

Çok Markalı Servis İstasyonları için Yapay Sinir Ağları ile Görüntü Tabanlı Araç Marka ve Modeli Tanıma Yazılımı

Gürkan TUNA¹

Trakya Üniversitesi, Edirne Teknik Bilimler Meslek Yüksekokulu, Edirne.

Özet

Bu çalışmada çok markalı araç servis istasyonları için görüntü tabanlı araç marka ve modeli tanıma uygulaması geliştirilmiştir. Akıllı otoyol sistemleri için geliştirilmiş uygulamalardan farklı olarak bu uygulamada tablet pc gibi düşük performanslı sistemlerde ve farklı işletim sistemleri yüklü kişisel bilgisayarda hızlı ve etkili bir biçimde çalışabilecek bir araç tanıma sistemi geliştirilmesi amaçlanmıştır. Bu nedenle uygulama Java ile geliştirilmiştir. Ağı eğitirken iyi bir eğitim setinin kullanılması önemlidir. Test sonuçlarından elde edilen verilere göre % 99'un üstünde başarı elde edilmiştir.

Anahtar kelimeler: Araç Tanıma, YSA, Görüntü İşleme, Prewitt Operatörü, Geriye Yayılım Algoritması.

A Vision Based Car Brand and Model Recognition Software with Artificial Neural Networks for Multi-Brand Service Stations

Abstract

In this study, a vision based vehicle type recognition software for multi-brand service stations has been developed. Different from the applications developed for intelligent motorway systems, in this application it is aimed to develop a vehicle recognition system to run on low performance systems like tablet PCs and PCs with different operating systems. So the application has been developed with Java. It is important to use a good training set while training the network. According to the data acquired from the test results, a success over % 99 has been attained.

Keywords: Vehicle Recognition, ANN, Image Processing, Prewitt Operator, Back Propagation Algorithm.

¹ Gürkan TUNA, gurkantuna@trakya.edu.tr

1. Giriş

Araç tipi tanıma uygulamaları akıllı otoyol sistemleri başta olmak üzere farklı alanlarda kullanılmaktadır. Araç tipi tanıma uygulamaları görüntü işleme teknikleri ile gerçekleştirilmekle beraber bu uygulamalarda YSA yaygın olarak kullanılmaya başlanmıştır. Farklı YSA tipleri kullanılan araç tipi tanıma uygulamaları geliştirilmiş olmasına rağmen genelde kullanılan ağ tipi İleri Beslemeli Geriye Yayılımlı YSA'dır.

Bu çalışmada çok markalı araç servis istasyonları için görüntü tabanlı araç marka ve modeli tanıma uygulaması geliştirilmiştir. Yazılım servis istasyonları için geliştirilmiş olup resim dosyalarından araç tanıma yapmaktadır. Akıllı otoyol sistemleri [1-5] için geliştirilmiş olan uygulamalar ücretlendirme esaslı yazılımlar olduğu için bu yazılımlar araçları marka ve model olarak sınıflandırmamakta, boyutlarına göre araba, kamyonet, kamyon, otobüs olarak sınıflandırmaktadırlar. Bu çalışma kapsamında geliştirilmiş olan uygulama araçları markalarına göre sınıflandırmakta ve yazılımın servis istasyonlarında kullanılması öngörülmektedir.

Akıllı otoyol sistemleri için geliştirilmiş uygulamalardan [1-5] farklı olarak bu uygulamanın düşük performanslı sistemlerde ve farklı işletim sistemleri yüklü kişisel bilgisayarda hızlı ve etkili bir biçimde çalışabilmesi amaçlanmıştır. Uygulama bu gereksinimleri sağlayan bir dil olduğu için Java ile geliştirilmiştir.

Makalede önce nesne tanıma, daha sonra yapay sinir ağları ele alınarak yapay sinir ağlarının nesne tanıma uygulamalarındaki önemi vurgulanmış ve geliştirilen uygulamada kullanımları açıklanmıştır. Makale Giriş bölümü ile başlamakta ve Nesnelerin tanınması bölümüyle devam etmektedir. Nesnelerin tanınması bölümünde örüntü tanıma, kenar tespiti ve yapay sinir ağlarının örüntü tanıma uygulamalarında kullanımı ele alınmıştır. Üçüncü bölümde geliştirilen yazılımın çalışma tekniği hakkında bilgiler verilerek bir örnek uygulama ile yazılımın çalışması gösterilmiştir. Makale Sonuçlar ve öneriler bölümü ile tamamlanmaktadır.

