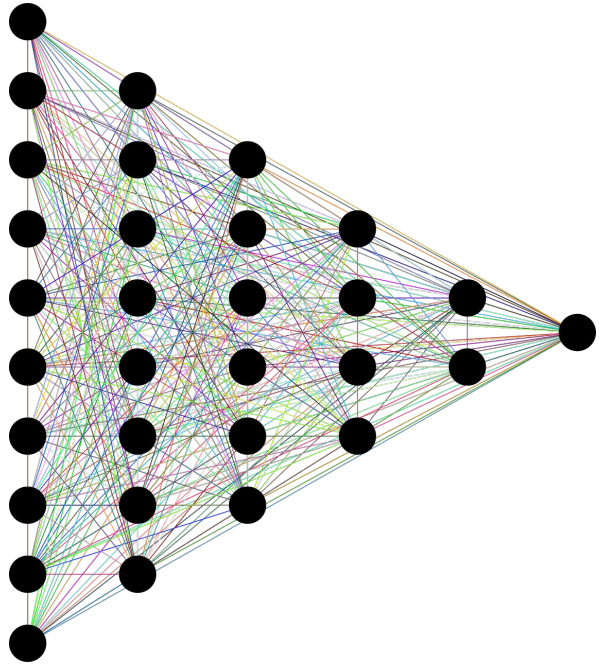
Sağ tarafta ise çıkış katmanı bulunur. Burası, algılanan bilgiye karşılık gelen sonuçları temsil eder. Örneğin, sol taraftan bir resim algılanırsa, sağ tarafta “kedi” gibi bir kavramla eşleşebilir.

Giriş ve çıkış katmanları arasında, çok sayıda nörondan oluşan ara yapılar yer alır. Bunlar genellikle katmanlar (layer) hâlinde düzenlenmiştir ve her katmanın kendine özgü bir görevi vardır.

Bu yapay sinir ağı yapısında üç temel katmandan söz edilir:

- ◆ **Giriş katmanı**, çevreden gelen veriyi alır.
- ◆ **Çıkış katmanı**, elde edilen sonuçları dışarıya verir.
- ◆ **Gizli katmanlar**, giriş ve çıkış katmanları arasında yer alır. Bu katman sayısı birden fazla olabilir ya da hiç olmayabilir.

Farklı görevler için farklı bağlantı türleri tercih edilir. Örneğin, bazı bağlantılar görsellerdeki parçaları tanımada daha iyiyken, bazıları da cümlelerdeki kelime sıralarını algılamada daha başarılıdır. "Derin öğrenme (deep learning)", "DeepMind" ya da metin çeviri programı "DeepL" gibi terimler, sinir ağlarının yapısıyla ilgilidir. Buradaki “deep (derin)” ifadesi, ağda birçok katmanın arka arkaya yer aldığını belirtir. Her katman, bir önceki katmandan gelen veriye tepki verir ve böylece öğrenme gerçekleşir.



Öğrenme Nasıl Gerçekleşir?

Basitçe anlatmak gerekirse, bir yapay sinir ağında öğrenme şu şekilde gerçekleşir: Sol taraftan bir uyarı (örneğin bir görsel) verilir ve sağ tarafta doğru kavramın ortaya çıkıp çıkmadığına bakılır. Eğer doğru sonuç elde edilmezse, bağlantıların gücü sağdan sola doğru yeniden ayarlanır ve işlem tekrar edilir. Bu döngü, ağ doğru cevabı verene kadar sürer.

Büyük bir sinir ağının sürekli olarak işlem yapması oldukça zaman alıcıdır. Bu nedenle yapay zekâya dair eğitimler doğrudan TXT 4.0 cihazı üzerinde değil, bilgisayarda gerçekleştirilir.

Yapay zekâ üzerine çalışmalar aslında onlarca yıldır devam ediyor. Ancak uzun süre boyunca bilgisayarlar ve ağ sistemleri yeterince güçlü değildi. Büyük sinir ağlarını sıfırdan eğitmek için milyonlarca görsel gerekir. Ağ, bu görselleri sürekli olarak soldan sağa işler ve ardından sağdan sola hatalarını düzeltir. Bu da ciddi işlem gücü gerektirir.

Örneklerimizde Yapay Zekâ Nasıl “Çalışır”

Bir yapay zekânın bir görev yapabilmesi için önce öğrenmesi, sonra da öğrendiklerini uygulayabilmesi gerekir. Bu süreç her zaman üç temel adımda ilerler:

1. Eğitim verilerinin toplanması: Yapay zekânın öğrenebilmesi için örnekler ihtiyacı vardır. Bu örnekler, farklı hayvanların ya da nesnelerin görselleri olabilir. Bu görselleri uygun şekilde etiketlememiz gerekir ki yapay zekâ neyi öğrenmesi gerektiğini anlayabilsin.

2. Yapay zekânın eğitilmesi: Bu adımda, yapay zekâya örnekleri gösterir ve her görselin neyi temsil ettiğini açıklarız. Böylece yapay zekâ, bir görsele karşılık gelen kavramı olabildiğince doğru şekilde eşleştirmeyi öğrenir.

3. Yapay zekânın kullanılması: Eğitim süreci tamamlandıktan sonra artık yapay zekâ ağı kullanılabilir. Bu noktada, daha önce görmediği yeni görselleri tanıma denemeleri yapabiliriz.

Ancak şunu çabucak fark edeceksiniz: Yapay zekâyı eğitmek, sanıldığı kadar kolay değildir. Doğru türde ve yeterli sayıda görsel gerekir; ayrıca ışık gibi çevresel koşullar da önemlidir. Bu noktaları, modellerle çalışarak ve denemeler yaparak kendiniz keşfedeceksiniz.

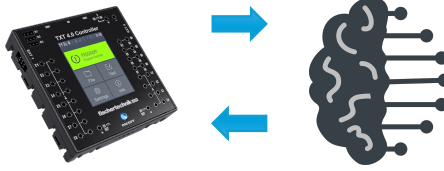
Eğitim sürecinin fazla zorlayıcı olmaması için, biz bu sette **önceden eğitilmiş bir yapay zekâ** kullanıyoruz. Bu yapay zekâ nesneleri tanıyabilme yeteneğine zaten sahiptir. Bizim yapmamız gereken, sadece tanıyacağı nesnelere uygun "isimleri" öğretmektir. Bu öğrenme yöntemine **aktararak öğrenme (transfer learning)** denir.

TXT 4.0 ile Yapay Zekâ

TXT 4.0 ile Yapay Zekâ Kullanımı

Yapay zekâ, günümüzde pek çok alanda yerini almış durumda ve günlük hayatımızda önemli bir rol oynuyor. Öğrencilerin makine öğrenmesi ve algoritmalar hakkında bilgi sahibi olması, bu teknolojiye karşı daha bilinçli ve sorumluluk sahibi bir bakış açısı geliştirmelerini sağlar.

TXT 4.0, öğrencilerin yapay zekâ ile ilk adımlarını atmalarını son derece kolaylaştırır. Bu sistem sayesinde, yapay zekânın arkasında neler olduğunu öğrenir ve kendi modellerinde bu teknolojiyi nasıl kullanabileceklerini keşfederler.



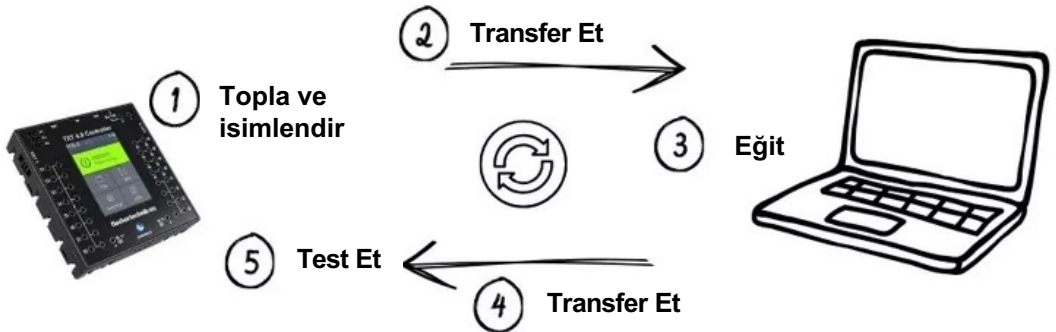
Deneylerimizi TXT 4.0 ile Nasıl Gerçekleştiriyoruz?

TXT ile yaptığımız deneylerde her zaman şu adımları izliyoruz: İlk olarak modeli kullanarak öğrenme için görseller topluyoruz. Bunun için, görüntüleri kaydeden ve gruplara ayıran bir program yazmamız gerekiyor.

Örneğin, bir bariyer kullanıldığında, görüntüler "bariyer açık" ve "bariyer kapalı" gibi gruplara ayrılır. Program, bu görüntüleri TXT cihazında toplar ve her birini uygun şekilde isimlendirir.

TXT 4.0 güçlü bir kontrolcü olsa da, yapay zekâ eğitimi için biraz daha fazla işlem gücüne ihtiyaç vardır. Bu nedenle, toplanan verileri bilgisayara ya da dizüstü bilgisayara aktarır ve yapay zekâyı orada eğitiriz. Eğitim işlemi tamamlandıktan sonra, sadece birkaç dakika içinde eğitilmiş model tekrar TXT cihazına yüklenebilir.

Artık farklı bir program kullanarak bu eğitilmiş yapay zekâyı TXT üzerinde çalıştırabiliriz.



Yapay Zekân Ne Kadar Başarılı?

Yapay Zekâ Değerlendirme Tablosu

Bir yapay zekânın nesneleri ne kadar iyi tanıdığını değerlendirmek için genellikle bir “doğruluk tablosu” kullanılır. Bu tabloda, yapay zekâyâ sadece belirli türde nesneler gösterilir ve doğru tanıdığı nesneler sayılır. Bunlara **“doğru olumlu” (true positive)** denir.

Yapay zekânın tanıyamadığı ama aslında doğru cevabı vermesi gereken nesneler ise **“yanlış olumsuz” (false negative)** olarak adlandırılır.

Bu iki sayı, gösterilen toplam nesne sayısına bölünerek bir oran elde edilir.

Eğer “doğru olumlu” oranı 1’e ne kadar yakınsa ve “yanlış olumsuz” oranı 0’a ne kadar yakınsa, yapay zekânızın nesneleri o kadar güvenilir bir şekilde tanıdığı anlamına gelir.

Doğruluk Tablosu

	"X" olarak algılandı	"X" olarak algılanmadı
"X"	doğru pozitif	yanlış negatif
"X" Değil	yanlış pozitif	doğru negatif

Örnek – Kedi Görselini Tanıma

100 kedi ve 100 köpek görseliyle test edilmiştir.

	Kedi olarak tanındı	Köpek olarak tanındı
Kedi	88 / 100	12 / 100
Köpek	18 / 100	82 / 100

Yapay Zekânın Tarihçesi

Yapay zekânın (YZ) son yıllardaki gelişimi, hem büyüleyici hem de benzersizdir. Sürücüsüz arabalar, sesli asistanlar, izlediğimiz filmlere göre yapılan öneriler, hatta karşımıza çıkan kişiye özel reklamlar... Artık YZ, hayatımızın neredeyse her alanında kullanılmaktadır.

Aslında bu fikir çok eskiye dayanır. 1726 yılında Jonathan Swift, *Gulliver'in Gezileri* adlı romanında, bilgiyi artırmak ve mekanik işleri geliştirmek amacıyla kullanılan “Engine” adlı bir makineden bahsetmiştir. Bu makine, günümüzdeki bilgisayarlara benzetilebilir.

İlk Somut Gelişmeler ve Yapay Zekâ Teriminin Kullanımı

Ancak akıllı makineler geliştirme konusunda ilk ciddi ilerlemeler 1950’li yıllarda yaşanmıştır:

- ◆ **1950:** Claude Shannon satranç oynayabilen ilk bilgisayar programlarını geliştirmeye başladı.
- ◆ **1950:** Alan Turing, “Hesaplama Makineleri ve Zekâ” başlıklı makalesinde, bir makinenin insan gibi düşünebildiğini anlamak için “Turing Testi”ni önerdi.
- ◆ **1955:** Allen Newell, Herbert Simon ve Cliff Shaw, ilk yapay zekâ programı olarak kabul edilen *Logic Theorist*’i geliştirdi.
- ◆ **1956:** John McCarthy ve ekibi Dartmouth Koleji’nde düzenledikleri bir konferansta “yapay zekâ” terimini ilk kez kullandı.

60’lar

1960’ların sonunda Joseph Weizenbaum, doktor-hasta görüşmelerini simüle eden ilk sohbet programlarından biri olan *ELIZA*’yı geliştirdi. Daha sonra bu sistemden elde edilen bilgiler “uzman sistemler” olarak adlandırılan yapay zekâlara aktarıldı. Örneğin, Edward H. Shortliffe tarafından geliştirilen *MYCIN*, doktorların teşhis kararlarına destek vermek amacıyla tasarlanmış bir sistemdir.

