

Elle Yazılmış Matematiksel İfadelerin İstatistiksel Olarak Tanınması Handwritten Mathematical Formula Recognition Using a Statistical Approach

Mehmet Çelik

Berrin Yanıkoğlu

Bilgisayar Mühendisliği Bölümü
Bilkent Üniversitesi
mcelik@cs.bilkent.edu

Mühendislik ve Doğa Bilimleri Fakültesi
Sabancı Üniversitesi
berrin@sabanciuniv.edu

ÖZETÇE

Bu çalışmada elle yazılan matematik ifadelerin tanınması için olasılıksal bir çerçeve anlatılmaktadır. Sistem, grameri kolayca genişletilebilir olduğundan esnek, çünkü kullandığı çizge grameri kural önceliği gerektirmez. Sistem aynı zamanda, bütün olası anlamları, erken veya katı karardan değerlendirildiği için, optimaldir. Bu makalede sisteme genel bakış ve kullandığımız içeriğe duyarlı çizge grameri ve ayrıştırma algoritmasının detayları anlatılmaktadır. Sistem gerçek zamanlı çalışmakta ve kısa ifadelerde %52 tanıma ve %88 yapı analizi başarısını göstermektedir.

ABSTRACT

We present a probabilistic framework for a mathematical expression recognition system. The system is flexible in that its grammar can be extended easily, thanks to its graph grammar which eliminates the need for specifying rule precedence. It is also optimal in the sense that all possible interpretations of the expressions are expanded, without making early commitments or hard decisions. The current system is able to recognize shorter expressions well and in real time. In this paper, we give an overview of the whole system and describe in detail our context sensitive graph grammar and the parsing process.

1. GİRİŞ

Elle yazılmış matematiksel ifadelerin otomatik olarak tanınması uzun zamandır üstünde çalışılan bir alandır. Artan sayıdaki kalemli cihazlarla bu alandaki araştırmalar yine önem kazanmaya başlamıştır. Elle yazılan girdi bilgisayarlarla etkileşimin doğal bir yoludur: bir kalem yazı yazmak, resim çizmek, düğmelere basmak, ve karmaşık denklemler yazmak için rahatlıkla kullanılabilir. Özellikle, matematiksel ifadeleri klavye ve fareyi kullanarak bilgisayara girmenin kolay bir yolu yoktur. Microsoft Equation Editor, Scientific Notebook gibi görsel arayüzlü programlar veya TeX/LaTeX dilleri matematiksel ifadeleri bilgisayara girmek için kullanılmaktadır, ancak bunlar kullanıcının dilin/arayüzün bilgisine sahip olmasını gerektirir. Ayrıca bu alternatifler matematiksel ifadeleri elle girmenin kolaylığından ve hızından çok uzaktır. Bu çalışmanın amacı, elektronik tabletlere yazılan matematiksel ifadelerin otomatik olarak tanınmasıdır. Matematiksel ifadeleri kalemle girmenin kolaylığı yanısıra kalemli cihazların sayısındaki artış göz önüne

alındığında, bu konu önemli bir araştırma alanı olarak karşımıza çıkar.

Tek boyutlu olarak bakabileceğimiz doğrusal metinlerin aksine, matematiksel ifadeler iki boyuta genişleyen anlamsal bir bütün oluştururlar (örn. bir toplama formülü alt ve üst indeksler ve ana formülden oluşan bir yapıdır). Matematiksel ifadelerin tanınması bu iki boyutlu yapının yapısal analizini ve ifadelerin içinde yer alan sembollerin tanınmasını içerir. Yapısal analiz örneğin bir toplama sembolünün alt ve üst endekslerini oluşturan sembollere karar verirken, sembol tanıma ise bu sembollerin hangi karakterlere karşılık geldiğini bulmayı amaçlar. Konu üzerinde yapılan araştırmalarda bu iki bölüm ayrı ayrı ele alındığı gibi tanıma işlemi tek bir aşamada da gerçekleştirilmekteydi.