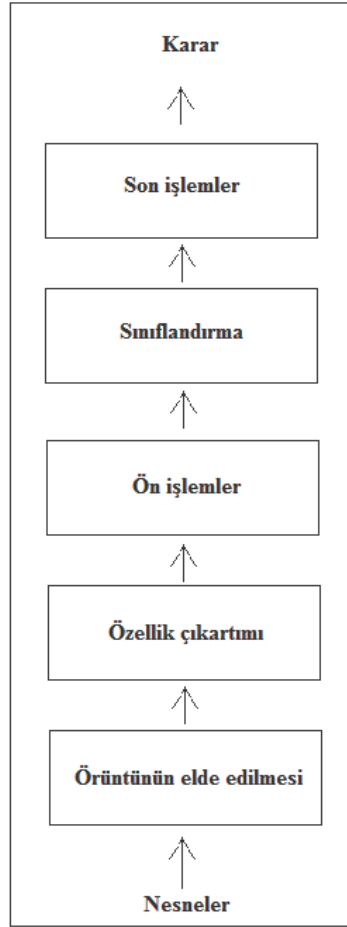
2. Nesnelerin tanınması

2.1 Örüntü Tanıma

Örüntü tanıma yapay zekânın alt dalı olan makine öğrenmesinin bir koludur. Örüntü tanımada amaç makinenin gördüğü örüntülerin özniteliklerini kullanarak daha önceden eğitildiği ya da öğrendiği şekilde bu örüntüleri sınıflandırmasını sağlamaktır. Şekil 1, örüntü tanıma adımlarını göstermektedir.

Örüntü tanımanın birinci aşaması örüntülerin elde edilmesidir. Farklı teknikler ile örüntüler elde edilebilir. İkinci aşama özellik çıkartımıdır. Bu aşamada amaç örüntüler arasındaki ayırt edici özelliklerin tespitidir. Araç tanıma uygulamasındaki amaç görüntülerden aracın tanınması olduğundan aracın kenar noktalarından hareketle aracın şeklinin bulunması amaçlanmaktadır. Dolayısıyla her aracın kenar noktalarının x ve y koordinatlarındaki değerlerinin elde edilmesi gerekmektedir. Bu aşamada kenar tespiti işlemi yapılacaktır. Üçüncü aşamada gerekirse giriş verilerine ön işlemler uygulanır. Dördüncü aşama sınıflandırmadır. Bu aşamada örüntü bir sınıfa atanır. Araç tanıma uygulamasının bu aşamasında İleri Beslemeli Geriye Yayılımlı YSA (İBGYYSA) ile

sınıflandırma yapılacaktır. Sınıflandırma yapılabilmesi için ağın önceden bir eğitim seti ile eğitilmesi gerekmektedir. Örüntü tanımanın son aşamasında gerekliyse karar verme aşamasından önce performansı arttırmak için son işlemler uygulanır.



Şekil 1. Örüntü tanıma adımları

Araç tanıma uygulamasının en önemli aşamaları araçların resimlerinden kenar noktalarının tespit edilmesi ve farklı araçlar kullanılarak ağın eğitilmesidir. Böylece sistem önceden girilmiş olan bir eğitim setine göre YSA kullanarak resmi girilen aracın hangi model olduğunu otomatik olarak tespit edecektir. Sistem resmi girilen aracın yüzde kaç oranında hangi model olabileceğini de belirtmektedir.

2.2 Kenar tespiti

Kenar tespiti görüntü işlemenin özellik bulma ve özellik çıkartma aşamalarında kullanılan bir yöntemdir. Kenar tespiti dijital bir görüntüde parlaklığının keskin olarak değiştiği yerler olan kesikliklerin olduğu noktaların bulunmasıdır [7]. Farklı yöntemler ile kenar tespiti yapılabilir. Bu uygulamada kenar tespiti için Prewitt operatörü kullanılmıştır.

Kenar tespitindeki en önemli kavram gradyendir. Gradyenin gösterimi aşağıdaki gibidir.

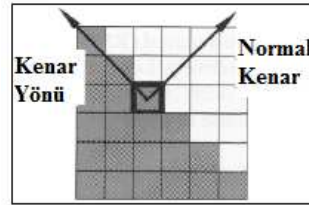
$$\nabla f = \begin{pmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{pmatrix}$$

$$\text{magnitude}(\nabla f) = \sqrt{\left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2} = \sqrt{M_x^2 + M_y^2}$$

Hesaplamaları azaltmak için genelde aşağıdaki gibi kabul edilir.

$$\text{magnitude}(\nabla f) \approx |M_x| + |M_y|$$

Gradyen değerinin büyüklüğü bir kenarın gücü hakkında bilgi verir. Gradyenin yönü daima kenara diktir. Buradaki temel fikir x ve y yönlerinde türevleri hesaplamak, gradyen değerini bulmak ve eşik değerine göre kontrol etmektir. Şekil 2, kenar ve gradyen ilişkisini göstermektedir.



Şekil 2. Kenar-gradyen ilişkisi

Gradyen aşağıdaki gibi hesaplanmaktadır.

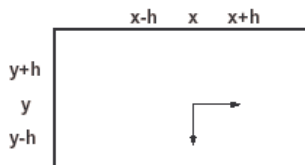
$$\frac{df}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x, y)}{h}$$

$$\frac{df}{dy} = \lim_{h \rightarrow 0} \frac{f(x, y+h) - f(x, y)}{h}$$

$$\frac{df}{dx} = \frac{f(x+h_x, y) - f(x, y)}{h_x} = f(x+1, y) - f(x, y) \quad (h_x=1)$$

$$\frac{df}{dy} = \frac{f(x, y+h_y) - f(x, y)}{h_y} = f(x, y+1) - f(x, y) \quad (h_y=1)$$

Piksel-koordinat yaklaşımı kullanılırsa (j, x yönünü ve i negatif y yönünü göstermektedir) aşağıdaki gibi hesaplamalar yapılır.



$$\frac{df}{dx} = f(i, j+1) - f(i, j)$$

$$\frac{df}{dy} = f(i-1, j) - f(i, j)$$

Aşağıda Z5'in gradyen değerine yaklaşmak için hesaplamalar yapılmaktadır.