80’ler

1980’lerde de önemli adımlar atıldı. **1984’te** geliştirilen *RB5X* adlı robot, kendi kendine öğrenen bir yazılıma sahipti ve geçmiş verilerden yola çıkarak geleceği tahmin edebiliyordu.

1986’da geliştirilen *NETtalk* (Terrence J. Sejnowski, Charles Rosenberg), sinir ağlarını kullanan ilk programlardan biriydi ve kelimeleri okuyup doğru şekilde telaffuz edebiliyordu. Ayrıca daha önce görmediği kelimelerde de öğrendiklerini uygulayabiliyordu.

90'lar

1990'larda, kayıtlı verileri kullanarak otomatik kararlar verebilen ilk algoritmalar geliştirildi. **1997'de**, IBM tarafından üretilen yapay zekâ satranç makinesi *Deep Blue*, dünya satranç şampiyonu Garry Kasparov'u yenerek büyük bir başarıya imza attı. Bu olay, insan zekâsının üstün olduğu düşünülen bir alanda makine başarısının simgesi oldu. Ancak bazı eleştirmenler, *Deep Blue*'nin aslında stratejik düşünmek yerine olası tüm hamleleri hesaplayarak kazandığını savunmuştur.

2000'ler

2016'da, *AlphaGo* adlı sistem, dünyanın en iyi Go oyuncularından Güney Koreli Lee Sedol'u yendi. Donanım ve yazılım alanındaki hızlı gelişmeler, yapay zekânın gündelik yaşama girmesini kolaylaştırdı. **2011'de** Apple'ın *Siri*'si, **2014'te** Microsoft'un *Cortana*'sı ve **2015'te** Amazon'un *Alexa*'sı piyasaya sürüldü. Günümüzde iş dünyasında YZ; otomasyon, derin öğrenme ve Nesnelerin İnterneti (IoT) gibi alanlarda öne çıkmaktadır. Endüstriyel robotlara ek olarak, hizmet alanında da birçok robot geliştirilmektedir.

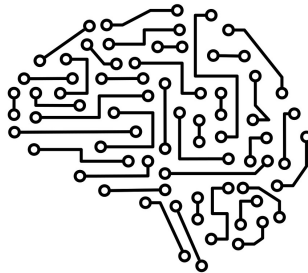
2020 Sonrası Dönem

2020 yılı, yapay zekâ için yeni bir dönemin başlangıcı oldu. OpenAI, *GPT-3*, *GPT-4*, *DALL-E* ve *GLIDE* gibi çığır açıcı yapay zekâ modelleri geliştirdi. Bu modeller; metin oluşturma, yazılım programlama ve görsel üretme gibi karmaşık görevleri yerine getirebilmektedir.

Bundan Sonrası Ne Olacak?

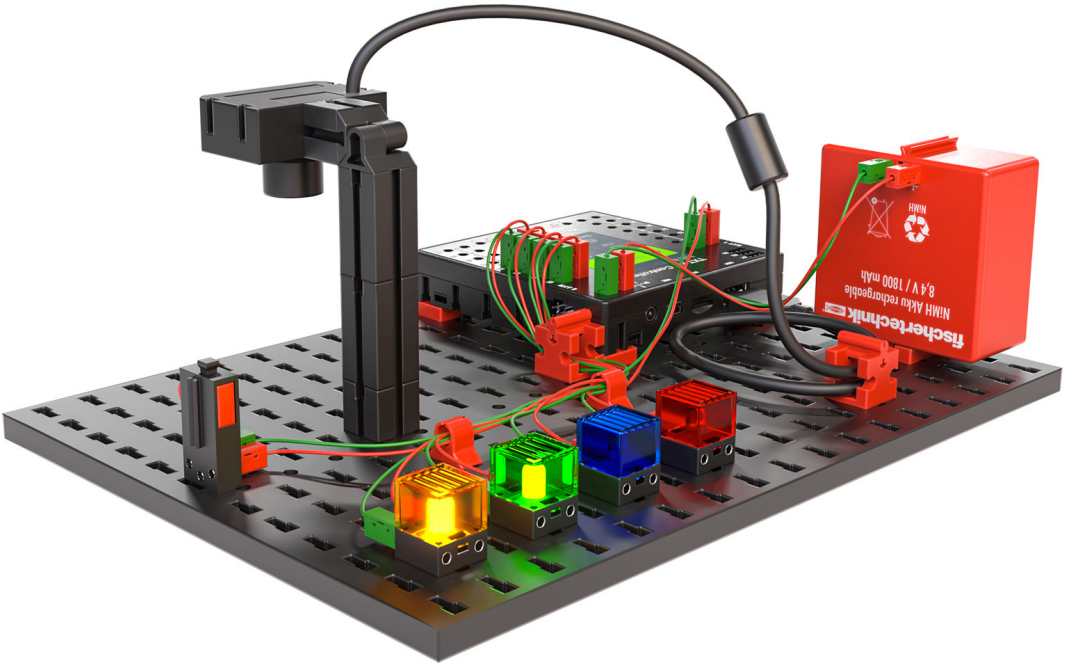
Tüm bu gelişmelere rağmen, yapay zekâ araştırmaları henüz yolun başındadır. Özellikle otomatik sürüş sistemleri ve tıp gibi hassas alanlarda kullanılabilmesi için daha güvenilir ve manipülasyona karşı dayanıklı hâle getirilmesi gerekmektedir.

Avrupa Birliği'nin yürürlüğe koyduğu YZ yasası ise, yapay zekânın nasıl çalıştığının insanlar tarafından anlaşılabilmesi için şeffaflık ilkesini zorunlu kılmaktadır.



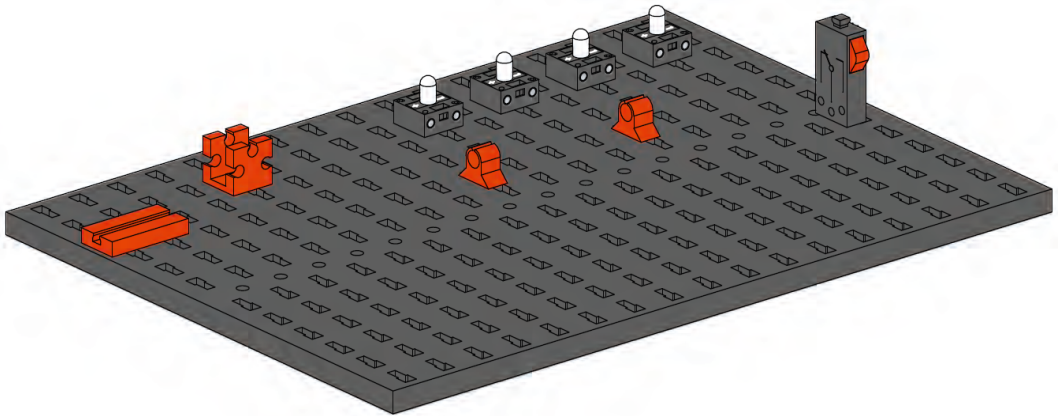
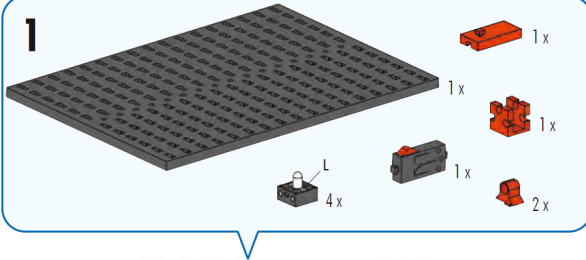
BÖLÜM 01

Makine Öğrenimi

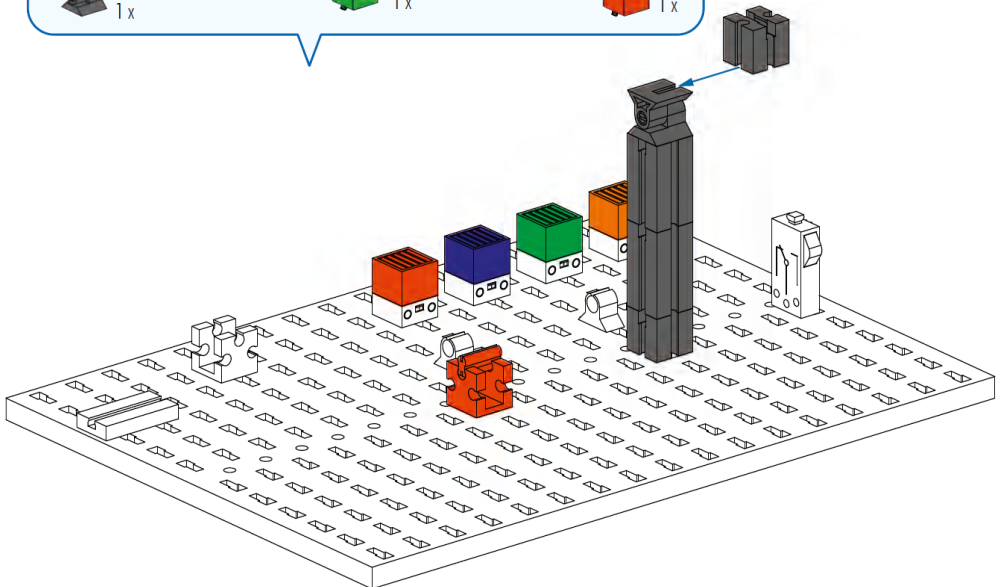
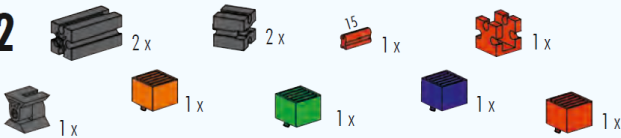