Matematiksel ifadenin tanınmasındaki belli başlı işlemler ve karşılaşılan zorluklar şöyle sıralanabilir:

- **sembollerin tanınması:** ayrıştırılmış sembollerin tanınması belli bir olgunluğa gelmiş bir çalışma alanı olsa da sembol (karakter) tanıma oranları hala

- **sembollerin gruplanması:** OCR'dan farklı olarak matematiksel ifadeler birtakım ek özelliklere sahiptir. Bitişik yazılan rakamların bir tek sayıyı belirtmesi (örn. 1599) ve bitişik bazı harflerin sin, tan gibi özel birtakım işlevler tanımlaması göz önünde bulundurulması gereken durumlardan sadece birkaçıdır. Ayrıca işlem önceliği de bu yönden dikkat edilmesi gereken unsurlardandır. Sembollerin gruplanması için olası matematiksel yapı bilgisine ek olarak (örn. fonksiyon isimleri), yakınlık, boyut benzerliği gibi faktörler kullanılmaktadır.

- **semboller arasındaki yerel ilişkilerin kurulması:** Sembollerin çevresine göre konumu ve boyu incelenerek bu ilişkiler kurulmaya çalışılmaktadır. Üstel ifadelerin tanınması gibi durumlarda sadece yerel değil sembollerin bütün ifade içindeki konumuna bakılması gerekebilir. Yapı analizi için 2B gramerler ve çizge yöntemleri ağırlıktadır.

2. Önceki Çalışmalar

Matematiksel ifadeleri tanıma konusundaki çalışmalar 1960'larda başlamış olsa da, 1990'lara kadar bu alanda yayınlanan makaleler çok az sayıdadır [1,2]. Son senelerde ise özellikle çevrimiçi matematiksel ifadelerin tanınması artan ilgi görmüştür [3-8]. Çevrimiçi ve çevrimdışı sistemlerin performansları aşağıda ayrı olarak verilse de iki işlem arasındaki

fark sadece sembol bölütleme ve tanıma aşamalarında kendini gösterir. Ana problemin asıl zorluğu yapı analizi olduğundan, çevrimdışı sistemler konusunda yapılan çalışmalar da aşağıda özetlenmiştir.

2.1. Çevrimdışı Sistemler

Bu konudaki bilinen ilk çalışmalardan birinde R. H. Anderson [1] hatasız bir simge tanıyıcı varsaymış ve 2B gramer için bir koordinat grameri sunmuştur. Daha sonra Belaid ve Haton [2] singeleri tanımak için, ilkelere bölütlemeye dayalı bir yöntem önermiştir.

Chou [3] yapı analizi için iki boyutlu bir olasılıksal ve bağlam-duyarsız gramer (two-dimensional stochastic context-free grammar) kullanımını öngörmüş, daha sonra Kopec ile beraber metin tanıma konusuna genişlettiği bu yaklaşım alanında en etkili çalışmalardan olmuştur. Olasılıksal bir gramerde her kuralın uygulanmasında bir olasılık hesaplanır; örneğin indis kuralında bir sembolün diğerine göre belli bir pozisyonda olması şartını aramak yerine, daha az olası olan durumlarda da kuralın uygulanmasına izin verip bir olasılık değeri atamak istatistiksel yaklaşımdır ve hataları azaltır.

Zannibbi ve meslektaşları iki boyutlu ifadeleri tanımlamak için ağaç dönüşümü tabanlı bir yöntem önermişlerdir [4]. Ağaç dönüşümünde yapraklarda bulunan birbirleriyle alakalı uç semboller gruplanır ve aralarındaki ilişkiyi tanımlayan yeni bir alt-ağaç yerlerini alır (örn. "2" ve "x" yaprakları yerine "2 - mult - x" yaprağı gelir). Ancak bu yöntem sözdizimsel (syntactic) kurallardan oluştuğu için gürültüye ve değişikliklere karşı çok hassastır. Nitekim 73 matematiksel ifade içeren Washington Üniversitesi veritabanında elde ettikleri sonuçlar en iyi durumda sadece %37 başarıdır.