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

$$\frac{dI}{dx} = z_6 - z_5$$

$$\frac{dI}{dy} = z_5 - z_8$$

$$\text{magnitude}(\nabla I) = \sqrt{(z_6 - z_5)^2 + (z_5 - z_8)^2}$$

Araç tanıma uygulamasında kenar tespiti için Prewitt kenar tespit yöntemi kullanılmaktadır. Yöntemi çalışması aşağıdaki gibidir.

$$\begin{array}{ccc} a_0 & a_1 & a_2 \\ a_7 & [i,j] & a_3 \\ a_6 & a_5 & a_4 \end{array}$$

Kısmi türevler hesaplanırsa;

$$M_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$M_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2)$$

C sabiti merkezdeki piksellere ağırlık verildiğini göstermektedir. C = 1 alınırsa Prewitt operatörü elde edilir. Operatör yatay ve dikey değişimleri tespit edebilmek için aşağıdaki maskları kullanılmaktadır [8].

$$M_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad M_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Şekil 3 ve Şekil 4 Prewitt operatörü kullanılarak yapılan bir kenar tespit uygulamasını göstermektedir.



Şekil 3. Prewitt operatörü uygulanacak resim



Şekil 4. Prewitt operatörü ile kenar noktalarının tespiti

2.3 Yapay sinir ağları ve örüntü tanıma işleminde kullanımları

YSA'lar insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir. Bu yetenekleri geleneksel programlama yöntemleri ile gerçekleştirmek oldukça zordur veya mümkün değildir. Birçok YSA tipi bulunmakla birlikte, en çok kullanılan sinir ağı yapısı, İleri Beslemeli Geri Yayılımlı modeldir. Belirsiz, gürültülü ve eksik bilgilerin işlenmesinde yapay sinir ağları başarıyla kullanılmaktadır. YSA'ların avantajları lineer olmayan özelliğe sahip olmaları, öğrenme kabiliyetleri, genelleme yapabilmeleri, adaptasyon kabiliyetleri, gürültüye karşı toleranslı olmaları ve donanım olarak gerçekleştirilebilmeleri olarak sıralanabilir.

Temel bir YSA hücresi biyolojik sinir hücresine göre çok daha basit bir yapıya sahiptir. Bir YSA hücresinde girişler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkışlar bulunur. İleri beslemeli YSA'da hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan gizli katmandaki hücrelere iletir. Bilgi, gizli katmanda ve çıkış katmanında işlenerek ağ çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli üç katmanlı YSA'nın gizli katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği

gösterilmiştir. YSA'ların eğitiminde geriye yayımlı öğrenme algoritması etkin olarak kullanılmaktadır. İBGYYSA'larda transfer fonksiyonları kullanılarak her bir katmanın giriş durumları elde edilir. Hem girişler hem de çıkışlar hakkında bilgi sahibi olduğu için son katmandaki ağırlıklar ayarlanır ve hata geriye yayılır. Böylece önceki katmandaki ağırlıklar güncellenir ve hata hesaplanır. Bu süreç giriş katmanına kadar devam eder [6].

Bu çalışma kapsamında geliştirilen araç tanıma uygulamasında İBGYYSA'nın görevi verilen bir eğitim setindeki araçların kenar tanıma yöntemiyle elde edilen kenar noktalarının x ve y koordinatlarına göre girilecek aracın modeline karar vermektir. Karar verme adımı olan tanıma işleminden önce eğitim setleri girilir, "Kenar belirleyici nokta" olarak adlandırılan iki kenar noktası arası değerler bulunur ve ağ eğitilir.