MODEL 1 Makine Öğrenimi

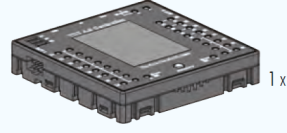
1



2

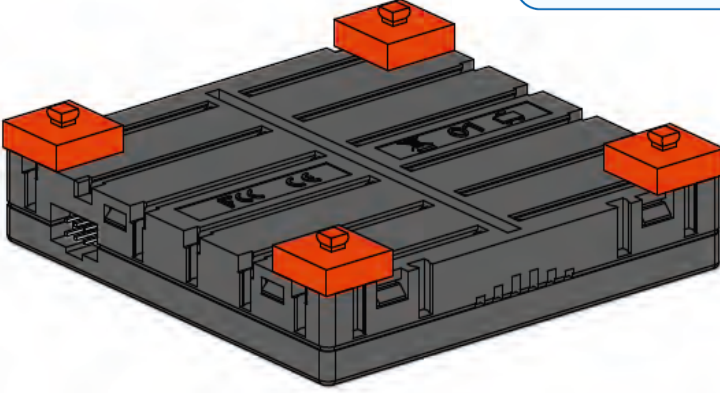


3



1 x

4 x



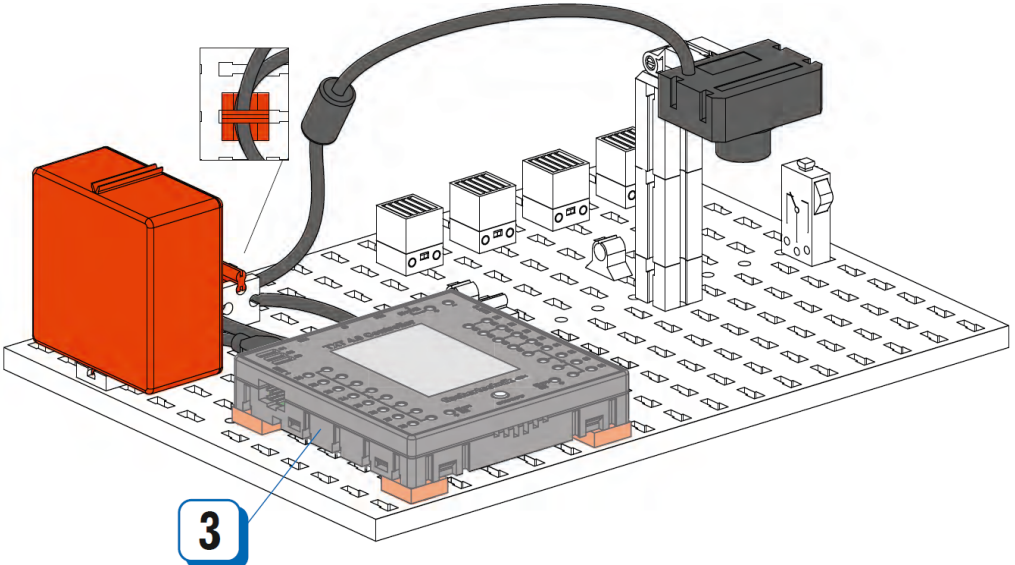
4



1 x

15
1 x

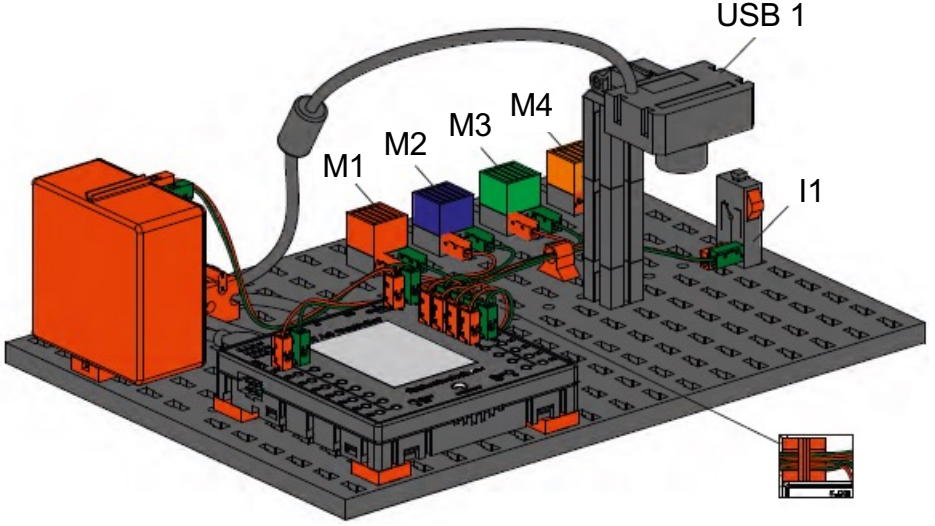
1 x



3

Model - 1

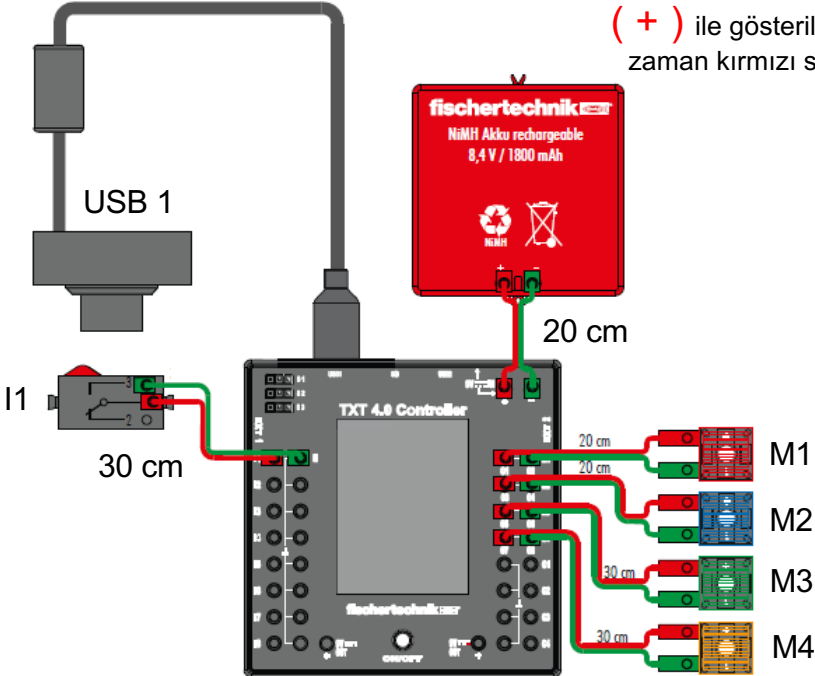
Makine Öğrenimi



Devre Şeması



(+) ile gösterilen alana her zaman kırmızı soket takılır.





Makine Öğrenimi İçin Verilen Görevler

Konu

Görevler – Makine Öğrenimi

Model kurulumu

Modeli deney talimatlarına göre oluşturun. Aşağıdaki görevlerde kamerayla çok fazla çalışacağız. Yapay zekamızın iyi eğitilebilmesi için, aşağıdaki deneyler için ışık koşullarının iyi olduğundan emin olmalısınız. Belki de modeli, nesneler için bir arka plan olarak bir kağıt veya karton parçası entegre edebileceğiniz şekilde genişletebilirsiniz. Gerekirse bunu daha sonra deneyde yapabilirsiniz.

Deneysel Görevler

Görev 1: Temel program

Bu görevle, öğrenme programımızın temel çerçevesi oluşturulacaktır.

Kontrol cihazında dört düğme sağlayan bir program oluşturmak için Robo Pro Kodlama'yı kullanın. Düğmeleri LED'lerin renklerine göre adlandırmanız yeterlidir. Düğmelerden birine bastığınızda, ilgili LED bir kontrol olarak kısa bir süre yanmalıdır.

Görev 2: Yapay zeka eğitimi için görseller toplayın

Bu görevde, görev 1'deki programımızı genişleterek, butonlardan birine basıldığında önce kamerayla fotoğraf çekilmesini sağlıyoruz.

Görüntü “xxxxxxx” dizinine kaydedilecektir. Görüntü dosyasını “<Colour>_<random-number>.png” olarak kaydedin. Ayrıca rastgele sayıyı saniye veya milisaniye cinsinden tam zamanla da değiştirebilirsiniz. Renk kodunu daha sonra bilgisayardaki eğitim programında kullanacağız.

Veriler başarıyla kaydedildikten sonra LED'lerin tekrar ilgili renkte yanmasını sağlayın.

Programı oluşturduğunuzda 4 farklı nesne veya resim arayın.

Şimdi programınızla her nesnenin bir dizi görüntüsünü kaydediyorsunuz.

Öğrenme için verileri böyle topluyoruz!

Görev 3: Bilgisayarda Eğitim

Lütfen dikkat! Bu aşamada işler biraz daha zorlaşıyor!

Bilgisayarınızı TXT 4.0 Denetleyicisine bağlayın. Bilgisayarınızda sözde “Güvenli Kabuk (ssh)” ile uzaktan erişimi açtığınızdan emin olun. Bunu nasıl yapacağınız konusunda TXT 4.0'ınız için talimatlara başvurabilirsiniz [1].

Şimdi öğrenme programını [2] PC'de çalıştırmamız gerekiyor. Program Python programlama diliyle yazılmıştır. Python henüz bilgisayarınıza yüklenmediyse, kurulum talimatlarını

[3] <http://python.org> adresinde bulabilirsiniz.

Python programına bir göz atın - üç bölümden oluşuyor:

İlk bölümde ilk programınız ile hazırladığınız resim dosyaları bilgisayarınıza yüklenir.

İkinci kısım ise yapay zekanızın eğitimi gerçekleştiriyor.

Üçüncü kısımda eğitilen ağ, bir sonraki görevde kullanılmak üzere tekrar denetleyiciye aktarılır.

Görev 4: Tanıma programımızın hazırlanması ve yapay zekanın kullanımı

Son adım olarak, öğrenilen AI'yı nesneleri tanımak için kullanan Robo Pro Coding'de ikinci bir program hazırlıyoruz. Bunun için, eğitilmiş modeli yüklemeniz gerekir (bunun nasıl yapılacağı hakkında Robo Pro Coding'deki [4] örnek programlara bakın).

Modelimiz için bir basmalı düğmeye de ihtiyacımız var. Basmalı düğmeye basıldığında, kamerayla bir resim çekilecek ve AI'mız hangi LED'in yanacağına karar verebilir - bunun için dosyaları adlandırırken daha önce seçtiğiniz metni döndürecektir. Eşleşen LED de programda açılmalıdır.

Program çalışıyorsa, modelin nesneleri ne kadar iyi tanıdığını kontrol edin. Her zaman yüzde yüz çalışmadığını göreceksiniz, ancak artık "öğretilbilir makinelerle" oynamaya hazırsınız ve görüntü tanımayı deneyebilirsiniz.

Görev 5: Yapay zekanız ne kadar iyi?

Yapay zekanın nesneleri tanımada ne kadar iyi olduğunu değerlendirmek için genellikle bir doğruluk tablosu kullanılır; bu, görüldüğünden daha karmaşıktır.

Bunu yapmak için, sadece bir türdeki AI nesnelerini göstermemiz gerekir, örneğin daha önce kırmızı LED ile "işaretledikleriniz". Sonra bu nesnelerden kaçının doğru tanındığını sayarız - bunlar "gerçek pozitifler"dir. Doğru tanınmayanlar "yanlış negatifler"dir, yani yanlışlıkla başka bir şey olarak tanınanlardır. Her iki sayıyı da gösterilen nesnelerin miktarına bölersiniz - bu size istatistik üzerine matematik derslerinden aşına olabileceğiniz bir oran değeri verir.

Şimdi deneyi diğer türdeki nesnelerle gerçekleştiriyoruz ve kaç tanesinin birinci türdeki nesneler olmadığı doğru bir şekilde tanındığını sayıyoruz - bunlar "gerçek negatifler", yani doğru bir şekilde tanınanlardır. Yapay zekamız tarafından kırmızı LED ile yanlış bir şekilde onaylanan nesneler "yanlış pozitifler"dir. Burada da elde edilen sayıları gösterilen toplam (yanlış) nesne sayısına bölersiniz ve yukarıdaki gibi bir oran değeri elde edersiniz.

İlk iki değer 1'e ne kadar yakınsa ve ikinci iki değer 0'a ne kadar yakınsa, AI'nız nesneleri doğru bir şekilde o kadar iyi tanıdıdır.

AI'nız dört tür nesnenin hepsinde ne kadar iyi?

Görev 6: Uzman görevi – Yapay zekanız ne kadar güvenilir?

Programınıza bir nesne gösterdiğinizde, yapay zeka genellikle birkaç değer döndürür ve bu da nesneleri tanımada ne kadar güvenilir olduğunu gösterir, örneğin kırmızı: 0,9, sarı: 0,1.

Yapay zekanın ne kadar güvenilir olduğunu farklı LED'lerin yanıp sönerek göstermesini sağlayacak şekilde değerleri kullanabilir misiniz?

Daha fazla bilgi

[1] Python kurulum programı, <https://www.python.org/>

[2] Robo Pro Coding: A4_TMachine_TModel_Load_AI



Makine Öğrenimi İçin Verilen Çözümler

Çözüm sayfası - Öğrebilen makine

Görev 1: Temel program

Robo Pro Coding'de şurada bir örnek çözüm var:

Proje -> Yeni -> Örnek -> STEM Eklenti AI ->

A1_TMachine_Basic_Program_AI

Görev 2:

Robo Pro Coding'de örnek bir çözüm aşağıdadır:

Proje -> Yeni -> Örnek -> STEM Eklenti AI ->

A2_TMachine_Collect_Images_AI

Görev 3:

Robo Pro Coding'de örnek bir çözüm aşağıdadır:

Proje -> Yeni -> Örnek -> STEM Eklenti AI ->

A3_TMachine_Collect_Images_AI

Görev 4:

Robo Pro Coding'de örnek bir çözüm aşağıdadır:

Proje -> Yeni -> Örnek -> STEM Eklenti AI ->

A4_TMachine_Recognise_Images_AI

Görev 6:

Robo Pro Coding'de örnek bir çözüm aşağıdadır:

Proje -> Yeni -> Örnek -> STEM Eklenti AI ->

A6_TMachine_AI_Reliable_AI

Ek

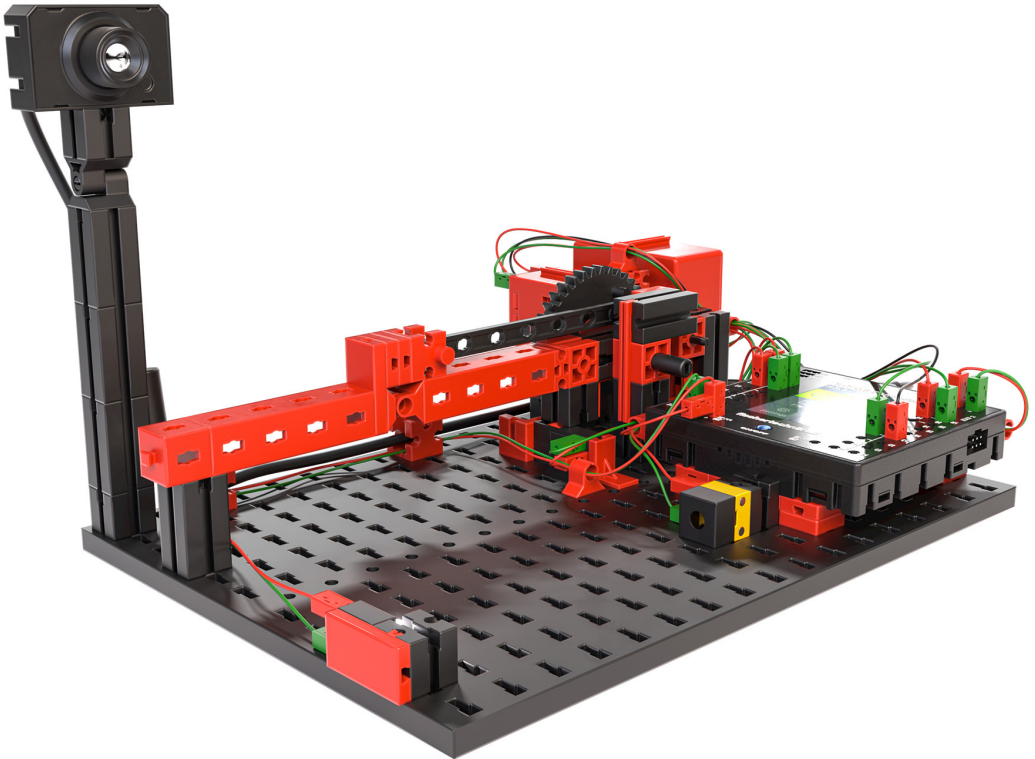
Makine öğrenimi

Daha fazla bilgi

Yapay zeka öğrenme/egitim programının yürütülmesi için Python yüklü bir bilgisayara ihtiyaç vardır.