2.2. Çevrimiçi Sistemler

Koschinski ve meslektaşları [5] Saklı Markov Modelleri (SMMler) kullanarak tanıdıkları sembolleri, hipotez ağı diye nitelendirdikleri soldan sağa tek boyutlu bir ağın düğümleri olarak göstermiş, daha sonra bu düğümler arasında sembolleri en iyi şekilde gruplayacak yolu bulmuşlardır. Bu yöntemde iki boyutlu bir çizge kullanılmadığından semboller arasındaki 2B ilişki ancak içkin olarak belirtilmiştir. Bu yaklaşım daha sonra Tapia ve Rojas tarafından da kullanılmış ve Destek Vektör Makinaları tabanlı bir sembol tanıma mekanizmasıyla E-chalk adı verilen elektronik tahta sistemi gerçekleştirilmiştir [6].

Garain ve Chaudhuri'nin çalışmasının çevrimiçi çalışmalarının arasında en başarılı ve önceli olduğu söylenebilir [7]. Bu çalışmada matematiksel ifadeler öncelikle iki farklı sınıflandırıcıdan geçirilip sembol tanıma işlemini tamamlandıktan sonra yapısal analiz uygulanır. Çevrimiçi olarak alınan ifadelerdeki sembollerin çizimleri ile ilgili zamansal bilgisi, çizim yönü ve şekilleri sınıflandırıcılar tarafından kullanılarak yüksek sembol tanıma oranları elde edilmiştir. Yapısal analiz aşamasında ise dikey ve yatay bölütleme ile ifade parçalara ayrılır ve oluşturulan bir gramer ile bölütler sıralı bir şekilde birleştirilir. Bu çalışmada çevrimiçi bir sistem önerildiğinden kök işareti gibi sembollerin etki alanlarından önce yazıldığı varsayımında bulunulmuştur. Sistem 5500 matematiksel ifadedeki yaklaşık 35000 yapıyı %98 başarıyla tanımıştır. Ancak matematiksel ifadelerin içerisindeki yapılar

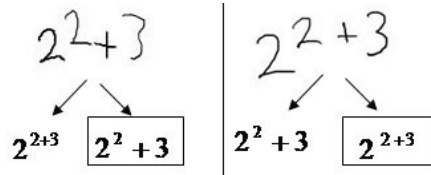
doğru tanındığı halde yanlış bölütleme sonucu oluşan hatalar sebebiyle 5500 ifadenin %75'i hatasız olarak tanınır. Ancak bu başarının değerlendirilmesinde kullanılan varsayımlar ve kullanıcıya yazım sırasında verilen geribesleme ve elde edilen kullanıcı onayı sayesinde sembollerin tanınmasında hiç hata olmaması dikkate alınmalıdır.

Bu makalede tanıtılan sistem ise, Büyükbayrak ve meslektaşlarının geliştirdiği Mathlet [8] sisteminde yapılan çalışmaların bir devamı olarak geliştirilmiştir.

3. Önerilen Yöntem

Bu çalışmada matematiksel ifade tanımda en başarılı yöntem olan çizge gramerler yenilikçi bir şekilde kullanılmıştır. Önerilen yöntemin diğer çizge gramer kullanan yöntemlerden ana farkı matematiksel ifadenin bütün olası yorumlarının istatistiksel olarak değerlendirilmesidir.

Şekil 1'de iki türlü yorumlanabilecek iki matematiksel ifade gösterilmektedir. Bunların doğru yorumunun bulunması ancak ayrıştırmanın tüm sembollerin kapsadığı zaman yapılabilir. Yani, üstlerle ilgili gramer kuralının uygulanması başta uygun gibi görülse de, tüm semboller ve bunların arasındaki uzamsal ilişkiler değerlendirilince, öne çıkan yorum olmayabilir.



Şekil 1. İki türlü yorumlanabilecek iki matematiksel ifade; kutularda tüm semboller görülünce daha olası olan yorum gösterilmektedir.