3. Araç tanıma yazılımı

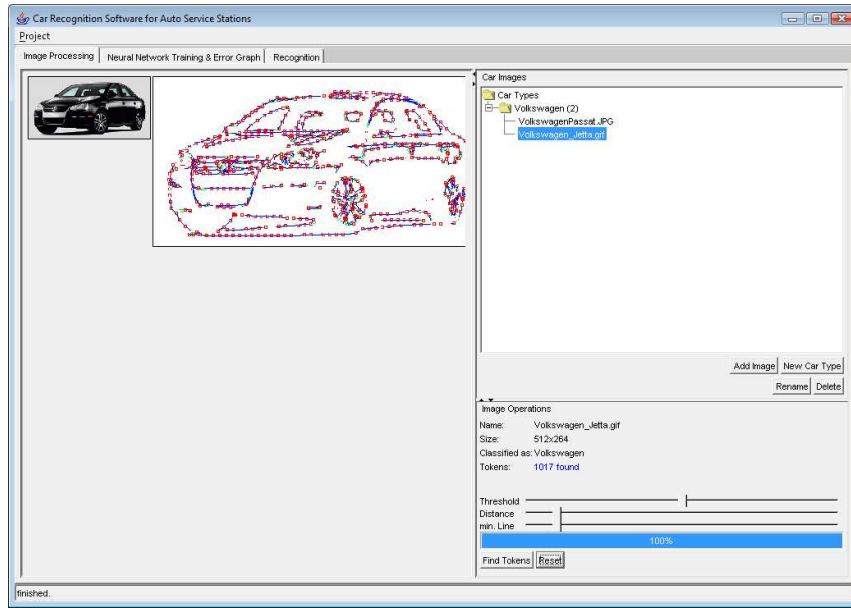
Araç tanıma yazılımı çok markalı servis istasyonlarının otomasyonu için geliştirilmiştir. Yazılımın amacı aracın servis noktasına kaydının yapılmasından sonra servis alanına giriş sırasında aracın sistem tarafından tanınarak araç sahibini ilgili servis noktasına yönlendirilmesidir. Böylece işlemler hızlanacak ve otomatik hale gelecektir.

Yazılım üç modülden oluşmaktadır. Bu modüller Image Processing (Görüntü İşleme), Neural Network Training & Error Graph (Yapay Sinir Ağı Eğitimi ve Hata Grafiği) ve Recognition (Tanıma) modülleridir.

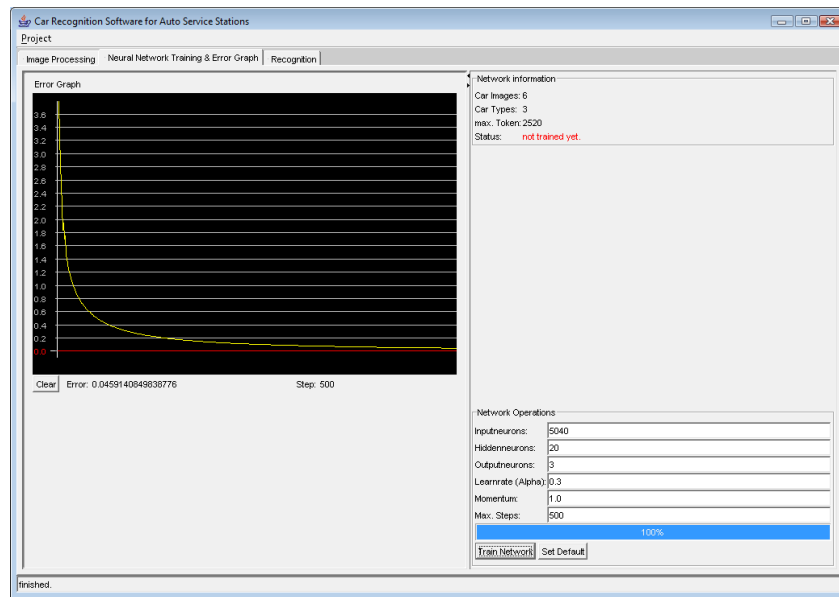
Image Processing modülü eğitim setini oluşturacak araçlara ait .jpeg, .gif veya .tiff uzantılı resim dosyalarının programa girilmesini ve resimlerdeki kenar noktalarından enar belirleyici noktaların elde edilmesini sağlayan modüldür. Bir kenar belirleyici nokta olduğu iki noktanın koordinatları, kosinüs ve sinüs açılarıyla temsil edilir. Bu modül araç tanıma işleminde ilk kullanılması gereken modüldür. Image Processing modülünde kenar tespitinde işlem yapılacak minimum mesafe, kenar belirleyici noktalar arası uzunluk ve Prewitt yöntemi kullanılarak yapılan kenar tespitindeki eşik değeri belirtilir. Şekil 5, modülün ekran görüntüsünü göstermektedir.

Neural Network Training & Error Graph modülü Image Processing modülü ile programa girilerek kenar belirleyici noktaları bulunmuş olan eğitim seti kullanılarak ağın eğitilmesini ve hata grafiğinin elde edilmesini sağlayan modüldür. Ağın eğitilmesi için en az 5 adet araç görüntüsünün Image Processing modülünde girilmiş ve kenar belirleyici noktalarının bulunmuş olması gereklidir. Bu modülde giriş katmanındaki nöron sayısı, çıkış katmanındaki nöron sayısı, gizli katmandaki nöron sayısı, öğrenme oranı, momentum ve maksimum adım sayısı belirtilir. Şekil 6, modülün ekran görüntüsünü göstermektedir.