BÖLÜM 02

Akıllı Bariyer

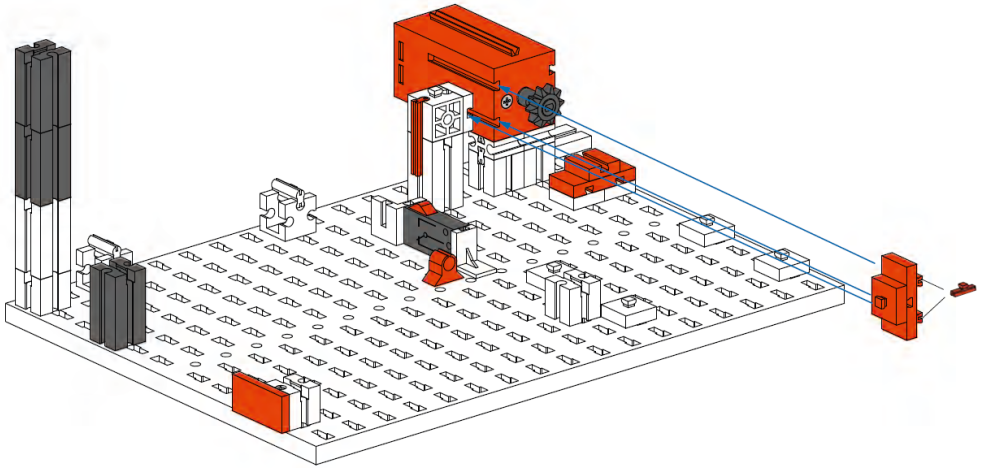
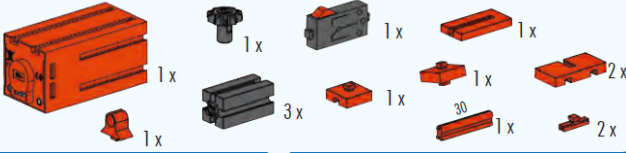


TASARIM

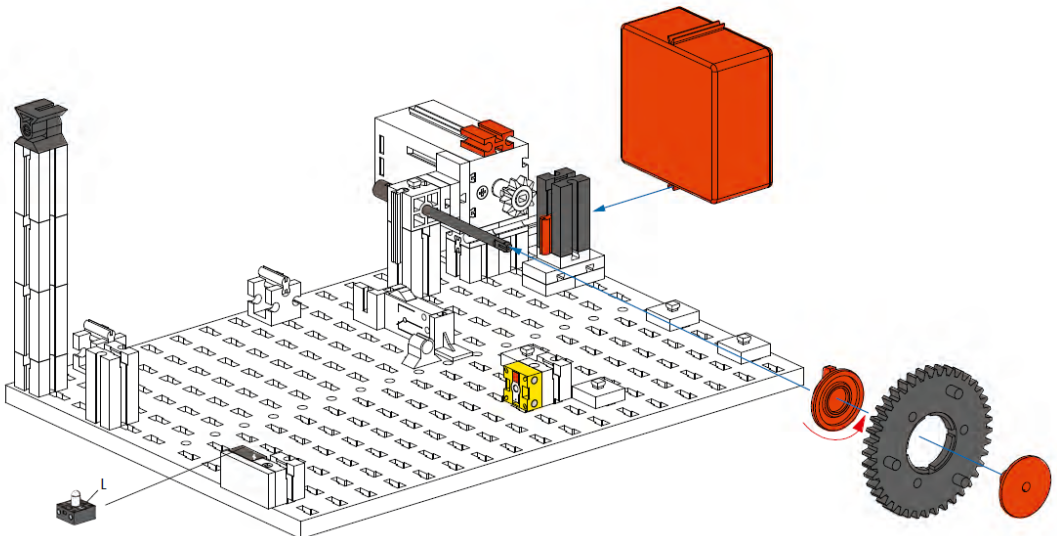
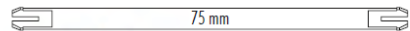
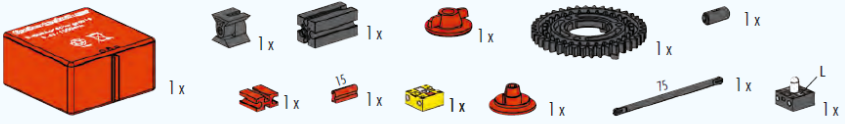
GÖREV

ÇÖZÜM

3



4



5



1x



1x



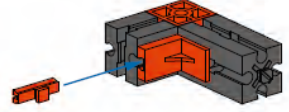
1x



1x



1x



6



1x



1x



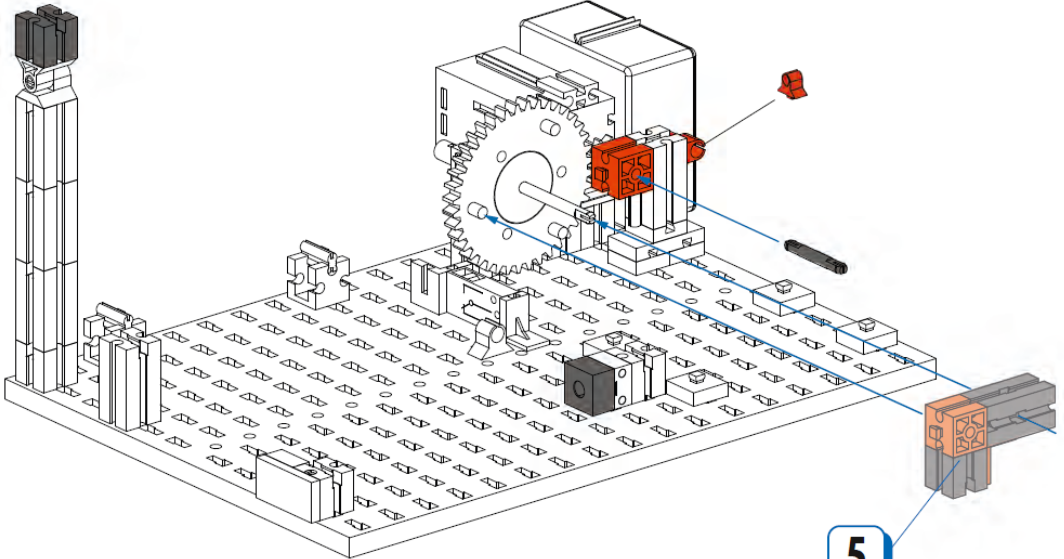
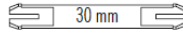
1x



1x



1x



5

7



1x



1x

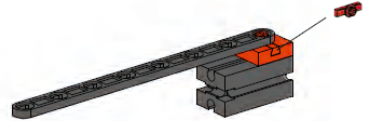


105

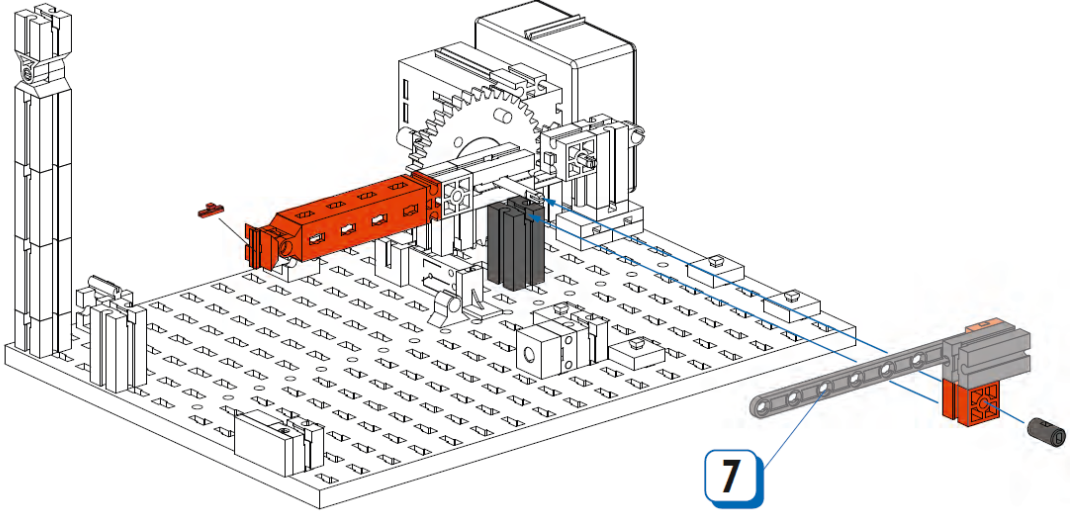
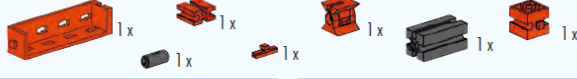
1x



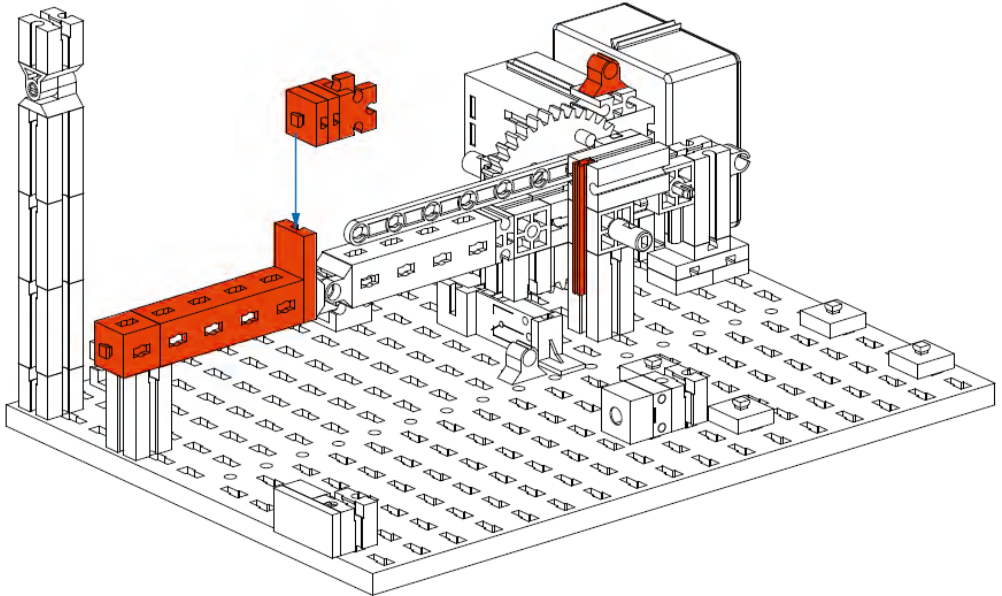
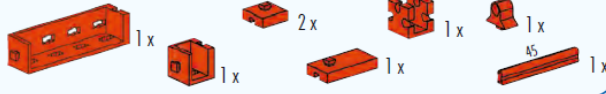
1x