Halbuki şu ana kadar varolan çizge gramer kullanan sistemlerde, ifadeye uygulanabilecek gramer kuralları, çizge dönüşüm (graph rewriting) yöntemleri ile mutlak şekilde uygulanmıştır. Örneğin üstlerle ilgili gramer kuralı ilk başta güzel veya uygulanabilir görülünce alt ve üst semboller birleştirilerek yerlerine yeni bir simge (token) konur. Önerilen sistemde ise bu gramer kuralı uygulanır, ona bir olabirlik atanır, ama bu sadece olası bir yorum olarak genişletilmiş çizgede tutulur (bkz. Şekil 2). Bu şekilde bütün olası yorumlar uygulanabilecek gramer kuralları oldukça genişletilmiş çizgeye eklenir. Ayrıştırmanın sonunda bütün semboller içeren yorumlar arasında en yüksek olabirliğe sahip olan bulunarak, ifadenin en olası yorumu olarak seçilir.

3.1. Bölütleme

Girilen matematiksel ifade öncelikle zamana göre bölütlenir; böylece aralarında belli bir süre geçen bütün vuruşlar (stroke) ayrı birer sembol gibi değerlendirilir. Daha sonra ise bu semboller uzamda önemli ölçüde keşiyorlarsa (örn. t'nin iki vuruşu) tekrar birleştirilirler. Bu şekilde ifadenin makul bir bölütlemesi elde edilir. Bu bölütleme ideal olmasa da %95-99 oranında doğru çalışmaktadır. Bölütlemenin ve çizge gramerinin aynı anda yapılması, yaratacağı hesaplama karmaşıklığından dolayı olur bir çözüm gibi durmamaktadır.

3.2. Sembol Tanıma

Bölütlenen semboller Destek Vektör Makinaları (DVM) ve Yayıp Sınır Ağları (YSA) tabanlı bir tanıma motoru ile tanınır ve her sembol için en olası 3 alternatif döndürülür. Bölütleme ve sembol tanıma bu makalenin kapsamı dışında olduğundan burada detayları belirtilmemiştir. Şu anki sistemde sembol tanıma başarısı %91'dir.

3.3. Çizge Gramerleri

Bir çizge grameri (graph grammar) bir çizgeyi başka bir çizgeye dönüştüren bir kurallar kümesidir. Örneğin bir kural $r = (g_l; g_r; C; E)$, uygulanabilirlik belirtimi (applicability predicate) C 'ye uyduğu takdirde, g_l alt-çizgesini g_r alt-çizgesine dönüştürür ve gömme kuralı (embedding rule) E 'ye göre ana çizgenin içine yerleştirir. Uygulanabilirlik belirtimi C bir kuralın uygulanması için gerekli olan (ve çok katı olmayan) şartların var olup olmadığını kontrol eder. Örneğin indis kuralının uygulanabilirlik belirtimi indis sembolünün kabaca alt sembole göre kabaca daha yukarıda, sağda ve daha küçükçe olmasını gerektirir.

3.4. Başlangıç Çizgesi ve Önerilen Çizge Grameri

Ayrıştırma işlemi başlamadan önce matematiksel ifadeden bir başlangıç çizgesi oluşturulur. Bu çizgede düğümler (node) tanınmış semboller ifade ederken, ayrıtlar (edge) bu semboller arasındaki komşuluk ilişkilerini belirtir. Burada komşuluk ilişkisi iki sembol arasında başka bir sembol ile kesişmeyen bir doğru çizilebilmesi şeklinde tanımlanmıştır.