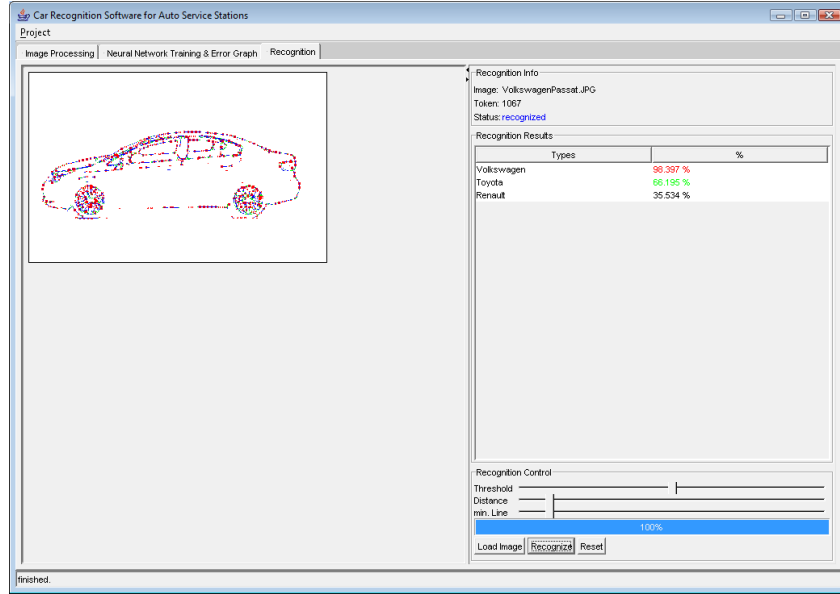
Recognition modülü ise girilecek bir araç görüntüsünün yazılıma daha önceden girilmiş olan eğitim seti kullanılarak hangi araçlara ait olabileceğini bulan modüldür. Bu modülde kenar tespitinde işlem yapılacak minimum mesafe, kenar belirleyici noktalar arası uzunluk ve Prewitt yöntemi kullanılarak yapılan kenar tespitindeki eşik değeri belirtilir. Şekil 7, modülün ekran görüntüsünü göstermektedir.



Şekil 5. Image processing modülü



Şekil 6. Neural network training & error graph modülü



Şekil 7. Recognition modülü

3.1 Araç tanıma yazılımı hakkında teknik bilgiler

Yazılım Fedora işletim sisteminde Borland JBuilder 2005 Foundation ile Java dilinde geliştirilmiştir.

Kenar Tespiti: Programın en önemli aşamalarından birisi girilmiş olan resim dosyasından kenar belirleyici nokta adı verilen değerlerin bulunmasıdır. Bu kenar belirleyici noktalar yapay sinir ağının hesaplamalarında temel teşkil etmektedir. Kenar tespiti için farklı yöntemler bulunmaktadır. Araç tanıma yazılımında kenar tespiti için Prewitt operatörü kullanılmaktadır. Prewitt kenar tespiti yönteminde yüksek gri seviyeleri iki nesne arasındaki kenarı işaret etmektedir [8]. Prewitt operatörü tarafından kullanılan filtre gradyen değerini hesaplamak için iki adet 3x3 lük şablon kullanılmaktadır.

$$\begin{array}{ccc}
 -1 & 0 & 1 \\
 1 & 1 & 1 \\
 -1 & 0 & 1 \\
 -1 & 0 & 1 \\
 -1 & -1 & -1 \\
 X & & Y
 \end{array}
 \begin{array}{l}
 +-----+ \\
 | a1 \ a2 \ a3 | \\
 | a4 \ a5 \ a6 | \\
 | a7 \ a8 \ a9 | \\
 +-----+
 \end{array}$$

a1..a9 : filtre penceresindeki her bir pikselin gri renk seviyeleri

$$X = -1*a1 + 1*a3 - 1*a4 + 1*a6 - 1*a7 + 1*a9$$

$$Y = 1*a1 + 1*a2 + 1*a3 - 1*a7 - 1*a8 - 1*a9$$

$$PREWITT \ gradient = \sqrt{x^2 + y^2}$$

Bütün pikseller filtrelenir. Kenar noktaları yakınılarında bulunan piksellerin filtrelenmesi için kenar piksel değerleri kopyalanarak artırılır.

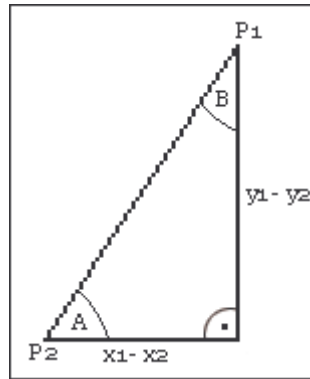
İnceltme: İnceltme algoritması eşik değerine bağlı olan kenarı tek çizgi çerçevesine indirgemek için kullanılır. Böylece kenar belirleyici noktalara bağlı tanıma yapmak kolaylaşır. Programdaki inceltme algoritması resim dosyasını ardı ardına işleyerek ve mevcut piksel durumunu belirli örüntülerle karşılaştırarak bulunmuş olan kenar çizgilerini bir piksel genişliğe indirir.

Geriye Yayımlı Yapay Sinir Ağı: Yapay Sinir Ağı programda önemli bir görev üstlenmektedir. Araç resimlerinden elde edilen kenar belirleyici noktalar yapay sinir ağının giriş verilerini oluşturmaktadır. Kenar belirleyici noktalar koordinat noktaları ve bu noktalar arasındaki kosinüs ve sinüs değerlerinden oluşmaktadır. Ağ yapısı için giriş katmanındaki nöronların sayısı kenar belirleyici noktaların sayısının iki katıdır. Programda çıkış nöronlarının sayısı araç markalarının sayısına bağlı olarak seçilir.