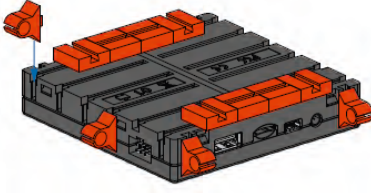


8

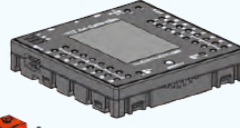


9





10



1 x

8 x

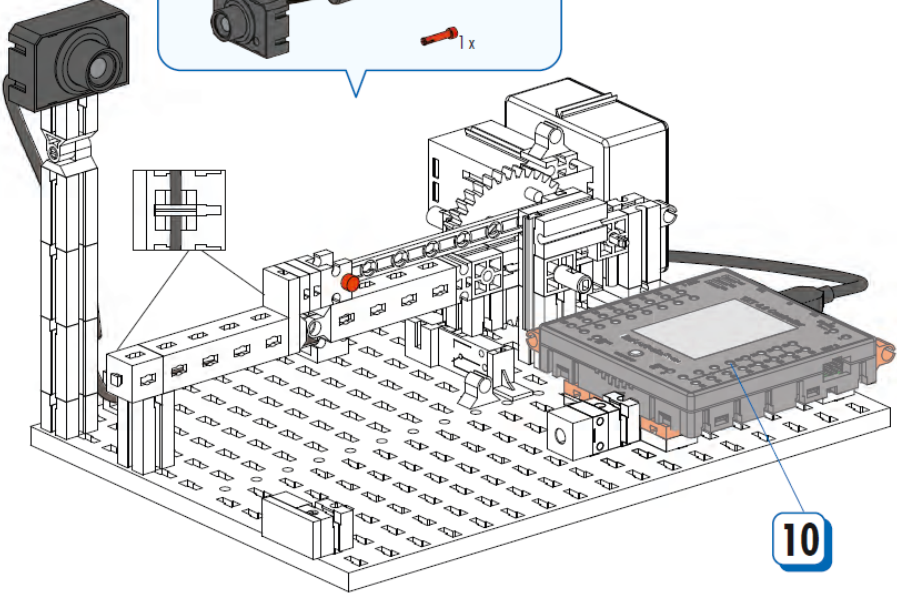
4 x

11

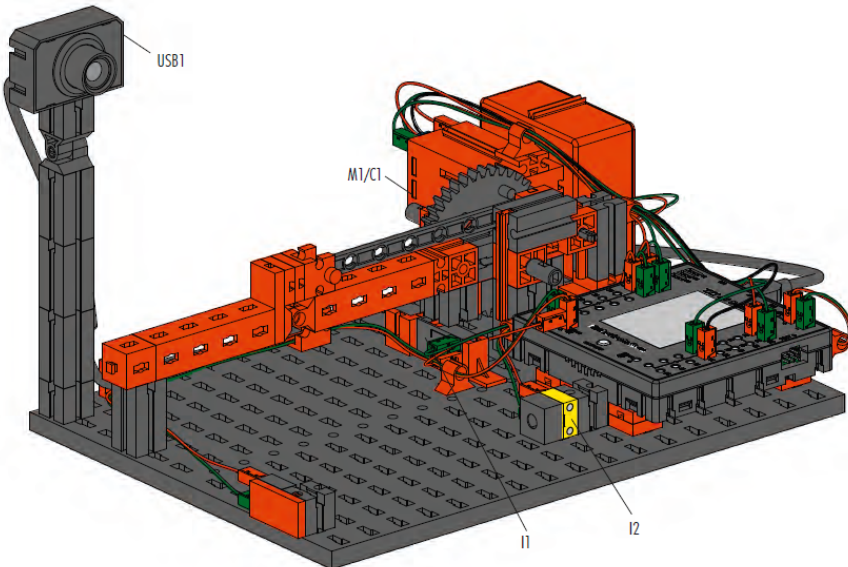


1 x

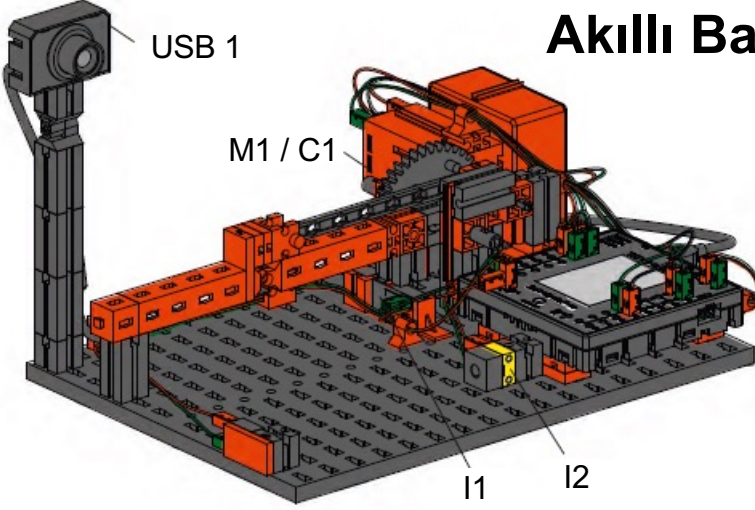
1 x



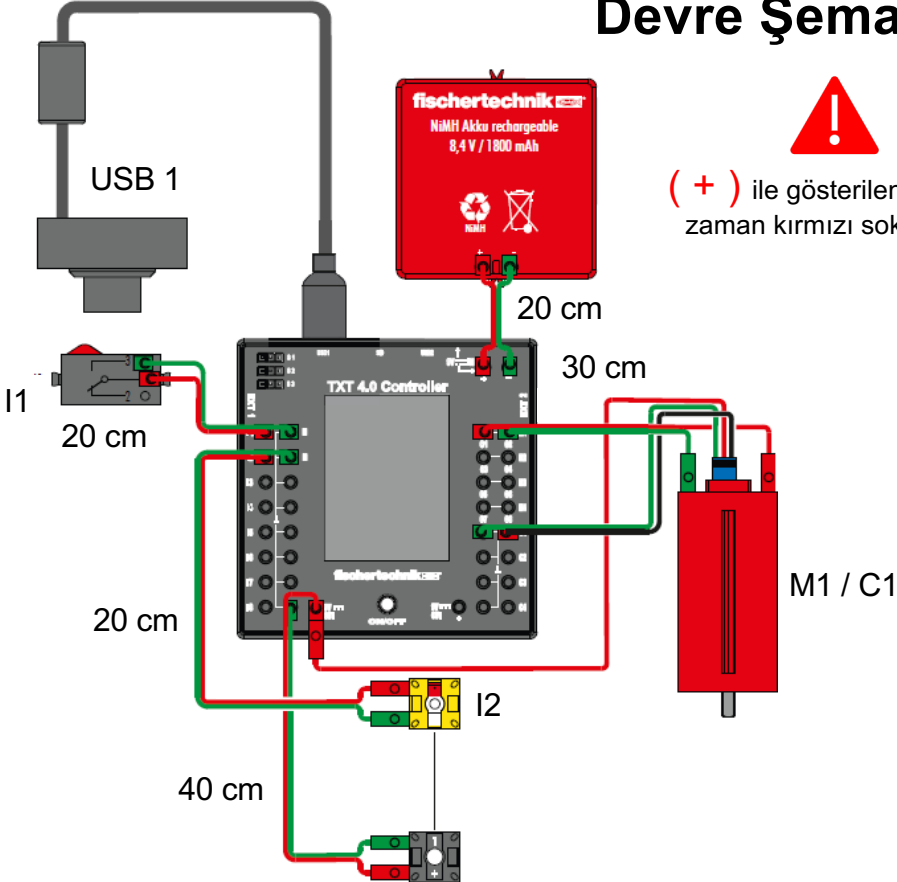
10



Model - 2 Akıllı Bariyer



Devre Şeması





Akıllı Bariyer İçin Verilen Görevler

Konu

Görevler – Akıllı Bariyer

Akıllı bariyer modelini düzgün bir şekilde kullanabilmeniz için, öncelikle “makine öğrenimi” modeline ve ilgili görevlere bakmanız mantıklıdır. Öğrenme sürecinin temelleri orada açıklanmıştır.

Model kurulumu

Kurulum adımlarına göre modeli oluşturun. Aşağıdaki görevlerde kamerayla çok çalışacağız. Yapay zekamızın iyi eğitilebilmesi için, aşağıdaki deneyler için ışık koşullarının iyi olduğundan emin olmalısınız.

DeneySEL Görevler

Görev 1: Temel program

Bu görevle bariyerimiz için temel bir çerçeve oluşturulacaktır.

Robo Pro Coding'de, ışık bariyerinin kesintiye uğramasına tepki veren ve daha sonra size (kontrol ekranındaki bir anahtar aracılığıyla) bariyerin önündeki nesneyi "yeşil" (bariyer açılmalı) veya "kırmızı" (bariyer kapalı kalmalı) olarak sınıflandırma olanağı sunan bir program hazırlayın.

Bu şekilde, öğrenebilen makinede olduğu gibi kamerayla fotoğraf çekerek ve ardından bunları buna göre kaydederek eğitim verileri toplarsınız.

Ayrıca "boş bir arka planı" "kırmızı" olarak sınıflandırmalısınız.

Görev 2: Bilgisayarda Eğitim

Modeli eğitmek için, öğrenebilen makine için kullandığınız iş akışının aynısını tekrar kullanabilirsiniz. Sonuç, bariyerin önündeki iki tür nesneyi ayırt edebilen AI modelidir,

Görev 3: Bariyer kontrolcümüzün oluşturulması

Artık AI'yı öğrenebilen makineyle aynı şekilde kullanabiliriz. Bunu yapmak için, bariyeri kontrol etmek için öğretilen AI'mızı kullanan ikinci bir program oluşturuyoruz.

Yeni program yaklaşık olarak şu şekilde çalışmalıdır:

Bir nesne ışık bariyerini keserse, kamerayla bir resim çekilmeli ve AI'ya iletilmelidir. AI'nın nesneyi "kırmızı" veya "yeşil" olarak sınıflandırmasına bağlı olarak, karşılık gelen LED yanmalı ve bariyer açılmalıdır - tabii ki yalnızca yeşilse - ve ardından birkaç saniye sonra tekrar kapanmalıdır.

Görev 4: Bariyer ne kadar iyi kontrol edilebilir?

Şimdi deney yapmaya başlayalım! Sadece bariyeri açmak için hangi "nesnelerin" iyi çalıştığını deneyin. Belki oklarla veya "sihirli" sembollerle birkaç işaret çizmek istersiniz. Ya da elinizle belirli bir hareket yapmayı deneyebilirsiniz. Ya da önüne farklı nesneler koyarsınız.

Bazı şeylerin veya hareketlerin diğerlerinden daha kolay tanındığını göreceksiniz. Bunun nedeni, farklı nesneleri tanıma ve sınıflandırma konusunda uzmanlaşmış bir AI kullanıyor olmamızdır. Metinleri (örneğin plakalar) ayırt etmede çok iyi değildir.

Sadece neyin iyi çalıştığını ve neyin çalışmadığını deneyin. Tanıma kalitesinin her zaman AI'nın daha önce ne kadar iyi eğitildiğine, yani örneğin öğrenmesi için ona doğru görüntüleri verip vermediğinize bağlı olduğunu unutmayın.

Daha fazla bilgi

Yapay zeka öğrenme/eğitim programının çalıştırılabilmesi için Python yüklü bir bilgisayara ihtiyaç vardır.

- [1] Python kurulum programı, <https://www.python.org/>
- [2] Robo Pro Coding: A4_TMachine_TModel_Load_AI



Akıllı Bariyer İçin Verilen Çözümler

Çözüm sayfası – Akıllı Bariyer

Görev 1: Temel program

Robo Pro Coding'de bir örnek çözüm var:

Proje -> Yeni -> Örnek -> STEM Eklenti AI ->

A1_Barrier_Basic_Program_AI

Görev 2:

Robo Pro Coding'de örnek bir çözüm aşağıdadır:

Proje -> Yeni -> Örnek -> STEM Eklenti AI ->

A2_Train_Barrier_AI

Görev 3:

Robo Pro Coding'de örnek bir çözüm aşağıdadır:

Proje -> Yeni -> Örnek -> STEM Eklenti AI ->

A3_Control_Barrier_AI

Ek

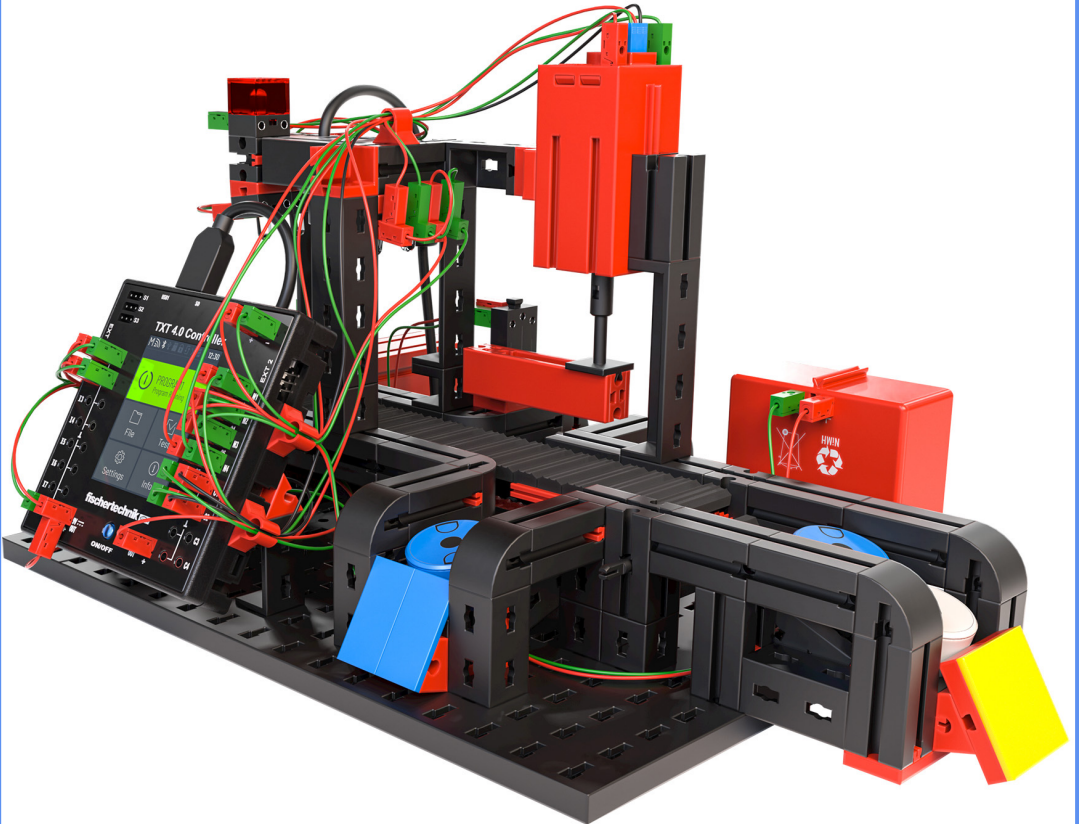
Akıllı Bariyer

Daha fazla bilgi

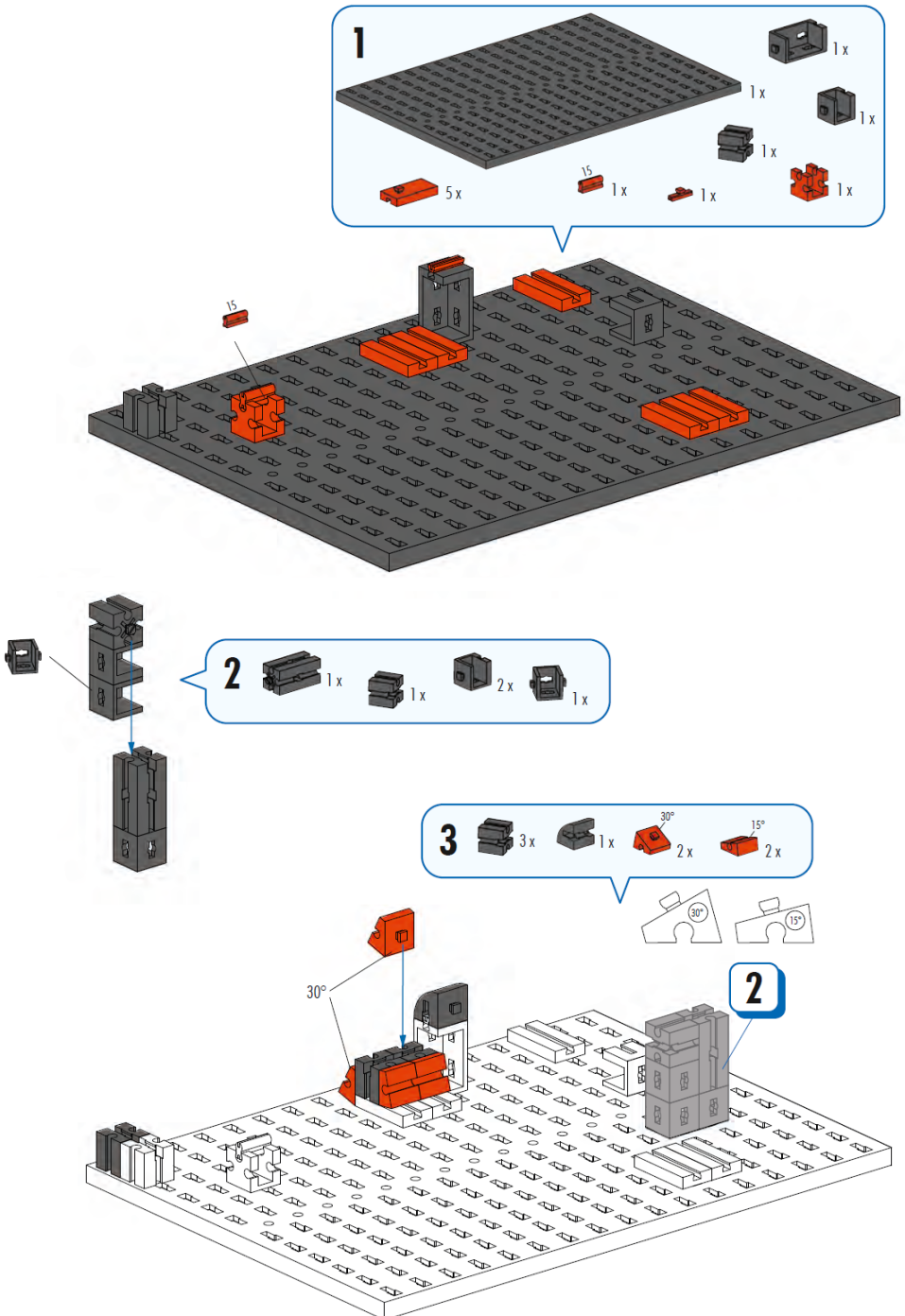
Yapay zeka öğrenme/eğitim programının yürütülmesi için Python yüklü bir bilgisayara ihtiyaç vardır.

BÖLÜM 03

AI Sıralama Hattı



MODEL 3 Al Sıralama Hattı

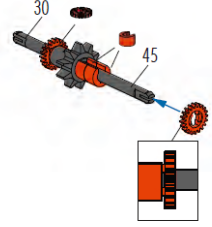


4

30 1 x 45 1 x 1 x 2 x 2 x

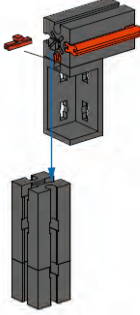
30 mm

45 mm



5

1 x 1 x 1 x 1 x 30 1 x 1 x

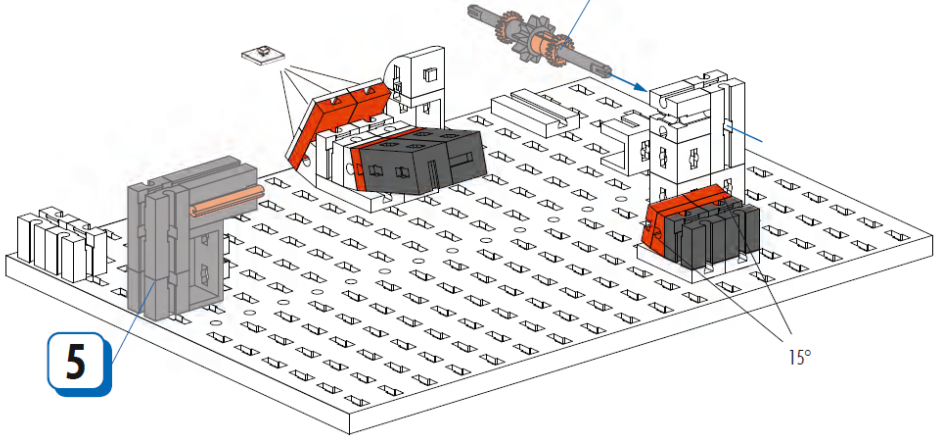


6

2 x 2 x 2 x 4 x 4 x 15° 2 x



4

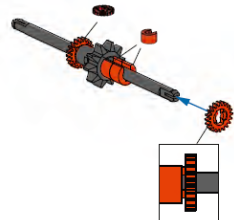


5

7

45 2 x 1 x 2 x 2 x

45 mm



TASARIM

GÖREV

ÇÖZÜM

 4 x
 30° 2 x



15 1x



1 x



30°

15 1x

 1 x

15

 1x 1 x 1x 1x

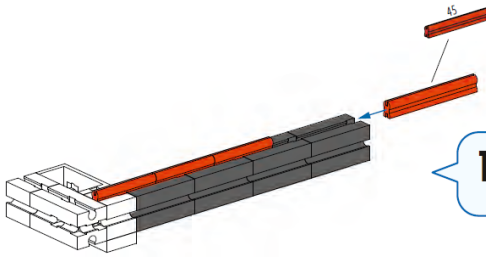
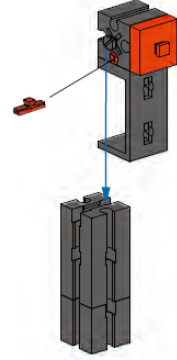
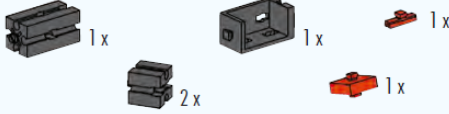
45 2x

 2 x

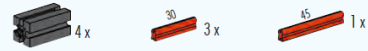
45

 2 x

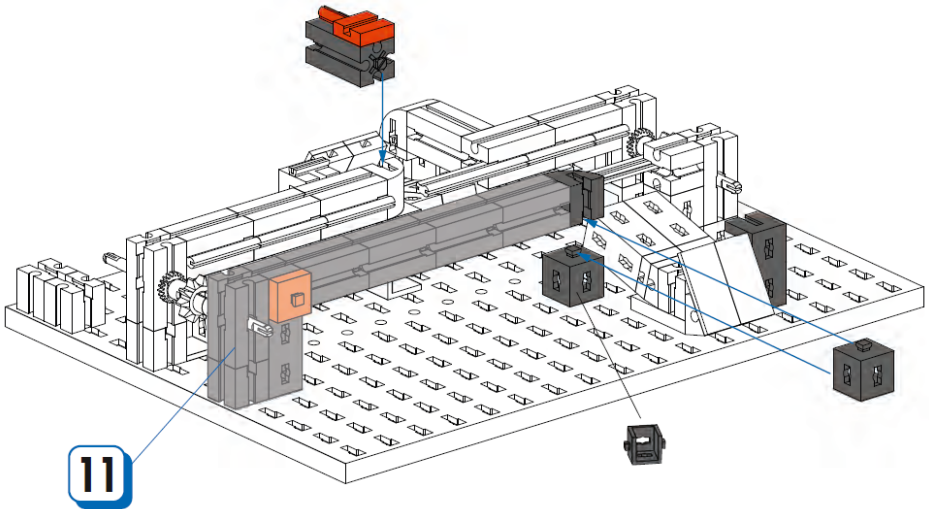
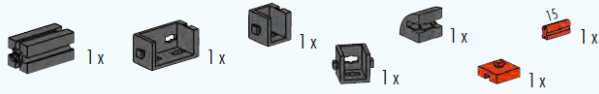
10



11

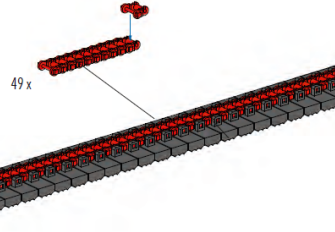


12



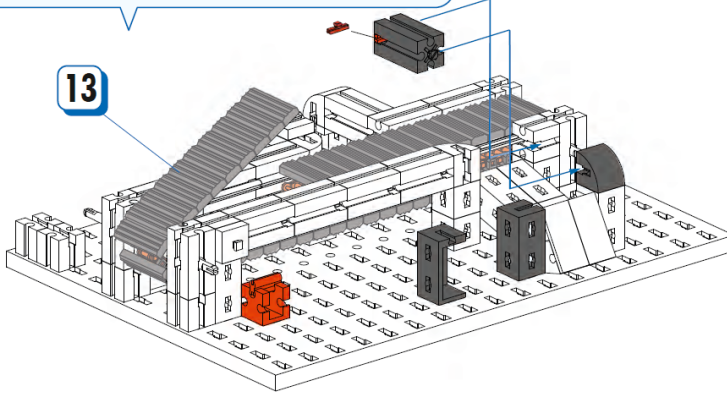
13

49 x 98 x



14

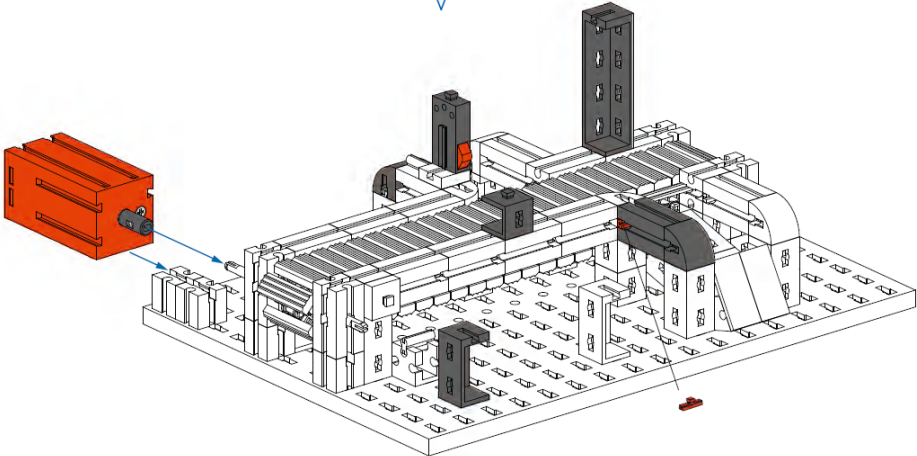
1 x 2 x 1 x 1 x 1 x 1 x



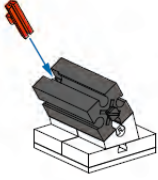
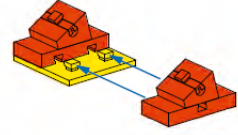
13