Ayrıştırıcı her gramer kuralının her bir çizge düğümüne uygulanabilirliğine bakar ve uyumlu olanları sırayla uygular. Ayrıştırma süreci uygulanabilir hiçbir çizge gramer kuralı kalmayınca kadar devam eder. Bir sembol bir gramer kuralının uygulanması için göz önünde bulundurulduğunda, ayrıştırıcı önce uygulanabilirlik belirtilerini kontrol etmektedir. Kuralların çoğunda bu semboller arası uzaklık ve açılımların kontrol edilmesi şeklinde olmaktadır. Bazı kurallar farklı kontroller de yapabilmektedir. Kullanılan kısıtlamalar matematik ifade için farklı seçeneklerin korunabilmesi için esnek tutulmuştur. Bir gramer kuralının uygulanması sonucu yeni bir düğüm yaratılır ve varolan çizgeye 3 farklı ayrıt türü ile bağlanır. Dolayısıyla sistemimizde düğüm ve ayrıtlar tam olarak şöyledir:

Düğüm: Her düğüm $n=(t,c,i,A)$ şeklindeki bir (tuple) ile tanımlanmaktadır. Bu tanımda t düğümün cinsini, c bu düğümü üretmiş olan gramer kuralını, i benzersiz bir tanımlayıcıyı (unique identifier) ve A da düğümün üzerinde taşıdığı nitelikler kümesini temsil etmektedir. Düğüm tipi, düğümün temsil ettiği sembolün sözlük karşılığından gelmektedir, mesela rakamlar, harfler vb. O düğümü hangi kuralın ürettiği bilgisi ise ayrıştırma ağacının çıkarılmasında kullanılmaktadır. Şekil 2'deki her bir kutu bir düğümü temsil etmektedir.

Ayrıtlar: Her ayrıt $e = (t,n1,n2)$ şeklindeki bir (tuple) şeklinde tanımlanmaktadır. Bu tanımda t ayrıtın cinsini, $n1$ ve $n2$ de aralarında bağlantı kurulan düğümleri temsil etmektedir. Şekil 3'te görülebileceği üzere, üç farklı ayrıt türü tanımlanmıştır.

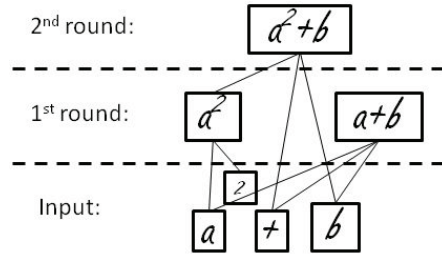
- Uzaysal ilişki ayrıtları iki düğümün uzaysal olarak

komşu olduğun belirtir.

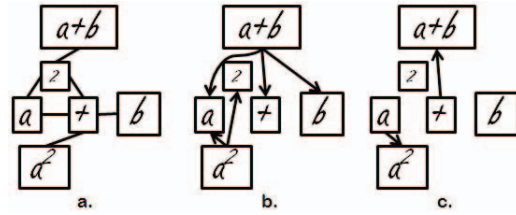
- Bileşen ayrıtları ayrıştırma sırasında oluşturulmuş düğüm ile bunun bileşenlerini bağlamaktadır.
- Üretim ayrıtları ayrıştırma sırasında yeni düğüm oluşmasına sebep olmuş düğümlerden, yeni oluşan düğüme yapılan bağlantılardır.

Aynı düğümün tekrar tekrar üretilmesine engel olmak için ise belirtilerden biri üretim ayrıtının olmamasının kontrol edilmesidir. Bu ayrıştırma işini daha karmaşık hale getirirse de gramer kuralları arasında öncelik tanımlanmasının önüne geçmektedir.

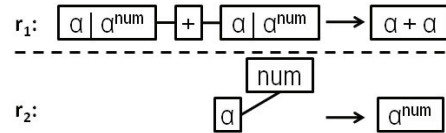
Tanımladığımız gramerde, uzaysal ilişki ayrıtları herhangi bir öznitelik taşımamaktadır. Uzaysal ilişkileri sağ-üst sağ-alt gibi sınıflandırmak yerine bu sınıflandırma belirtiler ile her kural için özel olarak yapılabilmektedir. Böylece her kural kendi uzaysal ilişki sınıfını tanımlayabilmektedir. Bu da gramere esneklik kazandırmaktadır. Yer darlığı yüzünden gramer kurallarımız burada listelenememiştir ama geniş bir matematik ifadeler kümesini temsil edecek şekilde tasarlanmıştır.