3.1.1 Yazılımın çalışma tekniği

Görüntünün İşlenmesi: Programın bu aşamasında farklı marka ve modellerden araçlara ait görüntüler eklenerek kenar tespiti ve inceltme algoritmasıyla kenar belirleyici noktalar bulunur ve her bir kenar belirleyici noktaya ait kosinüs ve sinüs değerleri hesaplanır.

Kenar belirleyici nokta: Programın işleyişi sırasında kullandığı temel veriler kenar belirleyici noktalardır. Kenar belirleyici noktaların elde edilmesinde amaç araç resimlerinden yapay sinir ağına giriş değerleri sağlamaktır. Kenar belirleyici noktaların koordinatları, kosinüs ve sinüs açıları tanıma işleminin temel verilerini oluşturmaktadır. Şekil 8'de kenar belirleyici noktanın daha ayrıntılı bir görüntüsü bulunmaktadır. Elde edilen tüm kenar belirleyici noktalar yapay sinir ağının hesaplamalarında kullanılacaktır.



Şekil 8. Kenar belirleyici noktayı oluşturan iki nokta

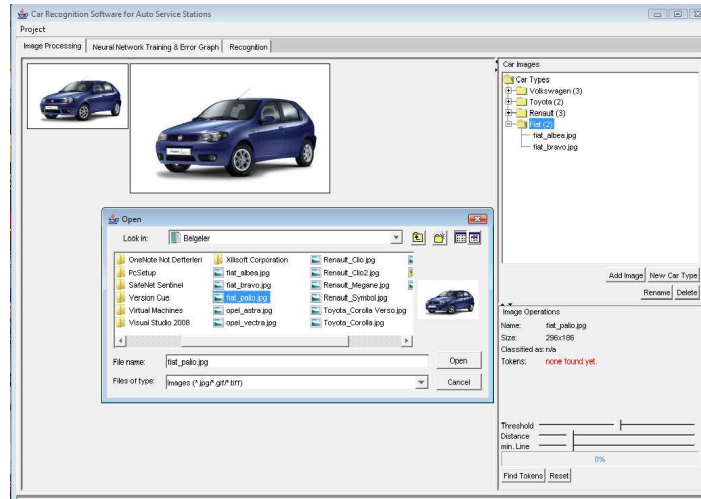
Eğitim: Bu aşama geriye yayılım algoritması ile gerçekleştirilir. Algoritma devir sayısı kere tekrar eder. Başlangıçta ağırlık dizilerine rastgele değerler atanır. Görüntü işlemeden gelen giriş verileri ile ağ eğitilir. Her bir devirde ağırlıklar güncellenir. Eğitim işlemi belirli bir hata değerinin altına düşünceye kadar devam eder.

Tanıma: Bu aşamada tanınacak olan aracın resmi programa yüklenir. Ağ tarafından aracın hangi markaya ait olduğu belirlenir.

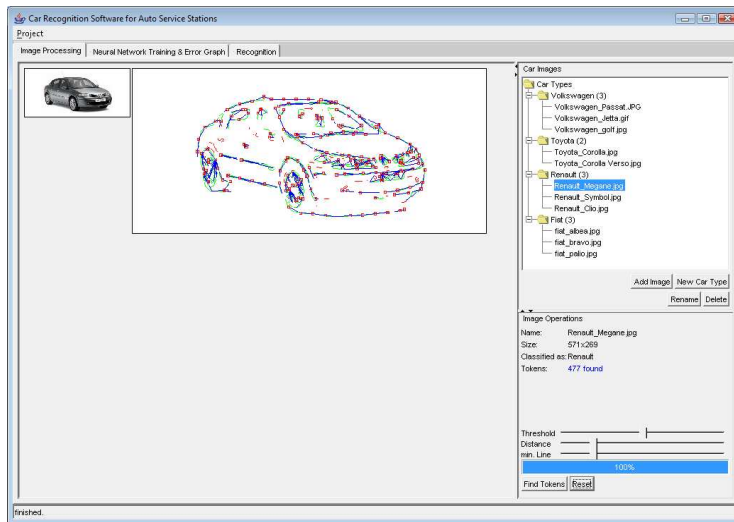
Geriye Yayılım: Bu aşama girilen eğitim setlerine göre ağırların eğitilmesini gerçekleştirir. Giriş verilerinden eğitim örneklerini alır ve karşılık gelen çıkış değerlerini elde eder. Gerçek değer ile arzu edilen değer arasındaki hatayı da hesaplar. Bu hata değeri ağırlıkların ayarlanmasında kullanılır.