15

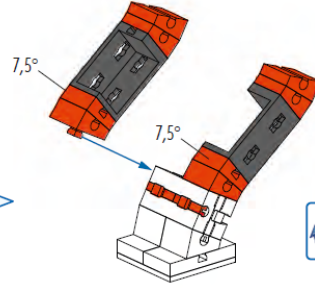
1 x 1 x 1 x 1 x 1 x 1 x 1 x 1 x



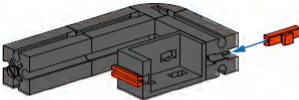
16



17



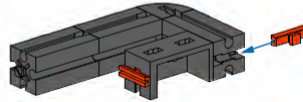
18



19



20



21

19

20

22



1x

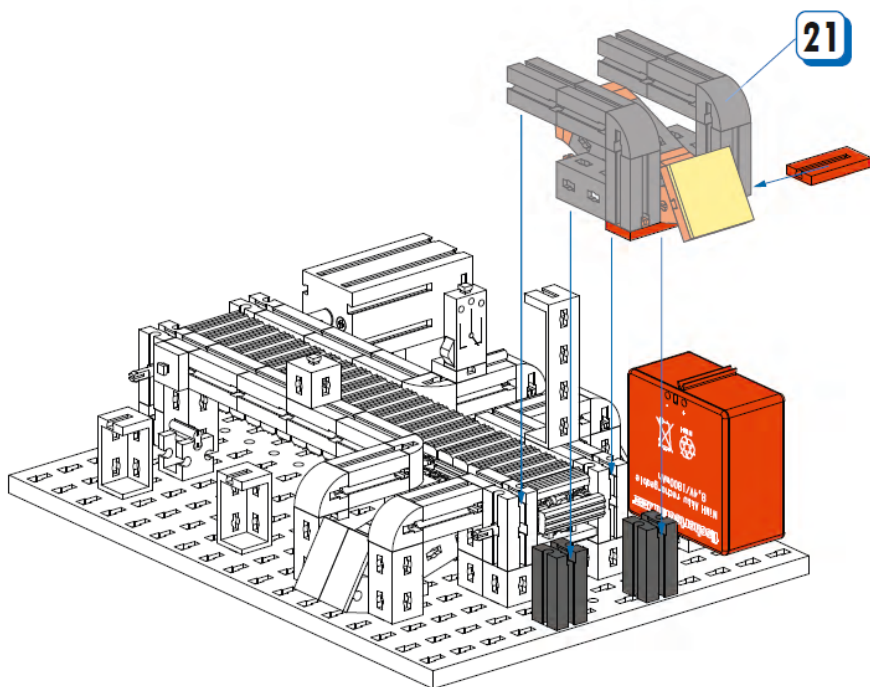


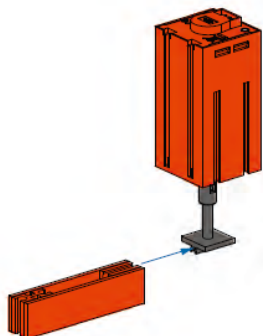
2x



2x

21





23



1x



2x



1x



2x



1x

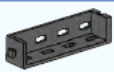
24



1x



2x



1x

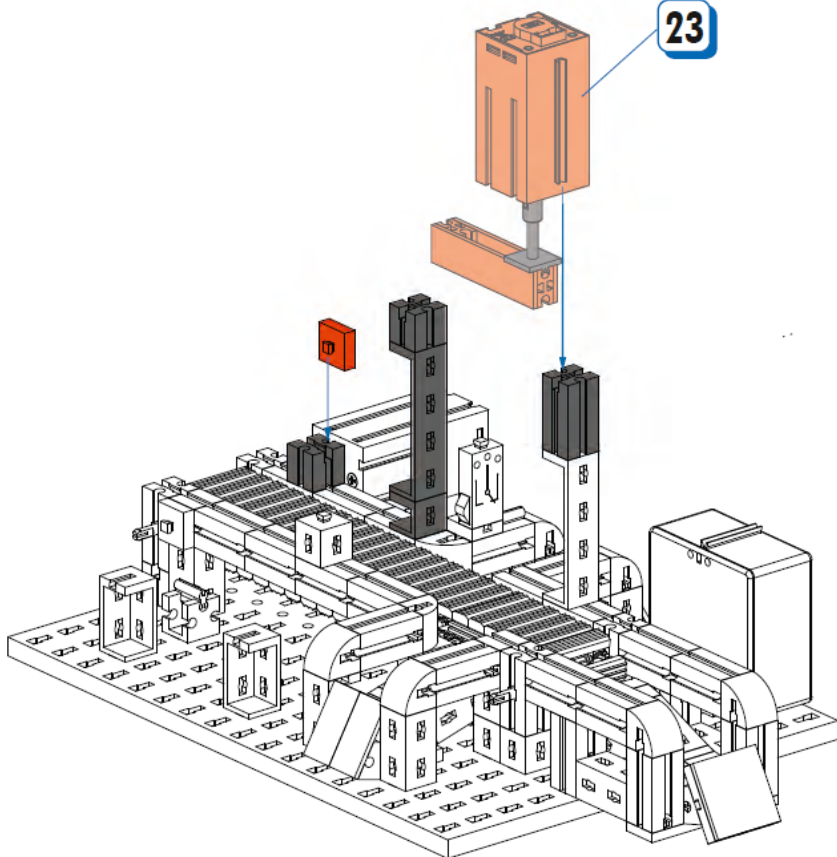


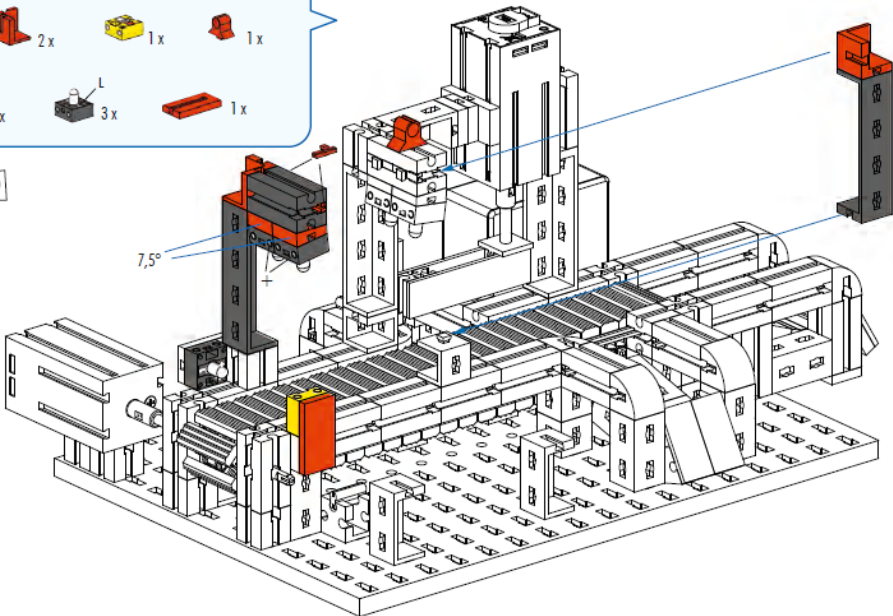
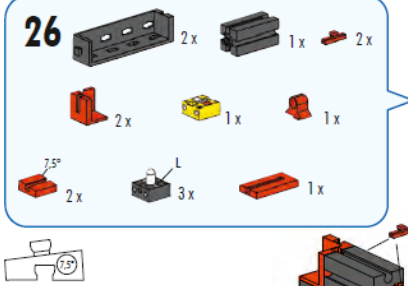
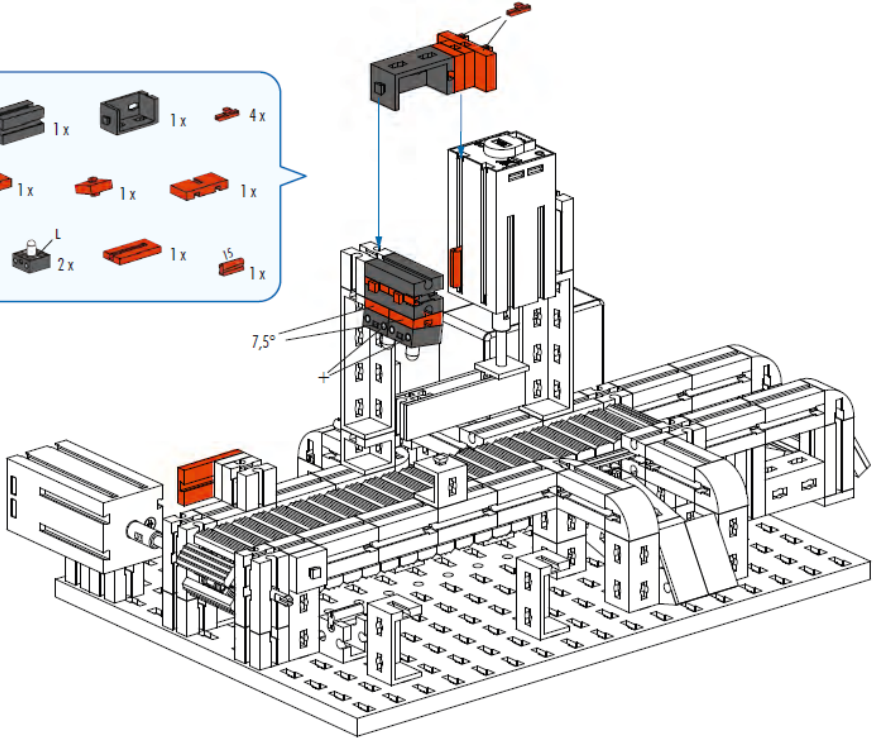
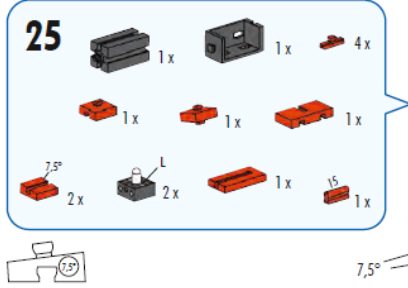
1x



1x

23

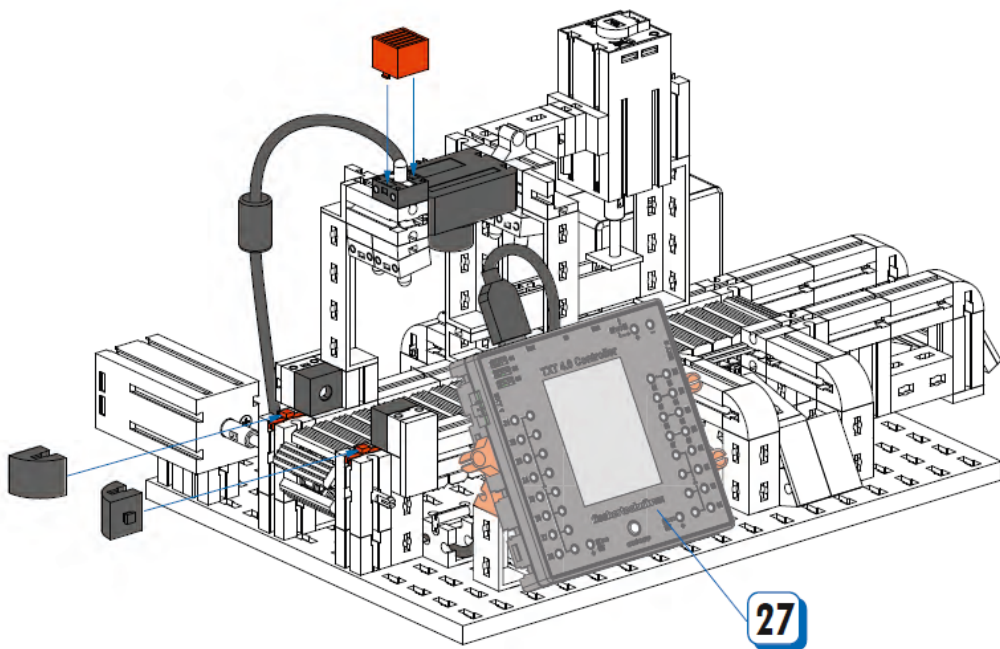
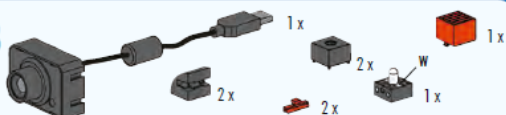
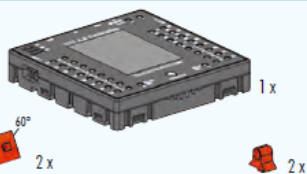
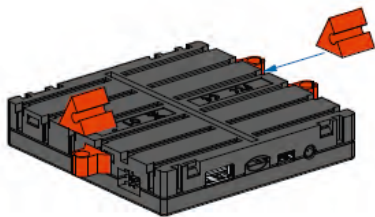