Şekil 2. İfadedeki komşuluk ilişkilerini gösteren başlangıç çizgesi gramer kurallarının uygulanması sonucu kademe kademe, başka kural uygulanamayana kadar genişletilir.



Şekil 3. İkinci aşamadan sonra elde edilen çizge, a) uzaysal ilişki ayrıtları, b) bileşen ayrıtları, c) üretim ayrıtları.



Şekil 4. Sistemde kullanılan gramerden 2 kural (+ ve indis).

3.5. Ayrıştırma Algoritması

Ayrıştırma algoritmamız yalnız bir aşağıdan yukarıya doğru yöntem izlemektedir. Temel olarak iki iş yapılmalıdır. Kuralla eşleşen bir çizge parçasının bulunması ve yeni çizge parçasının çizgeye eklenmesi. Gramer kurallarındaki çizgelerin hepsi

yıldız biçimli olduğu için, ayrıştırıcı önce merkez düğümün eşleşmesini kontrol etmekte ardından çevre düğümler kontrol edilmektedir. Bir eşleşmenin bulunmasının ardından, yeni bir düğüm oluşturulmakta ve çizgeye üretim ve bileşen ayrıtları ile eklenmektedir. Uzaysal ilişki ayrıtları mevcut çizge üzerinde uygulanabilecek hiçbir kural kalmadığında oluşturulmaktadır. Her düğüm kendisini oluşturan düğümlerden uzaysal ilişki ayrıtlarını da almaktadır, ayrıca yeni düğümler arasında bu ilişkiler ayrıca tespit edilmektedir. Bir ayrıştırma işlemi Şekil 2'de örneklenmiş ve bu işlem sırasında oluşan çizgenin bir durumu Şekil 3'te gösterilmiştir. Burada aynı sembol için birden fazla düğüm gösterilmesi şeklin okunabilirliğini arttırmak içindir.

Ayrıştırma işleminin çıktısı, girdi çizgesi ve buna ayrıştırma işlemi sırasında eklenen bütün düğümleri içeren bir çizgedir. Eğer girdi, uygulanan gramer tarafından tanımlanabiliyorsa, çıktı çizgesinin içinde en az bir tane tüm girdi sembollerini kapsayan düğüm oluşmuş olmalıdır.

3.6. Olasılık Hesabı

Sisteme girilen matematiksel ifadenin birden çok farklı yorumu olabiliyorsa (örn. Şekil 1'de her 2 ifade için ikişer farklı yorum gösterilmiştir), ayrıştırma sonucunda birden fazla düğüm tüm girdi sembollerini kapsayacaktır. Sistem bunlar arasında en olası olanı seçmek için semboller arasındaki uzamsal ilişkilerin kullanılan her gramer kuralına göre olasılığını değerlendirir. Örneğin Şekil 1'de sağ taraftaki ifadedeki 3 sembolünün yeri ve boyutu incelendiğinde, kare içine alınmış yorum öne çıkar. Bu olasılık değerleri, ayrıştırma sırasında ortalama log olasılık olarak ve bu amaçla topladığımız uzamsal istatistiklere bakılarak hesaplanır (komşu sembollerin en, boy ve uzaklık dağılımları) ve simge için saklanır. Böylece bir ifadenin olası bütün yorumları olabirliklerine göre sıralanıp, en olası olanı seçilir.

4. Sonuçlar

Matematiksel ifade tanıma sistemlerinde, kullanıcıya geribesleme verilmesi, ve matematiksel ifadenin uzunluğuna (içerdiği sembol sayısı) ve karmaşıklığına (düz bir satır veya pekçok seviyeli ifade) bağlı olarak hem tanıma hızı hem başarısı azalır. Tablo 1'de geliştirilen sistemin 5 farklı kullanıcıdan toplanmış 20'şer ifade (toplam 100 ifade) ile denenmesi sonucunda elde edilen başarı, doğru ifade tanıma, doğru yapısal analiz (ifadede sembol tanıma hatası olabilir), ve doğru sembol tanıma oranları olarak listelenmiştir.