3.2 Araç tanıma yazılımının kullanımı

Yazılım çalıştırıldığında ekrana ilk olarak Image Processing modülü gelir. Bu modülde araç marka ve modellerinin resim dosyalarından oluşan eğitim setinin yüklenmesi ve kenar belirleyici noktaların bulunması işlemi gerçekleştirilir. Şekil 9, 7 farklı markanın toplam 33 modeli için eğitim setinin yüklenmesi işlemini ve Şekil 10 ise kenar belirleyici noktaların bulunması işlemini göstermektedir. Otomobil, kamyonet, kamyon, otobüs gibi farklı tür araçlarla eğitim yapılması durumunda uygulama çok az nöronla bile aracın marka ve modelini belirleyebilmektedir. Dolayısıyla uygulamada sadece otomobiller kullanılmıştır. 33 otomobil resmi eğitim için kullanılmıştır. Tanıma işlemi ise 3 farklı otomobil resmi ile denenmiştir.

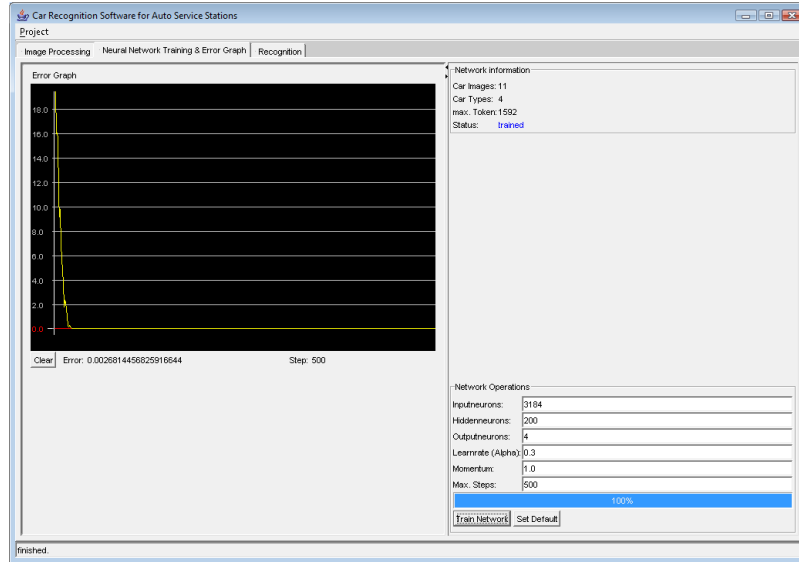


Şekil 9. Araçların yüklenmesi



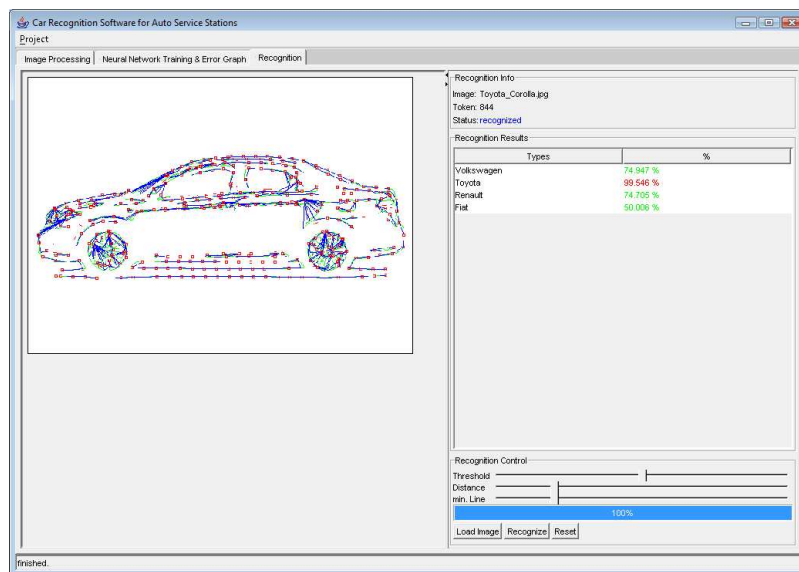
Şekil 10. Kenar belirleyici noktaların bulunması

İkinci aşamada kenar belirleyici noktaları bulunmuş olan giriş seti için ağın eğitilmesi işlemi gerçekleştirilmiştir. Bu işlem Neural Network Training & Error Graph modülü ile gerçekleştirilmiştir. Bu aşamada ağın eğitiminde kullanılan giriş nöronlarının sayısı, gizli katmanda bulunan nöronların sayısı, çıkış nöronlarının sayısı, öğrenme oranı, momentum ve maksimum adım değerleri belirlenmiştir. Şekil 11, ağın eğitilmesini göstermektedir.