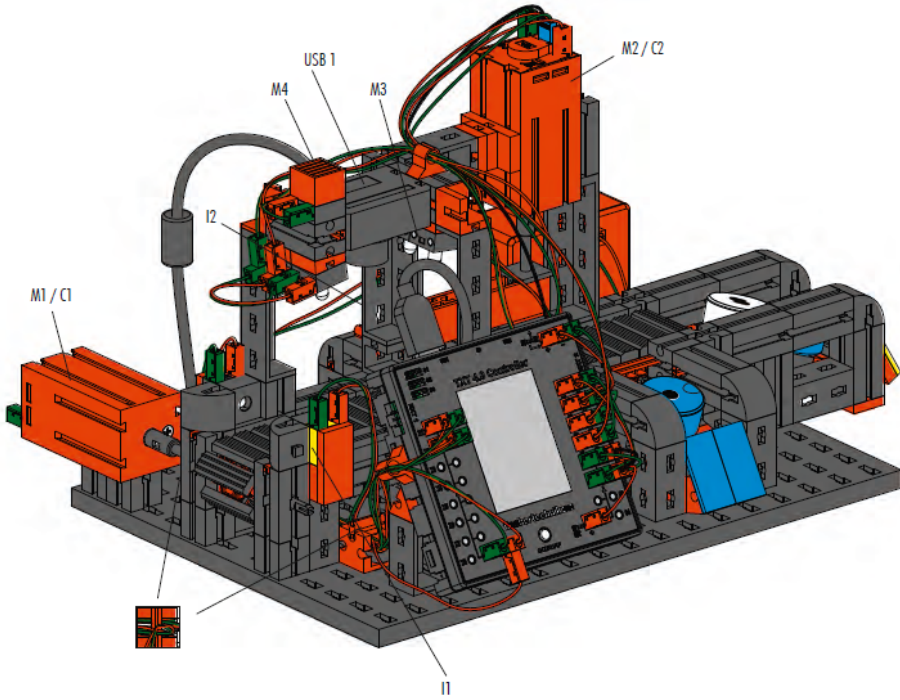
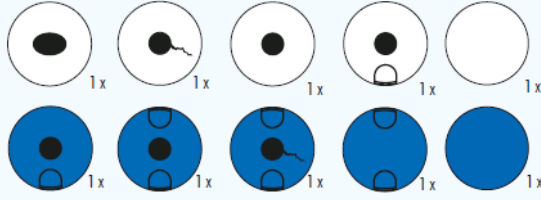
TASARIM

GÖREV

ÇÖZÜM



29



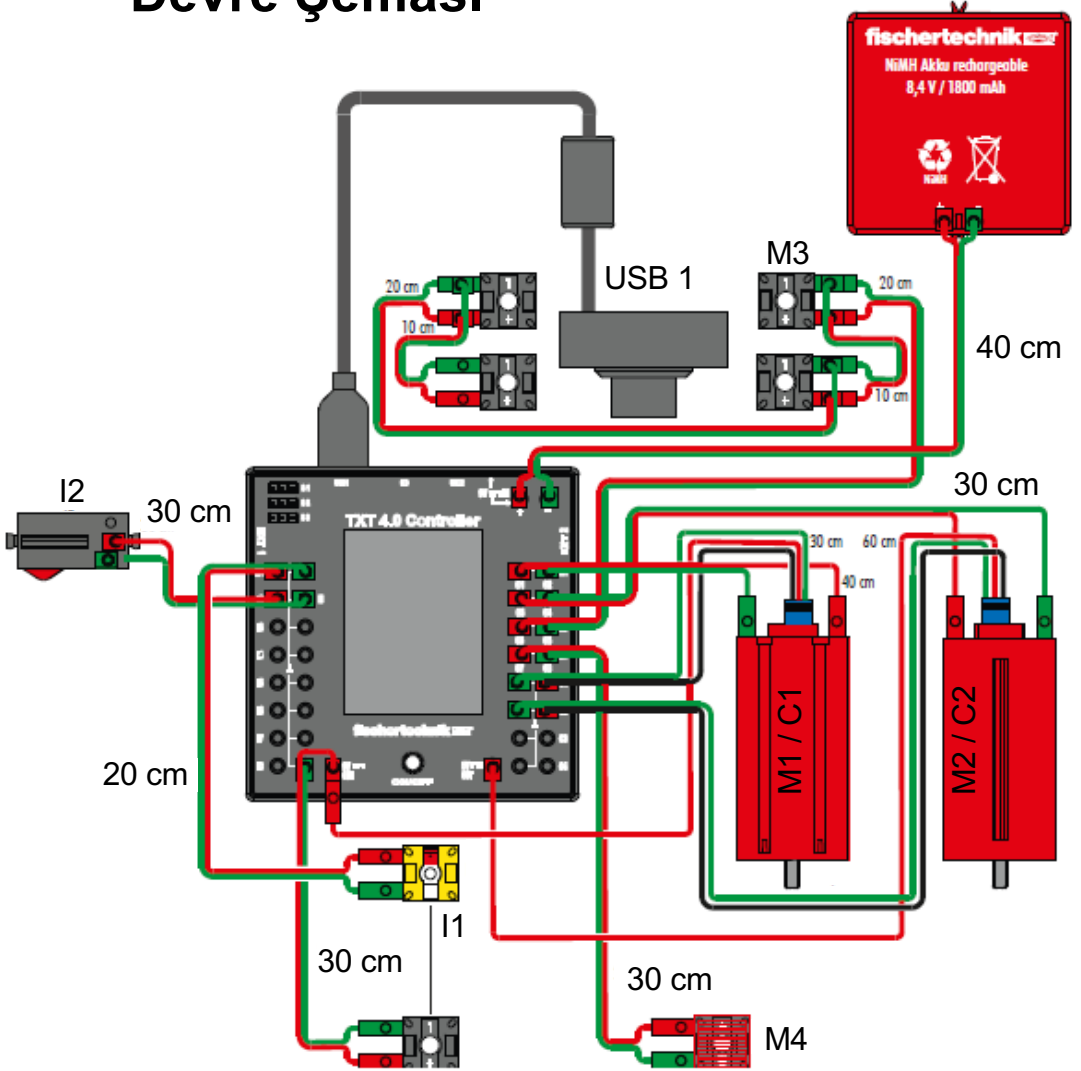
Model - 3

AI Sıralama Hattı



(+) ile gösterilen alana her zaman kırmızı soket takılır.

Devre Şeması





AI Sıralama Hattı İçin Verilen Görevler

Konu

Görevler – AI Sıralama Hattı

AI sıralama hattı modelini düzgün bir şekilde anlayabilmeniz için, öncelikle «makine öğrenimi» modeline ve ilgili görevlere bakmanız mantıklıdır. AI öğrenme sürecinin temelleri orada açıklanmıştır.

AI sıralama hattı, bir konveyör bant aracılığıyla bir kamerayı besler ve ardından iş parçalarının görünümüne göre otomatik olarak farklı alanlara ayrılır. Çekilen resim yapay zeka yardımıyla analiz edilir. Konveyör bant üzerinde iş parçaları kodlayıcı motorlar tarafından belirtilen alanlara gönderilir.

AI sıralama hattı modelinin özelliği, algoritmanın çözüm yolunu önceden öğrenmiş olmasıdır, yani mevcut veriler ve algoritma desenleri temelinde iş parçaları üzerindeki farklı desenleri tanır. AI'nın bir alt alanı olan makine öğrenimi (ML), henüz bilinmeyen görüntüleri analiz etmeyi de mümkün kılar.

Önemli!

Modelin çalışması sırasında, kameranın iş parçalarını iyi tanıyabilmesi için ışık koşullarının iyi ve tutarlı olduğundan emin olun.

TXT 4.0 Kontrol ünitesine güç verin ve ardından ON/OFF düğmesine basılı tutarak TXT 4.0 başlatın. Kontrolcü başladıktan sonra, önce Add_On_Sorting_Line_AI [1] programı kontrolcüye yüklenmelidir. Programı Robo Pro Kodlama'da şu şekilde bulabilirsiniz:

Proje -> Yeni -> Örnek -> STEM Add On AI -> Add_On_Sorting_Line_AI
Hiçbir program yüklenmemişse, “Dosya” altında Add_On_Sorting_Line_AI programını seçin ve “Yükle” aracılığıyla yükleyin.

Programı başlatmak için program başlat düğmesine kısaca basın.
Şimdi herhangi bir iş parçasını iş parçası yuvasına yerleştirebilirsiniz. şimdi bu iş parçası renk ve algılanan özelliklere göre ilgili sıralama bölmesine ayırır.
2 OK (OK) bölmesi ve bir NOK (Not OK) bölmesi arasında ayırım yapılır.

Denetleyici arayüzü

AI sıralama hattı, programın başlatıldığı ve çalışma durumunun görüntülenebildiği bir kontrolör arayüzü sağlar



Programcı başlangıç fişi	Program başlatma düğmesi
Kayıt defteri	Depolama bölgesi
Program durdurma düğmesi	Programı durdur düğmesi

Program dizisi

İş parçası yuvası

Eğer haznede iş parçası yoksa, ışık bariyeri kesintiye uğramaz. Bir iş parçası yerleştirildiğinde, ışık bariyeri kesintiye uğrar ve ardından konveyör bandı yavaşça ileri doğru hareket ettirilir. İş parçası ışık bariyerinden ayrıldığında, konveyör bandı durdurulur. Ardından konveyör bandı yalnızca tanımlanmış mesafelerle hareket ettirilir. İş parçasının konveyör bandındaki kesin konumu artık bilindiğinden, iş parçası kameranın altına hareket ettirilir.

Görüntü analizi

Kamera, görüntü analizi için iş parçasının bir resmini çeker. Bunun makine öğrenimi modeliyle analiz edilebilmesi için, bu model başlatılır ve ardından analiz edilir. Sonuç, algılanan özelliklerin bir listesidir. Bu liste şunları içerir:

1. Güven 0-1 (0-100%)
2. Konum [TopX, TopY, ButonX, ButonY]
3. Tip e.g. [CRACK,BOHO]

Listedeki ilk eleman en büyük olasılıkla tespit edilen elemandır. Bu nedenle, yalnızca bu eleman uygulamada daha fazla işlenir ve renk tanımaya verilir. Renk ve özellik OK ve NOK özelliklerine göre karşılaştırılır ve sınıflandırılır.

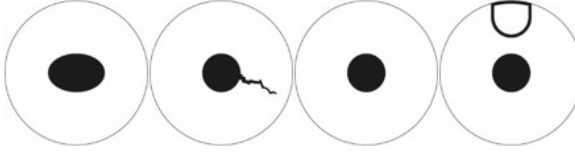
Sıralama

Algılanan özelliklere ve önceden tanımlanmış değerlere göre, iş parçası 3 bölmeden birine sınıflandırılır. Beyaz-OK, mavi-OK veya NOK (sarı). İş parçası konveyör bant tarafından taşınır ve ilgili bölmeye yönlendirilir.

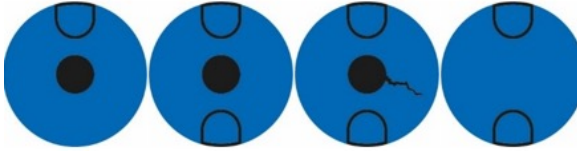
Deneyisel görevler

1. Renk başına 4 iş parçası çeşidi vardır. Sıralama hattı her birinden yalnızca 1 iş parçasını OK (OK veya geçti) olarak algılar.

Şekil 1 ve Şekil 2'de gösterildiği gibi etiketlerle 4 beyaz ve 4 mavi iş parçası hazırlayın ve ardından modelin hangi iş parçaları için eğitildiğini bulun.



Şekil 1: Beyaz iş parçaları



Şekil 2: Mavi iş parçaları

2. Diğer iş parçalarını kullanarak kendi yapay zekanızı sıralama hattı için eğitin.

Daha fazla bilgi

- [1] AI program model AI sıralama hattı: Add_On_AI
- [2] Talimatlar TXT 4.0 – Setting SSH
- [3] Python kurulum programı, <https://www.python.org/>



AI Sıralama Hattı İçin Verilen Çözümler

Çözüm sayfası – AI Sıralama Hattı

Görev 1:



Görev 2:

Robo Pro Coding'de örnek bir çözüm aşağıdadır:

Project -> New -> Example -> STEM Add On AI ->
A2_Sorting_New_AI

Ek

Daha fazla bilgi

- [1] AI program model AI sorting line: Add_On_AI
- [2] Talimatlar TXT 4.0 – Setting SSH
- [3] Python kurulum programı, <https://www.python.org/>