Bu sonuçlara bakınca küçük ifadelerin genellikle rahat tanıdığı ama üstünde biriken hatalar yüzünden performansın düştüğü görülmektedir. Bu çalışmada vurgulanan problem gramer kullanımı ile yapı analizidir. Bu açıdan sistemin kısa ifadelerde oldukça başarılı olduğu görülür (88%), ancak sembol tanıma hataları yapısal analizi de etkilediğinden (örn. bir '(' yanlış tanıdığı zaman yapısal analiz de bozulmaktadır), bu başarı ifade uzadıkça belirgin şekilde düşmektedir. Nitekim sembol tanıma başarısı tüm ifadeler üzerinde %79'dur ve bu ayrı sembollerde elde edilen sonuca göre (%91) oldukça düşüktür.

	Başarı	Oran
İfade Uzunluğu ≤ 10		25/100
Doğru Tanıma	52%	13/25
Doğru Yapısal Analiz	88%	22/25
Doğru sembol tanıma	91%	80/88
İfade Uzunluğu [11-30] arası		75/100
Doğru Tanıma	5,33%	4/75
Doğru Yapısal Analiz	37,33%	28/75
Doğru sembol tanıma	76%	369/476

Tablo 1. İfade uzunluğuna göre incelenmiş sonuçlar (5 kullanıcı x 20 ifade).

5. Gelecekteki Çalışmalar

Sembol tanıma motorunun iyileştirilmesi, sistemin genel olarak hızlandırılması ve iyileştirmesi, kullanıcı arayüzünde yapılabilecek olası geliştirmeler ile sistemin hem daha hızlı hem de daha başarılı olmasını beklemekteyiz. Örneğin %79 yerine %95'lik bir sembol tanıma başarısı toplam hatayı önemli ölçüde azaltacaktır. Bu konuda sınıflandırıcı birleştirme yöntemlerinden faydalanarak gerekli performans artışını yakalayabiliriz.

Sistem şu anda gerçek zamanlı çalışsa da (ifade tanıma ortalama yaklaşık 1sn), bazı uzun ifadelerde çok yavaşlamaktadır, bunun için olasılık açısından öne çıkan simgelerin daha önce değerlendirilmesi ele alınacaktır.

6. KAYNAKÇA

- [1] R. H. Anderson, "Syntax-directed recognition of hand-printed two-dimensional mathematics", Ph.D. Dissertation, Dept. Eng. Appl. Phys., Harvard Univ., Cambridge, MA, 1968.
- [2] A. Belaid and J. Haton, "A syntactic approach for handwritten mathematical formula recognition", IEEE PAMI, vol. 6, pp. 105-111, Jan. 1984.
- [3] P. A. Chou.: Recognition of equations using a twodimensional stochastic context-free grammar. In: Proc. SPIE Visual Commun. Image Process. IV, volume 1199, pp. 852-863, Philadelphia, PA, Nov. 1989.
- [4] R. Zanibbi, D. Blostein, and J. R. Cordy, "Recognizing mathematical expressions using tree transformation", IEEE PAMI, vol. 24, pp. 1455-1467, Nov. 2002.
- [5] M. Koschinski, H.-J. Winkler, and M. Lang, "Segmentation and recognition of symbols within handwritten mathematical expressions", in Proc. ICASSP, vol. 4, Detroit, MI, 1995, pp. 2439-2442.
- [6] E. Tapia and R. Rojas, "Recognition of on-line handwritten mathematical formulas in the E-chalk system", Proc. ICDAR, Edinburgh, U.K., 2003, pp. 980-984.
- [7] U. Garain and B. B. Chaudhuri, "Recognition of Online Handwritten Mathematical Expressions", IEEE Trans. on Sys., Man and Cybern., vol. 34, No.6, 2004, pp 2366-2375.
- [8] H. Büyükbayrak 'Online Handwritten Mathematical Expression Recognition', M.S. Thesis, Sabancı University, 2005.