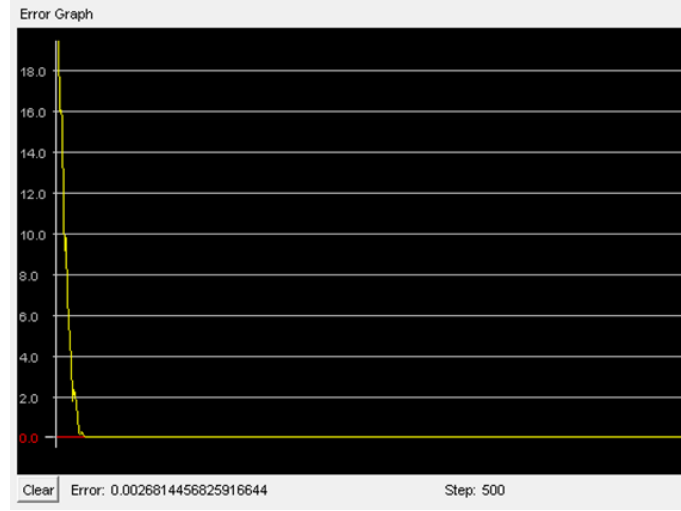
Şekil 11. Ağın eğitilmesi

Son aşamada tanıma işlemi gerçekleştirilmiştir. Recognition modülü kullanılarak eğitim setinde girilmiş olan Toyota Corolla, Volkswagen Passat ve Renault Megane marka otomobillere ait resimler yüklenmiş ve tanıma işlemi gerçekleştirilmiştir. Tanıma işlemi sonunda elde edilen başarı oranları Toyota Corolla için % 99.546, Volkswagen Passat için % 99.784 ve Renault Megane için % 98.351'dir. Bu işlem Şekil 12'de gösterilmektedir. Araç tanıma işlemi sonunda, tanıma işlemi uygulanan aracın olabileceği alternatif markalar ihtimalleriyle birlikte belirtilmektedir.



Şekil 12. Aracın tanınması

Uygulama katmanlardaki nöron sayısının, öğrenme oranının, momentum değerinin ve maksimum adım sayısının kullanıcı tarafından belirlenmesine izin vermektedir. Örnek çalışmada giriş katmanındaki nöron sayısı 3184, gizli katmandaki nöron sayısı 200, çıkış katmanındaki nöron sayısı 4, öğrenme oranı 0.3, momentum 1.0 ve maksimum adım sayısı 500 kullanılmıştır. Örnek uygulamada eğitim sonunda oluşan hata değeri 0,00268144'tür. Daha iyi bir eğitim seti kullanılarak ve parametrelerde değişim yapılarak daha düşük hata değerleri elde edilebilir. Şekil 13, hata değişimini göstermektedir.



Şekil 13. Hata değişim grafiği

4. Sonuçlar ve öneriler

Bu çalışmada çok markalı araç servis istasyonları için görüntü tabanlı araç marka ve modeli tanıma uygulaması geliştirilmiştir. Uygulamanın düşük kapasiteli sistemlerde ve farklı işletim sistemleri yüklü kişisel bilgisayarda hızlı ve etkili bir biçimde çalışabilmesi amaçlandığı için uygulama JAVA ile geliştirilmiştir.

Yazılım servis istasyonları için geliştirilmiş olup resim dosyalarından araç tanıma yapmaktadır. Yazılım farklı araç marka ve modelleri ile denenmiştir. Bozucu etkilerin az olduğu görüntü dosyalarında çok yüksek hassasiyetle aracın hangi marka olduğunu tespit edebilmektedir. Test sonuçlarından elde edilen verilere göre % 99'un üstünde başarı elde edilmiştir. Uygulamanın yüksek tanıma oranı elde edilebilmesi için iyi bir eğitim seti kullanılması ve eğitim setindeki bozucu etkilerin az olması gereklidir. Uygulamanın düşük işlemci ve düşük bellek kullanımı gerektirmesi ve platform bağımsız olması benzer uygulamalara göre en önemli avantajlarıdır.

Bir sonraki adım olarak direkt olarak kameradan görüntünün alınması sırasında aracın marka ve modelinin belirlenerek aracın servis istasyonuna girişte ilgili bölgeye sistem tarafından yönlendirilmesi hedeflenmektedir. Aracın kameranın önünde çok kısa bir süre durması görüntüsünün alınması için yeterli olacak ve aracın marka ve modeli belirlenebilecektir.

Kaynaklar

- [1] Zhang X., Jiang L., Vehicle Types Recognition Based on Neural Network, **Proceedings, CINC '09. International Conference on Computational Intelligence and Natural Computing**, 1, 3-6, (2009).
- [2] Wang, W., Vehicle Type Recognition Based on Radial Basis Function Neural Networks, **Proceedings, JCAI '09. International Joint Conference on Artificial Intelligence**, 444-447, (2009).
- [3] Wang, W., A Study on Contour Feature Algorithm for Vehicle Type Recognition, **Proceedings, JCAI '09. International Joint Conference on Artificial Intelligence**, 452-455, (2009).
- [4] Wu, W., Qisen Z., Mingjun, W., A method of vehicle classification using models and neural networks, **Proceedings, JCAI '09. International Joint Conference on Artificial Intelligence**, 4, 3022-3026, (2001).
- [5] Lan, L.W., Kuo, A.Y., Development of a fuzzy neural network color image vehicular detection, **Proceedings, The IEEE 5th International Conference on Intelligent Transportation Systems**, 88-93, (2002).
- [6] Haykin S., **Neural Networks**, 2nd ed., Pearson Education, New Jersey, (1999).
- [7] Liu, J., Sun, J., and Wang, S., Pattern Recognition: An Overview, **Proceedings, IJCSNS International Journal of Computer Science and Network Security**, 6, (2006).
- [8] Prewitt, <http://en.wikipedia.org/wiki/Prewitt> (4.5.2009).